הטכניון
מכון טכנולוגי לישראל
TECHNION
Israel Institute of Technology

ספריות הטכניון
The Technion Libraries

בית הספר ללימודי מוסמכים ע"ש ארווין וג'ואן ג'ייקובס
Irwin and Joan Jacobs Graduate School

©
All rights reserved to the author

This work, in whole or in part, may not be copied (in any media), printed, translated, stored in a retrieval system, transmitted via the internet or other electronic means, except for "fair use" of brief quotations for academic instruction, criticism, or research purposes only. Commercial use of this material is completely prohibited.

©
כל הזכויות שמורות למחבר/ת

אין להעתיק (במדיה כלשהי), להדפיס, לתרגם, לאחסן במאגר מידע, להפיץ באינטרנט, חיבור זה או כל חלק ממנו, למעט "שימוש הוגן" בקטעים קצרים מן החיבור למטרות לימוד, הוראה, ביקורת או מחקר. שימוש מסחרי בחומר הכלול בחיבור זה אסור בהחלט.

© Technion - Israel Institute of Technology, Elyachar Central Library

Network Optimization Methods in Passivity-Based Cooperative Control

Research thesis
In Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy

Miel Sharf

Submitted to the Senate of the Technion - Israel
Institute of Technology
Tishri 5780, Haifa, October 2019

© Technion - Israel Institute of Technology, Elyachar Central Library

The Research Thesis Was Done Under The Supervision of Prof. Daniel Zelazo in the Faculty of Aerospace Engineering

The Generous Financial Help Of The German-Israeli Foundation for Scientific Research and Development is Gratefully Acknowledged

To my mom, Yosefa.

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

Contents

1 Introduction **5**
1.1 Introduction and Focus 5
1.2 Background 8
1.2.1 Network Optimization 8
1.2.2 Diffusively Coupled Networks 9
1.2.3 The Role of Passivity in Cooperative Control 11
1.3 Contributions and Thesis Outline 15
1.4 Publications 19

2 A Network Optimization Framework for MIMO Systems **21**
2.1 Introduction 21
2.2 Cyclically Monotone Relations and Cooperative Control 22
2.2.1 Steady-States and Network Consistency 23
2.2.2 Connecting Steady-States to Network Optimization . . . 24
2.2.3 Convergence to the Steady-State 28
2.3 Examples of MEICMP Systems 30
2.3.1 Convex-Gradient Systems with Oscillatory Terms 30
2.3.2 Oscillatory Systems with Damping 34
2.3.3 Example: A Network of Planar Oscillators 36
2.4 Conclusions 38

3 A Network Optimization Framework for Passive-Short Agents **39**
3.1 Introduction 39
3.2 Shortage of Passivity and Failures of the Network Optimization Framework 40
3.2.1 Shortage of Passivity 40
3.2.2 Failure of the Network Optimization Framework 43
3.3 A Network Optimization Framework for Output Passive-Short Agents 47
3.3.1 Agent-Only Regularization 48
3.3.2 Network-Only Regularization 51
3.3.3 Hybrid Regularization 55
3.4 A Network Optimization Framework for General Passive-Short Agents 57

3.4.1 Monotonization of I/O Relations by Linear Transformations: A Geometric Approach 58
3.4.2 From Monotonization to Passivation and Implementation 63
3.4.3 Maximality of Input-Output Relations and the Network Optimization Framework 69
3.5 Case Studies 73
3.5.1 Unstable First Order Systems 73
3.5.2 Gradient Systems 74
3.5.3 Traffic Model 75
3.6 Conclusions 77

4 A Network Optimization Framework for Controller Synthesis 81
4.1 Introduction 81
4.2 Final-Value Synthesis for Multi-Agent Systems 82
4.2.1 Characterizing Forcible Steady-States 83
4.2.2 Forcing Global Asymptotic Convergence 84
4.2.3 Changing the Objective and "Formation Reconfiguration" 86
4.2.4 Plant Augmentation and Leading Agents for Non-achievable Steady States 88
4.2.5 Final-Value Synthesis for Other Signals 89
4.2.6 Case Studies 91
4.3 Clustering in Symmetric Multi-Agent Systems 94
4.3.1 The Static Automorphism Group of a Multi-Agent System 95
4.3.2 Steady-State Clustering in Multi-Agent Systems 97
4.3.3 Homogeneous Networks and Cluster Synthesis 100
4.4 Conclusions 108

5 Applications of the Network Optimization Framework in Data-Driven Control 111
5.1 Introduction 111
5.2 Problem Formulation and Verification of Passivity 112
5.3 Uniform Gain Amplification Methods 116
5.3.1 Theory 117
5.3.2 Data-Driven Determination of Gains 120
5.4 Non-Uniform Gain Amplification Methods 124
5.5 Case Studies 128
5.5.1 Velocity Coordination in Vehicles with Drag and Exogenous Forces 128
5.5.2 Clustering in Neural Networks 130
5.6 Conclusions 133

© Technion - Israel Institute of Technology, Elyachar Central Library

6 Applications of the Network Optimization Framework in Network Identification 135
6.1 Introduction . . . 135
6.2 Network Differentiation Using Constant Exogenous Inputs . . . 136
6.2.1 Motivation and Problem Definition . . . 137
6.2.2 Constructing Indication Vectors Using Randomization . . . 141
6.2.3 Constructing Indication Vectors Using Algebraic Methods . . . 144
6.3 Network Identification with Minimal Time Complexity . . . 147
6.3.1 An Algorithm for LTI Agents and Controllers . . . 149
6.3.2 An Algorithm for General MEIP Agents and Controllers Using Linearization . . . 154
6.4 Robustness and Practical Considerations . . . 160
6.4.1 Robustness to Noise and Disturbances . . . 160
6.4.2 Probing Inputs Supported on Subsets of Nodes . . . 162
6.4.3 Time Complexity Bounds for the Network Reconstruction Problem . . . 164
6.5 Case Studies . . . 168
6.5.1 Linear Agents and Controllers . . . 168
6.5.2 A Neural Network . . . 169
6.6 Conclusions . . . 170

7 Applications of the Network Optimization Framework in Fault Detection and Isolation 173
7.1 Introduction . . . 173
7.2 Problem Formulation and Assumptions . . . 174
7.3 Asymptotic Differentiation Between Networks . . . 176
7.4 Network Fault Detection and Isolation . . . 181
7.4.1 Fault Detection Over Networks . . . 181
7.4.2 Multi-Agent Synthesis in the Presence of an Adversary . . . 183
7.4.3 Network Fault Isolation in Multi-Agent Systems . . . 186
7.5 Online Assertion of Network Convergence . . . 189
7.5.1 Asserting Convergence Using High-Rate Sampling . . . 190
7.5.2 Asserting Convergence Using Convergence Profiles . . . 195
7.6 Case Studies . . . 200
7.6.1 Network FDI for LTI First Order Systems . . . 200
7.6.2 Network FDI for Velocity-Coordinating Vehicles . . . 202
7.7 Conclusions . . . 203

8 Summary 205
8.1 Conclusions . . . 205
8.2 Outlook . . . 207

© Technion - Israel Institute of Technology, Elyachar Central Library

A Convex Analysis and Optimization Theory **209**
A.1 Convex Sets and Convex Functions 209
A.2 Subdifferentials 212
A.3 Rockafellar's Theorem and Cyclically Monotone Relations 213

B Graph Theory and Algebraic Graph Theory **215**

C Dynamical Systems, Passivity and Stability **217**
C.1 Stability and Lyapunov Theory 217
C.2 Passivity 219

D Complexity Theory for Matrix Multiplication and Inversion **223**

E Tools From Algebraic Number Theory **225**

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

List of Tables

3.1 Elementary matrices and their realizations. 66
3.2 Comparison of hybrid regularization terms for networks with output-passive short agents. 74

6.1 Network detection algorithm performance for LTI systems 169
6.2 Network detection algorithm performance for neural networks . . 170

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

List of Figures

1.1 Block diagram of a diffusively-coupled network 10
1.2 Examples of steady-state input-output relations. 13

2.1 Consistency of network steady-states. 24
2.2 Block diagram of a general feedback interconnection. 29
2.3 Case study: A network of MIMO oscillators - Positions. 37
2.4 Case study: A network of MIMO oscillators - Distance from value forecasted by the network optimization framework 37

3.1 An example of the absence of integral functions. 45
3.2 Steady-state relations and integral functions for output passive-short systems 46
3.3 Failure of the network optimization framework for passive-short agents. 47
3.4 Agent-based regularization. 50
3.5 Network-based regularization. 53
3.6 Hybrid regularization. 56
3.7 Passivation, monotonization and convexification by transformation. 57
3.8 Example of a symmetric double-cone and a symmetric section. . 59
3.9 The transformed system $\tilde{\Sigma}$ after the linear transformation T. . . 65
3.10 Comparison of regularization terms. 75
3.11 Steady-state relations and integral functions under transformation. 76
3.12 Trajectories of regularized network of gradient systems. 76
3.13 Regularization in a traffic control model. 78

4.1 The formation reconfiguration scheme. 87
4.2 Agent augmentation overcoming non-achievable steady-states. . . 88
4.3 Formation control of damped oscillators. 92
4.4 Network structure of damped MIMO oscillators. 93
4.5 Formation control of damped MIMO oscillators. 94
4.6 An example of the exchangeability graph. 99
4.7 Trajectories of a statically homogeneous network with non-identical agents. 101
4.8 Cluster synthesis - The closed-loop system trajectories. 102
4.9 First example of a graph solving the cluster synthesis problem. . 107

4.10 Second example of a graph solving the cluster synthesis problem. 107
4.11 Third example of a graph solving the cluster synthesis problem. . 108
4.12 Fourth example of a graph solving the cluster synthesis problem. 108

5.1 Block diagram of the diffusively-coupled network $(\mathcal{G}, \Sigma, \Pi, A)$. . . 114
5.2 Experimental setup for the data-driven control algorithm. 123
5.3 Experimental data for data-driven synthesis of a vehicle network. 130
5.4 Results of data-driven control for a vehicle network. 131
5.5 Iterative data-driven control for vehicle network. 131
5.6 Results of data-driven control for a neural network. 132
5.7 Iterative data-driven control for neural network. 132

6.1 Network differentiation in neural networks. 147
6.2 Network reconstruction of an LTI network. 169
6.3 Network reconstruction of a neural network. 170

7.1 Examples of r-connected graphs. 179
7.2 Underlying graphs for case studies in fault isolation. 201
7.3 First set of scenarios for fault isolation in LTI systems case study. 202
7.4 Second set of scenarios for fault isolation in LTI systems case study. 202
7.5 First set of scenarios for fault isolation in vehicle networks case study 204
7.6 First set of scenarios for fault isolation in vehicle networks case study 204

C.1 Interconnections preserving passivity. 220

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

Acknowledgements

This thesis presents the results of my research at the Technion - Israel Institute of Technology, which was my home throughout my bachelor's, master's and doctorate degrees. During my time at the Technion, I was fortunate to find a number of people who became mentors, teachers, and friends to me.

First of all, I want to thank Prof. Daniel Zelazo. He was a wonderful, motivating and supportive advisor and he led me well on my way through the academic world. His advice on professional matters was invaluable to me, and I thank him for the freedom to pursue my own research ideas. I would also like to thank Prof. Moshe Idan, Prof. Leonid Mirkin and Prof. Michael Margaliot (Tel Aviv University) for their interest in my work, for their encouraging comments, and for being members of my doctoral examination committee. In particular, I sincerely thank Prof. Leonid Mirkin for the insightful and constructive discussions we had more than once throughout my doctoral studies. I would also like to thank my collaborators, Prof. Anoop Jain (Indian Institute of Technology - Jodhpur), Anne Koch (IST, University of Stuttgart), and Frank Allgöwer (IST, University of Stuttgart) for stimulating, insightful, and fruitful discussions.

Moreover, I would like to thank Prof. Uri Bader (Weizmann Institute of Science) and Prof. Shahar Mendelson (The Australian National University and LPSM, Sorbonne University) for guiding me through my first years in academia, and providing advice whenever I needed it. I can certainly say that you both have shaped my views and philosophy about mathematics and its applications, which was of tremendous help during my doctoral studies. I would also like to thank the German-Israeli Foundation for Scientific Research and Development (GIF) for its generous financial support in my research, as well as the IAAC, IEEE and MDPI for their conference-related financial support, which allowed me to present my works in different venues.

Last, but most certainly not least, I would like to thank my mom, Yosefa Sharf. Thank you for you invaluable help, and the constant support throughout the years of my doctoral studies. You escorted me through this wild journey, and this work belongs to you just as much as it belongs to me. **אני אוהב אותך**.

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

Abstract

Cooperative control and multi-agent networks have been subject to extensive research over the last few years, exhibiting both a rich theoretical framework as well as a wide range of applications. In this venue, researchers have tried to establish a unified theory for networks of dynamical systems. Two recurring themes that appear in many theories include graph theory and energy-based control, i.e. the notion of passivity. Passivity was first applied to the world of multi-agent systems by Arcak, and since then many different variants of passivity were suggested to tackle cooperative control problems, including incremental passivity and equilibrium-independent passivity (EIP).

In 2014, Bürger, Zelazo and Allgöwer introduced the notion of maximally equilibrium-independent passive systems (MEIP), in which passivity with respect to all steady-state inputs is assumed, and the collection of all steady-state input-output pairs is a monotone relation. They showed that the steady-state limit of a diffusively-coupled multi-agent network, with MEIP agents and controllers, can be found by solving a pair of dual network optimization problems, known as the optimal potential and optimal flow problems, which have been studied in the field of network optimization for decades. Thus a network optimization framework for analysis of multi-agent systems was established. However, it has a few main drawbacks. First, it requires the agents to be single-input-single-output systems, limiting the application to many real-world systems. Second, it requires that the agents are passive with respect to any steady-state they possess, excluding systems like generators and other passive-short systems. Lastly, the result they present is purely an analysis result, giving no method for synthesizing controllers.

The research presented in this thesis confronts all three problems. First, the notion of MEIP is extended to include multiple-input-multiple-output systems by applying the notion of cyclically monotone relations introduced by Rockafellar, and a generalized version of the network optimization framework is presented. Second, networks with passive-short agents are treated. In this case, the associated network optimization problems are non-convex, and it is shown that convexifying them results in a passivizing transformation for the agents, validating the augmented network optimization framework. Lastly, we apply the framework to solve various problems in cooperative control, including final-value synthesis, model-free synthesis, network identification, and fault detection and isolation.

© Technion - Israel Institute of Technology, Elyachar Central Library

Table of Notation

In this dissertation, we will also use the following notation:

Notation	Meaning
$\|A\|$	The cardinality (size) of the set A.
$\mathbb{R}$	The set of real numbers.
$\mathbb{Q}$	The set of rational numbers.
$\mathbb{Z}$	The set of integers.
$\mathbf{0}$, $\mathbf{0}_d$	The all-zero vector of length d.
$\mathbb{1}, \mathbb{1}_d$	The all-one vector of length d.
e_i	The i-th standard basis vector.
Id_d	The identity matrix of size $d \times d$.
$\ker(A)$	The kernel of the linear transformation A.
$\mathrm{Im}(A)$	The image of the linear transformation A.
$U^\perp$	The orthogonal complement of the linear subspace U.
Proj_U	The orthogonal projection operator on U.
$A \otimes B$	The Kronecker product of the matrices A and B.
$\|\|x\|\|$	The Euclidean norm of the vector x.
$A \geq B$	The matrix $A - B$ is positive semi-definite.
$A > B$	The matrix $A - B$ is positive-definite.
$\underline{\sigma}(A)$	The minimal singular value of A
$\overline{\sigma}(A)$	The maximal singular value of A
$\lambda_{\min}$	The minimal eigenvalue of A
$\lambda_{\max}$	The maximal eigenvalue of A
$\mathbb{V}$	A set of vertices.
$\mathbb{E}$	A set of pairs of vertices, called edges.
$\mathcal{G} = (\mathbb{V}, \mathbb{E})$	A graph $\mathcal{G}$ with vertices $\mathbb{V}$ and edges $\mathbb{E}$.
$\mathcal{E}$, $\mathcal{E}_\mathcal{G}$	The incidence matrix of the graph $\mathcal{G}$.
$\mathcal{E}_{\mathcal{G},d}$	The incidence operator $\mathcal{E}_{\mathcal{G},d} = \mathcal{E}_\mathcal{G} \otimes \mathrm{Id}_d$ on the graph $\mathcal{G}$.
$\lambda_2(\mathcal{G})$	The algebraic connectivity of the graph $\mathcal{G}$.
Σ_i	The i-th agent in a multi-agent system.
Σ	The collection of all agents in a multi-agent system.
Π_e	The e-th controller in a multi-agent system.
Π	The collection of all controllers in a multi-agent system.
$(\mathcal{G}, \Sigma, \Pi)$	A diffusively coupled system with agents Σ, controllers Π, and interaction graph $\mathcal{G}$.
$M\Sigma, \Sigma M$	The cascade of the dynamical system Σ and the linear map M.
∇F	The gradient of the function F.
$\mathrm{Hess}(U)$	The Hessian matrix of the function U
∂F	The subgradient of the function F.
$\mathcal{C}^q$	The space of functions which are continuously differentiable q times.
$I_\mathcal{D}$	The indicator function of the set $\mathcal{D}$, defined as $\begin{cases} 0 & x \in \mathcal{D} \\ \infty & x \notin \mathcal{D} \end{cases}$.
I_c	The indicator function of the set $\mathcal{D} = \{c\}$.
$K^\star$	The Legendre transform of the function K, defined as $K^\star(\mathrm{y}) = \sup_\mathrm{u}\{\mathrm{y}^\top \mathrm{u} - K(u)\}$.

© Technion - Israel Institute of Technology, Elyachar Central Library

Notation	Meaning
$O(f)$	A function g growing no faster than f in the specified limit, i.e. $\limsup_{x\to\infty}\frac{g(x)}{f(x)}<\infty$ or $\limsup_{x\to x_0}\frac{g(x)}{f(x)}<\infty$, depending on the context.
$o(f)$	A function g growing strictly slower than f in the specified limit, i.e. $\lim_{x\to\infty}\frac{g(x)}{f(x)}=0$ or $\lim_{x\to x_0}\frac{g(x)}{f(x)}=0$, depending on the context.
$\Omega(f)$	A function g growing no slower than f in the specified limit, i.e. $\liminf_{x\to\infty}\frac{g(x)}{f(x)}>0$ or $\liminf_{x\to x_0}\frac{g(x)}{f(x)}>0$, depending on the context.
$\omega(f)$	A function g growing strictly faster than f in the specified limit, i.e. $\lim_{x\to\infty}\frac{g(x)}{f(x)}=\infty$ or $\lim_{x\to x_0}\frac{g(x)}{f(x)}=\infty$, depending on the context.
$\Theta(f)$	A function g growing as fast as f in the specified limit, i.e. $\liminf_{x\to\infty}\frac{g(x)}{f(x)}>0$ and $\limsup_{x\to\infty}\frac{g(x)}{f(x)}<\infty$, or $\liminf_{x\to x_0}\frac{g(x)}{f(x)}>0$ and $\limsup_{x\to x_0}\frac{g(x)}{f(x)}<\infty$, depending on the context.

Moreover, italicized letters (e.g. $y_i(t)$ or y_i) will denote time-dependent signals, whereas normal font letters (e.g. y_i) will denote constant vectors. We will also use the following acronyms:

Acronym	Meaning
CM	Cyclically Monotone
EIP	Equilibrium-Independent Passive (or Passivity)
EIPS	Equilibrium-Independent Passive Short
EI-IPS	Equilibrium-Independent Input-Passive Short
EI-IOPS	Equilibrium-Independent Output-Passive Short
EI-OPS	Equilibrium-Independent Input- and Output-Passive Short
FDI	Fault Detection and Isolation
GNSS	Global Navigation Satellite System
HROPP	Hybrid Regularized Optimal Potential Problem
I/O	Input/Output
IP	Incremental Passivity
LTI	Linear and Time-Invariant
MEICMP	Maximally Equilibrium-Independent Cyclically Monotone Passive (or Passivity)
MEIP	Maximally Equilibrium-Independent Passive (or Passivity)
MIMO	Multiple-Input Multiple-Output
NROPP	Network-Regularized Optimal Potential Problem
ODE	Ordinary Differential Equation
OFP	Optimal Flow Problem
OPP	Optimal Potential Problem
PQI	Projective Quadratic Inequality
ROPP	Regularized Optimal Potential Problem
SISO	Single-Input Single-Output
TF	Transfer Function
UAV	Unmanned Aerial Vehicle

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

Chapter 1

Introduction

1.1 Introduction and Focus

The study of cooperative control and multi-agent networks has been in the pinnacle of control research for the last few years, exhibiting both a rich theoretical framework as well as a wide range of applications [83, 105, 107]. Examples include robotics [68], neural networks and neuroscience [18, 115, 127, 130], power grids [2, 157], traffic engineering [8], gene regulation [54, 74], communication networks [29, 112], physics [14, 156], ecology [94, 159], and even behavioral sciences [126, 177] and finance [87, 102].

In this venue, researchers have tried to establish a unified theory for networks of dynamical systems. Two recurring themes that appear in many theories include the study of graph theory, which emerged as an important tool in the modeling and analysis of these systems [92], and passivity theory [75, 133], as it brings a powerful framework to analyze the dynamic behavior of these interconnected systems. Passivity theory, a system-theoretic notion which is related to energy conservation, is a widespread tool to synthesize controllers for a wide range of control problems [4,10,78,93,147,151,160,166], including cyber-physical systems, energy systems and robotics. It enables an analysis of the networked system that decouples the dynamics of the agents in the ensemble, the structure of the information exchange network, and the protocols used to couple interacting agents [6, 7].

Passivity for multi-agent systems was first pursued in [5], where it was used to study group coordination problems. Several variants of passivity theory were used in various contexts like coordinated control of robotic systems [32, 62], synchronization problems [129, 148], port-Hamiltonian systems on graphs [162], distributed optimization [154], cyber-physical systems [4, 151], micro-grid control [40] and teleoperation of unmanned aerial vehicle (UAV) swarms [50].

One important variant of passivity particularly useful for the analysis of multi-agent systems is *equilibrium-independent passivity* (EIP), introduced in [64]. For EIP systems, passivity is verified with respect to any steady-state

© Technion - Israel Institute of Technology, Elyachar Central Library

input-output pair, allowing one to show convergence results without specifying the limit beforehand [144]. A generalization of this property, known as *maximal equilibrium-independent passivity* (MEIP), was introduced in [23] for single-input single-output (SISO) systems, allowing one to prove convergence using energy methods for a much wider class of systems, including nonlinear first-order integrators and other marginally-stable systems, which are not EIP.

The main result of [23] showed that the asymptotic behavior of these networked systems is *(inverse) optimal* with respect to a family of network optimization problems. In fact, the steady-state input-output signals of both the dynamical systems and the controllers comprising the networked system can be associated to the optimization variables of either an *optimal flow* or an *optimal potential* problem; these are the two canonical dual network optimization problems described in [122]. Thus, [23] built a network optimization framework for understanding problems in cooperative control and multi-agent systems, which can give network-level intuition for handling these problems.

Although promising, the network optimization framework suggested in [23] has several important drawbacks. First, it can only be applied to single-input-single-output systems, and not to multiple-input-multiple-output (MIMO) systems, as the definition of MEIP relied on the notion of monotonicity, which is not canonically extendable to multiple dimensions. Second, it requires the agents to be passive (or even output-strictly passive) with respect to all steady-state input-output pairs they possess, thus restricting the application of the framework for many important cases. Third, the paper [23] describes an analysis result. For such analysis to be practically useful, one must also develop synthesis procedures to design controllers for networked systems to achieve the desired coordination goals.

This thesis aims to extend the network optimization framework for cooperative control in order to cope with the challenges described above. There are a few main tools that are used throughout the thesis. The tools of convex analysis are repeatedly applied to move between steady-state equations for the multi-agent systems and the associated network optimization problems. Namely, strict convexity is used to verify uniqueness of the steady-state, subdifferential calculus is used to simplify nested optimization problems, and convex regularization terms are used to cope with lack of passivity. Graph theory and algebraic graph theory are also used throughout the thesis, as analyzing the underlying network structure is vital for understanding the behavior of a multi-agent system. Other tools are used in specific chapters of the thesis to solve specific problems. First, the notion of cyclic monotonicity, which was first introduced by Rockafellar in [119] is used to generalize the network optimization framework to MIMO agents. Second, the notion of system transformation and feedback passivation (or passification) is repeatedly used when dealing when passive-short agents. Third, the notions of randomization and absolute continuity are used together in some application examples, namely network detection and fault detection and isolation, as it can be proved that certain algorithms work "with almost any input," but constructing a deterministic input that is valid for the algorithm can be near-impossible and time-consuming. Other tools which are used more

© Technion - Israel Institute of Technology, Elyachar Central Library

sparingly include group theory, field theory, computational complexity theory, iterative control, Lyapunov stability, and matrix analysis.

Notation

First, we use some standard notation from set theory, as used in [44] . The cardinality (or size) of a set A is denoted by $|A|$. The set $\mathbb{R}$ denotes the real numbers, the set $\mathbb{Q}$ denotes the rational numbers, and the set $\mathbb{Z}$ denotes the integers.

Second, we use standard notation from linear algebra and matrix analysis [66]. The vector $\mathbf{0}_d$ denotes the d-dimensional zero vector. The vector $\mathbb{1}_d$ denotes the d-dimensional all-ones vector. In both cases, the subscript may be omitted when the dimension is clear. The vector e_i will denote the i-th standard basis vector, i.e., $\mathrm{e}_i \in \mathbb{R}^n$ and $(\mathrm{e}_i)_j = \delta_{ij}$, where δ_{ij} is Dirac's delta. The identity matrix of size $d \times d$ will be denoted Id_d. Given a linear map A, we will denote its kernel by $\ker(A)$ and its image as $\mathrm{Im}(A)$. If A is a map from a linear space to itself, then we denote its minimal singular value by $\underline{\sigma}(A)$, and its maximal singular value by $\overline{\sigma}(A)$. Similarly, if all of A's eigenvalues are real, we'll let $\lambda_{\min}(A)$ be the minimal eigenvalue of A, and $\lambda_{\max}(A)$ be the maximal value of A. For two matrices A, B, we'll let $A \otimes B$ denote the corresponding Kronecker product. If A, B are both square matrices of the same dimension, we'll write $A \geq B$ if $A - B$ is positive semi-definite. Moreover, we'll write $A > B$ if $A - B$ is positive-definite. If U is a linear subspace of $\mathbb{R}^n$, we'll let $U^{\perp}$ be its orthogonal complement, and Proj_U be the orthogonal projection on it .The Euclidean norm of a vector x will be denoted by $\|x\|$. Moreover, if X, Y are two sets inside the same vector space, we let $X + Y$ be their Minkowski sum, defined as $X + Y = \{x + y : \ x \in X, y \in Y\}$.

Next, we use some nomenclature from graph theory [15, 57]. A graph $\mathcal{G}$ is a pair $\mathcal{G} = (\mathbb{V}, \mathbb{E})$, where $\mathbb{V}$ is the set of nodes and $\mathbb{E}$ is the set of edges. Each edge $e \in \mathbb{E}$ consists of two vertices $i, j \in \mathbb{V}$, and will be oriented arbitrarily, say from i to j; we write $e = (i, j)$ in this case. If there is an edge between i and j, we'll write $i \sim j$. The incidence matrix $\mathcal{E}_{\mathcal{G}}$ of $\mathcal{G}$ is a $|\mathbb{V}| \times |\mathbb{E}|$ matrix such that for each edge $e = (i, j)$, $(\mathcal{E}_{\mathcal{G}})_{ie} = -1, (\mathcal{E}_{\mathcal{G}})_{je} = 1$ and all other entries in e's column are zero. In some cases, we'll omit the subscript and refer to the matrix as $\mathcal{E}$. Given some positive integer d, the incidence operator $\mathcal{E}_{\mathcal{G},d}$ is defined as the Kronecker product $\mathcal{E}_{\mathcal{G},d} = \mathcal{E}_{\mathcal{G}} \otimes \mathrm{Id}_d$. The Laplacian of the graph $\mathcal{G}$ is defined as the matrix $\mathcal{E}_{\mathcal{G}}\mathcal{E}_{\mathcal{G}}^{\top}$. This is a positive semi-definite matrix, and its second lowest eigenvalue is denoted as $\lambda_2(\mathcal{G})$, which is known as the algebraic connectivity of the graph $\mathcal{G}$.

We also use notation from analysis and convex analysis [17, 121]. For a smooth function F, we let ∇F be its gradient. We denote the collection of all function which are q times continuously differentiable by $\mathcal{C}^q$. If K is a convex function, we let ∂K denote its subgradient, and let $K^{\star}$ be its Legendre transform, defined as $K^{\star}(\mathrm{y}) = \sup_u \{y^{\top} u - K(u)\}$. Moreover, for a set $\mathcal{D}$ we let

$I_{\mathcal{D}}$ be the corresponding indicator function, i.e.,

$$I_{\mathcal{D}}(x) = \begin{cases} 0 & x \in \mathcal{D} \\ \infty & x \notin \mathcal{D} \end{cases}.$$

For the set $\mathcal{D} = \{c\}$, the corresponding indicator function will be denoted I_c.

Lastly, if Υ is some dynamical system and M is a linear map, we'll denote the cascade of M and Υ by $M\Upsilon$ or ΥM, where the rightmost operator is applied first.

We also go over several acronyms repeatedly used throughout the thesis. Namely, "ordinary differential equation" will be abbreviated as ODE, and "input/output" will be abbreviated as I/O. Single-input single-output systems will be called SISO systems, where multiple-input multiple-output systems will be called MIMO systems. Lastly, linear and time-invariant systems will be denoted as LTI systems, and their transfer function will be abbreviated as TF.

1.2 Background

In this section, we present the subject of network optimization, introduce the network dynamic model used throughout the thesis, and present an overview of the role of passivity in cooperative control.

1.2.1 Network Optimization

The field of network optimization is one of the gems of mathematics, lying at the intersection of two major subjects - namely graph theory and optimization theory. Network optimization deals with algorithmically finding optimal solutions to optimization problems defined on graphs, e.g. shortest path problems, maximal flow problems and routing problem [11, 122]. Network optimization is widely used in theoretical computer science, supply chain management, operations research and communication networks. This section presents the network optimization notions required for this thesis .

Consider a graph $\mathcal{G} = (\mathbb{V}, \mathbb{E})$ with an incidence matrix $\mathcal{E}$. One classic example of a network is an electrical network, in which the nodes are devices and the edges are wires. A flow on a network is a vector $\mu = [\mu_1, .., \mu_{|\mathbb{E}|}]^\top$, which can be thought of as a vector of electrical currents running through the edges. In a similar fashion, we look at $\mathrm{u} = [\mathrm{u}_1, ..., \mathrm{u}_{|\mathbb{V}|}]^\top$ as a divergence vector, which adds up the in/out flow through each node. In that setting, Kirchhoff's current law is represented by $\mathrm{u} + \mathcal{E}\mu = \mathbf{0}$. We can also think about $\mathrm{y} \in \mathbb{R}^{|\mathbb{V}|}$ as the potential of the network. To each edge $e = (i, j)$ one can associate a potential difference $\zeta_e = \mathrm{y}_j - \mathrm{y}_i$. The stacked potential difference vector $\zeta \in \mathbb{R}^{|\mathbb{E}|}$ can also be expressed by $\zeta = \mathcal{E}^\top \mathrm{y}$. These connections yield the *conversion formula*, relating all four variables by $\mathrm{y}^\top \mathrm{u} = -\zeta^\top \mu$.

We consider a few important optimization problems over networks that will each be given an interpretation in the cooperative control setting. The first

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

attempts to optimize the flow and divergence in a network, subject to a conservation of flow constraint. We present a flow tariff, giving a cost $C_e^{\text{flow}}(\mu_e)$ to a flow of volume μ_e, and a divergence tariff, giving a cost $C_i^{\text{div}}(\mathrm{u}_i)$ to each divergence u_i. In this case, one tries to minimize:

$$\begin{array}{ll} \min & \sum_{i\in\mathbb{V}} C_i^{\text{div}}(\mathrm{u}_i) + \sum_{e\in\mathbb{E}} C_e^{\text{flow}}(\mu_e) \\ \text{s.t.} & \mathrm{u} + \mathcal{E}\mu = \mathbf{0} \end{array},$$

which is known as the *optimal flow problem*. If the tariffs are convex functions, one can consider the dual problem, in the convex optimization sense. Indeed, if one defines tariffs for tension and potential using the Legendre transform, one obtains

$$\begin{aligned} C_i^{\text{pot}}(\mathrm{y}_i) = (C_i^{\text{div}})^\star(\mathrm{y}_i) = \min_{\mathrm{u}_i}\left(\mathrm{u}_i^\top \mathrm{y}_i - C_i^{\text{div}}(\mathrm{u}_i)\right); \\ C_e^{\text{ten}}(\zeta_e) = (C_e^{\text{flow}})^\star(\zeta_e) = \min_{\mu_e}\left(\mu_e^\top \zeta_e - C_e^{\text{flow}}(\mu_e)\right). \end{aligned}$$

This gives the dual problem, known as the *optimal potential problem*:

$$\begin{array}{ll} \min & \sum_{i\in\mathbb{V}} C_i^{\text{pot}}(\mathrm{y}_i) + \sum_{e\in\mathbb{E}} C_e^{\text{ten}}(\zeta_e) \\ \text{s.t.} & \mathcal{E}^\top \mathrm{y} = \zeta \end{array}.$$

Both of these problems are convex, and can be easily solved using gradient descent or other convex optimization techniques. See Appendix A for more details.

1.2.2 Diffusively Coupled Networks

We now present the network model used throughout this thesis, for which we will find a connection to network optimization. We consider a population of agents interacting over a network, described by the graph $\mathcal{G} = (\mathbb{V}, \mathbb{E})$. The agents are represented by the vertices $\mathbb{V}$, and pairs of interacting agents are represented by edges $\mathbb{E}$. Each specific edge contains information about the coupling (i.e., the controllers), which are allowed to be dynamic. We assume a *diffusive coupling* structure, where the inputs to the edge controllers are the differences between the outputs of the adjacent agents, and the control input of each agent is the (directed) sum of the edge controller outputs.

Each agent in the network is modeled as a multiple-input multiple-output (MIMO) dynamical system of the form

$$\Sigma_i : \begin{cases} \dot{x}_i(t) = f_i(x_i(t), u_i(t), \mathrm{w}_i), \\ y_i(t) = h_i(x_i(t), u_i(t), \mathrm{w}_i) \end{cases} \quad i \in \mathbb{V}, \tag{1.1}$$

where $x_i(t) \in \mathbb{R}^{p_i}$ is the state, $u_i(t) \in \mathbb{R}^d$ is the input, $y_i(t) \in \mathbb{R}^d$ is the output, and w_i is a constant exogenous input. Note that each agent need not

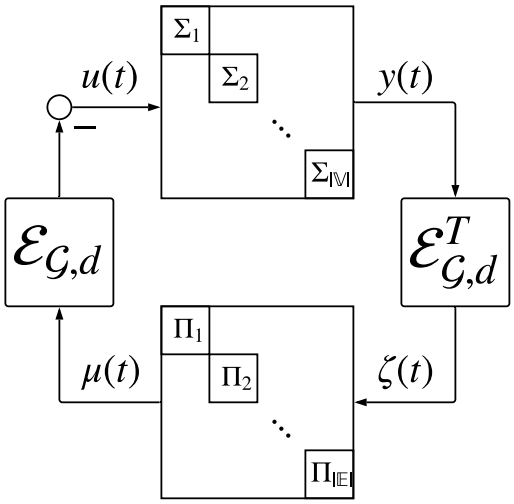


Figure 1.1: Block diagram of the diffusively-coupled network $(\mathcal{G}, \Sigma, \Pi)$.

have the same state dimension, but we require all agents have the same number of inputs and outputs, d. Let $u(t) = [u_1(t)^\top, \ldots, u_{|\mathbb{V}|}(t)^\top]^\top$ and $y(t) = [y_1(t)^\top, \ldots, y_{|\mathbb{V}|}(t)^\top]^\top$ be the concatenation of the input and output vectors. Similarly, $x(t) \in \mathbb{R}^{\sum_{i=1}^{|\mathbb{V}|} p_i}$ is the stacked state vector, and w the stacked exogenous input.

The agents are diffusively coupled over the network by dynamic systems that we consider as the network controllers. For the edge $e = (i, j)$, we denote the difference between the outputs of the adjacent nodes as $\zeta_e(t) = y_j(t) - y_i(t)$. The stacked vector $\zeta(t)$ can be compactly expressed using the incidence operator of the graph as $\zeta(t) = \mathcal{E}^\top_{\mathcal{G},d} y(t)$. These, in turn, drive the edge controllers described by the dynamics

$$\Pi_e : \begin{cases} \dot{\eta}_e(t) = \phi_e(\eta_e(t), \zeta_e(t)), \\ \mu_e(t) = \psi_e(\eta_e(t), \zeta_e(t)) \end{cases} \quad e \in \mathbb{E}. \tag{1.2}$$

These edge controllers regulate the relative output of the corresponding agents, and can be implemented either by the agents, or using a central server communicating with the agents, e.g. using cloud computing. The output of these controllers will yield an input to the nodal dynamical systems as $u(t) = -\mathcal{E}_{\mathcal{G},d}\mu(t)$, with $\mu(t)$ the stacked vector of controller outputs. We denote the complete network system by the triple $(\mathcal{G}, \Sigma, \Pi)$, where Σ and Π are the parallel interconnection of the agent and controller systems, and $\mathcal{G}$ is the underlying network; see Figure 1.1.

The diffusive coupling structure includes many types of networks, including the Kuramoto model [43], traffic models [8], and neural networks [65]. We

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

illustrate possible uses of diffusively-coupled networks in the following example.

Example 1.1. *Consider the gradient system* $\dot{x} = -\nabla F(x)$*, where*

$$F(x) = \sum_{(i,j)\in\mathbb{E}} F_{ij}(x_i - x_j),$$

and F_{ij} *are smooth* $\mathcal{C}^1$ *functions. This is an example of a diffusively coupled network, where the agents are single integrators* $\dot{x}_i = u_i,\ y_i = x_i$*, and the controllers are static nonlinearities* ∇F_{ij}*. This system drives the agents to minimize the function* $F(x)$*, and the agents converge to a local minimum of* F*.*

First, consider the consensus problem, in which we want the agents converge to a state in which $x_i = x_j,\ \forall i,j \in \mathbb{V}$ *[107]. It can be achieved using the described diffusively-coupled network by choosing* $F_{ij}(\zeta_e) = \frac{1}{2}\|\zeta_e\|^2$*. The functions* F_{ij} *are convex, meaning that the agents must converge to a global minimum, in which consensus is achieved.*

Second, consider the distance-based formation control problem, in which we want the agents converge to a state in which $\|x_i - x_j\| = d_{ij},\ \forall i,j \in \mathbb{V}$*, where the distances* d_{ij} *are given. It can be achieved using the described diffusively-coupled network by choosing* $F_{ij}(\zeta_e) = \frac{1}{4}(\|\zeta_e\|^2 - d_{ij}^2)^2$*. This is the distance-based formation control protocol described in [105].*

Lastly, suppose that the states x_i *are real numbers, and we wish to force the agents to have the same phase, i.e.,* $x_i = x_j \mod 2\pi$*. This can be achieved using the described diffusively-coupled network by choosing* $F_{ij}(\zeta_e) = 1 - \cos(\zeta_e)$*, which gives the Kuramoto model for the case in which oscillators revolve at the same velocity, i.e.,* $\omega_i = 0$ *[43].*

1.2.3 The Role of Passivity in Cooperative Control

Passivity theory has taken an outstanding role in the analysis of cooperative control systems, and in particular those with the diffusive coupling structure of Figure 1.1. We dedicate this section to consider a few variants of passivity used to prove various analysis results for multi-agent systems. For an introduction to passivity and its relation to stability, we refer the reader to Appendix C. The main advantage of using passivity theory is that it allows to decouple the system into three different layers - namely the agent dynamics, the coupling dynamics, and the network connecting the two. This concept is clearly seen in the following theorem:

Theorem 1.1 ([23])**.** *Consider the network system* $(\mathcal{G}, \Sigma, \Pi)$ *comprised of agents and controllers. Suppose that there are constant vectors* $\mathrm{u_i}, \mathrm{y_i}, \zeta_\mathrm{e}$ *and* μ_e *such that*

i) the systems Σ_i *are output strictly-passive with respect to* $\mathrm{u_i}$ *and* $\mathrm{y_i}$*;*

ii) the systems Π_e *are passive with respect to* ζ_e *and* μ_e*;*

iii) the stacked vectors $\mathrm{u}, \mathrm{y}, \zeta$ *and* μ *satisfy* $\mathrm{u} = -\mathcal{E}_{\mathcal{G},d}\mu$ *and* $\zeta = \mathcal{E}_{\mathcal{G},d}^\top \mathrm{y}$*;*

where $\mathrm{u} = [\mathrm{u}_1^\top, \ldots, \mathrm{u}_{|\mathbb{V}|}^\top]^\top$, $\mathrm{y} = [\mathrm{y}_1^\top, \ldots, \mathrm{y}_{|\mathbb{V}|}^\top]^\top$, $\zeta = [\zeta_1^\top, \ldots, \zeta_{|\mathbb{E}|}^\top]^\top$, *and* $\mu = [\mu_1^\top, \ldots, \mu_{|\mathbb{E}|}^\top]^\top$. *Then the output vector* $y(t)$ *converges to* y *as* $t \to \infty$.

The first condition involves the agent dynamics, the second the controllers, and the third the underlying network. We note that the version of the theorem proved in [23] deals with SISO systems, but the exact same argument also proves the result for MIMO systems.

The first paper to fully embrace passivity theory to analyze cooperative control problems was [5]. It led to many variants of passivity in the literature proven to be useful for the analysis of cooperative control problems. *Incremental passivity* (IP), introduced in [110], allows one to consider the passivity property with respect to certain trajectories, rather than fixed equilibria. Indeed, incremental passivity was used in [129, 148] to prove various synchronization and analysis results for multi-agent systems. However, IP is restrictive, as it demands the passivation inequality to hold for any two trajectories.

Other variants of passivity focus on the collection of all equilibria of a system. In this direction, the notion of steady-state input-output maps is useful. In the following, we focus on dynamical systems of the form

$$\Sigma : \begin{cases} \dot{x}(t) = f(x(t), u(t)) \\ y(t) = h(x(t), u(t)) \end{cases} . \tag{1.3}$$

Definition 1.1. *Consider the dynamical system* (1.3) *with input* $u \in \mathcal{U}$ *and output* $y \in \mathcal{Y}$. *The* steady-state input-output relation *associated with* (1.3) *is the set* $k \subset \mathcal{U} \times \mathcal{Y}$ *consisting of all steady-state input-output pairs* (u, y) *of the system.*

Remark 1.1. *Even if the relation* k *is not a function, we can always think of it as a* set-valued function. *Namely, for a steady-state input* u, *let* $k_y(\mathrm{u}) = \{\mathrm{y} : (\mathrm{u}, \mathrm{y}) \in k\}$, *and for a steady-state output* y, *let* $k^{-1}(\mathrm{y}) = \{\mathrm{u} : \ (\mathrm{u}, \mathrm{y}) \in k\}$.

Example 1.2. *We consider the following four SISO dynamical systems:*

$$\Sigma_1 : \begin{cases} \dot{x} = -x + u \\ y = x \end{cases}, \Sigma_2 : \begin{cases} \dot{x} = u \\ y = x \end{cases}$$

$$\Sigma_3 : \begin{cases} \dot{x} = -x + u \\ y = \tanh(x) \end{cases}, \Sigma_4 : \begin{cases} \dot{x} = -x^3 + u \\ y = x \end{cases}$$

We compute the steady-state input-output relation for each of the systems. For Σ_1, *in steady-state, we have* $\dot{x} = 0$, *so* $\mathrm{y} = \mathrm{x} = \mathrm{u}$. *Thus the steady-state input-output relation is* $k_1 = \{(\mathrm{u}, \mathrm{y}) : \ \mathrm{u} = \mathrm{y} \in \mathbb{R}\}$. *For* Σ_2, *in steady-state, we have* $\dot{x} = 0$, *so* $\mathrm{u} = 0$. *Moreover, the corresponding steady-state output* y *can take any value, depending on initial conditions. Thus the steady-state input-output relation is* $k_2 = \{(\mathrm{u}, \mathrm{y}) : \ \mathrm{u} = 0, \mathrm{y} \in \mathbb{R}\}$. *For* Σ_3, *in steady-state, we have* $\dot{x} = 0$, *so* $\mathrm{y} = \tanh(\mathrm{x}) = \tanh(\mathrm{u})$. *Thus the steady-state input-output relation is* $k_3 = \{(\mathrm{u}, \mathrm{y}) : \ \mathrm{y} = \tanh(\mathrm{u}) \in \mathbb{R}\}$. *Lastly, for* Σ_4, *in steady-state,*

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

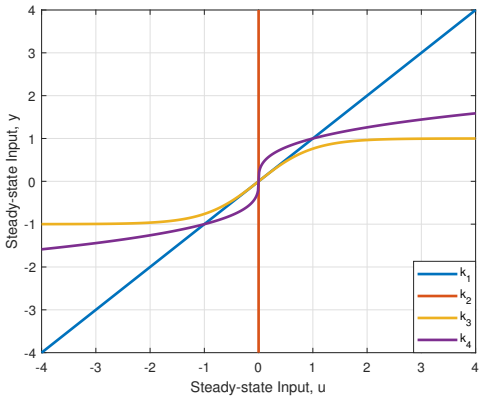


Figure 1.2: The steady-state input-output relations considered in Example 1.2.

we have $\dot{x} = 0$*, so* $y = x = \sqrt[3]{u}$*. Thus the steady-state input-output relation is* $k_4 = \{(u, y) : \ y = \sqrt[3]{u} \in \mathbb{R}\}$*. The relations* k_1, k_2, k_3 *and* k_4 *can be seen in Figure 1.2*

With this definition, we now introduce the next variant of passivity termed *equilibrium-independent passivity* (EIP) [64]. A key feature of EIP is the assumption that for any steady-state input u there is exactly one steady-state output y. This implies that the steady-state output y can be expressed as a function of the steady-state input u, which is assumed to be continuous. Thus, with a slight abuse of notation we can consider the the relation k_y as a *function* $k_y : u \mapsto y$, i.e., $y = k_y(u)$. In general, this is less restrictive than IP, and allows to prove analysis results for MIMO systems. However, there are IP systems which are not EIP. The epitome of these kind of systems is the single integrator, which can be verified to be IP, but not EIP. The steady-state input $u = \mathbf{0}$ has multiple different steady-state outputs (depending on the initial condition of the system), and thus the input-output relation is no longer a function.

The last variant of passivity we review is *maximal equilibrium-independent passivity* (MEIP) [23]. It is a variant of EIP that attempts to remedy the exclusiveness of the single integrator and similar systems. However, it is only defined in the case of SISO systems, as it relies on the notion of monotone relations:

Definition 1.2 ([23])**.** *Consider a relation* $R \subseteq \mathbb{R} \times \mathbb{R}$*. We say that* R *is a* monotone relation *if for every two elements* (u_1, y_1) *and* (u_2, y_2)*, we have that* $(u_2 - u_1)(y_2 - y_1) \geq 0$*. We say that* R *is* maximally monotone *if it is monotone and is not contained in a larger monotone relation.*

In other words, increasing the first component u implies that the second component y cannot decrease. We now present the definition of MEIP.

Definition 1.3 ([23])**.** *The SISO system* (1.3) *is said to be (output-strictly)* maximal equilibrium-independent passive *(MEIP) if:*

i) The system is (output-strictly) passive with respect to any steady-state input-output pair (u, y) *it has.*

ii) The collection k_y *of all steady-state input-output pairs* (u, y) *is maximally monotone.*

This is indeed a generalization of EIP, as the function k_y of an EIP system is monotone ascending [64]. It can also be shown that the single SISO integrator is MEIP. However, the problem of finding a MIMO analogue of MEIP, or a variant of EIP that will include marginally-stable systems like the single integrator, has not been addressed in the literature.

The main result of [23] showed that a diffusively-coupled system $(\mathcal{G}, \Sigma, \Pi)$ in which the agents are output-strictly MEIP and the controllers are MEIP must converge to a closed-loop steady-state, which is the minimizer of two dual convex instances of the optimal potential problem and the optimal flow problem. This is a generalization of Theorem 1.1 that allows one to check for convergence of a multi-agent system without specifying an a priori limit. Thus, one can compute the steady-state limit of $(\mathcal{G}, \Sigma, \Pi)$ with relative ease.

More precisely, it is known that any maximal monotone relation in $\mathbb{R}^2$ is equal to the subgradient of some convex function $\mathbb{R} \to \mathbb{R}$, which is unique up to an additive constant [121]. Thus, if we let k_i and γ_e be the agents' and controllers' steady-state input-output relations, then there exist convex functions K_i, Γ_e such that $\partial K_i = k_i$ and $\partial \Gamma_e = \gamma_e$. The functions K_i, Γ_e are said to be the *integral functions* of k_i and γ_e, respectively. We also define the convex functions $K(u) = \sum_{i \in \mathbb{V}} K_i(u_i)$ and $\Gamma(\zeta) = \sum_{e \in \mathbb{E}} \Gamma_e(\zeta_e)$, and let $K^\star, \Gamma^\star$ be their Legendre duals. It is straight forward to check that $\partial K = k$ and $\partial \Gamma = \gamma$, where k and γ are the stacked relations achieved from concatenating k_i and γ_e, respectively. The following theorem is proved in [23]:

Theorem 1.2. *Consider a diffusively-coupled system* $(\mathcal{G}, \Sigma, \Pi)$*, and assume that the agents are output-strictly MEIP and that the controllers are MEIP. Let* K *and* Γ *be the sum of the integral functions for the agents and for the controllers, respectively. Then the signals* $u(t), y(t), \zeta(t), \mu(t)$ *converge to steady-states* $\hat{\mathrm{u}}, \hat{\mathrm{y}}, \hat{\zeta}, \hat{\mu}$*, which are optimal primal-dual solutions of the following pair of network optimization problems:*

Optimal Potential Problem (OPP)		**Optimal Flow Problem** (OFP)	
$\min\limits_{\mathrm{y},\zeta}$	$K^\star(\mathrm{y}) + \Gamma(\zeta)$	$\min\limits_{\mathrm{u},\mu}$	$K(\mathrm{u}) + \Gamma^\star(\mu)$
s.t.	$\mathcal{E}^\top \mathrm{y} = \zeta$	s.t.	$\mathrm{u} = -\mathcal{E}\mu.$

It should be noted that the network optimization framework allows for more than just finding the steady-state of a diffusively-coupled system, as it gives "network-level intuition" for problems in multi-agent systems. One example of this network-level intuition can be seen below:

Example 1.3. *Consider a collection of output-strictly MEIP agents interacting over a graph* $\mathcal{G} = (\mathbb{V}, \mathbb{E})$*. We fix arbitrary vectors* $\mathrm{v}_1, ..., \mathrm{v}_{|\mathbb{E}|}$*, and consider the*

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

single integrator controllers with these biases, i.e., the controller on the edge $e \in \mathbb{E}$ *is given by* $\dot{\eta}_e = \zeta_e - \mathrm{v}_e,\ \mu_e = \eta_e$. *We ask ourselves whether or not the closed-loop diffusely-coupled network* $(\mathcal{G}, \Sigma, \Pi)$ *will converge to a steady-state. Intuitively, one could guess that the closed-loop system converges to a steady-state if and only if the agents have a steady-state output* y *such that* $\mathrm{y}_i - \mathrm{y}_j = \mathrm{v}_e$ *for all* $e = (i, j) \in \mathbb{E}$. *To prove this claim classically, one needs to consider a Lyapunov function which is the sum of the storage functions for the agents and the controllers, prove that it is a Lyapunov function, and then invoke LaSalle's invariance principle, which is quite cumbersome as we do not know the steady-state output in advance. Alternatively, one could use the network optimization framework to prove the claim with ease. Indeed, the integral function of the controllers is easily computed as* $\Gamma_e(\zeta_e) = I_{\mathrm{v}_e}(\zeta_e)$. *Thus (OPP) is just the minimization problem of* $K^\star(\mathrm{y}) + I_{\mathrm{v}}(\zeta)$, *where* $\zeta = \mathcal{E}_{\mathcal{G}}^\top \mathrm{y}$ *and* $\mathrm{v} = [\mathrm{v}_1^\top, ..., \mathrm{v}_{|\mathbb{E}|}^\top]^\top$ *is the stacked bias vector. To have a non-infinite value, we must have* $\zeta = \mathrm{v}$, *so we demand that* $\mathrm{v} \in \mathrm{Im}(\mathcal{E}_{\mathcal{G}}^\top)$. *Moreover, if we restrict the variable* y *by demanding that* $\mathcal{E}_{\mathcal{G}}^\top \mathrm{y} = \zeta$, *then (OPP) has a non-infinite value if and only if* $K^\star(\mathrm{y})$ *is finite, i.e., the agents have the steady-state output* y. *Thus the closed-loop system* $(\mathcal{G}, \Sigma, \Pi)$ *converges if and only if there's a steady-state output* y *such that* $\mathcal{E}_{\mathcal{G}}^\top \mathrm{y} = \mathrm{v}$, *which is exactly as conjectured.*

1.3 Contributions and Thesis Outline

The following overview presents the outline of this thesis and briefly summarizes its contributions. The first two chapters deal with theoretical extensions to the network optimization framework, while the following chapters deal with various applications of the framework to problems in multi-agent systems.

Chapter 2 - A Network Optimization Framework for MIMO Systems

In this chapter we introduce a generalization of the network optimization framework of [23] to diffusively-coupled systems with MIMO agents and controllers. This is done using the notion of *cyclically monotone relations*. Namely,

- We use the notion of cyclically monotone relations to define *maximally equilibrium-independent cyclically-monotone passive* systems (MEICMP), which is a generalization of MEIP to MIMO systems.
- We derive a network optimization framework for diffusively-coupled systems with output-strictly MEICMP agents and MEICMP controllers, generalizing the network optimization framework of [23], using a different approach than the one presented therein.
- We explore the notion of MEICMP systems. Namely, we show that it holds for a large class of gradient systems with oscillatory terms, as well as classifying which linear systems possessing this property.

Chapter 3 - A Network Optimization Framework for Passive-Short Agents

In this chapter we introduce a generalization of the network optimization framework to passive-short systems. It is shown that regularization of the network optimization problems corresponds to passivation of the non-passive agents, validating the network optimization framework. Namely,

- It is shown that the original network optimization framework of [23] fails for passive-short agents, and three main causes for the failure are identified.
- It is shown that for output-passive short agents, the network optimization problem can still be defined, but it is no longer convex. It is also shown that convexifying the optimization problem corresponds to passivizing the multi-agent system, where the regularizing term gives rise to a corresponding feedback term. Three different regularization approaches are presented.
- For the case of general passive-short agents, the network optimization problem might no longer be defined. The convexification technique is replaced by a monotonization approach, which is shown to also passivize the agents, rendering the network optimization framework valid in this case as well.

Chapter 4 - A Network Optimization Framework for Controller Synthesis

In this chapter, we introduce a first possible application of the analysis result of the framework, studying the final-value synthesis problem. In this problem, we are given agents and the underlying interaction graph, and wish to design edge controllers to force the closed loop diffusively-coupled system to some desired steady-state. We will also consider a problem in which only the agents are given, and we can also design the underlying interaction graph.

- It is shown that the general final-value synthesis problem can be solved for any desired steady-state and for any underlying interaction graph, and an efficient algorithm for its solution is presented. Moreover, it is shown that given any collection of MEIP controllers, one can slightly augment them to achieve a solution to the final-value synthesis problem.
- The special case of clustering is studied, where it is shown that steady-state clusters can be understood using symmetries of the multi-agent system through the notion of exchangeability.
- The problem of cluster synthesis, i.e., forcing the system to cluster with prescribed cluster sizes at prescribed locations, is studied. We focus on the case of homogeneous networks, in which the agents are identical to

© Technion - Israel Institute of Technology, Elyachar Central Library

one another and the controllers are also required to be identical. We show that graphs forcing the agents to a desired clustering structure exist, and give bounds on the number of edges needed for their construction.

Chapter 5 - Applications of the Network Optimization Framework in Data-Driven Control

In this chapter, we show that the network optimization framework can be used to derive data-driven control algorithms for multi-agent systems. Namely, we consider the problem of final-value synthesis for given relative outputs from the previous chapter, but now assume that exact models for the agents are unknown. We show how measured data can be used to derive a data-driven solution to the problem, without estimating a model for the agents. Our approach is through amplification, cascading fixed edge controllers with adjustable positive gains.

- We first discuss methods for determining passivity and MEIP without exact models. We present known results for verifying passivity using data, and show that even an obscure model can be enough to prove MEIP.
- It is shown that for a vast class of controllers, a solution for the final-value synthesis problem is achieved for large enough gains. This is done by essentially recasting one of the network optimization problems as a robust optimization problem. It is then shown that for large gains, the closed-loop system behaves similarly to an augmented diffusively-coupled system, where the agents are replaced by single integrators.
- Two data-driven approaches for choosing the gains are studied. The first uses experiments on each agent to calculate a uniform gain on the edges, while the second is an iterative scheme augmenting the gains in-run using data from the closed-loop system. Convergence and stability guarantees are presented for both approaches. The two approaches are compared for two case-studies.

Chapter 6 - Applications of the Network Optimization Framework in Network Identification

In this chapter, we study the problem of network identification. In this problem, we are given a diffusively-coupled system $(\mathcal{G}, \Sigma, \Pi)$, with known agents and controllers, and are required to compute the graph $\mathcal{G}$. We also discuss the related problem of network differentiation, where it is required to differentiate between two networked systems having the same agents and controllers, but different underlying graphs

- We discuss the notion of indication vectors for multi-agent systems, which are constant exogenous inputs forcing systems with different underlying graphs to different steady-state outputs. We show different ways of constructing them, using randomization and algebraic methods.

© Technion - Israel Institute of Technology, Elyachar Central Library

- A network identification algorithm for networks with MEIP agents and controllers is suggested, with a polynomial time complexity in the number of agents. It is first built for networks of LTI agents and controllers, where the connection between exogenous inputs and steady-state output is given by a matrix, whose off-diagonal entries correspond to edges in the underlying graph. We later generalize it to general MEIP agents and controllers using linearization, and bounds on the error are derived.

- We use complexity theory methods to derive a lower bound on the time complexity of any algorithm solving the network identification problem with probability $p > 0$ and a bounded error, showing that the developed algorithm is optimal in sense of time complexity.

Chapter 7 - Applications of the Network Optimization Framework in Fault Detection and Isolation

In this chapter, we study the problem of network fault detection and isolation. In this problem, one must achieve some control goal (in this case, final-value synthesis), while faults may occur throughout the network. We study the case of network faults, in which the underlying graph changes, which can happen due to malfunctioning communication systems or a cyber attack on the agents.

- We first define the notion of edge-indication vectors, which are close relatives of the indication vectors from Chapter 6. We show that they can be found using randomization, while also exhibiting their ability to provide a solution to the synthesis problem

- We show how these edge-indication vectors can be applied to solve the fault detection and fault isolation problems, while also providing a solution to an adversarial game. The solutions are developed under the assumption of the existence of a "convergence assertion protocol", which checks whether a diffusively-coupled system converges to a conjectured steady-state.

- Finally, We use the passivity of the agents and controllers to show how convergence assertion protocols can be built. Two methods are presented, one relying on high-rate sampling and the other relying on a numerical connection between the output y_i and the storage function S_i of each of the agents.

Chapter 8 - Conclusions and Outlook

This final chapter provides some conclusive remarks, both summarizing the thesis and hinting at possible future directions of research.

Supplementary material is provided in several appendices, referenced at appropriate places, with the aim to make this thesis as self-contained as possible.

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

1.4 Publications

The research presented in this thesis has been presented in the following publications

- M. Sharf, and D. Zelazo, **"Network Optimization Approach to Cooperative Control Synthesis"**, IEEE Control Systems Letters, 1(1): 86-91, 2017
- M. Sharf, and D. Zelazo, **"Analysis and Synthesis of MIMO Multi-Agent Systems Using Network Optimization"**, IEEE IEEE Transactions on Automatic Control, vol. 64, no. 11, pp. 4512-4524, Nov. 2019.
- A. Jain, M. Sharf, and D. Zelazo, **"Regularization and Feedback Passivation in Cooperative Control of Passivity-Short Systems: A Network Optimization Perspective"**, IEEE Control Systems Letters, 2(4):731-736, 2018
- M. Sharf, and D. Zelazo, **"Network Identification: A Passivity and Network Optimization Approach"**, Proc. 57th IEEE Conference on Decision and Control, pp. 2107-2113 (Miami Beach, FL, USA, December 2018)
- M. Sharf and D. Zelazo, **"Network Feedback Passivation of Passivity-Short Multi-Agent Systems"**, IEEE Control Systems Letters, 3(3):607-612, 2019
- M. Sharf, and D. Zelazo, **"Symmetry-Induced Clustering in Multi-Agent Systems using Network Optimization and Passivity"**, Proc. 27th Mediterranean Conference on Control and Automation (MED), pp. 19-24 (Akko, Israel, July 2019)

Some material in this thesis is also contained in the following manuscripts, submitted for review:

- M. Sharf, A. Jain, and D. Zelazo, **"A Geometric Method for Passivation and Cooperative Control of Equilibrium-Independent Passivity-Short Systems"**, submitted to IEEE Transactions on Automatic Control (arXiv pre-print available online).
- M. Sharf, and D. Zelazo, **"Network Identification for Diffusively-Coupled Systems with Minimal Time Complexity"**, submitted to IEEE Transactions on the Control of Network Systems (arXiv pre-print online).
- M. Sharf, A. Koch, D. Zelazo and F. Allgöwer, **"Model-Free Practical Cooperative Control for Diffusively Coupled Systems"**, submitted to IEEE Transactions on Automatic Control (arXiv pre-print available online).

- M. Sharf, and D. Zelazo, **"A Data-Driven and Model-Based Approach to Fault Detection and Isolation in Networked Systems"**, submitted to IEEE Transactions on Automatic Control (arXiv pre-print available online).

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

Chapter 2

A Network Optimization Framework for MIMO Systems

This section is a review of [138]. We present a variant of passivity generalizing MEIP for MIMO systems, culminating in a network optimization framework for MIMO multi-agent systems. This generalization will allow us to extend the applications presented later from SISO to MIMO systems, provided the passivity requirements hold. Namely, this generalization will allow us to generalize the results of Section 3.3 and of Chapters 4, 6 and 7 to MIMO systems, with little to no effort required.

2.1 Introduction

As we saw in Section 1.2, the notion of EIP is defined for multiple-input multiple-output (MIMO) systems, but does not apply to the single integrator, even though it is passive with respect to any steady-state it has. EIP does not hold also for other marginally-stable systems, e.g. nonlinear integrators. Moreover, the results of [64] prove that a network of EIP systems is stable only for a positive linear feedback law of the form $u = (K \otimes \mathrm{Id}_m)y$, and they do not suggest a method of computing the closed-loop steady-state without running the closed-loop system.

The notion of MEIP tries to generalize EIP. It holds for some marginally stable systems such as the single integrator and other nonlinear integrators. The results of [23] proves that a network of output-strictly MEIP systems is stable under a feedback of general MEIP controllers, and the network optimization framework provides a way to compute the steady-state of the network by solving the corresponding network optimization problems, either analytically or using any appropriate numerical algorithm, such as gradient descent.

© Technion - Israel Institute of Technology, Elyachar Central Library

However, MEIP, as presented in [23], is only defined for SISO agents, meaning that we cannot apply it for general MIMO agents. The main reason that MEIP is undefined for MIMO systems is that it requires that the steady-state relations to be *monotone* (see Definition 1.2). The notion of monotonicity has no clear generalization for MIMO relations (or multi-variate functions), meaning that generalizing MEIP to MIMO systems is non-trivial.

The main analytic tool required to study MIMO systems in this context is the notion of *cyclically monotone* (CM) relations, first defined in [119]. The key result due to [119] shows that CM relations are contained in the sub-gradient of a convex function. These provide the "correct" generalization of monotonicity from scalar functions to multi-variate functions, allowing a complete generalization of the results of [23] to square MIMO systems.

For the rest of this chapter, we consider a diffusively-couple network $(\mathcal{G}, \Sigma, \Pi)$, with agents Σ_i and edge controllers Π_e governed by the equations:

$$\Sigma_i : \begin{cases} \dot{x}_i(t) = f_i(x_i(t), u_i(t), \mathrm{w}_i), \\ y_i(t) = h_i(x_i(t), u_i(t), \mathrm{w}_i) \end{cases} \quad , \quad \Pi_e : \begin{cases} \dot{\eta}_e(t) = \phi_e(\eta_e(t), \zeta_e(t)), \\ \mu_e(t) = \psi_e(\eta_e(t), \zeta_e(t)) \end{cases}$$

where the dimensions of the input u_i, output y_i, controller input ζ_e and controller output μ_e are all equal to d. The agents and edge controllers are coupled by the equations $\zeta(t) = \mathcal{E}_{\mathcal{G},d} y(t) = (\mathcal{E} \otimes \mathrm{Id}_d) y(t)$ and $u(t) + \mathcal{E}_{\mathcal{G},d}\mu(t) = u(t) + (\mathcal{E} \otimes \mathrm{Id}_d)\mu(t)$. The chapter is composed of two sections. The first studies the notion of CM relations, and links it to systems theory by defining the notion of *maximal equilibrium-independent cyclically monotone passive* (MEICMP) systems, which are the MIMO generalization of MEIP systems. It then proves that a diffusively-coupled network comprised of (MIMO) agents that are (output-strictly) MEICMP with (MIMO) controllers that are also MEICMP converges to a steady-state. Moreover, we show that the steady-states of the system are the optimal solutions of a pair of dual network optimization problems. The second section gives examples of systems with cyclically monotone input-output relations, and presents a case study.

2.2 Cyclically Monotone Relations and Cooperative Control

In [23], the concept of monotone relations is used to provide convergence results for a networked system $(\mathcal{G}, \Sigma, \Pi)$ comprised of SISO agents. However, many applications deal with MIMO systems, necessitating a need to extend this work for network systems consisting of MIMO agents. We consider a MIMO multi-agent network $(\mathcal{G}, \Sigma, \Pi)$, with each agent having an input and an output of dimension d. We begin by considering the steady-states of the system.

© Technion - Israel Institute of Technology, Elyachar Central Library

2.2.1 Steady-States and Network Consistency

Consider a steady-state $(\mathrm{u}, \mathrm{y}, \zeta, \mu)$ of the closed loop system $(\mathcal{G}, \Sigma, \Pi)$, presented in Figure 1.1. We know that for every $i = 1, ..., |\mathbb{V}|$, $(\mathrm{u}_i, \mathrm{y}_i)$ is a steady-state input-output pair of the i-th agent Σ_i. Similarly, for every $e \in \mathbb{E}$, (ζ_e, μ_e) is a steady-state input-output pair of the e-th controller Π_e. The network interconnection between the systems Σ and Π imposes an additional consistency constraint between these steady-state values. This motivates us to consider the steady-state input-output relations for each of the agents and the controllers.

In this direction, we denote the steady-state input-output relation of the i-th agent by k_i, and the relation for the e-th controller by γ_e. That is, $k_i \subset \mathbb{R}^d \times \mathbb{R}^d$ and $\gamma_e \subset \mathbb{R}^d \times \mathbb{R}^d$. We denote the stacked relation for the agents and controllers as k and γ, respectively. As in Remark 1.1, we can consider these input-output relations as set-valued functions. The consistency constraints for the steady-state of the closed-loop system can be written as follows:

Proposition 2.1. *Let* $\mathrm{u} \in \mathbb{R}^{d|\mathbb{V}|}, \mathrm{y} \in \mathbb{R}^{d|\mathbb{V}|}, \zeta \in \mathbb{R}^{d|\mathbb{E}|}, \mu \in \mathbb{R}^{d|\mathbb{E}|}$ *be any four constant vectors. Then* $(\mathrm{u}, \mathrm{y}, \zeta, \mu)$ *is a steady-state of the closed-loop system* $(\mathcal{G}, \Sigma, \Pi)$ *if and only if*

$$\begin{aligned} (\mathrm{u}, \mathrm{y}) \in k, \quad (\zeta, \mu) \in \gamma, \\ \zeta = \mathcal{E}_{\mathcal{G},d}^{\top}\mathrm{y}, \quad \mathrm{u} = -\mathcal{E}_{\mathcal{G},d}\mu. \end{aligned} \tag{2.1}$$

Proof. Follows directly from the interconnection of the network, and from the definitions of k and γ. □

We wish to manipulate the conditions in (2.1) to reduce the steady-state characterization from a system with four constraints to one. This can be done by trying to compute u from y in two different methods. First, we "go around the loop", taking the direct path of the block diagram in Figure 1.1. Second, we can take the "inverse" route and use the inverse relation k^{-1}. Similarly, we can compute ζ from μ in two different methods, going around the loop or using the inverse relation γ^{-1}. This idea can be seen in Figure 2.1. We use this intuition to prove the following proposition.

Proposition 2.2. *Let* $\mathrm{y} \in \mathbb{R}^{d|\mathbb{V}|}$ *be any vector. Then the following conditions are equivalent:*

i) The zero vector $\mathbf{0}$ *belongs to the set* $k^{-1}(\mathrm{y}) + \mathcal{E}_{\mathcal{G},d}\gamma(\mathcal{E}_{\mathcal{G},d}^{\top}\mathrm{y})$.

ii) There exists vectors u, ζ *and* μ *such that* $(\mathrm{u}, \mathrm{y}, \zeta, \mu)$ *is a steady-state of the closed-loop network* $(\mathcal{G}, \Sigma, \Pi)$.

Proof. First, assume the existence of u, ζ and μ. By Proposition 2.1, it follows that $\mathrm{u} \in k^{-1}(\mathrm{y})$, $\zeta = \mathcal{E}_{\mathcal{G},d}^{\top}\mathrm{y}$, $\mu \in \gamma(\zeta)$, and $\mathrm{u} = -\mathcal{E}_{\mathcal{G},d}\mu$. Thus,

$$\mathbf{0} = \mathrm{u} + \mathcal{E}_{\mathcal{G},d}\mu \in k^{-1}(\mathrm{y}) + \mathcal{E}_{\mathcal{G},d}\gamma(\zeta) = k^{-1}(\mathrm{y}) + \mathcal{E}_{\mathcal{G},d}\gamma(\mathcal{E}_{\mathcal{G},d}^{\top}\mathrm{y}).$$

Conversely, if $\mathbf{0} \in k^{-1}(\mathrm{y}) + \mathcal{E}_{\mathcal{G},d}\gamma(\mathcal{E}_{\mathcal{G},d}^{\top}\mathrm{y})$, then we know that there are some $\mathrm{u} \in k^{-1}(\mathrm{y})$ and $\mu \in \gamma(\mathcal{E}_{\mathcal{G},d}^{\top}\mathrm{y})$ such that $\mathrm{u} + \mathcal{E}_{\mathcal{G},d}\mu = \mathbf{0}$. Thus, by Proposition 2.1, the 4-tuple $(\mathrm{u}, \mathrm{y}, \zeta = \mathcal{E}_{\mathcal{G},d}^{\top}\mathrm{y}, \mu)$ is a steady-state of the closed-loop system. □

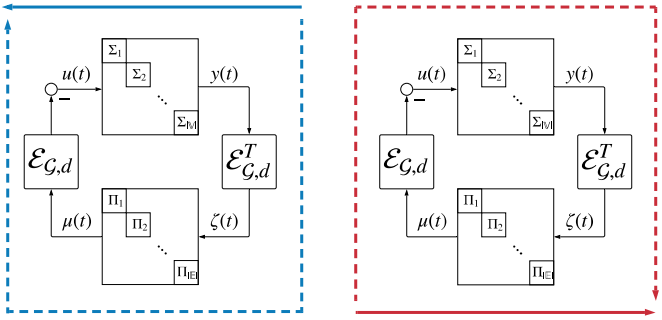


Figure 2.1: Converting Proposition 2.1 into one condition. Going around the loop in the direct path, and taking the "inverse" path using the inverse relation, must give the same result.

By the same methods, we can also reduce the conditions (2.1) to an inclusion in the edge-variables μ.

Proposition 2.3. *Let $\mu \in \mathbb{R}^{d|\mathbb{E}|}$ be any vector. Then the following conditions are equivalent:*

i) The zero vector $\mathbf{0}$ belongs to the set $\gamma^{-1}(\mu) - \mathcal{E}_{\mathcal{G},d}^{\top} k(-\mathcal{E}_{\mathcal{G},d}\mu)$.

ii) There exists vectors u, y and ζ such that (u, y, ζ, μ) is a steady-state of the closed-loop network $(\mathcal{G}, \Sigma, \Pi)$.

Proof. Same as the proof of Proposition 2.2. □

2.2.2 Connecting Steady-States to Network Optimization

So far, we showed that the steady-states of the closed-loop system can be understood using the following two conditions:

$$\begin{cases} \mathbf{0} \in k^{-1}(y) + \mathcal{E}_{\mathcal{G},d}\gamma(\mathcal{E}_{\mathcal{G},d}^{\top} y) \\ \mathbf{0} \in \gamma^{-1}(\mu) - \mathcal{E}_{\mathcal{G},d}^{\top} k(-\mathcal{E}_{\mathcal{G},d}\mu). \end{cases} \tag{2.2}$$

These conditions highlight the connection between agents, the controllers, the underlying network structure, and their impact on the steady-states of the closed-loop system. However, these conditions are highly nonlinear, and would be difficult to solve even if they were equations instead of inclusions. One method of dealing with nonlinear equations of the form $g(x) = \mathbf{0}$ for some function g, is to consider its *integral function* instead. Suppose there is a function G such that $g = \nabla G$. In that case, we can find a solution to $g(x) = \mathbf{0}$ by solving the unconstrained minimization problem, $\min_x G(x)$. If, in addition, the function G is convex, the solution to the minimization problem can often be computed efficiently (i.e., in polynomial time), e.g. by using gradient descent methods.

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

In general, convex functions need not be smooth, or even differentiable. In this case, the notion of the *subdifferential* of a convex function can be employed. The subdifferential of the convex function G at the point x is denoted $\partial G(x)$, and consists of all vectors v such that

$$G(y) \geq G(x) + v^\top (y - x), \ \forall y.$$

See [121] and Appendix A for more on subdifferentials. Note that the subdifferential ∂G is a set-valued map. Also, analogously to the differentiable case, x is a minimum point of G if and only if $\mathbf{0} \in \partial G(x)$. Thus, if we are able to require that k_i and γ_e are gradients of convex functions (i.e., their integral functions are convex), then the nonlinear inclusions in (2.2) may be solved using convex optimization. In fact, such relations have been characterized due to Rockafellar [119]:

Definition 2.1 (Cyclic Monotonicity)**.** *Let $d \geq 1$ be an integer, and consider a subset R of $\mathbb{R}^d \times \mathbb{R}^d$. We say that R is a* cyclically monotone *(CM) relation if for any $N \geq 1$ and any pairs $(\mathrm{u}_1, \mathrm{y}_1), \ldots, (\mathrm{u}_\mathrm{N}, \mathrm{y}_\mathrm{N}) \in R$ of d-vectors, the following inequality holds,*

$$\sum_{i=1}^{N} \mathrm{y}_i^\top (\mathrm{u}_i - \mathrm{u}_{i-1}) \geq 0. \tag{2.3}$$

Here, we use the convention that $\mathrm{u}_0 = \mathrm{u}_N$. We say that R is strictly cyclically monotonic *(SCM) if the inequality (2.3) is strict whenever at least two u_i-s are distinct. We term the relation as* maximal *CM (or maximal SCM) if it is not strictly contained in a larger CM (SCM) relation.*

Remark 2.1. *This is a generalization of the concept of monotone relations for SISO systems. We note that for all dimensions d, cyclic monotonicity implies monotonicity. Indeed, taking $N = 2$, the inequality* (2.3) *can be written as:*

$$0 \leq \mathrm{y}_1^\top (\mathrm{u}_1 - \mathrm{u}_2) + \mathrm{y}_2^\top (\mathrm{u}_2 - \mathrm{u}_1) = (\mathrm{y}_2 - \mathrm{y}_1)^\top (\mathrm{u}_2 - \mathrm{u}_1),$$

which holds for all pairs $(\mathrm{u}_1, \mathrm{y}_1), (\mathrm{u}_2, \mathrm{y}_2)$, coinciding with the definition of a monotone relation [121].

We now present Rockafellar's result establishing the connection between cyclic monotonicity and convex functions.

Theorem 2.1 ([119])**.** *A relation $R \subset \mathbb{R}^n \times \mathbb{R}^n$ is the subgradient of a convex function if and only if it is maximal CM. Moreover, it is the sub-gradient of a strictly convex function if and only if it is maximal SCM. The convex function is unique up to an additive scalar.*

Remark 2.2. *If R is maximally CM, and f is a convex function such that $R = \partial f$, then f is the* integral function *of R.*

© Technion - Israel Institute of Technology, Elyachar Central Library

Remark 2.3. *If R is a maximally CM relation, an integral function f can be found as [119]:*

$$f(\mathrm{u}) = \sup\left\{ \sum_{i=0}^{m} \mathrm{y}_i^\top (\mathrm{u}_{i+1} - \mathrm{u}_i) \right\}, \tag{2.4}$$

with the convention that $\mathrm{u}_{m+1} = \mathrm{u}_0$, and the supremum is taken over all integers $m \geq 0$ and all pairs $(\mathrm{u}_0, \mathrm{y}_0), ..., (\mathrm{u}_\mathrm{m}, \mathrm{y}_\mathrm{m}) \in R$. Another method is to use an analogue of the path integral formula for a potential function, namely choosing some u_0 arbitrarily, and defining

$$f(\mathrm{u}) = \int_\gamma \sup\{\mathrm{y} \cdot d\mathbf{l} : \ (\mathrm{u}, \mathrm{y}) \in R\}, \tag{2.5}$$

where γ is a curve defined on $[a, b]$ connecting u_0 to u. Formally, the integral is defined as

$$f(\mathrm{u}) = \int_a^b \sup\{\mathrm{y} \cdot \gamma'(s) : \ (\mathrm{u}, \mathrm{y}) \in R\} ds. \tag{2.6}$$

Rockafellar's Theorem gives us a way to check that a relation is the subdifferential of a convex function. If we want to state the conditions in (2.2) as the solutions of convex minimization problems, we need to assume that the input-output relations appearing are CM. This, together with Theorem 1.1, motivates the following system-theoretic definition:

Definition 2.2. *A system Σ is* maximal equilibrium-independent cyclically monotone (output strictly) passive *(MEICMP) if*

i) for every steady-state input-output pair (u, y), the system Σ is (output strictly) passive with respect to u and y;

ii) the set of all steady-state input-output pairs, R, is maximally (strictly) cyclically monotonic.

If the relation is strictly cyclically-monotone, then we say that the system is maximal equilibrium-independent strictly cyclically monotone (output strictly) passive *(MEISCMP).*

Remark 2.4. *It can be shown that when $d = 1$, a relation is cyclically monotone if and only if it is monotone. Thus, a SISO system is MEIP if and only if it is MEICMP [119, 121].*

Now, suppose that the agents Σ_i and the controllers Π_e are all MEICMP with steady-state input maps k_i and γ_e. We let K_i and Γ_e be the associated integral functions, which are convex functions, as a result of Theorem 2.1. We let $K = \sum_i K_i$ and $\Gamma = \sum_e \Gamma_e$ be their sum, so that $\partial K = k$ and $\partial \Gamma = \gamma$. As these are convex functions, we can look at the dual convex functions $K^\star$ and $\Gamma^\star$, namely

$$K^\star(\mathrm{y}) = -\inf_{\mathrm{u}} K(\mathrm{u}) - \mathrm{y}^\top \mathrm{u},$$

© Technion - Israel Institute of Technology, Elyachar Central Library

and similarly for $\Gamma^\star$. These are convex functions that satisfy $\partial K^\star = k^{-1}$ and $\partial \Gamma^\star = \gamma^{-1}$ (see Appendix A). The functions $K, K^\star, \Gamma, \Gamma^\star$ allow us to convert the conditions (2.2) to the unconstrained minimization problems of $K^\star(\mathrm{y})+\Gamma(\mathcal{E}_{\mathcal{G},d}^\top \mathrm{y})$ and $K(-\mathcal{E}_{\mathcal{G},d}\mu)+\Gamma^\star(\mu)$. Recalling that $\mathrm{u} = -\mathcal{E}_{\mathcal{G},d}\mu$ and that $\zeta = \mathcal{E}_{\mathcal{G},d}^\top \mathrm{y}$, we can state the minimization problems in the following form:

Optimal Potential Problem (OPP)		**Optimal Flow Problem** (OFP)	
$\min_{\mathrm{y},\zeta}$	$K^\star(\mathrm{y}) + \Gamma(\zeta)$	$\min_{\mathrm{u},\mu}$	$K(\mathrm{u}) + \Gamma^\star(\mu)$
s.t.	$\mathcal{E}_{\mathcal{G},d}^\top \mathrm{y} = \zeta$	s.t.	$\mathrm{u} = -\mathcal{E}_{\mathcal{G},d}\mu.$

These static optimization problems, known as the *optimal potential problem* and *optimal flow problem*, are two fundamental problems in the field of network optimization, which have been widely studied in computer science, mathematics, and operations research for many years [122]. A well-known instance of these problems is the *maximum-flow/minimum-cut problems*, which are widely used by algorithmists and by supply chain designers [36].

We conclude this subsection by stating the connection between the steady-states of the closed-loop network and the network optimization problems.

Theorem 2.2. *Consider a network system $(\mathcal{G}, \Sigma, \Pi)$ and suppose that both the agents and controllers are maximally equilibrium-independent cyclically-monotone passive. Let K and Γ be the sum of the integral functions for the agents and for the controllers, respectively. For any 4-tuple of vectors $(\mathrm{u}, \mathrm{y}, \zeta, \mu)$, the following conditions are equivalent:*

i) $(\mathrm{u}, \mathrm{y}, \zeta, \mu)$ *is a steady-state of the closed-loop;*

ii) (u, μ) *and* (y, ζ) *are dual optimal solutions of (OFP) and (OPP) respectively.*

Proof. We know that a convex function F is minimized at a point x if and only if $\mathbf{0} \in \partial F(x)$. Applying this to the objective functions of (OPP) and (OFP) implies that they are minimized exactly when the following inclusions hold,

$$\begin{cases} \mathbf{0} \in k^{-1}(\mathrm{y}) + \mathcal{E}_{\mathcal{G},d}\gamma(\mathcal{E}_{\mathcal{G},d}^\top \mathrm{y}) \\ \mathbf{0} \in \gamma^{-1}(\mu) - \mathcal{E}_{\mathcal{G},d}^\top k(-\mathcal{E}_{\mathcal{G},d}\mu). \end{cases} \tag{2.7}$$

Thus, Propositions 2.2 and 2.3 imply that if $(\mathrm{u}, \mathrm{y}, \zeta, \mu)$ is a steady-state of the closed-loop, then (u, μ) and (y, ζ) are optimal solutions of (OPP) and (OFP). The duality between them follows from $\mathrm{y} = k(\mathrm{u})$, $\mu = \gamma(\zeta)$. Conversely, if (u, μ) and (y, ζ) are dual optimal solutions, then y minimizes $K^\star(y) + \Gamma(\mathcal{E}_{\mathcal{G},d}^\top y)$ and μ minimizes $K(-\mathcal{E}_{\mathcal{G},d}\mu) + \Gamma^\star(\mu)$. Again, a convex function is minimized only where $\mathbf{0}$ is in its subdifferential, so we get the same inclusions (2.7). By Propositions 2.2 and 2.3 we get that $(\mathrm{u}, \mathrm{y}, \zeta, \mu)$ must be a steady-state of the closed-loop. □

© Technion - Israel Institute of Technology, Elyachar Central Library

Remark 2.5. *The problems (OPP) and (OFP) are special as they are convex duals of each other; the cost functions* $K^\star(\mathrm{y}) + \Gamma(\zeta)$ *and* $K(\mathrm{u}) + \Gamma^\star(\mu)$ *are dual [121]. Consequently, if* (y, ζ) *is an optimal solution of (OPP), then* (u, μ) *is an optimal solution of (OFP) if and only if* $\mu \in \gamma(\zeta)$, $\mathrm{u} \in k^{-1}(\mathrm{y})$ *and* $\mathrm{u} = -\mathcal{E}_{\mathcal{G},d}\mu$. *Thus, solving (OPP) and (OFP) on their own gives a viable method to understand the steady-states of* $(\mathcal{G}, \Sigma, \Pi)$.

2.2.3 Convergence to the Steady-State

Up to now, we dealt with the steady-states of the closed-loop system, but we did not prove that the system converges to a steady-state. We now address this point.

Theorem 2.3. *Consider the network system* $(\mathcal{G}, \Sigma, \Pi)$*, and suppose all node dynamics are maximally equilibrium-independent cyclically monotone output-strictly passive and that the controller dynamics are maximally equilibrium-independent cyclically monotone passive. Then there exists constant vectors* $\mathrm{u}, \mathrm{y}, \mu, \zeta$ *such that* $\lim_{t\to\infty} u(t) = \mathrm{u}$, $\lim_{t\to\infty} y(t) = \mathrm{y}$, $\lim_{t\to\infty} \mu(t) = \mu$, *and* $\lim_{t\to\infty} \zeta(t) = \zeta$. *Moreover,* (u, μ) *and* (y, ζ) *form optimal dual solutions to (OPP) and (OFP).*

We will give a proof of Theorem 2.3 for the case in which the controllers are given by the following form:

$$\Pi_e : \begin{cases} \dot{\eta_e} = \zeta_e \\ \mu_e = \psi_e(\eta_e). \end{cases} \tag{2.8}$$

The proof for the general case is analogous but more involved, and is not considered here to improve streamlining and readability.

Proof. Our assumption implies that the optimization problems (OPP) and (OFP) have dual optimal solutions, meaning that a steady-state solution exists. The equilibrium-independent passivity assumption implies that there are storage functions S_i (for $i \in \mathbb{V}$) and W_e (for $e \in \mathbb{E}$), such that

$$\begin{cases} \dot{S}_i \ \leq -\rho_i ||y_i(t) - \mathrm{y}_i||^2 + (y_i(t) - \mathrm{y}_i)^\top (u_i(t) - \mathrm{u}_i) \\ \dot{W}_e \leq (\mu_e(t) - \mu_e^\top (\zeta_e(t) - \zeta_e) \end{cases}. \tag{2.9}$$

Theorem 1.1 implies that $y(t)$ converges to y, implying that $\zeta(t)$ converges to $\mathbf{0} = \zeta = \mathcal{E}^\top \mathrm{y}$, as the integral function of the controllers is I_0 (see Example 1.3). Integrating implies that $\eta(t)$ converges to some η, as $\dot{\eta} = \zeta$. In turn, this implies that $\mu(t)$ converges to $\mu = \psi(\eta)$ and that $u(t)$ converges to $\mathrm{u} = -\mathcal{E}\mu$. It is clear that (u, y) is a steady-state input-output pair, and furthermore that $(\mathrm{u}, \mathrm{y}, \zeta, \mu)$ satisfy the conditions in Proposition 2.1, meaning that it is also a steady-state of the closed-loop and thus gives rise to an optimal solution of (OPP) and (OFP). This concludes the proof of the theorem. □

© Technion - Israel Institute of Technology, Elyachar Central Library

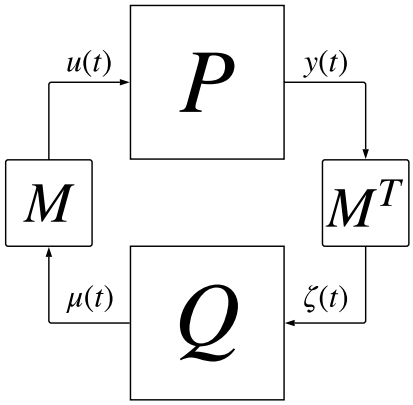


Figure 2.2: Block diagram of a general feedback interconnection.

Remark 2.6. *As a consequence of Remark 2.4, Theorem 2.3 also holds for output-strictly MEIP SISO agents and MEIP SISO controllers, which is the analysis result of Theorem 1.2. The result presented here is therefore more general, and the proof derivation, relying on integrating steady-state equations (or inclusions), provides a different approach than what was presented in [23].*

Remark 2.7. *The proof of Theorem 2.3 consisted of two parts. The first shows that if there is a steady-state I/O pair* (u, y) *for the agents* Σ*, a steady-state I/O pair* (ζ, μ) *for the controllers* Π*, and* $\zeta = \mathcal{E}_{\mathcal{G},d}^{\top}\mathrm{y}, \mathrm{u} = -\mathcal{E}_{\mathcal{G},d}\mu$*, then the closed-loop system converges. This part is based on the output-strict passivity of* Σ*, the passivity of* Π *and Theorem 1.1. The second part (for (OPP)) shows that the steady-state equation* $\mathbf{0} \in k^{-1}(\mathrm{y}) + \mathcal{E}_{\mathcal{G},d}\gamma(\mathcal{E}_{\mathcal{G},d}^{\top}\mathrm{y})$ *is equivalent to the minimization of* $K^{\star}(\mathrm{y}) + \Gamma(\mathcal{E}_{\mathcal{G},d}^{\top}\mathrm{y})$*. This part is based on the convexity of the integral function* $K^{\star}(\mathrm{y}) + \Gamma(\mathcal{E}_{\mathcal{G},d}^{\top}\mathrm{y})$*.*

The feedback configuration in Figure 1.1 can be thought of more abstractly as the symmetric feedback configuration of two MIMO systems P *and* Q *with the matrix* M*, as shown in Figure 2.2. This added layer of abstraction, in which we treat the stacked agents and controllers as stacked dynamical systems and study their I/O steady-state behavior, will be of great importance later. The reason is that* P*, in one case, will be a feedback connection of the agents* Σ *with some network control law, coupling the agents together, and forcing us to consider them as a single, indecomposable system.*

To conclude this section, we showed that under certain passivity requirements, the analysis problem for multi-agent systems can be solved - the system converges to a steady-state dictated by the network optimization problems (OPP) and (OFP). This connection gives a novel network interpretation to multi-agent systems, allowing for network-motivated intuition of multi-agent systems. In the next section, we'll consider different examples for MIMO systems with CM input-output steady-state relations.

© Technion - Israel Institute of Technology, Elyachar Central Library

2.3 Examples of MEICMP Systems

In this section, we focus on giving examples for MEICMP systems, showing that this property holds for many systems found in the literature. We focus on two classes of examples, the first being convex-gradient systems with oscillatory terms, generalizing reaction-diffusion systems, gradient descent algorithms and more, and the second being oscillatory systems with damping, which are a natural extension of oscillators like springs and pendulums. We conclude the section with a simulation of a network of oscillatory systems with damping.

2.3.1 Convex-Gradient Systems with Oscillatory Terms

Many systems can be divided into two parts - an oscillatory term and a damping term. These include physical systems such as reaction-diffusion systems, Euler-Lagrange systems and port-Hamiltonian systems, as well as examples coming from optimization theory, in which gradient descent algorithms play a vital role [17, 161, 162, 175]. Incremental passivity of these system has been studied in [71]. Mathematically, these systems can be represented as

$$\dot{x} = -\nabla\psi(x) + Jx + Bu, \tag{2.10}$$

where $x \in \mathbb{R}^n$ is the state of the system, $u \in \mathbb{R}^p$ is the input, representing various forces (both control and exogenous ones) acting on the system, $\psi : \mathbb{R}^n \to \mathbb{R}$ is a function representing the gradient part (and the sign is chosen to give ψ a potential-energy interpretation), J is a skew-symmetric matrix representing the oscillatory part, and $B \in \mathbb{R}^{n\times p}$ is the input matrix. Our goal is to show that for a wide class of measurements $y = h(x, u)$, this system is MEICMP. We first focus on stability of this system.

On many occasions, the function ψ is convex, and even strictly convex. For example, $\psi = \frac{\alpha}{2}x^2$ gives a linear damping term.

Theorem 2.4. *Assume that the system (2.10) is given, and that ψ is a strictly convex function such that $\lim_{||x||\to\infty} \frac{\psi(x)}{||x||} = \infty$ (i.e., $\psi(x) = o(||x||)$ as $x \to \infty$). Suppose furthermore that u is constant. Then there exists a unique x_0, which depends on u, such that all solutions converge to x_0 as $t \to \infty$.*

Remark 2.8. *The function ψ in* (2.10) *can be given the interpretation of potential energy. One might ask if assuming that ψ is radially unbounded is not enough, using intuition from Lyapunov theory [75]. We consider the system* (2.10) *in dimension $n = p = 1$, where $\psi(x) = \int_0^x \tanh(s)ds$, $J = 0$ and $B = 1$. In this case, the closed-loop system can be written as $\dot{x} = -\tanh(x) + u$. It is clear that if $u > 1$ is any constant input, then any solution $x(t)$ will converge to ∞. Moreover, we note that in this case ψ is radially unbounded, as $\psi(x) \geq 0.1|x| - c$ for some constant c. Thus, the assumption that $\psi(x) = o(||x||)$ as $x \to \infty$ is essential in Theorem 2.4.*

© Technion - Israel Institute of Technology, Elyachar Central Library

The proof of the theorem is quite convoluted, so it will be postponed for the time being. We now deal with the question of cyclic monotonicity. Consider the system

$$\begin{cases} \dot{x} = -\nabla\psi(x) + Jx + Bu \\ y = Cx + \rho(u), \end{cases} \tag{2.11}$$

where ψ is a strictly convex function such that $\lim_{\|x\|\to\infty} \frac{\psi(x)}{\|x\|} = \infty$ and J is a skew-symmetric matrix. By Theorem 2.4, the state of the system converges as $t \to \infty$ whenever u is constant, so the steady-state input-output relation can be defined.

Theorem 2.5. *Consider a system of the form* (2.11). *Suppose that B and C are invertible, and that ψ is a strictly convex function. Then the function $B^{-1}\nabla\psi C^{-1} - B^{-1}JC^{-1}$ is invertible, and the input-output relation of the system is CM if the function $(B^{-1}\nabla\psi C^{-1} - B^{-1}JC^{-1})^{-1} + \rho$ is the gradient of a convex function. Furthermore, if this map is the gradient of a strictly convex function, then the input-output relation is SCM.*

Proof. We first explain why the function $B^{-1}\nabla\psi C^{-1} - B^{-1}JC^{-1}$ is invertible. As B, C are invertible matrices, we show that $\nabla\psi - J$ can be inverted, i.e. that it is one-to-one. First, we assume that there are two points $x, y \in \mathbb{R}^n$ such that $\nabla\psi(x) - Jx = \nabla\psi(y) - Jy$, which is equivalent to $\nabla\psi(x) - \nabla\psi(y) = J(x - y)$. Recalling that J is skew-symmetric, multiplying both sides by $(x - y)^\top$ on the left implies that $(x-y)^\top(\nabla\psi(x) - \nabla\psi(y)) = 0$. However, as ψ is strictly convex, $\nabla\psi$ is strictly monotone, so we conclude that $x = y$ [121].

As for the second part of the theorem, in steady state we have $\dot{x} = \mathbf{0}$. Thus, if the steady-state input is u_{ss} and the state is x_{ss}, then they relate by $\nabla\psi(\mathrm{x}_{ss}) - J\mathrm{x}_{ss} = B\mathrm{u}_{ss}$. As B is invertible, we have

$$B^{-1}\nabla\psi(\mathrm{x}_{ss}) - B^{-1}J\mathrm{x}_{ss} = \mathrm{u}_{ss}.$$

However, if ρ is the zero function, we have $\mathrm{y}_{ss}^{\rho=0} = C\mathrm{x}_{ss}$, so we have the relation

$$B^{-1}\nabla\psi(C^{-1}\mathrm{y}_{ss}^{\rho=0}) - B^{-1}JC^{-1}\mathrm{y}_{ss}^{\rho=0} = \mathrm{u}_{ss}.$$

Thus,

$$\mathrm{y}_{ss}^{\rho=0} = (B^{-1}\nabla\psi C^{-1} - B^{-1}JC^{-1})^{-1}(\mathrm{u}_{ss}).$$

In the case of general ρ, we have the following input-output relation:

$$\mathrm{y}_{ss} = (B^{-1}\nabla\psi C^{-1} - B^{-1}JC^{-1})^{-1}(\mathrm{u}_{ss}) + \rho(\mathrm{u}_{ss}). \tag{2.12}$$

□

Corollary 2.1. *Consider a system of the form* (2.11). *If $C = B^\top = \mathrm{Id}$ and ρ satisfies $(\nabla\psi - J)^{-1} + \rho = \nabla\chi$ for some convex function χ, then the steady-state input-output relation is CM.*

Proof. This follows directly from (2.12) and $C = B^\top = \mathrm{Id}$. □

Corollary 2.2. *Consider a system of the form* (2.11). *If* $J = 0, B = C^\top$ *and* $\rho(u)$ *is the gradient of a convex function, then the steady-state input-output relation is CM.*

Proof. The only thing that needs to be shown is that $B^{-1}\nabla\psi(C^{-1}u)$ is the gradient of a convex function. Note that this is enough, as the inverse of the gradient function of a convex function is itself the gradient of a convex function (due to duality of convex functions). To do this, we define $\mu(x) = \psi(C^{-1}x)$. Then μ is convex as ψ is, and the gradient of μ is given by the chain rule. Its i-th entry is given by

$$\begin{aligned}\frac{\partial\mu}{\partial x_i} &= \sum_{j=1}^{n}\frac{\partial\psi}{\partial x_j}(C^{-1}x)\cdot\frac{\partial(C^{-1}x)_j}{\partial x_i} = \sum_{j=1}^{n}\frac{\partial\psi}{\partial x_j}(C^{-1}x)\cdot(C^{-1})_{ji}\\ &= \sum_{j=1}^{n}(C^{-1})_{ji}\frac{\partial\psi}{\partial x_j}(C^{-1}x) = [(C^{-1})^\top\nabla\psi(C^{-1}x)]_i = [B^{-1}\nabla\psi(C^{-1}x)]_i,\end{aligned}$$

meaning that $\nabla\mu(x) = B^{-1}\nabla\psi(C^{-1}x)$, proving the last part. □

Remark 2.9. *Theorem 2.5 can be stated more easily for linear systems. Suppose that* B, C *and* J *are as above. Suppose further that* ψ *has the form* $\psi(x) = x^\top Ax$ *where* $A > 0$, *and suppose we only seek for linear maps* ρ *of the form* $\rho(u) = Tu$ *for some matrix* T. *The dynamical system now has the form,*

$$\begin{cases}\dot{x} = -(A-J)x + Bu\\ y = Cx + Tu\end{cases}. \tag{2.13}$$

We now require ρ *to satisfy*

$$(B^{-1}\nabla\psi C^{-1} - B^{-1}JC^{-1})^{-1} + \rho = \nabla\chi,$$

for some convex function χ. *If we again seek linear* $\rho(u) = Tu$, *then the left-hand side of the equation is a linear map, so* $\nabla\chi$ *must also be a linear map. Due to convexity of* χ, *this is only possible if* $\nabla\chi(u) = Du$ *for some* $D \geq 0$. *We end up with following equation,* $(B^{-1}AC^{-1} - B^{-1}JC^{-1})^{-1} + T \geq 0$. *After some algebraic manipulation, we obtain*

$$C(A-J)^{-1}B + T \geq 0, \tag{2.14}$$

Thus we conclude that a linear system

$$\begin{cases}\dot{x} = Ax + Bu\\ y = Cx + Tu\end{cases}, \tag{2.15}$$

where A *is Hurwitz, is MEICMP if and only if* $-CA^{-1}B+T$ *is a positive-definite symmetric matrix.*

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

To conclude this subsection, we return to the proof of Theorem 2.4. The proof is rather lengthy, and requires two lemmas. The idea is to try and construct a quadratic Lyapunov function of the form $V(x) = \frac{1}{2}(x - x_0)^\top (x - x_0)$, where the point x_0 is a fixed point of the flow. Thus, we need to find a point x_0 which satisfies $\nabla\psi(x_0) - Bu = Jx_0$. The following two lemmas will assure that such a point exists.

Lemma 2.1. *Let χ be a strictly convex function, and suppose that $\frac{\chi(x)}{||x||} \to \infty$ as $||x|| \to \infty$. Then there exists some $\rho > 0$, such that for every point $x \in \mathbb{R}$ satisfying $\|x\| = \rho$, the inequality $\langle x, \nabla\chi(x)\rangle \geq 0$ holds.*

Proof. Fix some arbitrary unit vector $\theta \in \mathbb{R}^n$, and consider the convex function $f_\theta(r) = \chi(r\theta)$ and its derivative $\frac{df_\theta}{dr} = \nabla\chi(r\theta)^\top\theta$. Note that because χ grows faster than any linear function, the same can be said about f_θ, and in particular, it's derivative tends to infinity. Furthermore, the function f_θ is strictly convex, so $\frac{df_\theta}{dr}$ is strictly ascending, Thus there is some r_θ such that $\frac{df_\theta}{dr} > 0$ if $r > r_\theta$ and $\frac{df_\theta}{dr} < 0$ if $r < r_\theta$.

Our task now is to show that r_θ is a bounded function of θ. Suppose not, and let θ_n be a sequence of unit vectors such that $r_{\theta_n} \to \infty$. Passing to a sub-sequence, we may assume without loss of generality that $\theta_n \to \theta$ for some unit vector $\theta \in \mathbb{R}^n$. There is some N such that if $n \geq N$ then $r_{\theta_n} > r_\theta + 1 = t$. In particular, $\frac{df_{\theta_n}}{dr}|_{r=t} \leq 0$ for $n \geq N$ but $\frac{df_\theta}{dr}|_{r=t} > 0$. This is impossible, as the first expression is equal to $\nabla\chi(t\theta_n)^\top\theta_n$, which converges to the second expression, which is $\nabla\chi(t\theta)^\top\theta$. Thus, there is some $\rho > 0$ such that $r_\theta < \rho$ for all unit vectors θ, meaning that if x is a vector of norm ρ, then for $\theta = \frac{x}{||x||}$:

$$\langle\nabla\chi(x), x\rangle = \rho\langle\nabla\chi(\rho\theta), \theta\rangle = \rho\frac{df_\theta}{dr}(\rho) \geq 0. \tag{2.16}$$

□

Lemma 2.2. *Let $Q : \mathbb{R}^n \to \mathbb{R}^n$ be a continuous vector field, and let $\rho > 0$. Suppose that for any vector x satisfying $\|x\| = \rho$, the inequality $\langle Q(x), x\rangle \geq 0$ holds. Then there exists some point y satisfying $\|y\| \leq \rho$ such that $Q(y) = \mathbf{0}$.*

In order to prove the lemma, we use a theorem from algebraic topology.

Theorem 2.6 (Brouwer's Fixed Point Theorem [63])**.** *Let D be a closed ball inside $\mathbb{R}^n$, and let $f : D \to D$ be a continuous map. Then f has a fixed point.*

We now prove the lemma.

Proof. Suppose, heading toward contradiction, that Q does not vanish at any point in the ball $D = \{\|x\| \leq \rho\}$. We define a map $F : D \to D$ by

$$F(x) = -\rho\frac{Q(x)}{\|Q(x)\|}. \tag{2.17}$$

This is a continuous map (as Q never vanishes), and the norm of $F(x)$ is always equal to ρ, so $F(x)$ is indeed in D. Thus, we can apply Brouwer's fixed point theorem to F and get a fixed point, called y.

We know that y satisfies $F(y) = y$, i.e., $-\rho\frac{Q(y)}{\|Q(y)\|} = y$. On one hand, taking the norm of the last equation implies that $\|y\| = \rho$. On the other hand, rearranging it implies that $Q(y) = -\frac{\|Q(y)\|}{\rho}y = \lambda y$ where λ is some negative scalar (as $Q(y) \neq \mathbf{0}$). Thus, we found a point y of norm ρ such that $\langle Q(y), y\rangle = \lambda\|y\|^2 < 0$ for some $\lambda < 0$, which contradicts our assumption. Thus Q has a zero inside the ball $D = \{||x|| \leq \rho\}$. □

We are now ready to prove Theorem 2.4.

Proof. First, because u is constant, we can absorb the constant term Bu inside the gradient $\nabla\psi(x)$ by adding the linear term $(Bu)^\top x$ to $\psi(x)$. This does not change the fact that ψ is strictly convex, nor the fact that it ascends faster than any linear function. Thus we may assume that $Bu = \mathbf{0}$ for the remainder of the proof.

Now, we define the vector field $Q(x) = \nabla\psi(x) - Jx$. Note that because J is skew-symmetric, for all $x \in \mathbb{R}$,

$$\langle \nabla\psi(x) - Jx, x\rangle = \langle \nabla\psi(x), x\rangle. \tag{2.18}$$

Thus, by the Lemma 2.1, there's some $\rho > 0$ such that $\langle Q(x), x\rangle \geq 0$ for any vector x satisfying $\|x\| \leq \rho$, and by Lemma 2.2 we can find some point $x_0 \in \mathbb{R}$ such that $Q(x_0) = 0$, or equivalently, $Jx_0 = \nabla\psi(x_0)$. We claim that any solution to the ODE converges to x_0. Indeed, define $F(x) = \frac{1}{2}\|x - x_0\|^2$. Then F is non-negative, vanishing only at x_0, and furthermore,

$$\begin{aligned}\dot{F} &= (x - x_0)^\top \dot{x} = (x - x_0)^\top(-\nabla\psi(x) + Jx)\\ &= (x - x_0)^\top(-\nabla\psi(x) + \nabla\psi(x_0) + J(x - x_0))\\ &= -(x - x_0)^\top(\nabla\psi(x) - \nabla\psi(x_0)) \leq 0,\end{aligned} \tag{2.19}$$

where the last inequality is true because ψ is convex and Theorem 2.1. Furthermore, $\dot{F}$ is negative if $x \neq x_0$ because ψ is strictly convex and Theorem 2.1. The uniqueness of x_0 follows from the fact that the flow globally asymptotically converges to x_0. This completes the proof. □

2.3.2 Oscillatory Systems with Damping

We consider a damped oscillator with a linear forcing term of the form $\ddot{x} + \zeta\dot{x} + \omega^2 x = Bu$ where B is a constant matrix, u is the input vector, and $\zeta > 0$ is the damping factor. This system can also be represented via the set of first order ODEs:

$$\begin{cases}\dot{q} = \omega p\\ \dot{p} = -\omega q - \zeta p + Bu\end{cases}. \tag{2.20}$$

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

One can easily generalize this formulation to more complex methods of damping:

$$\begin{cases} \dot{q} = Mp \\ \dot{p} = -M^\top q - \nabla\psi(p) + Bu \end{cases}. \tag{2.21}$$

We are usually interested in the position as the output, i.e., $y = q$ for this system. We wish to find a condition that will assure this system is stable and MEICMP. We first prove the following result.

Theorem 2.7. *Consider a system of the form* (2.21)*, and suppose that* M *is invertible. Suppose furthermore that* ψ *is a strictly convex function such that* $\lim_{\|x\|\to\infty} \frac{\psi(x)}{\|x\|} = \infty$*. Then the system is stable for constant inputs. Furthermore, if the system is injected with the constant input signal* u*, then there is some* q_0 *such that all trajectories of the system satisfy* $q \to q_0, p \to p_0 = \mathbf{0}$ *as* $t \to \infty$*. Even further,* $q_0 = (M^\top)^{-1}Bu - (M^\top)^{-1}\nabla\psi(p_0)$

Proof. As above, the assumption on ψ allows us to absorb the linear term inside ψ, so we can assume $B\mathrm{u} = \mathbf{0}$. Now, we take $p_0 = \mathbf{0}$ and $q_0 = -(M^\top)^{-1}\nabla\psi(p_0)$. We note that the following relations hold:

$$Mp_0 = \mathbf{0},\ M^\top q_0 = -\nabla\psi(p_0),\ p_0^\top \nabla\psi(p_0) = \mathbf{0}. \tag{2.22}$$

Now, consider the following Lyapunov function candidate,

$$F(p,q) = \frac{1}{2}(p-p_0)^\top(p-p_0) + \frac{1}{2}(q-q_0)^\top(q-q_0). \tag{2.23}$$

It's clear that $F \geq 0$ and that $F = 0$ if and only if $p = p_0$ and $q = q_0$. Furthermore, the derivative of F along the trajectories is given by:

$$\begin{aligned} \dot{F} &= (p-p_0)^\top\dot{p} + (q-q_0)^\top\dot{q} \\ &= (p-p_0)^\top(-M^\top q - \nabla\psi(p)) + (q-q_0)^\top Mp \\ &= -(p-p_0)^\top\nabla\psi(p) - (Mp_o)^\top q - (M^\top q_0)^\top p \\ &\overset{(2.22)}{=} -(p-p_0)^\top\nabla\psi(p) + p^\top\nabla\psi(p_0) - p_0^\top\nabla\psi(p_0) \\ &= -(p-p_0)^\top(\nabla\psi(p) - \nabla\psi(p_0)). \end{aligned}$$

The last expression is non-positive, and furthermore is strictly negative if $p \neq p_0$ (as ψ is strictly convex). Thus, it follows from Barbalat's lemma that $p \to p_0 = \mathbf{0}$ as $t \to \infty$, because F is non-negative. Now, the equation driving p is $\dot{p} = -M^\top q - \nabla\psi(p)$, which can be rewritten as

$$q = -(M^\top)^{-1}(\dot{p} + \nabla\psi(p)). \tag{2.24}$$

As $t \to \infty$, the right hand side tends to $-(M^\top)^{-1}(\nabla\psi(p_0)) = q_0$, concluding the proof of the claim. □

© Technion - Israel Institute of Technology, Elyachar Central Library

Not only have we proved that the system is stable, we also found the input-output steady-state relation, which turns out to be linear. Thus, we can apply Remark 2.9 to conclude the following corollary.

Corollary 2.3. *The system* (2.21) *is MEICMP if and only if the matrix* $(M^\top)^{-1}B$ *is positive semi-definite. Furthermore, it is MEISCMP if and only if this matrix is positive definite.*

2.3.3 Example: A Network of Planar Oscillators

We now demonstrate these results for oscillatory systems with damping by a simulation. We consider a network of six damped MIMO oscillators,

$$\Sigma_i \begin{cases} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \Omega_i x_2 \\ -D_i x_2 - \Omega_i^\top (x_1 - \xi_i) + \Omega_i^{-1} u \end{bmatrix} \\ y = x_1 \end{cases}$$

where ξ_i is the equilibrium point of the oscillator, Ω_i is a matrix consisting of the self frequencies, and D_i is a damping matrix, which is positive-definite. The values of ξ_i were chosen as normally distributed random variables with mean $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$ and covariance $\begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}$. Moreover, the matrices D_i were chosen as random positive-definite matrices by writing $D_i = U_i S_i U_i^\top$, where S_i is a diagonal matrix whose diagonal entries distributed according to a Rayleigh distribution, and U_i is a random orthogonal matrix chosen according to the Haar probability measure on the space of unitary matrices [46]. Similarly, the connecting matrix Ω_i was chosen to be diagonal with Rayleigh-distributed diagonal entries. The underlying graph is the complete bipartite graph on 6 vertices, meaning that two agents i, j are connected if and only if i is odd and j is even or vice versa.

The steady-state input-output relation of Σ_i is given by $k_i(u_i) = (\Omega_i \Omega_i^\top)^{-1} u + \mathrm{x_i}$, so that $k_i^{-1}(y) = (\Omega_i \Omega_i^\top)(y - \mathrm{x_i})$. The corresponding integral function is $K_i^\star(y_i) = \frac{1}{2} y^\top \Omega_i \Omega_i^\top y - y^\top \Omega_i \Omega_i^\top \mathrm{x}_i$, which is strictly convex. Moreover, we consider network controllers of the form:

$$\begin{cases} \dot{\eta}_e = -\eta_e + \zeta_e \\ \zeta_e = \psi(\eta_e). \end{cases}$$

The function ψ is given as

$$\psi(x) = \arcsin\left(\frac{\log^2\left(\frac{e^x+1}{2}\right)\mathrm{sgn}(x)}{\log^2\left(\frac{e^x+1}{2}\right) + 1} \right),$$

where $\mathrm{sgn}(x)$ is the sign function. This function was chosen to demonstrate that the network optimization framework holds even for highly-nonlinear systems. One can verify that $\psi(0) = 0$ and that ψ is a monotone ascending function, and that the associated integral function is given by:

$$\Gamma_e(\zeta_e) = \int_0^{\zeta_e} \arcsin\left(\frac{\log^2\left(\frac{e^x+1}{2}\right)\mathrm{sgn}(x)}{\log^2\left(\frac{e^x+1}{2}\right) + 1} \right) dx.$$

© Technion - Israel Institute of Technology, Elyachar Central Library

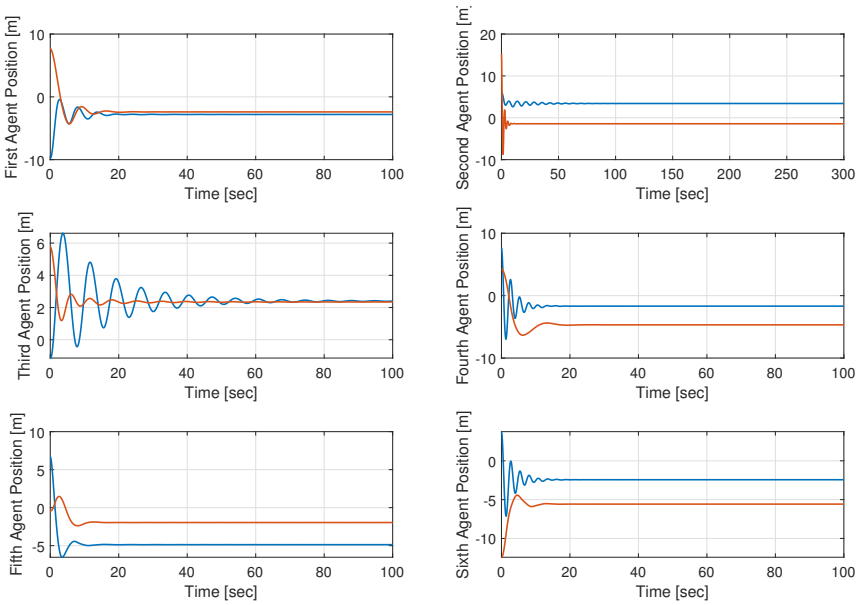


Figure 2.3: Positions of the oscillators in the case study.

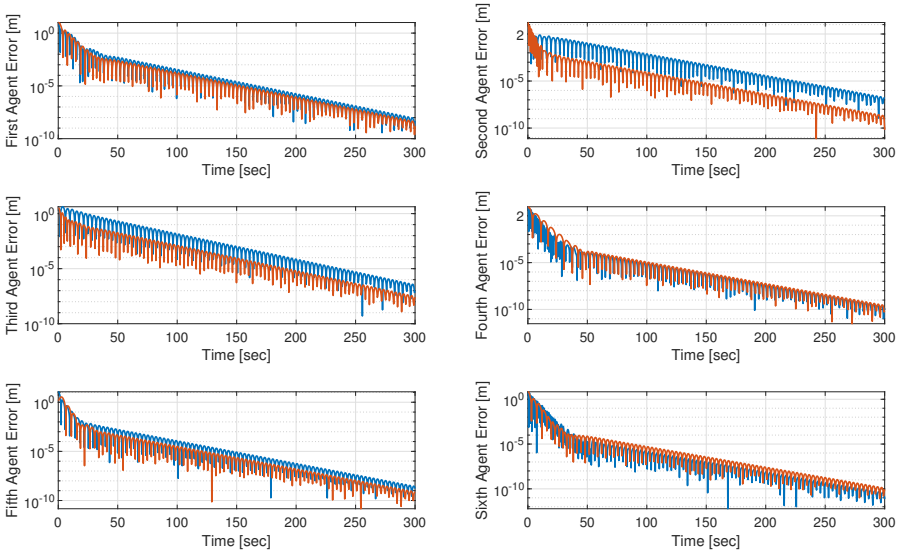


Figure 2.4: Distance of the oscillators' positions from the projected steady-state value.

We run the system for a total of 300 seconds, checking that its limit corresponds to the minimizer of (OPP). The output of the system in the first 100

seconds can be seen in Figure 2.3, exhibiting the positions of the agents $y(t)$. We omit the behavior after the first 100 seconds, as no real change in the position can be seen with the naked eye. The blue line represents first coordinate, and the red one represents the second coordinate. We also solve (OPP) using gradient descent, running the algorithm with random initial conditions and a total of 8000 steps of varying step-sizes. The distance of the positions of the oscillators can be seen in Figure 2.4, where we see that the network converges to the forecasted value, up to minuscule errors arising from stopping both the network system and the gradient descent algorithm after finite time.

2.4 Conclusions

We have found a profound connection between passivity-based cooperative control of MIMO systems and network optimization theory in the spirit of Rockafellar [122]. This was done by introducing the notion of maximal equilibrium-independent cyclically monotone passive systems, and showing that such systems converge to a solution of a collection of network optimization problems, bonded by duality. In other words, we have established inverse optimality and duality results for general networks of maximal equilibrium-independent cyclically monotone passive systems. This connection creates a dictionary between system signals (like outputs and inputs) and network optimization variables (potentials and node divergences, respectively). We have also studied two classes of nonlinear systems, proving that they are maximal equilibrium-independent cyclically monotone passive under certain coercivity assumptions, and exemplified the connection between the network optimization framework and multi-agent systems by a case study of planar damped oscillators. This significant extension of the framework connecting multi-agent systems and cooperative control to network optimization will allow us to consider the applications appearing in Chapters 4, 6 and 7 also for MIMO systems, with little to no effort required.

In the next chapter, we'll consider another extension of the network optimization framework of [23], this time to Passive-short systems.

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

Chapter 3

A Network Optimization Framework for Passive-Short Agents

This section is a review of [72, 134, 140]. We demonstrate that the presented network optimization framework cannot be used for passive-short agents, either because it is undefined, or because it predicts a wrong limit for the closed-loop system. The failure of the network optimization framework can be understood by the integral function for the agents being non-convex (when defined), or by the steady-state input-output relation for the agents being non-monotone. We will use this motivation to generalize the network optimization framework by augmenting the network optimization problems associated with the closed-loop system, and interpret the augmentation as a transformation of the agents.

3.1 Introduction

As we saw in Section 1.2, the notion of passivity is vital for different approaches studying large-scale multi-agent systems, including the network optimization framework. In practice, however, many systems are not well-behaved, and possess a shortage of passivity (or non-passiveness) in their operation [61, 113, 158, 167]. Motivated by this fact, we consider dynamical systems that do not fulfill the passivity requirements, and are characterized by their *shortage of passivity*. In the literature, passivity indices are used to quantify the excess or shortage of passivity in a system and are often compensated using passivation[1] methods such as feedback, feed-forward, or a combination of such schemes [24, 168, 179].

However, the methods of dealing with shortage of passivity only consider passivity with respect to a specific steady-state input-output pair (u, y). These methods give no way of asserting that the multi-agent system converges without

[1] Also referred to as feedback passification [47, 48].

© Technion - Israel Institute of Technology, Elyachar Central Library

specifying a specific steady-state limit. We want to use the network optimization framework presented in Sections 1.2 and 2.2, assuming we have an equilibrium-independent shortage of passivity. However, the network optimization framework does not apply for passive-short systems unless augmented, as we'll see in Section 3.2. The failure of the framework can be understood in terms of non-convexity of the integral functions for the agents, if they are defined, and by the non-monotonicity of the steady-state input-output relations of the agents otherwise. Thus, one might try and convexify the integral functions, or monotonize the steady-state relations, and understand what the system-theoretic meaning of these augmentations are. The goal of this chapter is to present the methods of augmenting the network optimization framework, validating it for passive-short agents, and to give these augmentations a system-theoretic meaning.

The structure of this chapter is as follows. In Section 3.2, we show three examples of the failure of the network optimization framework. In Section 3.3, we consider augmentations of the network optimization framework for networks of output-passive short agents. In this case, the integral functions are defined, so we can discuss the network optimization problems and try to convexify them. We show multiple methods of convexification, which translate into multiple system-theoretic transformations for the agents. In Section 3.4, we study networks of agents with general shortage of passivity, for which the integral functions need not be defined. In this case, we focus on monotonization of the steady-state input-output relations, and present the equivalent system-theoretic transformation for the agents. We conclude this section by a few case studies.

3.2 Shortage of Passivity and Failures of the Network Optimization Framework

In this section, we first define and study the notions of shortage of passivity and equilibrium-independent shortage of passivity, and then discuss some examples in which the network optimization framework fails, highlighting the different reasons which can cause the failure.

3.2.1 Shortage of Passivity

We first give a definition of shortage of passivity and equilibrium-independent shortage of passivity.

Definition 3.1. *A dynamical system $\Sigma : u \mapsto y$ is* output ρ-passive *with respect to the steady-state* (u, y)*, for some number $\rho \in \mathbb{R}$, if there exists a non-negative $\mathcal{C}^1$-smooth function $S(x)$ of the state x of Σ such that the inequality*

$$\frac{d}{dt}S(x(t)) = \nabla S(x(t))\dot{x}(t) \leq (y(t) - \mathrm{y})^\top (u(t) - \mathrm{u}) - \rho\|y(t) - \mathrm{y}\|^2 \qquad (3.1)$$

holds for any trajectory $(u(t), x(t), y(t))$.

© Technion - Israel Institute of Technology, Elyachar Central Library

We say that the system is input ν-passive *with respect to the steady-state* (u,y)*, for some number* $\nu \in \mathbb{R}$*, if there exists a non-negative* $\mathcal{C}^1$*-smooth function* $S(x)$ *such that the inequality*

$$\frac{d}{dt}S(x(t)) = \nabla S(x(t))\dot{x}(t) \leq (y(t)-\mathrm{y})^\top (u(t)-\mathrm{u}) - \nu\|u(t)-\mathrm{u}\|^2 \tag{3.2}$$

holds for any trajectory $(u(t),x(t),y(t))$*.*

Lastly, we say the system is input-output (ρ,ν)-passive short *with respect to the steady-state* (u,y)*, for some numbers* $\rho,\nu \in \mathbb{R}$ *such that* $\rho\nu < \frac{1}{4}$*, if there exists a non-negative* $\mathcal{C}^1$*-smooth function* $S(x)$ *such that the inequality*

$$\frac{d}{dt}S(x(t)) \leq (y(t)-\mathrm{y})^\top (u(t)-\mathrm{u}) - \rho\|y(t)-\mathrm{y}\|^2 - \nu\|u(t)-\mathrm{u}\|^2 \tag{3.3}$$

holds for any trajectory $(u(t),x(t),y(t))$*.*

Remark 3.1. *It should be noted that the numbers* ρ,ν *in the definition above are not unique, as decreasing them makes the inequality easier to satisfy. This motivates the definition of the equilibrium-independent passivity indices analogously to the output-feedback passivity index (OFP) and the input-feedthrough passivity index (IFP) in [167]. Moreover, the definition above unifies both strictly-passive, passive, and passive-short systems. The case* $\rho,\nu > 0$ *corresponds to strict passivity,* $\rho,\nu = 0$ *corresponds to passivity, and* $\rho,\nu < 0$ *corresponds to shortage of passivity. Thus, it will allow to consider networks of systems where some are passive and some are passive-short, without needing to specify the exact passivity assumption. Moreover, it allows to consider input-output* (ρ,ν)*-passivity systems for* $\rho > 0$ *and* $\nu < 0$ *(or vice versa) with no additional effort needed.*

Remark 3.2. *The demand that* $\rho\nu < \frac{1}{4}$ *is essential. Indeed, if* $\rho\nu \geq \frac{1}{4}$*, the right-hand side of* (3.3) *is always non-negative. For example, if* $\rho = \nu = -\frac{1}{2}$*, then the right-hand side is equal to* $\|u(t)-\mathrm{u}+y(t)-\mathrm{y}\|^2 \geq 0$*. In that case, any static nonlinearity would be input-output* (ρ,ν)*-passive, which is obviously too broad for such a definition.*

Remark 3.3. *Observe that input-output* (ρ,ν)*-passive systems capture both output* ρ*-passive and input* ν*-passive systems by setting either* $\rho = 0$ *or* $\nu = 0$*.*

The notion of input-output (ρ,ν)-passivity is quite extensive, and includes many different systems. As a specific case, we claim that it includes all finite-$\mathcal{L}_2$-gain systems, which are defined below:

Definition 3.2. *A dynamical system* $\Sigma : u \mapsto y$ *has* finite $\mathcal{L}_2$-gain *with respect to the steady-state* (u,y) *if there exists a number* $\beta > 0$ *and non-negative* $\mathcal{C}^1$*-smooth function* $S(x)$ *of the state* x *of* Σ *such that the inequality*

$$\frac{d}{dt}S(x(t)) = \nabla S(x(t))\dot{x}(t) \leq -\|y(t)-\mathrm{y}\|^2 + \beta^2\|u(t)-\mathrm{u}\|^2 \tag{3.4}$$

holds for any trajectory $(u(t),x(t),y(t))$*. The minimal number* β *for which the inequality holds for any trajectory is called the* $\mathcal{L}_2$-gain *of the system* Σ*, and is it equal to the operator norm of* $\Sigma : \mathcal{L}_2 \to \mathcal{L}_2$*.*

© Technion - Israel Institute of Technology, Elyachar Central Library

Theorem 3.1. *Let $\Sigma : u \mapsto y$ be a finite $\mathcal{L}_2$-gain system with gain β with respect to the steady-state input-output pair (u, y). Then, Σ is input-output (ρ, ν)-passive with respect to (u, y), with passivity indices $\rho = -g(\beta) < 0, \nu \leq -\left(1 + \frac{1}{4(g(\beta)+\beta^2)}\right)$, where $g(\beta)$ is any positive function of the $\mathcal{L}_2$-gain β.*

Proof. By definition, there exists a storage function $S(x) > 0$ such that

$$\dot{S}(x) \leq \begin{bmatrix} u - \mathrm{u} \\ y - \mathrm{y} \end{bmatrix}^\top \begin{bmatrix} -\beta^2 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u - \mathrm{u} \\ y - \mathrm{y} \end{bmatrix}.$$

On the other hand, it follows from Definition 3.1 that Σ is input-output (ρ, ν)-passive if there exists a storage function $\bar{S}(x) > 0$ such that

$$\dot{\bar{S}}(x) \leq \begin{bmatrix} u - \mathrm{u} \\ y - \mathrm{y} \end{bmatrix}' \begin{bmatrix} -\rho & \frac{1}{2} \\ \frac{1}{2} & -\nu \end{bmatrix} \begin{bmatrix} u - \mathrm{u} \\ y - \mathrm{y} \end{bmatrix},$$

where $\rho < 0$, $\nu < 0$ and $\rho\nu < \frac{1}{4}$.

Thus, it's enough to show the existence of $\rho < 0, \nu < 0$ such that

$$\begin{bmatrix} -\rho & \frac{1}{2} \\ \frac{1}{2} & -\nu \end{bmatrix} \geq \begin{bmatrix} -\beta^2 & 0 \\ 0 & 1 \end{bmatrix},$$

and $\rho\nu < \frac{1}{4}$, as we can choose $\bar{S}(x) = S(x)$. Indeed, we need to show that

$$\begin{bmatrix} -\rho + \beta^2 & \frac{1}{2} \\ \frac{1}{2} & -\nu - 1 \end{bmatrix} \geq 0.$$

It is sufficient to show that both the upper left element $-\rho + \beta^2$ and the determinant are positive. Since $\rho < 0$, it is clear that $-\rho + \beta^2 > 0$. As for the determinant, we have

$$\det \begin{bmatrix} -\rho + \beta^2 & \frac{1}{2} \\ \frac{1}{2} & -\nu - 1 \end{bmatrix} = (-\rho + \beta^2)(-\nu - 1) - \frac{1}{4}.$$

Assigning $\rho = -g(\beta) < 0$, where $g(\beta)$ is any positive function of the gain β, one can see that the determinant is non-negative as long as $\nu \leq -1 + \frac{1}{4(g(\beta)+\beta^2)}$, which proves the claim. □

Remark 3.4. *One can easily check that the result above is not true in the opposite direction, that is, if the system Σ is input-output (ρ, ν)-passive, it does not necessarily have a finite $\mathcal{L}_2$-gain. This is due of the fact that the 2×2 matrix above cannot be negative semi-definite as $-\rho + \beta^2 > 0$. See also the example following this remark.*

Example 3.1. *Consider the (SISO) linear system $\Sigma : \dot{x} = x + u;\ y = x$. One can easily verify using the storage function $S(x) = \frac{1}{2}(x - \mathrm{x}^2)$ that this system is output- ρ-passive with respect to any steady-state I/O pair (u, y), for $\rho = -1$. However, the system does not have a finite $\mathcal{L}_2$-gain and is not even $\mathcal{L}_2$-stable.*

© Technion - Israel Institute of Technology, Elyachar Central Library

Indeed, if we apply an input signal of the form $u(t) = \begin{cases} 1 & t < 1 \\ 0 & t \geq 1 \end{cases}$*, then the output is given via the following convolution integral*

$$y(t) = \int_0^t e^{t-\tau} u(\tau) d\tau = \int_0^1 e^{t-\tau} d\tau = (e-1)e^{t-1},$$

which is not even bounded and therefore $y(t) \notin \mathcal{L}_2$*, even though the input* $u(t) \in \mathcal{L}_2$*. Thus, the system* Σ *is output* ρ*-passive with respect to any equilibrium, but does not have a finite* $\mathcal{L}_2$*-gain with respect to any steady-state input-output pair.*

To conclude this subsection, we define the notion of equilibrium-independent input-output (ρ, ν)-passivity, which will be our main focus throughout this chapter.

Definition 3.3. *A dynamical system* $\Sigma : u \mapsto y$ *is said to be:*

i) equilibrium-independent output ρ-passive *(EI-OP(ρ)) if* (3.1) *holds with respect to any steady-state input-output pair;*

ii) equilibrium-independent input ν-passive *(EI-IP(ν)) if* (3.2) *holds with respect to any steady-state input-output pair;*

iii) equilibrium-independent input-output (ρ, ν)-passive *(EI-IOP(ρ, ν)) if* $\rho\nu < \frac{1}{4}$ *and* (3.3) *holds with respect to any steady-state input-output pair.*

3.2.2 Failure of the Network Optimization Framework

We can now consider the network optimization framework for EI-IOP(ρ, ν) agents. We claim that the network optimization framework fails in this case. We present three examples of this failure. The first shows that the closed-loop system need not converge, no matter which controllers we choose. The second shows that the integral functions for the agents need not be defined. The third shows that even if the closed-loop system converges, and the integral functions are defined, the limit predicted by the network optimization framework need not be equal to the actual limit of the closed-loop system.

Example 3.2. *Consider a collection of* n *agents, each having the model* $\dot{x_i} = x_i + u_i$*,* $y_i = x_i$*. It's easy to check that the agents are EI-OP(ρ) with parameter* $\rho = -1$*. We claim that for any graph* $\mathcal{G}$ *and any collection of controllers* Π*, the diffusively-coupled network* $(\mathcal{G}, \Sigma, \Pi)$ *diverges. Indeed, take any graph* $\mathcal{G}$ *and network controllers* Π*, and consider an initial condition for which* $x_i(0) = 1$*. Let* $\mu(t)$ *be the controller output, so that the closed-loop system has the dynamics* $\dot{x} = x - \mathcal{E}_{\mathcal{G}}\mu(t)$*. In particular,* $\mathbb{1}^\top \dot{x} = \mathbb{1}^\top x$*, meaning that the mode* $\mathbb{1}^\top x$ *is equal to* $\mathbb{1}^\top x(0) \exp(t) = n \exp(t)$*. Thus the state* x*, and the output* y*, do not converge.*

© Technion - Israel Institute of Technology, Elyachar Central Library

Example 3.3. *Consider a SISO dynamical system Σ given by*

$$\Sigma: \ \dot{x} = -\sqrt[3]{x} + 0.5x + 0.5u; \quad y = 0.5x - 0.5u, \tag{3.5}$$

where the input is u and the output is y. By using a change of variables of the form

$$\begin{bmatrix} \tilde{u} \\ \tilde{y} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} u \\ y \end{bmatrix},$$

we obtain the following system

$$\tilde{\Sigma}: \ \dot{x} = -\sqrt[3]{x} + \tilde{u}; \quad \tilde{y} = x, \tag{3.6}$$

where $\tilde{u}$ and $\tilde{y}$ are, respectively, the input and output of the transformed system $\tilde{\Sigma}$. The matrix transformation $T = \left[\begin{smallmatrix} 1 & 1 \\ 1 & 2 \end{smallmatrix}\right]$ also connects the steady-state relations of the two systems, that is, if $(\tilde{\mathrm{u}}, \tilde{\mathrm{y}})$ is a steady-state input-output pair for $\tilde{\Sigma}$, then (u, y) is a steady-state input-output pair of Σ, where

$$\begin{bmatrix} \tilde{\mathrm{u}} \\ \tilde{\mathrm{y}} \end{bmatrix} = T \begin{bmatrix} \mathrm{u} \\ \mathrm{y} \end{bmatrix}.$$

It is easy to verify that $\tilde{\Sigma}$ is MEIP with storage function $S(x) = \frac{1}{2}(x - \mathrm{x})^2$, with $\mathrm{x} = \tilde{\mathrm{y}}$ for any steady-state I/O pair $(\tilde{\mathrm{u}}, \tilde{\mathrm{y}})$. Let $R(x) = \frac{1}{3}S(x)$ be the storage function for the original system Σ, we obtain

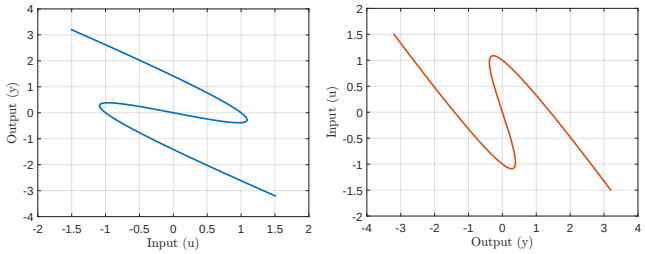
$$\begin{aligned} \dot{R} = \frac{1}{3}\dot{S} &\le \frac{1}{3}(\tilde{u} - \tilde{\mathrm{u}})(\tilde{y} - \tilde{\mathrm{y}}) = \frac{1}{3}\begin{bmatrix} \tilde{u} - \tilde{\mathrm{u}} \\ \tilde{y} - \tilde{\mathrm{y}} \end{bmatrix}' \begin{bmatrix} 0 & \frac{1}{2} \\ \frac{1}{2} & 0 \end{bmatrix} \begin{bmatrix} \tilde{u} - \tilde{\mathrm{u}} \\ \tilde{y} - \tilde{\mathrm{y}} \end{bmatrix} \\ &= \frac{1}{3}\begin{bmatrix} u - \mathrm{u} \\ y - \mathrm{y} \end{bmatrix}' T' \begin{bmatrix} 0 & \frac{1}{2} \\ \frac{1}{2} & 0 \end{bmatrix} T \begin{bmatrix} u - \mathrm{u} \\ y - \mathrm{y} \end{bmatrix} = \frac{1}{3}\begin{bmatrix} u - \mathrm{u} \\ y - \mathrm{y} \end{bmatrix}' \begin{bmatrix} 1 & \frac{3}{2} \\ \frac{3}{2} & 2 \end{bmatrix} \begin{bmatrix} u - \mathrm{u} \\ y - \mathrm{y} \end{bmatrix} \\ &= (u - \mathrm{u})(y - \mathrm{y}) + \frac{1}{3}(u - \mathrm{u})^2 + \frac{2}{3}(y - \mathrm{y})^2, \end{aligned}$$

i.e., the system Σ is EI-IOP(ρ, ν) with passivity indices $\rho = -2/3$ and $\nu = -1/3$. Utilizing the connection between the steady-state input-output relations of the two systems, one can easily see that the steady-state relation of Σ is given by the planar curve $\mathrm{u} = 2\sigma - \sigma^3$; $\mathrm{y} = \sigma^3 - \sigma$, parameterized by a variable σ, as shown in Figure 3.1. It is clear from Figure 3.1 that both steady-state input-output relation and its inverse are non-monotone, and furthermore that they possess no integral function. In other words, Σ is EI-IOP(ρ, ν) for some ρ, nu, but one cannot associate an integral function with it, so the network optimization framework cannot be defined.

Example 3.4. *Consider a class of networked nonlinear SISO gradient systems, which are diffusively-coupled networks with agents described by*

$$\Sigma_i: \ \dot{x}_i = -\frac{\partial U(x_i)}{\partial x_i} + u_i; \quad y_i = x_i, \quad i = 1, \ldots, |\mathcal{V}|, \tag{3.7}$$

© Technion - Israel Institute of Technology, Elyachar Central Library



(a) The relation (b) The inverse relation

Figure 3.1: Steady-state relations of the system in Example 3.3.

where U is a $\mathcal{C}^1$ scalar potential function with $U(\sigma) > 0,\ \sigma \neq 0$ and $U(0) = 0$. Moreover, we take identical static gains as controllers Π_e : $\mu_e = G\zeta_e$, where $G > 0$, meaning we have a control law of the form

$$u_i = G\sum_{j\sim i}(x_j - x_i), \quad i = \ldots, |\mathcal{V}|, \tag{3.8}$$

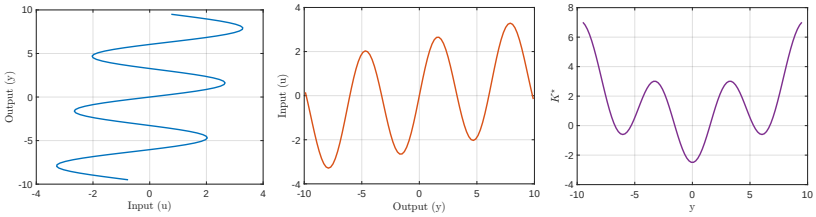
Such classes of systems are important because of their applications in both biological and multi-agent systems, and are inspired from [131]. As discussed in [131], (3.7) loosely describes the dynamics of a group of bacteria performing chemotaxis (where x_i is the position of the bacteria) in response to the chemical stimulus such as directing their movements according to the concentration of chemicals in their environment to find food (for example, glucose) by swimming towards the highest concentration of food molecules. Other possible applications include networks of robots that must efficiently climb gradients to search for a source by measuring its signal strength in a spatially distributed environment.

Proposition 3.1. *Consider the gradient system (3.7), where the Hessian matrix $\mathrm{Hess}(U)$ of the potential U satisfies $\mathrm{Hess}(U) \geq \rho \mathrm{Id}_{|\mathcal{V}|}$ for some ρ. Then the system is EI-OP(ρ).*

Proof. Take any steady-state input-output pair (u, y) of the system. Consider the storage function $S(x) = \frac{1}{2}\|x - \mathrm{x}\|^2$. Taking the derivative of S along the system dynamics (3.7) yields $\dot{S} = (x-\mathrm{x})^\top(-f(x)+u)$, where we denote $f(x) = \nabla U(x)$ for notational convenience. Defining $\varphi(x) = f(x) - \rho x$, we can write $\dot{S} = (x-\mathrm{x})^\top(-\varphi(x) - \rho x + u)$. Adding and subtracting $\varphi(\mathrm{x})$ and $\rho\mathrm{x}$ and using the fact that $\mathrm{u} = f(\mathrm{x}), \mathrm{y} = \mathrm{x}$ and $\varphi(\mathrm{x}) = f(\mathrm{x}) - \rho\mathrm{x}$ at equilibrium, we can get $\dot{S} = -(x-\mathrm{x})^\top((\varphi(x) - \varphi(\mathrm{x})) - \rho(y-\mathrm{y})^\top(y-\mathrm{y}) + (y-\mathrm{y})(u-\mathrm{u}))$. By assumption, $\mathrm{Hess}(U) \geq \rho I_{|\mathcal{V}|}$, meaning that it is easy to check that $\nabla\varphi(\mathrm{x}) = \mathrm{Hess}(U)(\mathrm{x}) - \rho\mathrm{Id} \geq 0$, implying that $-(x-\mathrm{x})^\top((\varphi(x)-\varphi(\mathrm{x})) \leq 0$. Thus, we can conclude that $\dot{S} \leq -\rho(y-\mathrm{y})^\top(y-\mathrm{y}) + (y-\mathrm{y})^\top(u-\mathrm{u})$, and hence the network (3.7) is EI-OPS with passivity parameter ρ. □

We now consider the network optimization framework for the diffusively-coupled systems $(\mathcal{G}, \Sigma, \Pi)$. One can compute that the steady-state input-output

© Technion - Israel Institute of Technology, Elyachar Central Library



(a) k_i (b) k_i^{-1} (c) $K_i^\star$

Figure 3.2: Steady-state maps and the associated integral function of the EI-OPS system Σ_i of Example 3.4. Both k_i and k_i^{-1} are non-monotone and the dual integral function $K_i^\star$ is non-convex.

relation for the controller is $\gamma_e(\zeta_e) = G\zeta_e$*, implying that the integral function for the controllers is given by* $\Gamma(\zeta) = \frac{G}{2}\zeta^2$*. Take the potential* U *as* $U(x_i) = r_1(1-\cos x_i) + \frac{1}{2}r_2 x_i^2, r_1 > 0, r_2 > 0$*,* $\forall i \in |\mathcal{V}|$*. We have that* $\mathrm{Hess}(U)(x_i) = r_1\cos x_i + r_2 \geq r_2 - r_1$*, so the agents are EI-OP(*ρ*) with parameter* $\rho = r_2 - r_1$*. The steady-state I/O map* k_i *of the systems* Σ_i *is given by the planar curve* $\mathrm{u}_i = r_1\sin\sigma + r_2\sigma$*;* $\mathrm{y}_i = \sigma$*, parameterized by the variable* σ*. It can be seen in Figure 3.2.*

It can be seen that k_i^{-1} *is a smooth function, so the integral function* $K_i^\star$ *can be defined so that* $\nabla K_i^\star = k_i^{-1}$*. We note that* $K_i^\star$ *is non-convex. However, we do note that* $K_i^\star$ *has a global minimum at* $y_i = 0$*. Thus, the minimum of the optimization problem (OPP), which is the unconstrained optimization problem of* $\sum_i K_i^\star(\mathrm{y}_i) + \Gamma(\mathcal{E}_\mathcal{G}^\top y)$*, will occur at* $y = \mathbf{0}$*.*

We choose $r_1 = 2.5, r_2 = 0.1$ *for all agents, and* $G = 1$*. The closed-loop system was run, and the resulting trajectories can be seen in Figure 3.3. As can be seen, the system converges to a steady-state different from* $\mathbf{0}$*, which was the steady-state predicted by the network optimization framework.*

A more thorough examination of the steady-state of the trajectories in Example 3.4 shows that all agents converge to local minima of the function $K_i^\star$, as seen in Figure 3.2. Actually, the closed-loop system converges to a local minimum of the cost function of (OPP), which is not the global minimum. One can show with methods similar to the ones used in Section 2.2 that if the closed-loop system does converge to a steady-state, it must be a critical point of the optimization problem. This is true as the critical point of the problem, by definition, satisfies the equations (2.2). For convex functions, all critical points are global minima, so the failure of the network optimization framework can be attributed to non-convexity of the network optimization problems, provided the agents are EI-IOP(ρ, ν).

© Technion - Israel Institute of Technology, Elyachar Central Library

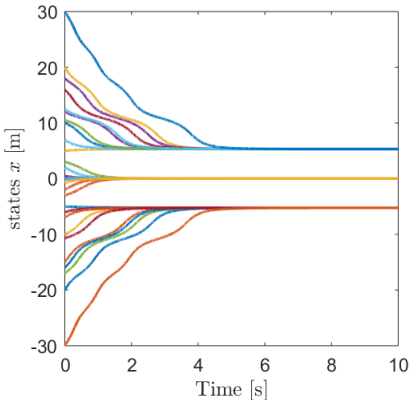


Figure 3.3: The trajectories of the closed-loop system from Example 3.4. The steady-state limit is not $\mathbf{0}$, which was the steady-state limit predicted by (OPP).

3.3 A Network Optimization Framework for Output Passive-Short Agents

Consider a diffusively-coupled network $(\mathcal{G}, \Sigma, \Pi)$. Assume that the agents are SISO and EI-OP(ρ_i) with passivity parameters $\rho_1, \cdots \rho_{|\mathbb{V}|}$, and that the controllers are MEIP. Motivated by the previous section, we try and regularize the network optimization problem (OPP) to make the network optimization framework valid. We consider the following technical assumption:

Assumption 3.1. *Each SISO dynamical system Σ_i is EI-OP(ρ_i). Moreover, we assume that the inverse steady-state relation k_i^{-1} is a function defined over $\mathbb{R}$. In this case, one can choose the integral functions for k_i^{-1} by taking any steady-state output y_0 and defining $K_i^\star(\mathrm{y}) = \int_{\mathrm{y}_0}^{\mathrm{y}} k_i^{-1}(\tilde{\mathrm{y}})d\tilde{\mathrm{y}}$, so that $K_i^\star$ are differentiable and $\nabla K_i^\star = k_i^{-1}$.*

Remark 3.5. *The technical assumption makes the presentation more streamlined, but it is not essential in order to prove that the agents have integral functions in this case. However, proving this claim without the assumption requires an extension of the notion of subdifferentials, which we avoid.*

Thus, if Assumption 3.1 holds, integral functions exist, and we can write the (non-convex) problem (OPP). We try and convexify it in order to amend the network optimization problem. In the next three subsections, we consider various methods of convexification of (OPP), resulting in different augmentations for the agents. The first will lead to a term that always successfully convexifies (OPP), but requires the agents to sense their own output. The second uses the network to regularize the agents, and only requires the agents to sense their output relative to their neighbors, $y_i - y_j$, but successfully convexifies (OPP) only if the average of the passivity indices ρ is positive. The third is a hybrid of

© Technion - Israel Institute of Technology, Elyachar Central Library

the previous two approaches, which always successfully convexifies (OPP), and requires no more than one agent to sense its own output.

3.3.1 Agent-Only Regularization

Consider the optimal potential problem (OPP), and add a Tikhonov type regularization term of the form $\frac{1}{2}\|y\|_\beta^2 = \frac{1}{2}\sum_{i=1}^{|\mathcal{V}|}\beta_i y_i^2$ [17]. The new optimization problem reads

$$\begin{aligned} \min_{\mathrm{y},\zeta} \quad & K^\star(\mathrm{y}) + \Gamma(\zeta) + \frac{1}{2}\mathrm{y}^\top \mathrm{diag}(\beta)\mathrm{y} \\ s.t. \quad & \mathcal{E}^\top \mathrm{y} = \zeta, \end{aligned} \tag{ROPP}$$

where $\beta = [\beta_1, \ldots, \beta_{|\mathcal{V}|}]^\top$ is a design parameter that will be specified later. By presenting this regularization term, we have two questions to answer. First, how can one interpret this augmentation on the system-theoretic level. Second, does this regularization term actually convexifies (OPP). Toward answering the first question, we consider the agents' augmented integral function from (ROPP):

$$\Lambda_i^\star(\mathrm{y}) = K_i^\star(\mathrm{y}) + \frac{1}{2}\beta_i \mathrm{y}_i^2. \tag{3.9}$$

We claim that this regularization term can be understood as an output-feedback term for each individual agent. Indeed:

Theorem 3.2. *Consider the SISO dynamical system Σ_i and suppose that Assumption 3.1 is satisfied. Let $\Lambda_i^\star(\mathrm{y})$, given by (3.9) be the augmented function of the (ROPP). Then the output-feedback control of the form*

$$u_i = v_i - \beta_i y_i, \tag{3.10}$$

with new exogenous input v_i for each $i \in \mathcal{V}$, gives rise to the integral function $\Lambda_i^\star(\mathrm{y})$, in the sense of Assumption 3.1. In other words, the augmented agent $\tilde{\Sigma}_i$ given by

$$\tilde{\Sigma}_i : \begin{cases} \dot{x}_i = f_i(x_i, v_i - \beta_i y_i), \\ y_i = h_i(x_i), \end{cases} \tag{3.11}$$

has a steady-state input-output relation, and $\Lambda_i^\star$ is the integral function of the inverse steady-state input-output relation.

Proof. Note that $\Lambda_i^\star(\mathrm{y})$ is a differentiable function, as it is the sum of the differentiable function $K_i^\star(\mathrm{y})$ and the quadratic function $\frac{1}{2}\beta_i \mathrm{y}_i^2$. Thus we can define an input-output relation λ_i such that, $\partial\Lambda_i^\star(\mathrm{y}_i) = \lambda_i^{-1}(\mathrm{v}_i)$ for any v_i, where λ_i^{-1} is the inverse relation of λ_i. Consequently, it follows from (3.9) that

$$\lambda_i^{-1}(\mathrm{y}_i) = k_i^{-1}(\mathrm{y}_i) + \beta_i \mathrm{y}_i, \tag{3.12}$$

© Technion - Israel Institute of Technology, Elyachar Central Library

If $(\mathrm{u_i}, \mathrm{y_i})$ is a steady-state input-output pair of Σ_i, and $\mathrm{v}_i = \lambda_i^{-1}(\mathrm{y}_i)$, then we get from (3.12) that in steady-state,

$$\mathrm{v}_i = \mathrm{u}_i + \beta_i \mathrm{y}_i, \tag{3.13}$$

meaning that $(\mathrm{v_i}, \mathrm{y_i})$ is a steady-state input-output pair of agent augmented by the output-feedback as in (3.10). This completes the proof. □

Example 3.5. *Consider the SISO agent of the form* $\dot{x} = -x + \sqrt[3]{x} + u; y = \sqrt[3]{x}$. *The steady-state input-output relation of the system is given by all pairs* (u, y) *so that* $\mathrm{u} = \mathrm{y}^3 - \mathrm{y}$, *as in steady-state,* $\mathrm{x} = \mathrm{y}^3$. *This implies that* $k^{-1}(\mathrm{y}) = \mathrm{y}^3 - \mathrm{y}$, *and thus the integral function is given by* $K^\star(\mathrm{y}) = \frac{1}{4}\mathrm{y}^4 - \frac{1}{2}\mathrm{y}^2$, *which is non-convex.*

We consider the augmented function $\Lambda^\star(\mathrm{y}) = \frac{1}{4}\mathrm{y}^4$, *which is convex, and can be found by adding a Tikhonov regularization term of the form* $\frac{1}{2}\beta\mathrm{y}^2$ *to* $K^\star(y)$, *as suggested above. By the proposition, it should induce the output-feedback* $u(t) = v(t) - y(t)$. *We verify that the closed-loop system of the agent and the feedback possesses* $\Lambda^\star$ *as an integral function. Indeed, the closed-loop system is* $\dot{x} = -x + \sqrt[3]{x} + u = -x + \sqrt[3]{x} + v - \sqrt[3]{x} = -x + v$, *where* $y = \sqrt[3]{x}$. *The new system has a steady-state input output relation* λ *satisfying* $\lambda^{-1}(\mathrm{y}) = \mathrm{y}^3$, *as the new steady-state input output relation consists of all pairs* (u, y) *such that* $\mathrm{u} = \mathrm{y}^3$. *Thus the integral function of the new system is* $\frac{1}{4}\mathrm{y}^4$, *as predicted by the theorem.*

This answers the first question - the regularization term corresponds to an output-feedback term on each agent $i \in \mathbb{V}$, with constant gain equal to β_i. It remains to show that the regularized problem (ROPP) is convex. We show that each function $\Lambda_i^\star$ is convex, and in fact, that the augmented agents $\tilde{\Sigma}_i$ are output-strictly MEIP.

Theorem 3.3. *Consider the SISO dynamical system* Σ_i *with steady-state input-output relation* k_i *and suppose that Assumption 3.1 is satisfied. Let* σ_i *be the passivity parameter of* Σ_i. *Then for any* $\beta_i > -\sigma_i$, *the feedback* (3.10) *defines a new dynamical system* $\tilde{\Sigma}_i$, *which is output-strictly MEIP. In particular, the function* $\Lambda_i^\star$ *is convex.*

Proof. Substituting (3.10) and (3.13) in (3.1), yields

$$\dot{S}_i \leq -\rho_i(y_i - \lambda_i(\mathrm{v}_i))^2 + (y_i - \lambda_i(\mathrm{v}_i))(v_i - \mathrm{v}_i), \tag{3.14}$$

where $\rho_i = \beta_i + \sigma_i > 0$ since $\beta_i > -\sigma_i$. Thus, the new dynamical system $\tilde{\Sigma}_i$ is output-strictly passive with respect to any steady-state input-output pair it possesses, so it is output-strictly MEIP if and only if its input-output relation, λ_i, is maximally monotone, which we prove next.

Plugging in v_i equal to another arbitrary constant $\hat{\mathrm{v}}_i$ and $y_i = \hat{\mathrm{y}}_i = \lambda_i(\hat{\mathrm{v}}_i)$, it follows from (3.14) that λ_i is a monotone relation, so we only need to consider its maximality. Rockafellar's Theorem [119] implies that λ_i is contained in the subdifferential of some convex function P_i, meaning that λ_i^{-1} is contained in the

© Technion - Israel Institute of Technology, Elyachar Central Library

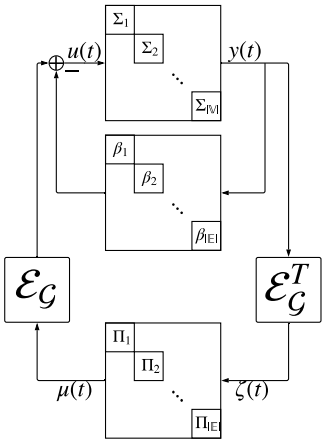


Figure 3.4: Block diagram for the closed-loop system induced by the agent-based regularization term.

subdifferential of the convex function $P_i^\star$. But $\lambda_i^{-1} = \nabla \Lambda_i^\star = \partial \Lambda_i^\star$, so uniqueness of the subgradient implies that $\Lambda_i^\star = P_i^\star$ up to an additive constant. In that case, $\Lambda_i^\star$ must be also convex, so $\lambda_i^{-1} = \partial \Lambda_i^\star$ implies that λ_i^{-1}, and hence λ_i, is actually *maximally* monotone. This concludes the proof. □

As the augmented agents $\tilde{\Sigma}_i$ are MEIP, we may use the network optimization framework for MEIP agents, namely Theorem 1.2, to conclude the following network optimization framework for EI-OP(ρ) systems:

Theorem 3.4. *Let $\{\Sigma_i\}_{i\in\mathbb{V}}$ be agents satisfying Assumption 3.1 with passivity indices $\rho_1, \cdots, \rho_{|\mathbb{V}|}$. Let $\{\Pi_e\}_{e\in\mathbb{E}}$ be MEIP controllers with stacked integral function $\Gamma(\zeta)$. Consider the closed-loop network with the controller input $\zeta = \mathcal{E}_{\mathcal{G}} y$ and the control input $u = -\mathcal{E}_{\mathcal{G}}\mu - \mathrm{diag}(\beta)y$. Then the closed-loop system converges to a steady-state. Moreover, the steady-state output* y *and $\zeta = \mathcal{E}_{\mathcal{G}}$*y *are optimal solutions to the problem (ROPP).*

The closed-loop system can be seen in Figure 3.4.

This output-feedback is the classical method of passivizing output-passive short systems [75], which we recover by considering the agent-based regularization term to (OPP). However, it has two drawbacks when considering multi-agent systems. First, all agents need to be able to sense their own output y_i in order to implement the induced control law. In some cases, agents cannot sense their own output in a global frame of reference, but only the output relative to other agents, $y_i - y_j$. This is the case for a multi-agent system comprised of robots operating in an area with no GNSS (Global Navigation Satellite System) reception. The agents can sense their relative position and/or velocity, but they do not know their own position and velocity in a global frame of reference. Second, even if the agents can sense their own output, they need not be amenable

© Technion - Israel Institute of Technology, Elyachar Central Library

to the network designer, and agree to implement the self-feedback considered. This is the case in open networks. Moreover, an adversary might inject an agent to the network which claims it implemented the self-feedback, but didn't, which will cause the multi-agent system to possibly diverge, or converge to a different limit than forecasted by the network optimization framework, implying that this augmentation is susceptible to attacks. Both problems can be addressed by considering a feedback term which relies on the network alone. In this direction, we explore network-only regularization terms in the next section.

3.3.2 Network-Only Regularization

We consider a different Tikhonov-type regularization term, of the form of $\frac{1}{2}\sum_{e\in\mathbb{E}}\beta_e\zeta_e^2$, depending only on the network variables ζ. It gives rise to the network-regularized optimal potential problem (NROPP):

$$\begin{aligned}\min_{\mathrm{y},\zeta}\quad & K^\star(\mathrm{y})+\Gamma(\zeta)+\frac{1}{2}\zeta^\top B\zeta\\ \text{s.t.}\quad & \mathcal{E}_{\mathcal{G}}^\top \mathrm{y}=\zeta,\end{aligned}\tag{NROPP}$$

where $B=\mathrm{diag}(\beta)=\mathrm{diag}\{\beta_1,\ldots,\beta_{|\mathbb{E}|}\}$ is a design parameter that will be appropriately chosen to make (NROPP) convex. We can consider the cost function of (NROPP) as the sum of two functions - the first is $\Gamma(\zeta)$, which is known to be convex. The second is $K^\star(\mathrm{y})+\frac{1}{2}\zeta^\top\mathrm{diag}(\beta)\zeta$. Recalling that $\zeta=\mathcal{E}_{\mathcal{G}}^\top\mathrm{y}$, we denote the latter as

$$\Lambda_N^\star(\mathrm{y})=K^\star(\mathrm{y})+0.5\mathrm{y}^\top\mathcal{E}_{\mathcal{G}}\mathrm{diag}(\beta)\mathcal{E}_{\mathcal{G}}^\top\mathrm{y}.\tag{3.15}$$

The following theorem proves that this new, regularized integral function for the agents is induced by a network consensus-type feedback.

Proposition 3.2. *Consider the agents Σ_i satisfying Assumption 3.1. Let $\Lambda_N^\star$, given by* (3.15)*, be the network-regularized integral function for the agents. Then $\Lambda_N^\star$ is differentiable. Moreover, consider the MIMO system $\tilde{\Sigma}$ given by the parallel interconnection of the agents $\{\Sigma_i\}_{i\in\mathbb{V}}$ with an output-feedback control of the form*

$$u=v-\mathcal{E}_{\mathcal{G}}\mathrm{diag}(\beta)\mathcal{E}_{\mathcal{G}}^\top y,\tag{3.16}$$

with some new exogenous input $v\in\mathbb{R}^n$, i.e., $\tilde{\Sigma}$ is modeled as the feedback-interconnection of $\dot{x}=f(x,u)$, $y=h(x,u)$ and (3.16)*. Let λ_N be its input-output steady-state relation. Then λ_N^{-1} is a function, and $\nabla\Lambda_N^\star=\lambda_N^{-1}$.*

Proof. The proof is similar to the proof of Theorem 3.2. The function $\Lambda_N^\star$ is differentiable as a sum of the differentiable functions $K^\star$ and $\frac{1}{2}\mathrm{y}^\top\mathcal{E}_{\mathcal{G}}\mathrm{diag}(\beta)\mathcal{E}_{\mathcal{G}}^\top\mathrm{y}$. Its derivative is given by

$$\nabla\Lambda_N^\star(\mathrm{y})=k^{-1}(\mathrm{y})+\mathcal{E}_{\mathcal{G}}\mathrm{diag}(\beta)\mathcal{E}_{\mathcal{G}}^\top\mathrm{y}.\tag{3.17}$$

© Technion - Israel Institute of Technology, Elyachar Central Library

If (u,y) is a steady-state input-output steady-state pair for the agents, and we denote $\mathrm{v} = \nabla\Lambda_N^\star(\mathrm{y})$, then (3.17) is equivalent to

$$\mathrm{v} = \mathrm{u} + \mathcal{E}_\mathcal{G}\mathrm{diag}(\beta)\mathcal{E}_\mathcal{G}^\top \mathrm{y}. \tag{3.18}$$

Rearranging the terms, we conclude that (v,y) is a steady-state input-output pair for the closed-loop system $\tilde{\Sigma}$ given by the agents with the network feedback as in (3.16), as $\mathrm{u} = \mathrm{v} - \mathcal{E}_\mathcal{G}\mathrm{diag}(\beta)\mathcal{E}_\mathcal{G}^\top \mathrm{y}$ and y are a steady-state input-output pair for the agents Σ. This completes the proof. □

In other words, Proposition 3.2 gives the following interpretation of (NROPP). It is the optimal potential problem (OPP) for the closed-loop system which is the feedback connection of the controllers Π with the augmented agents $\tilde{\Sigma}$, as seen in Figure 3.5. In the spirit of feedback connection of passive systems, and because the controllers Π are MEIP, we wish to understand when $\tilde{\Sigma}$ is passive.

Proposition 3.3. *Suppose that $R+\mathcal{E}_\mathcal{G}\mathrm{diag}(\beta)\mathcal{E}_\mathcal{G}^\top$ is positive-semi definite, where $R = \mathrm{diag}\{\rho_1,\ldots,\rho_{|\mathbb{V}|}\}$ with ρ_i the passivity index of Σ_i. Then $\tilde{\Sigma}$ is passive with respect to any steady-state input-output pair. Moreover, if the matrix is positive-definite, then $\tilde{\Sigma}$ is output-strict passivity.*

Proof. We take a steady-state input-output pair (v,y) for $\tilde{\Sigma}$, so that (u,y) is a steady-state input-output pair of Σ where $\mathrm{v} = \mathrm{u} + \mathcal{E}_\mathcal{G}\mathrm{diag}(\beta)\mathcal{E}_\mathcal{G}^\top \mathrm{y}$. If $S(x) = \sum_i S(x_i)$ is the sum of the storage functions for the agents Σ_i, then summing (3.1) over the agents gives

$$\dot{S} \le -(y-\mathrm{y})^\top R(y-\mathrm{y}) + (y-\mathrm{y})^\top(u-\mathrm{u}).$$

Substituting $u = v - \mathcal{E}_\mathcal{G}\mathrm{diag}(\beta)\mathcal{E}_\mathcal{G}^\top \mathrm{y}$ gives

$$\dot{S} \le -(y-\mathrm{y})^\top R(y-\mathrm{y}) + (y-\mathrm{y})^\top(v-\mathrm{v}) - (y-\mathrm{y})^\top\mathcal{E}_\mathcal{G}\mathrm{diag}(\beta)\mathcal{E}_\mathcal{G}^\top(y-\mathrm{y}).$$

Grouping R and $\mathcal{E}_\mathcal{G}\mathrm{diag}(\beta)\mathcal{E}_\mathcal{G}^\top$ completes the proof. □

We conclude the following theorem.

Theorem 3.5. *Let $\{\Sigma_i\}_{i\in\mathbb{V}}$ be agents satisfying Assumption 3.1 with passivity indices $\rho_1,...,\rho_{|\mathbb{V}|}$. Let $\{\Pi_e\}_{e\in\mathbb{E}}$ be MEIP controllers with stacked integral function Γ. Consider the closed-loop system with the controller input $\zeta = \mathcal{E}_\mathcal{G}^\top y$ and the control input $u = -\mathcal{E}_\mathcal{G}\mu - \mathcal{E}_\mathcal{G}\mathrm{diag}(\beta)\zeta$. If $R + \mathcal{E}_\mathcal{G}\mathrm{diag}(\beta)\mathcal{E}_\mathcal{G}^\top$ is positive-definite, then the closed-loop system converges to a steady-state. Moreover, the steady-state output* y *and $\zeta = \mathcal{E}_\mathcal{G}^\top \mathrm{y}$ are the optimal solutions to the problem (NROPP).*

Proof. By the discussion above, the closed-loop system is a feedback connection of the network-regularized agents $\tilde{\Sigma}$, which are output-strictly passive with respect to any steady-state they have, and the controllers Π, which are MEIP. Moreover, the augmented agents' steady-state input-output relation λ_N^{-1} is the gradient of the function $\Lambda_N^\star$. The proof now follows from Theorem 1.2 and Remark 2.7. □

© Technion - Israel Institute of Technology, Elyachar Central Library

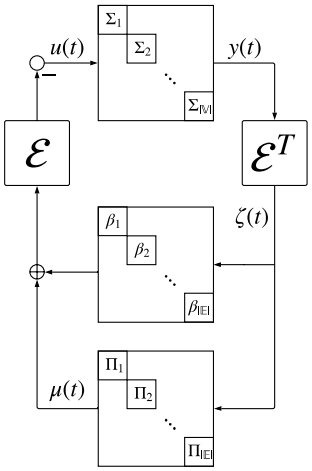


Figure 3.5: Block diagram for the closed-loop system induced by the network-based regularization term.

We now ask ourselves how to ensure $R + \mathcal{E}_\mathcal{G}\text{diag}(\beta)\mathcal{E}_\mathcal{G}^\top$ is positive-definite by appropriately choosing the gains β_e. To answer that question, we prove the following.

Theorem 3.6. *Let $\rho_1, ..., \rho_{|\mathbb{V}|}$ be any real numbers and assume $\mathcal{G}$ is connected. There exists some $\beta_1, ..., \beta_{|\mathbb{E}|}$ such that* $\text{diag}(\rho)+\mathcal{E}_\mathcal{G}\text{diag}(\beta)\mathcal{E}_\mathcal{G}^\top$ *is positive definite if and only if $\sum_{i\in\mathbb{V}} \rho_i$ is strictly positive.*

Proof. Suppose first that there exist some $\beta_1, ..., \beta_{|\mathbb{E}|}$ such that $X = R + \mathcal{E}_\mathcal{G}\text{diag}(\beta)\mathcal{E}_\mathcal{G}^\top$ is positive definite. Then $\mathbb{1}_{|\mathbb{V}|}^\top X \mathbb{1}_{|\mathbb{V}|} > 0$. However, $\mathcal{E}_\mathcal{G}^\top \mathbb{1}_{|\mathbb{V}|} = \mathbf{0}$, so $0 < \mathbb{1}_{|\mathbb{V}|}^\top X \mathbb{1}_{|\mathbb{V}|} = \mathbb{1}_{|\mathbb{V}|}^\top R \mathbb{1}_{|\mathbb{V}|} = \sum_i \rho_i$.

As for the other direction, suppose that $\sum_i \rho_i > 0$. We show that if b is large enough, then $R + b\mathcal{E}_\mathcal{G}\mathcal{E}_\mathcal{G}^\top$ is positive definite, which will conclude the proof as we can choose $\beta_e = b$. As the matrix in question is symmetric, it's enough to show that for any $x \in \mathbb{R}^{|\mathbb{V}|}$, $x^\top(R + b\mathcal{E}_\mathcal{G}\mathcal{E}_\mathcal{G}^\top)x \geq 0$.

We can write any vector $x \in \mathbb{R}^{|\mathbb{V}|}$ as $x = \alpha \mathbb{1}_{|\mathbb{V}|} + \mathcal{E}_\mathcal{G} z$ for $z \in \mathbb{R}^{|\mathbb{E}|}$ orthogonal to $\ker(\mathcal{E}_\mathcal{G})$. The quadratic form is

$$x^\top(R + b\mathcal{E}_\mathcal{G}\mathcal{E}_\mathcal{G}^\top)x = \alpha^2 \sum_i \rho_i + z^\top \mathcal{E}_\mathcal{G}^\top R(2\alpha \mathbb{1}_{|\mathbb{V}|} + \mathcal{E}_\mathcal{G} z) + b||\mathcal{E}_\mathcal{G}^\top \mathcal{E}_\mathcal{G} z||^2,$$

where we use $\mathcal{E}_\mathcal{G}^\top \mathbb{1}_{|\mathbb{V}|} = \mathbf{0}$. Now, $\mathcal{E}_\mathcal{G}^\top \mathcal{E}_\mathcal{G}$ is a positive semi-definite matrix, and z is orthogonal to its kernel, as $\ker(\mathcal{E}_\mathcal{G}^\top \mathcal{E}_\mathcal{G}) = \ker(\mathcal{E}_\mathcal{G})$. Thus $||\mathcal{E}_\mathcal{G}^\top \mathcal{E}_\mathcal{G} z|| \geq \lambda_{\min,\neq 0}(\mathcal{E}_\mathcal{G}^\top \mathcal{E}_\mathcal{G})||z||$, where $\lambda_{\min,\neq 0}(\mathcal{E}_\mathcal{G}^\top \mathcal{E}_\mathcal{G})$ is the minimal non-zero eigenvalue. Moreover, $\mathcal{E}_\mathcal{G}^\top \mathcal{E}_\mathcal{G}$ and $\mathcal{E}_\mathcal{G}\mathcal{E}_\mathcal{G}^\top$ share nonzero eigenvalues [66], hence the minimal nonzero eigenvalue of $\mathcal{E}_\mathcal{G}^\top \mathcal{E}_\mathcal{G}$ is $\lambda_2(\mathcal{G})$, the second lowest eigenvalue of the graph

© Technion - Israel Institute of Technology, Elyachar Central Library

Laplacian. Therefore the quadratic form is bounded from below by

$$\begin{aligned}
&\alpha^2 \sum_i \rho_i + 2\alpha z^\top \mathcal{E}_\mathcal{G}^\top R \mathbb{1}_{|\mathbb{V}|} + z^\top \mathcal{E}_\mathcal{G}^\top R \mathcal{E}_\mathcal{G} z + b\lambda_2^2(\mathcal{G})||z||^2 \\
&= \left\| \frac{\sqrt{\sum_i \rho_i}}{\sqrt{|\mathbb{V}|}} \alpha \mathbb{1}_{|\mathbb{V}|} + \frac{\sqrt{|\mathbb{V}|}}{\sqrt{\sum_i \rho_i}} R\mathcal{E}_\mathcal{G} z \right\|^2 \\
&+ z^\top \left(\mathcal{E}_\mathcal{G}^\top R \mathcal{E}_\mathcal{G} - \frac{|\mathbb{V}|}{\sum_i \rho_i} \mathcal{E}_\mathcal{G}^\top R^2 \mathcal{E}_\mathcal{G} + b\lambda_2(\mathcal{G})^2 \mathrm{Id} \right) z.
\end{aligned}$$

The first summand is non-negative, as it is a norm, and the second is positive if the symmetric matrix multiplying $z^\top$ and z is positive-definite (and $z \neq 0$), which is guaranteed if

$$b > \frac{\lambda_{\max}\left(\frac{|\mathbb{V}|}{\sum_i \rho_i} \mathcal{E}_\mathcal{G}^\top \mathrm{diag}(\rho)^2 \mathcal{E}_\mathcal{G} - \mathcal{E}_\mathcal{G}^\top \mathrm{diag}(\rho) \mathcal{E}_\mathcal{G} \right)}{\lambda_2(\mathcal{G})^2} := \mathbf{b},$$

where $\lambda_{\max}(\cdot)$ is the largest eigenvalue of a matrix. Note that in the case $z = 0$, $\alpha \neq 0$ and the norm is positive, so $x^\top (R + b\mathcal{E}_\mathcal{G}\mathcal{E}_\mathcal{G}^\top)x$ is positive as well. This completes the proof.

□

Remark 3.6. *Note that if $\mathcal{G}$ is not connected, the result of Theorem 3.6 holds if we require that the sum on each connected component is positive.*

Example 3.6. *One might expect that if we only demand positive-semi definiteness in Theorem 3.6, we might be able to accommodate $\sum_{i \in \mathbb{V}_c} \rho_i = 0$. However, this is not the case. Consider a two-agent case with $\rho_1 = 1$ and $\rho_2 = -1$. There is only one edge in the case,*

$$\mathrm{diag}(\rho) + \mathcal{E}_\mathcal{G} \mathrm{diag}(\beta) \mathcal{E}_\mathcal{G}^\top = \begin{bmatrix} 1+\beta & -\beta \\ -\beta & -1+\beta \end{bmatrix}.$$

This matrix can never be positive semi-definite, as its determinant is equal to $-1 < 0$. Thus, the agents cannot be passivized using the network.

Theorem 3.6 does not only characterize the diffusively-coupled systems that can be regularized using the network, but it also gives a prescription for network regularization, namely have a uniform gain of size $\mathbf{b} + \epsilon$, for some $\epsilon > 0$, over all edges. However, it shows that not all diffusively-coupled systems satisfying Assumption 3.1 can be network-passivized. This can be problematic in some applications, e.g. open networks in which the sign of the sum $\sum_i \rho_i$ might be volatile, meaning that there might be long periods in which we cannot provide a solution. For that reason, we consider a more general approach in the next subsection, fusing both agent-based and network-based regularization terms.

© Technion - Israel Institute of Technology, Elyachar Central Library

3.3.3 Hybrid Regularization

We return to the problem of regularizing the non-convex network optimization problem (OPP). Motivated by Subsections 3.3.1 and 3.3.2, we consider a quadratic regularization term containing both agent-based and netword-based regularization terms Namely, we consider the Hybrid-Regularized Optimal Potential Problem (HROPP),

$$\begin{aligned} \min_{\mathrm{y},\zeta} \quad & K^\star(\mathrm{y}) + \Gamma(\zeta) + \frac{1}{2}\zeta^\top \mathrm{diag}(\beta)\zeta + \frac{1}{2}\mathrm{y}^\top \mathrm{diag}(\alpha)\mathrm{y} \\ s.t. \quad & \mathcal{E}_\mathcal{G}^\top \mathrm{y} = \zeta. \end{aligned} \tag{HROPP}$$

As we'll see, unlike in Subsection 3.3.1, the vector $\alpha = [\alpha_1, ..., \alpha_{|\mathbb{V}|}]^\top$ can be very sparse. Namely, we can prove the following theorems.

Theorem 3.7. *Consider the agents Σ_i satisfying Assumption 3.1. Let $\Lambda_H^\star(\mathrm{y}) = K^\star(\mathrm{y}) + \frac{1}{2}\mathrm{y}^\top \mathcal{E}_\mathcal{G}\mathrm{diag}(\beta)\mathcal{E}_\mathcal{G}^\top \mathrm{y} + \frac{1}{2}\mathrm{y}^\top \mathrm{diag}(\alpha)\mathrm{y}$ be the augmented agents' integral function. Then $\Lambda_H^\star$ is differentiable. Moreover, consider the MIMO system $\tilde{\Sigma}$ given by the agents with the output-feedback control*

$$u = v - \mathcal{E}_\mathcal{G}\mathrm{diag}(\beta)\mathcal{E}_\mathcal{G}^\top y - \mathrm{diag}(\alpha)y, \tag{3.19}$$

with a new exogenous input $v \in \mathbb{R}^n$. Let λ_H be its steady-state input-output relation. Then λ_H^{-1} is a function, and $\nabla\Lambda_H^\star = \lambda_H^{-1}$.

Proof. Similar to the proof of Proposition 3.2. □

Theorem 3.8. *Let $\{\Sigma_i\}_{i\in\mathbb{V}}$ be agents satisfying Assumption 3.1. Let $\{\Pi_e\}_{e\in\mathbb{E}}$ be MEIP controllers with stacked integral function Γ. Consider the closed-loop system with the controller input $\zeta = \mathcal{E}_\mathcal{G}^\top y$ and the control input $u = -\mathcal{E}_\mathcal{G}\mu - \mathcal{E}_\mathcal{G}\mathrm{diag}(\beta)\zeta - \mathrm{diag}(\alpha)y$. If the matrix $\mathrm{diag}(\rho+\alpha) + \mathcal{E}_\mathcal{G}\mathrm{diag}(\beta)\mathcal{E}_\mathcal{G}^\top$ is positive-definite, then the closed-loop system converges. Moreover, the steady-state output y and $\zeta = \mathcal{E}_\mathcal{G}^\top \mathrm{y}$ are the optimal solutions to the problem (HROPP).*

Proof. Similar to the proof of Theorem 3.5. □

Corollary 3.1. *(Almost Network-Only Regularization) Let $\{\Sigma_i\}_{i\in\mathbb{V}}$ be agents satisfying Assumption 3.1, and suppose that the graph $\mathcal{G}$ is connected. Let $\mathbb{V}_{sr} \subseteq \mathbb{V}$ be any nonempty subset of the agents. Let $\{\Pi_e\}_{e\in\mathbb{E}}$ be MEIP controllers with stacked integral function Γ. Consider the closed-loop system with the controller input $\zeta = \mathcal{E}_\mathcal{G}^\top y$ and the control input $u = -\mathcal{E}_\mathcal{G}\mu - \mathcal{E}_\mathcal{G}\mathrm{diag}(\beta)\zeta - \mathrm{diag}(\alpha)y$. Then there exist vectors $\alpha \in \mathbb{R}^{|\mathbb{V}|}, \beta \in \mathbb{R}^{|\mathbb{E}|}$ such that:*

i) For any vertex $i \notin \mathbb{V}_{sr}, \alpha_i = 0$.

ii) The closed-loop system converges to a steady-state.

iii) The steady-state output y and $\zeta = \mathcal{E}_\mathcal{G}^\top \mathrm{y}$ are the optimal solutions to the optimization problem (HROPP).

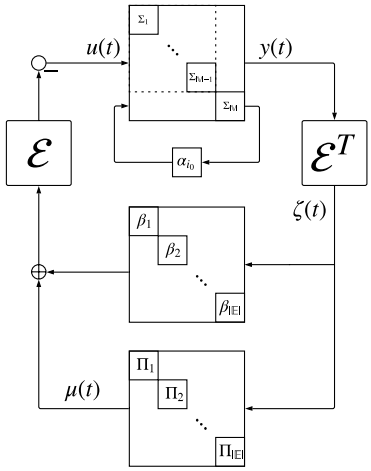


Figure 3.6: Block diagram for the closed-loop system induced by the hybrid regularization term with one self-regulating agent $\Sigma_{|\mathbb{V}|}$.

An example of the closed-loop system for a single self-regulating agent $\Sigma_{|\mathbb{V}|}$ can be seen in Figure 3.6.

Proof. By Theorem 3.8, it's enough to find some α, β satisfying the first condition such that $\mathrm{diag}(\rho+\alpha)+\mathcal{E}_{\mathcal{G}}\mathrm{diag}(\beta)\mathcal{E}_{\mathcal{G}}^{\top}$ is positive-definite. Fixing α, Theorem 3.6 implies that there is some β such that $\mathrm{diag}(\rho+\alpha)+\mathcal{E}_{\mathcal{G}}\mathrm{diag}(\beta)\mathcal{E}_{\mathcal{G}}^{\top}$ is positive-definite if and only if $\sum_i(\rho_i+\alpha_i)>0$, or $\sum_i \alpha_i > -\sum_i \rho_i$. Taking any $i_0 \in \mathbb{V}_{sr}$ and choosing $\alpha_i = 1-\sum_i \rho_i$ for $i = i_0$, and $\alpha_i = 0$ otherwise, completes the proof. □

Remark 3.7. *The set $\mathbb{V}_{sr}$ in the theorem can be thought of the set of vertices that can sense their own output, and are amenable to the network designer (i.e., self-regularizable agents). The theorem shows the strength of the hybrid approach for regularization of (OPP). We can choose almost all α_i-s to be zero - namely one agent is enough. In practice, this solution is less restrictive than the one offered in Subsections 3.3.1 and 3.3.2, as it makes no assumptions on the passivity indices of the agents, and requires only one agent to implement a self-feedback.*

Remark 3.8. *In all of the regularization procedures discussed in this section, we considered quadratic terms around* $\mathbf{0}$*. One could also consider quadratic terms of the form $\alpha_i(\mathrm{y}_i - v)^2$, or $\beta_e(\zeta_e - v)^2$, for some number v, which will give the same result using the same exact analysis. These regularization terms are of use when we wish to steer the system toward a certain steady-state, v. For example, in a traffic control model, using the term $\alpha_i \mathrm{y}_i^2$ would steer the i-th agent toward zero-velocity (as 0 is the minimizer of $\alpha_i \mathrm{y}_i^2$), which can be dangerous. This is*

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

demonstrated in the case study presented in Subsection 3.5.3. This idea is used in Chapter 5 to give a model-free solution to a certain synthesis problem.

3.4 A Network Optimization Framework for General Passive-Short Agents

Until now, we focused on the case that all agents are EI-OP(ρ_i), for some $\rho_i \in \mathbb{R}$, and explained how to rectify (OPP) so that it becomes convex, which resulted in a passivizing transformation for the agents, rendering the network optimization framework valid. One can act similarly in the case of EI-IP(ν_i) agents, replacing (OPP) with (OFP). In this section, we wish to broaden our viewpoint and consider general input-output (ρ, ν)-passive agents.

Consider Example 3.3, in which the agents are EI-IOP(ρ, ν) with passivity indices $\rho = -2/3$ and $\nu = -1/3$. In that example, the steady-state input-output relation of each of the agents was non-monotone enough so that the notion of integral functions could not have be defined, even when omitting the convexity requirement. Thus, the network optimization framework can't be directly applied, nor be defined. In particular, the convexification approach used in the previous section can not be applied.

We take a different approach. Instead of convexifying the integral function (which is undefined), we opt to monotonize the steady-state input-output relation, a first step in transforming the agent to being MEIP, which is the route for applying the network optimization framework. The idea can be seen pictorially in Figure 3.7. We start by studying how steady-state input-output relations alter when applying linear transformations, and how can they be made monotone.

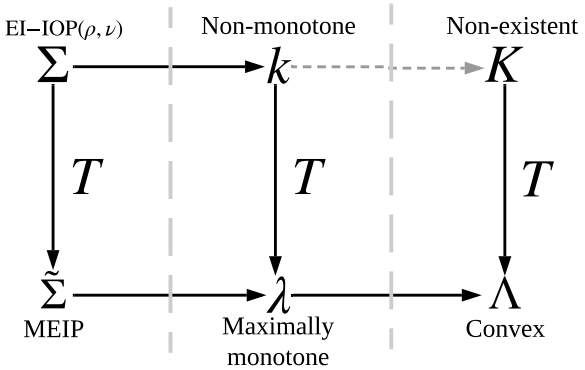


Figure 3.7: Passivation, monotonization and convexification by transformation.

© Technion - Israel Institute of Technology, Elyachar Central Library

3.4.1 Monotonization of I/O Relations by Linear Transformations: A Geometric Approach

Consider the dynamical system $\Sigma : u \mapsto y$ and suppose it is EI-IOP(ρ, ν). Let k be the steady-state I/O relation of Σ. Our goal is to find a monotonizing transformation $T : (\mathrm{u}, \mathrm{y}) \mapsto (\tilde{\mathrm{u}}, \tilde{\mathrm{y}})$ for k. Namely, we look for a linear transformation T of the form $\begin{bmatrix}\tilde{\mathrm{u}}\\ \tilde{\mathrm{y}}\end{bmatrix} = T\begin{bmatrix}\mathrm{u}\\ \mathrm{y}\end{bmatrix}$. We can use the EI-IOP$(\rho, \nu)$ assumption to deduce some information on k.

Proposition 3.4. *Let k be the steady-state I/O relation of Σ, which is EI-IOP(ρ, ν). Then for any two points $(\mathrm{u}_1, \mathrm{y}_1)$, $(\mathrm{u}_2, \mathrm{y}_2)$ in k, the following inequality holds:*

$$0 \leq -\rho(\mathrm{y}_1 - \mathrm{y}_2)^2 + (\mathrm{u}_1 - \mathrm{u}_2)(\mathrm{y}_1 - \mathrm{y}_2) - \nu(\mathrm{u}_1 - \mathrm{u}_2)^2. \tag{3.20}$$

Proof. Consider inequality (3.3) for the steady-state input-output pair $(\mathrm{u}_1, \mathrm{y}_1)$, and let $S(x)$ be the corresponding storage function. The steady-state input-output pair $(\mathrm{u}_2, \mathrm{y}_2)$ corresponds to some steady state x_2, so that $(\mathrm{u}_2, \mathrm{x}_2, \mathrm{y}_2)$ is an (equilibrium) trajectory of the system. Inserting this into (3.3), and noting that $S(\mathrm{x}_2)$ is a fixed number, we conclude the inequality (3.20) holds. □

The proposition suggests the following definition:

Definition 3.4. *A* projective quadratic inequality *(PQI) is a an inequality in the variables ξ, χ of the form*

$$0 \leq a\xi^2 + b\xi\chi + c\chi^2, \tag{3.21}$$

for some numbers a, b, c, not all zero. The inequality is called non-trivial if $b^2 - 4ac > 0$. The associated solution set of the PQI is the set of all point $(\xi, \chi) \in \mathbb{R}^2$ satisfying the inequality.

By Definition 3.4, it is clear that (3.20) is a PQI. Indeed, plugging $\xi = \mathrm{u}_1 - \mathrm{u}_2$ and $\chi = \mathrm{y}_1 - \mathrm{y}_2$ and choosing a, b, c correctly verifies this. The demand $\rho\nu < \frac{1}{4}$ is equivalent to the non-triviality of the PQI. For example, monotonicity of the steady-state k can be written as $0 \leq (\mathrm{u}_1 - \mathrm{u}_2)(\mathrm{y}_1 - \mathrm{y}_2)$, which can be transformed to a PQI by choosing $a = c = 0$ and $b = 1$ in (3.21). Similarly, strict monotonicity can be modeled by taking $b = 1$ and $a \leq 0, c < 0$ or $b = 1$ and $a < 0, c \leq 0$.

As for transformations, the linear transformation T of the form

$$\begin{bmatrix}\tilde{\mathrm{u}}\\ \tilde{\mathrm{y}}\end{bmatrix} = T\begin{bmatrix}\mathrm{u}\\ \mathrm{y}\end{bmatrix}$$

can be written as

$$\begin{aligned}\tilde{\mathrm{u}} &= T_{11}\mathrm{u} + T_{12}\mathrm{y}\\ \tilde{\mathrm{y}} &= T_{21}\mathrm{u} + T_{22}\mathrm{y}.\end{aligned}$$

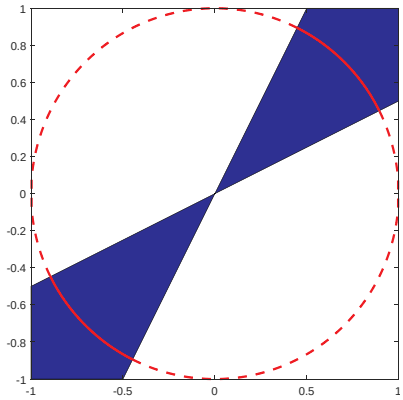


Figure 3.8: A double cone (in blue), and the associated symmetric section (in solid red). The parts of $\mathbb{S}^1$ outside the symmetric section are presented by the dashed red line.

Plugging it inside (3.20) gives some other PQI. More precisely, if we let $f(\xi,\chi) = a\xi^2 + b\xi\chi + c\chi^2$, and T is a linear map, then T maps to PQI $f(\xi,\chi) \geq 0$ to $f(T^{-1}(\tilde{\xi},\tilde{\chi})) \geq 0$. Our goal is to find a map T which transforms an inequality of the form in Definition 3.4 to the PQI which corresponds to monotonicity. Thus, we are compelled to consider the action of the group of linear transformations on the collection of PQIs.

Let $\mathcal{A}$ be the solution set of the original PQI. The connection between the original and transformed PQI described above shows that the solution set of the new PQI is $T(\mathcal{A}) = \{T(\xi,\chi) : \ (\xi,\chi) \in A\}$, implying we can study the effect of linear transformations on PQIs by studying their actions on the solution sets. The action of the group of linear transformations on the collection of PQIs can be understood algebraically, but we use the notion of the solution set to understand it geometrically. We begin by giving a geometric characterization of the solution sets.

Note 3.1. *In this subsection, we abuse notation and identify the point* $(\cos\theta, \sin\theta)$ *on the unit circle* $\mathbb{S}^1$ *with the angle* θ *in some segment of length* 2π.

Definition 3.5. *A* symmetric section S *on the unit circle* $\mathbb{S}^1 \subseteq \mathbb{R}^2$ *is defined as the union of two closed disjoint sections that are opposite to each other, i.e.,* $S = B \cup (B + \pi)$ *where* B *is a closed section of angle* $< \pi$. *A* symmetric double-cone *is defined as* $A = \{\lambda s : \ \lambda > 0, s \in \mathbb{R}\}$ *for some symmetric section* S.

An example of a symmetric section and the associated symmetric double-cone can be seen in Figure 3.8.

Theorem 3.9. *The solution set of any non-trivial PQI is a symmetric double-cone. Moreover, any symmetric double-cone is the solution set of some non-trivial PQI, which is unique up to a positive multiplicative constant.*

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

The proof of the theorem will be postponed to the end of this subsection. This theorem presents a geometric interpretation of the steady-state condition (3.20). The relation between cones and measures of passivity is best known for static systems through the notion of sector-bounded nonlinearities [75]. It was expanded to more general systems in [173], and later in [90]. We consider a different branch of this connection, focusing on the steady-state relation rather on trajectories. In turn, it allows us to have intuition when constructing monotonizing maps. In particular, we have the following result.

Theorem 3.10. *Let (ξ_1, χ_1) and (ξ_2, χ_2) be two non-colinear solutions of $a_1\xi^2 + \xi\chi + c_1\chi^2 = 0$. Moreover, let (ξ_3, χ_3) and (ξ_4, χ_4) be two non-colinear solutions of $a_2\xi^2 + \xi\chi + c_2\chi^2 = 0$. Define*

$$T_1 = \begin{bmatrix} \xi_3 & \xi_4 \\ \chi_3 & \chi_4 \end{bmatrix} \begin{bmatrix} \xi_1 & \xi_2 \\ \chi_1 & \chi_2 \end{bmatrix}^{-1}, \quad T_2 = \begin{bmatrix} \xi_3 & -\xi_4 \\ \chi_3 & -\chi_4 \end{bmatrix} \begin{bmatrix} \xi_1 & \xi_2 \\ \chi_1 & \chi_2 \end{bmatrix}^{-1}.$$

Then one of the maps T_1, T_2 transforms the PQI $a_1\xi^2 + \xi\chi + c_1\chi^2 \geq 0$ to the PQI $\tau a_2\xi^2 + \tau\xi\chi + \tau c_2\chi^2 \geq 0$ for some $\tau > 0$.

The non-colinear solutions of the equations correspond to the straight lines forming the boundary of the symmetric double-cone, thus can be found geometrically. Moreover, as will be evident from the proof, knowing which one of T_1 and T_2 works is possible by checking the PQIs on $(\xi_1 + \xi_2, \chi_1 + \chi_2)$ and $(\xi_3 + \xi_4, \chi_3 + \chi_4)$. Namely, if they both satisfy or both don't satisfy the PQIs, then T_1 works, and otherwise T_2 works.

Proof. It's enough to show that either T_1 or T_2 maps the solution set of $a_1\xi^2 + b_1\xi\chi + c_1\chi^2 \geq 0$ to the solution set of $a_2\xi^2 + b_2\xi\chi + c_2\chi^2 \geq 0$. Let $\mathcal{A}_1$ be the solution set of the first PQI, and let $\mathcal{A}_2$ be the solution set of the second PQI. We note that $T_1(\mathcal{A}_1)$ and $T_2(\mathcal{A}_1)$ are symmetric double-cones, whose boundary is the image of the boundary of $\mathcal{A}_1$ under T_1 and T_2 respectively, i.e., they are the image of $\text{span}\{(\xi_1, \chi_1)\} \cup \text{span}\{(\xi_2, \chi_2)\}$ under T_1, T_2. We note that T_1 maps (ξ_1, χ_1),(ξ_2, χ_2) to (ξ_3, χ_3),(ξ_4, χ_4) correspondingly, and that T_2 maps (ξ_1, χ_1),(ξ_2, χ_2) to (ξ_3, χ_3),$(-\xi_4, -\chi_4)$ correspondingly. Thus, $\text{span}\{(\xi_1, \chi_1)\} \cup \text{span}\{(\xi_2, \chi_2)\}$ is mapped by T_1 and T_2 to $\text{span}\{(\xi_3, \chi_3)\} \cup \text{span}\{(\xi_4, \chi_4)\}$, so that $T_1(\mathcal{A}_1), T_2(\mathcal{A}_1)$ have the same boundary as $\mathcal{A}_2$. As T_1, T_2 are homeomorphisms, they map interior points to interior points, so it's enough to show that some point in the interior of $\mathcal{A}_1$ is mapped to a point in $\mathcal{A}_2$ either by T_1 or by T_2, or that a point in the interior of $\mathbb{R}^2 \setminus \mathcal{A}_1$ is mapped to a point in $\mathbb{R}^2 \setminus \mathcal{A}_2$ either by T_1 or by T_2.

Consider the point $(\xi_1 + \xi_2, \chi_1 + \chi_2)$. By non-colinearity, this point cannot be on the boundary of $\mathcal{A}_1$, which is equal to $\text{span}\{(\xi_1, \chi_1)\} \cup \text{span}\{(\xi_2, \chi_2)\}$. Thus, it's either in the interior of $\mathcal{A}_1$ or in the interior of its complement. We assume the prior case, as the proof for the other is similar. The point $(\xi_1 + \xi_2, \chi_1 + \chi_2)$ is mapped to $(\xi_3 \pm \xi_4,\ \chi_3 \pm \chi_4)$ by T_1,T_2. By non-colinearity, these points do not lie on the boundary of $\mathcal{A}_2$. Moreover, the line passing through them is parallel to $\text{span}\{(\xi_4, \chi_4)\}$ which is part of the boundary of $\mathcal{A}_2$, and their average

is (ξ_3, χ_3), which is on the boundary. Thus one point is in the interior of $\mathcal{A}_2$, and one is in the interior of its complement. This completes the proof. □

Example 3.7. *Consider the system Σ studied in Example 3.3, in which the steady-state input-output relation was non-monotone. There, we saw that the system is EI-IOP(ρ, ν) with parameters $\rho = -2/3$ and $\nu = -1/3$. The corresponding PQI is $0 \leq \frac{1}{3}\xi^2 + \xi\chi + \frac{2}{3}\chi^2$. We use Theorem 3.10 to find a monotonizing transformation. That is, we seek a transformation mapping the given PQI to the PQI defining monotonicity, i.e., $\xi\chi \geq 0$. We take $(\xi_3, \chi_3) = (1, 0)$ and $(\xi_4, \chi_4) = (0, 1)$, as these are non-colinear solutions to $\xi\chi = 0$. As for the points corresponding to the original PQI, $0 = \frac{1}{3}\xi^2 + \xi\chi + \frac{2}{3}\chi^2$ can be rewritten as $\frac{1}{3}(\xi+\chi)(\xi+2\chi) = 0$. Thus we can take $(\xi_1, \chi_1) = (2, -1)$ and $(\xi_2, \chi_2) = (-1, 1)$. It's easy to check that $(\xi_1 + \chi_1, \xi_2 + \chi_2) = (1, 0)$ satisfies the original PQI $0 \leq \frac{2}{3}\xi^2 + \xi\chi + \frac{1}{3}\chi^2$, so the map T_1^{-1}, where T_1 is defined as in the Theorem 3.10, should monotonize the steady-state relation. Plugging in T_1, we get*

$$T_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix},$$

so that

$$T_1^{-1} = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix},$$

and

$$\begin{bmatrix} \xi \\ \chi \end{bmatrix} = T_1^{-1} \begin{bmatrix} \tilde{\xi} \\ \tilde{\chi} \end{bmatrix}.$$

Then,

$$\begin{aligned} 0 \leq & \frac{1}{3}\xi^2 + \xi\chi + \frac{2}{3}\chi^2 = \frac{1}{3}(2\hat{\xi} - \hat{\chi})^2 + (2\hat{\xi} - \hat{\chi})(-\hat{\xi} + \hat{\chi}) + \frac{2}{3}(-\hat{\xi} + \hat{\chi})^2 \\ = & \left(\frac{4}{3} - 2 + \frac{2}{3}\right)\hat{\xi}^2 + \left(-\frac{4}{3} + 3 - \frac{4}{3}\right)\hat{\xi}\hat{\chi} + \left(\frac{1}{3} - 1 + \frac{2}{3}\right)\hat{\chi}^2 = \frac{1}{3}\hat{\xi}\hat{\chi} \end{aligned}$$

so the transformed PQI can also be written as $0 \leq \hat{\xi}\hat{\chi}$, which corresponds to monotonicity. To get the transformed steady-state relation, we recall that the steady-state relation of Σ is given by the planar curve $\mathrm{u} = 2\sigma - \sigma^3$; $\mathrm{y} = \sigma^3 - \sigma$, parameterized by a variable σ. The transformed relation is given by:

$$\begin{bmatrix} \tilde{\mathrm{u}} \\ \tilde{\mathrm{y}} \end{bmatrix} = T_1 \begin{bmatrix} \mathrm{u} \\ \mathrm{y} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 2\sigma - \sigma^3 \\ \sigma^3 - \sigma \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma^3 \end{bmatrix},$$

and can be modeled as $\tilde{\mathrm{y}} = \tilde{\mathrm{u}}^3$, which is a monotone relation.

The above theorem prescribes a monotonizing transformation for the relation k. Moreover, it prescribes a transformation which forces strict monotonicity, which can be viewed as the PQI $-\rho\xi^2 + \xi\chi - \nu\chi^2 \geq 0$ for some $\rho, \nu \geq 0$ which are not both zero. Before moving on, we repay our debt and prove Theorem 3.9.

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

Proof. Consider a PQI $a\xi^2+b\xi\chi+c\chi^2 \geq 0$. If $a=c=0$ and $b\neq 0$, the solution set is either the union of the first and third quadrants, or the union of the second and fourth quadrants (depending whether $b>0$ or $b<0$). Moreover, it is a symmetric double-cone in both these cases. Thus, we can assume that either $a\neq 0$ or $c\neq 0$. By switching the roles of ξ and χ, we may assume, without loss of generality, that $a\neq 0$. Note that if (ξ,χ) is a solution to the PQI, and $\lambda\in\mathbb{R}$, then $(\lambda\xi,\lambda\chi)$ is also a solution to the PQI. Thus, it's enough to show that the intersection of the solution set with the unit circle is a symmetric section. Writing a general point in $\mathbb{S}^1$ as $(\cos\theta,\sin\theta)$, the inequality becomes

$$a\cos^2\theta + b\cos\theta\sin\theta + c\sin^2\theta \geq 0.$$

We assume, for a moment, that $\cos\theta\neq 0$, and divide by $\cos^2\theta$. The inequality becomes

$$a\tan^2\theta + b\tan\theta + c \geq 0.$$

We denote $t_\pm = \frac{-b\pm\sqrt{b^2-4ac}}{2a}$ and consider two possible scenarios:

i) $a<0$: In that case, the inequality holds only when $\tan\theta$ is between t_+ and t_-. As tan is a monotone ascending function in $(-\pi/2,\pi/2)$ and $(\pi/2,1.5\pi)$, and tends to infinite values at the limits of said intervals, we conclude that the inequality holds only when θ is inside $I_1\cup I_2$, where I_1,I_2 are the closed intervals which are the image of $[t_-,t_+]$ under $\arctan(x)$ and $\arctan(x)+\pi$, so that $I_2=I_1+\pi$. Note that as $a<0$, any point at which $\cos\theta=0$ does not satisfy the inequality. Thus the intersection of the solution set with $\mathbb{S}^1$ is a symmetric section.

ii) $a>0$: In that case, the inequality holds only when $\tan\theta$ is outside the interval (t_-,t_+). Similarly to the previous case, $\tan\theta\in(t_-,t_+)$ can be written as $B\cup(B+\pi)$ where B is an *open* section of angle $<\pi$. Thus its complement, which is the intersection of the solution set with $\mathbb{S}^1$, is a symmetric section.

Conversely, consider a symmetric double-cone A, and let $S=B\cup(B+\pi)$ be the associated symmetric section. Let $C\cup(C+\pi)$ be the complement of S inside $\mathbb{S}^1$, where C is an open section. We first claim that $\cos\theta\neq 0$ either on B or on C. Indeed, $B\cup C$ is a half-open half-circle, and the only points at which $\cos\theta=0$ are $\theta=\pm\pi/2$. Thus, $B\cup C$ can only contain one of them. Moreover, B and C are disjoint, so at least one does not include points at which $\cos\theta\neq 0$. Now, we consider two possible cases.

i) B (hence S) contains no points at which $\cos\theta=0$. Then tan maps B continuously into some interval $I=[t_-,t_+]$. Thus $\theta\in S$ if and only if $-(\tan\theta-t_-)(\tan\theta-t_+)\geq 0$. Inverting the process from the first part of the proof, we can rewrite this as a PQI, such that the intersection of its solution set with $\mathbb{S}^1$ is equal to S, meaning that its solution set is equal to A. Non-triviality of the PQI follows from the fact that $t_\pm$ are two distinct solutions to the associated equation.

ii) C contains no points at which $\cos\theta = 0$. Then tan maps C continuously into some interval $I = (t_-, t_+)$. Thus, $\theta \in C \cup (C + \pi)$ if and only if $(\tan\theta - t_-)(\tan\theta - t_+) < 0$. Equivalently, $\theta \in S$ if and only if $(\tan\theta - t_-)(\tan\theta - t_+) \geq 0$. Inverting the process from the first part of the proof, we can rewrite this as a PQI, such that the intersection of its solution set with $\mathbb{S}^1$ is equal to S, meaning that its solution set is equal to A. Non-triviality follows from the fact that $t_\pm$ are two distinct solutions to the associated equation.

As for uniqueness, note that if the non-trivial PQIs $a_1\xi^2 + b_1\xi\chi + c_1\chi^2 \geq 0$ and $a_2\xi^2 + b_2\xi\chi + c_2\chi^2 \geq 0$ define the same solution set, then the equations $a_1\xi^2 + b_1\xi\chi + c_1\chi^2 = 0$ and $a_2\xi^2 + b_2\xi\chi + c_2\chi^2 = 0$ have the same solutions (as the boundaries of the solution sets). In particular, for $\xi = \tau\chi$, both equations $\chi^2(a_1\tau^2 + b_1\tau + c_1) = 0$ and $\chi^2(a_2\tau^2 + b_2\tau + c_2) = 0$ have the same solutions. If a_1 or a_2 is non-zero, then dividing by χ^2, as $b_1^2 - 4a_1c_1 > 0$ and $b_2^2 - 4a_2c_2 > 0$, both equations have two solutions, t_-, t_+. Thus, we can write:

$$a_1\tau^2 + b_1\tau + c_1 = a_1(\tau - t_-)(\tau - t_+),$$
$$a_2\tau^2 + b_2\tau + c_2 = a_2(\tau - t_-)(\tau - t_+),$$

implying the original PQIs are the same up to scalar, which must be positive due to the direction of the inequalities. If $a_1 = a_2 = 0$, then we must have $b_1, b_2 \neq 0$, as otherwise $b_1^2 - 4a_1c_1 = 0$ or $b_2^2 - 4a_2c_2 = 0$. Now, for $\chi = 1$, we get that the equations $b_1\xi + c_1 = 0$ and $b_2\xi + c_2 = 0$ have the same solutions, implying that (b_1, c_1) and (b_2, c_2) are equal up to a multiplicative scalar. As $a_1 = a_2 = 0$, we conclude the same about the original PQIs. Moreover, the scalar has to be positive due to the direction of the original PQIs. This completes the proof. □

3.4.2 From Monotonization to Passivation and Implementation

Until now, we found a map $T : \mathbb{R}^2 \to \mathbb{R}^2$, monotonizing the steady-state relation k. We claim that the map T, in fact, augments the system Σ to be output-strictly passive with respect to any equilibrium it has, by defining a new input and output by

$$\begin{bmatrix}\tilde{u}\\ \tilde{y}\end{bmatrix} = T\begin{bmatrix}u\\ y\end{bmatrix}.$$

Proposition 3.5. *Suppose that Σ is EI-IOP(ρ, ν), and let T be a linear map transforming the PQI $\rho\xi^2 + \xi\chi + \nu\chi^2 \geq 0$ to $\rho'\xi^2 + \xi\chi + \nu'\chi^2 \geq 0$, as built in Theorem 3.10. Consider the augmented system with input and output. defined by*

$$\begin{bmatrix}\tilde{u}\\ \tilde{y}\end{bmatrix} = T\begin{bmatrix}u\\ y\end{bmatrix}.$$

Then the augmented system is EI-IOP(ρ', ν'). In particular, if the map T monotonizes the relation k, then it passivizes Σ.

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

Proof. The inequality defining EI-IOP(ρ,ν) for ρ,ν is the PQI $\rho\xi^2+\xi\chi+\nu\chi^2\geq 0$, where we put $\xi=y(t)-\mathrm{y}$ and $\chi=u(t)=\mathrm{u}$ for some trajectory $(u(t),y(t))$ and steady-state I/O pair (u,y). The proposition now follows from noting that

$$\begin{bmatrix}\xi\\ \chi\end{bmatrix}=T^{-1}\begin{bmatrix}\tilde{\xi}\\ \tilde{\chi}\end{bmatrix},$$

satisfies the PQI $\rho'\xi^2+\xi\chi+\nu'\chi^2\geq 0$, $\tilde{\xi}=\tilde{y}(t)-\tilde{\mathrm{y}}$ and $\tilde{\chi}=\tilde{u}(t)-\tilde{\mathrm{u}}$. □

The proposition shows that the monotonizing linear transformation from Theorem 3.10 augments the plant Σ to another plant, $\tilde{\Sigma}$, which is passive with respect to any steady-state I/O pair it has.

Remark 3.9. *Proposition 3.5, together with Subsection 3.4.1, prescribes a linear transformation that passivizes the agent with respect to all equilibria. We note that the same procedure can be applied to "classical" passivity, in which one only looks at passivity with respect to a single equilibrium. However, our approach is entirely geometric and does no rely on algebraic manipulations. As we saw in Section 3.2 that, our approach is more general than the one in [169], as our approach does not demand finite-$\mathcal{L}_2$-gain stability of the passive-short system Σ.*

The main upshot of the geometric approach over the classical algebraic approach, besides its intuition stemming from sector-bounded nonlinearities and geometry, is its simplicity and generality. The algebraic approach, like the one presented in [169], relies on a collection of inequalities between the entries of the passivizing matrix, depending on the passivity parameters of the original system and the desired passivity parameters of the augmented system. Even if one shows that these inequalities have a solution theoretically, finding a solution can be hard, especially if the passivity parameters of the plant are not exactly known. On the contrary, the geometric approach relies on choosing four points by solving two quadratic equations, as described in Theorem 3.10, and checking if two specific PQIs are satisfied at specific points, which is simpler for a computer to do.

Moreover, it is evident from the discussion above that if the linear transformation forces k to become *strictly* monotone, then the augmented agent is *strictly* passive. For the remainder of this section, we show that the augmentation can be easily implemented using standard control tools, namely gains, feedback and feed-through. We also wish to connect the steady-state I/O relation of the augmented system $\tilde{\Sigma}$, denoted by λ, to k.

In this direction, take any linear map $T:\mathbb{R}^2\to\mathbb{R}^2$ of the form

$$T=\begin{bmatrix}a & b\\ c & d\end{bmatrix},$$

where we assume that $\det(T)\neq 0$. It defines the plant augmentation of the form

$$\begin{bmatrix}\tilde{u}\\ \tilde{y}\end{bmatrix}=T\begin{bmatrix}u\\ y\end{bmatrix}. \tag{3.22}$$

© Technion - Israel Institute of Technology, Elyachar Central Library

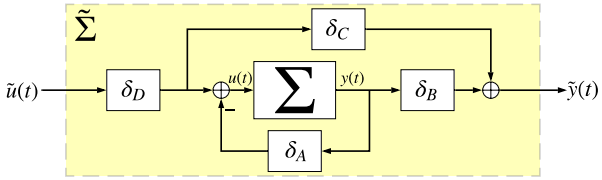


Figure 3.9: The transformed system $\tilde{\Sigma}$ after the linear transformation T. If $T = \left[\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right]$, then $\delta_A = b/a, \delta_B = d - \frac{b}{a}c, \delta_C = c$ and $\delta_D = a$.

For simplicity of presentation, we assume that $a \neq 0$.[2] We note that T can be written as a product of elementary matrices, and the effect of each elementary matrix on Σ_i can be easily understood. By applying each elementary transformation sequentially, the effect of their product, that is the transformation T, can be realized. Table 3.1 summaries these elementary transformations and their effect on the system Σ. Following Table 3.1, T can be written as

$$T = \begin{bmatrix} a & b \\ c & d \end{bmatrix} = L_D L_C L_B L_A, \tag{3.23}$$

with $\delta_A = b/a, \delta_B = d - \frac{b}{a}c, \delta_C = c$ and $\delta_D = a$. The product of these matrices can be seen as the sequential transformation to the original system Σ, which can be understood as a loop-transformation, illustrated in Figure 3.9.

Remark 3.10. *Writing $T = L_D L_C L_B L_A$ allows us to give a closed form description of the transformed system. Suppose the original system is given by $\dot{x} = f(x,u);\ y = h(x)$. Applying L_A gives a new input v, and the augmented system $\dot{x} = f(x, v - \delta_A h(x)); y = h(x)$. Applying L_B on this system gives $\dot{x} = f(x, v - \delta_A h(x)); y = \delta_B h(x)$. Applying L_C then gives $\dot{x} = f(x, v - \delta_A h(x)); y = \delta_B h(x) + \delta_C v$, and applying L_D finally gives $\dot{x} = f(x, \delta_D v - \delta_A h(x)); y = \delta_B h(x) + \delta_C \delta_D v$.*

Example 3.8. *Suppose T is given by $T = \left[\begin{smallmatrix} 1 & 0 \\ c & d \end{smallmatrix}\right]$, or $\tilde{u} = u$ and $\tilde{y} = cu + dy$. The steady-state I/O relations λ and k of the systems $\tilde{\Sigma}$ and Σ respectively, are connected as*

$$\lambda(\tilde{u}) = c\tilde{u} + dk(\tilde{u}). \tag{3.24}$$

On the other hand, we note that

$$\begin{bmatrix} 1 & 0 \\ c & d \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ c & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & d \end{bmatrix} = L_C L_B,$$

as defined in Table 3.1, with $\delta_C = c$, and $\delta_B = d$. Define

$$\begin{bmatrix} u_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & d \end{bmatrix} \begin{bmatrix} u \\ y \end{bmatrix},$$

[2] We note that by switching the names of (ξ_3, χ_3) and (ξ_4, χ_4) in Theorem 3.10, we switch the two columns of T. Thus we can always assume that $a \neq 0$, as $a = b = 0$ cannot hold due to the determinant condition.

Technion - Israel Institute of Technology, Elyachar Central Li

Elementary Transformation	*Relation between I/O of* Σ_i *and* $\tilde{\Sigma}_i$	*Effect on Steady-State Relations*	*Realization*	*Effect on Integral Functions*
$L_A = \begin{bmatrix} 1 & \delta_A \\ 0 & 1 \end{bmatrix}$	$\tilde{u} = u + \delta_A y$ $\tilde{y} = y$	$\lambda_A^{-1}(\tilde{\mathrm{y}}) = k^{-1}(\tilde{\mathrm{y}}) + \delta_A \tilde{\mathrm{y}}$	*output-feedback*	$\Lambda^\star(\mathrm{y}) = K^\star(\mathrm{y}) + \frac{1}{2}\delta_A \mathrm{y}^2$
$L_B = \begin{bmatrix} 1 & 0 \\ 0 & \delta_B \end{bmatrix}$	$\tilde{u} = u$ $\tilde{y} = \delta_B y$	$\lambda_B(\mathrm{u}) = \delta_B k(\mathrm{u})$ *or* $\lambda_B^{-1}(\tilde{\mathrm{y}}) = k^{-1}(\frac{1}{\delta_B}\tilde{\mathrm{y}})$	*post-gain*	$\Lambda^\star(\mathrm{y}) = \frac{1}{\delta_B} K^\star(\frac{1}{\delta_B}\mathrm{y})$ *or* $\Lambda(\mathrm{u}) = \delta_B K(\mathrm{u})$
$L_C = \begin{bmatrix} 1 & 0 \\ \delta_C & 1 \end{bmatrix}$	$\tilde{u} = u$ $\tilde{y} = y + \delta_C u$	$\lambda_C(\tilde{\mathrm{u}}) = k(\tilde{\mathrm{u}}) + \delta_C \tilde{\mathrm{u}}$	*input-feedthrough*	$\Lambda(\mathrm{u}) = K(\mathrm{u}) + \frac{1}{2}\delta_C \mathrm{u}^2$
$L_D = \begin{bmatrix} \delta_D & 0 \\ 0 & 1 \end{bmatrix}$	$\tilde{u} = \delta_D u$ $\tilde{y} = y$	$\lambda_D^{-1}(\mathrm{y}) = \delta_D k^{-1}(\mathrm{y})$ *or* $\lambda_D(\tilde{\mathrm{u}}) = k(\frac{1}{\delta_D}\tilde{\mathrm{u}})$	*pre-gain*	$\Lambda^\star(\mathrm{y}) = \delta_D K^\star(\mathrm{y})$ *or* $\Lambda(\mathrm{u}) = \frac{1}{\delta_D} K(\frac{1}{\delta_D}\mathrm{u})$

Table 3.1: Elementary matrices and their realizations.

such that

$$\begin{bmatrix}\tilde{\mathrm{u}}\\ \tilde{\mathrm{y}}\end{bmatrix}=\begin{bmatrix}1 & 0\\ c & 1\end{bmatrix}\begin{bmatrix}\mathrm{u}_1\\ \mathrm{y}_1\end{bmatrix}.$$

Note that both transformations, from (u,y) *to* $(\mathrm{u}_1,\mathrm{y}_1)$ *and from* $(\mathrm{u}_1,\mathrm{y}_1)$ *to* $(\tilde{\mathrm{u}},\tilde{\mathrm{y}})$*, are given by the elementary matrices* L_B *and* L_C*, respectively. Let* κ *be the steady-state input-output relation from* u_1 *to* y_1*. Then,*

$$\kappa(\mathrm{u}_1)=dk(\mathrm{u}_1),\lambda(\tilde{\mathrm{u}})=\kappa(\tilde{\mathrm{u}})+c\tilde{\mathrm{u}}=dk(\tilde{\mathrm{u}})+c\tilde{\mathrm{u}},$$

which is the same as (3.24)*. Thus, if we write* T *as a product of elementary matrices, the new I/O relation* λ *can be seen as the old I/O relation* k*, after applying the elementary linear transformations sequentially.*

Proposition 3.6. *Let* k *and* λ *be the steady-state I/O relations of* Σ *and* $\tilde{\Sigma}$*, respectively, where* $\tilde{\Sigma}$ *is the result of applying the transformation* T *in* (3.23) *on* Σ*, where* $\delta_A=b/a,\delta_B=d-\frac{b}{a}c,\delta_C=c$ *and* $\delta_D=a$*. Assume that* κ_1 *is the steady-state I/O relation for some system* $\Sigma_1:u_1\mapsto y_1$*, obtained after the transformation* $L_A=\left[\begin{smallmatrix}1 & \delta_A\\ 0 & 1\end{smallmatrix}\right]$ *on the original system* Σ*. Then, the relation between* λ *and* k *is given by*

$$\lambda(\tilde{\mathrm{u}})=\left(d-\frac{b}{a}c\right)\kappa_1\left(\frac{1}{a}\tilde{\mathrm{u}}\right)+\frac{c}{a}\tilde{\mathrm{u}},\tag{3.25}$$

where the inverse of κ_1 *is*

$$(\kappa_1)^{-1}(\mathrm{y}_1)=k^{-1}(\mathrm{y}_1)+\frac{b}{a}\mathrm{y}_1.\tag{3.26}$$

Proof. Denote the steady-state I/O relations after the first, the second, and the third elementary matrix transformations, sequentially in (3.23), as $\kappa_1,\kappa_2,\kappa_3$, corresponding to the steady-state I/O pairs $(\mathrm{u}_1,\mathrm{y}_1)$, $(\mathrm{u}_2,\mathrm{y}_i)$ and $(\mathrm{u}_3,\mathrm{y}_3)$. The first transformation,

$$\begin{bmatrix}\mathrm{u}_1\\ \mathrm{y}_1\end{bmatrix}=L_A\begin{bmatrix}\mathrm{u}\\ \mathrm{y}\end{bmatrix}=\begin{bmatrix}1 & \frac{b}{a}\\ 0 & 1\end{bmatrix}\begin{bmatrix}\mathrm{u}\\ \mathrm{y}\end{bmatrix},$$

has the steady-state inverse I/O relation $\kappa_1^{-1}(\mathrm{y}_1)=k^{-1}(\mathrm{y}_1)+\frac{b}{a}\mathrm{y}_1$. The second transformation,

$$\begin{bmatrix}\mathrm{u}_2\\ \mathrm{y}_2\end{bmatrix}=L_B\begin{bmatrix}\mathrm{u}_1\\ \mathrm{y}_1\end{bmatrix}=\begin{bmatrix}1 & 0\\ 0 & d-\frac{b}{a}c\end{bmatrix}\begin{bmatrix}\mathrm{u}_1\\ \mathrm{y}_1\end{bmatrix},$$

has the steady-state I/O relation $\kappa_2(\mathrm{u}_2)=(d-\frac{b}{a}c)\kappa_1(\mathrm{u}_2)$. The third transformation,

$$\begin{bmatrix}\mathrm{u}_3\\ \mathrm{y}_3\end{bmatrix}=L_C\begin{bmatrix}\mathrm{u}_2\\ \mathrm{y}_2\end{bmatrix}=\begin{bmatrix}1 & 0\\ c & 1\end{bmatrix}\begin{bmatrix}\mathrm{u}_2\\ \mathrm{y}_2\end{bmatrix},$$

has steady-state I/O relation $\kappa_3(\mathrm{u}_3)=\kappa_2(\mathrm{u}_3)+c\mathrm{u}_3$. Finally, the fourth transformation,

$$\begin{bmatrix}\tilde{\mathrm{u}}\\ \tilde{\mathrm{y}}\end{bmatrix}=L_D\begin{bmatrix}\mathrm{u}_3\\ \mathrm{y}_3\end{bmatrix}=\begin{bmatrix}a & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}\mathrm{u}_3\\ \mathrm{y}_3\end{bmatrix},$$

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

has the steady-state I/O relation λ of $\tilde{\Sigma}$, and $\lambda(\tilde{\mathrm{u}}) = \kappa_3(\frac{1}{a}\tilde{\mathrm{u}})$. Substituting back for κ_3 and then for κ_2, we get the desired result. □

Example 3.9. *Consider the system in Example 3.3. The steady-state I/O relation λ of $\tilde{\Sigma}$ consists of all pairs $(\tilde{\mathrm{u}}, \tilde{\mathrm{u}}^3)$. We use Proposition 3.6 to verify this result. According to Proposition 3.6, for the given matrix transformation $T = \left[\begin{smallmatrix} 1 & 1 \\ 1 & 2 \end{smallmatrix}\right]$, λ is given by $\lambda(\tilde{\mathrm{u}}) = \kappa_1(\tilde{\mathrm{u}}) + \tilde{\mathrm{u}}$. After the first transformation $L_A = \left[\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right]$, the steady-state I/O pairs of the system Σ_1 are $\mathrm{u}_1 = \mathrm{u} + \mathrm{y}$, and $\mathrm{y}_1 = \mathrm{y}$. Substituting $\mathrm{u} = 2\sigma - \sigma^3$, and $\mathrm{y} = \sigma^3 - \sigma$ as obtained in Example 3.3 yields $\mathrm{u}_1 = \sigma$ and hence $\kappa_1(\mathrm{u}_1) = \mathrm{y}_1 = \mathrm{u}_1^3 - \mathrm{u}_1$. This implies that $\kappa_1(\tilde{\mathrm{u}}_1) = \mathrm{u}_1^3 - \mathrm{u}_1$, which on substitution yields $\lambda(\tilde{\mathrm{u}}) = \tilde{\mathrm{u}}^3$, as expected.*

Corollary 3.2. *For the conditions given in Proposition 3.6, if additionally $b = 0$, the relation between the relations λ and k, is given by*

$$\lambda(\tilde{u}) = dk\left(\frac{1}{a}\tilde{u}\right) + \frac{c}{a}\tilde{u}. \tag{3.27}$$

One can easily verify Example 3.8 using Corollary 3.2.

Proposition 3.6 connects the steady-state I/O relations of the new and the old system. In some cases,i.e., when $\rho, \nu \geq 0$, we know that the original system posses an integral functions. We can integrate the steady-state transformation, and obtain a connection between the original and the new integral function. For example, integrating the steady-state equation for output-feedback $\lambda^{-1}(\tilde{\mathrm{y}}) = k^{-1}(\tilde{\mathrm{y}}) + \delta\tilde{\mathrm{y}}$ results in $K^\star(\tilde{\mathrm{y}}) = \Lambda^\star(\tilde{\mathrm{y}}) + \frac{\delta}{2}\tilde{\mathrm{y}}^2$, where $K^\star, \Lambda^\star$ are the integral functions of k^{-1}, λ^{-1} respectively. Similarly, input-feedthrough corresponds to a quadratic term added to the integral function K of k, and pre- and post-gain correspond to scaling the integral function. These connections are summarized Table 3.1.

Example 3.10. *Consider Example 3.5. We saw that the steady-state input-output relation for the system is $\mathrm{u} = k^{-1}(\mathrm{y}) = \mathrm{y}^3 - \mathrm{y}$, so the corresponding integral function is $K^\star(\mathrm{y}) = \frac{1}{4}\mathrm{y}^4 - \frac{1}{2}\mathrm{y}^2$. Consider the input-output transformation $T = \left[\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right]$, which can be written as $\tilde{u} = u + y = u + \sqrt[3]{x}, \tilde{y} = y$, meaning that $u = -\sqrt[3]{x} + \tilde{u}$. Thus, the augmented system $\tilde{\Sigma}$ can be realized by the state-space model $\dot{x} = -x + \tilde{u}, \tilde{y} = \sqrt[3]{x}$, which has a steady-state input-output relation of $\tilde{\mathrm{u}} = \lambda^{-1}(\tilde{\mathrm{y}}) = \tilde{\mathrm{y}}^3$, and the corresponding integral function is $\Lambda^\star(\tilde{\mathrm{y}}) = \frac{1}{4}\tilde{\mathrm{y}}^4$. It is evident that $\Lambda^\star(\mathrm{y}) = K^\star(\mathrm{y}) + \frac{1}{2}\mathrm{y}^2$, as forecast by the discussion above and Table 3.1.*

Up to now, we showed how to monotonize the non-monotone relation k using a transformation, and how to implement the monotonizing transformation, connecting the old and new steady-state relations via a simple procedure. In the next section, we deal with the last ingredient missing for MEIP, namely maximality of the acquired monotone relation.

© Technion - Israel Institute of Technology, Elyachar Central Library

3.4.3 Maximality of Input-Output Relations and the Network Optimization Framework

As we saw, the map T monotonizes the steady-state relation k, i.e., the steady-state input-output relation λ of the augmented agent $\tilde{\Sigma}$ is monotone. However, it does not guarantee that λ is *maximally monotone*, which is essential for the network optimization framework. We explore a possible way to assure that λ is maximally monotone, under which we prove the network optimization framework. We begin by introducing the following property of relations.

Definition 3.6 (Cursive Relations)**.** *A set $A \subset \mathbb{R}^2$ is called* cursive *if there exists a curve $\alpha : \mathbb{R} \to \mathbb{R}^2$ such that the following conditions hold:*

i) The set A is the image of α.

ii) The map α is continuous.

iii) The map α satisfies $\lim_{|t|\to\infty} \|\alpha(t)\| = \infty$, where $\|\cdot\|$ is the Euclidean norm.

iv) The set $\{t \in \mathbb{R} : \ \exists s \neq t, \ \alpha(s) = \alpha(t)\}$ has measure zero.

A relation Υ is called cursive if the set $\{(p, q) \in \mathbb{R}^2 : \ q \in \Upsilon(p)\}$ is cursive.

Intuitively speaking, a relation is cursive if it can be drawn on a piece of paper without lifting the pen. The third requirement demands that the drawing will be infinite (in both time directions), and the fourth allows the pen to cross its own path, but forbids it from going over the same line twice. This intuition is the reason we call these relations cursive relations.

Under the assumption that the steady-state I/O relation k of Σ is cursive (which is usually the case in dynamical systems of the form (1.1)), we prove the maximality of λ in the following theorem.

Theorem 3.11. *Let k and λ be the steady-state I/O relations of the original system Σ and the transformed system $\tilde{\Sigma}$ under the transformation T, respectively. Suppose that k is a cursive relation and T is chosen to monotonize k as in Theorem 3.10. Then,*

i) λ is a maximally *monotone relation, and*

ii) $\tilde{\Sigma}$ is MEIP.

Moreover, if T transforms k to a strictly monotone relation, then $\tilde{\Sigma}$ is output-strictly MEIP.

Before proving the theorem, we prove the following lemma.

Lemma 3.1. *A cursive monotone relation Υ must be maximally monotone.*

Proof. Let $A_\Upsilon \subseteq \mathbb{R}^2$ be the set associated with Υ, which is cursive by assumption. Let α be the corresponding curve. If Υ is not maximal, then there is

© Technion - Israel Institute of Technology, Elyachar Central Library

some point $(p_0, q_0) \notin A_\Upsilon$ so that $\Upsilon \cup \{(p_0, q_0)\}$ is a monotone relation. By monotonicity, we find that

$$A_\Upsilon \subseteq \{(p,q) \in \mathbb{R},\ (p \geq p_0 \text{ and } q \geq q_0) \text{ or } (p \leq p_0 \text{ and } q \leq q_0), (p,q) \neq (p_0, q_0)\}.$$

The set on the right hand side has two connected components, namely $\{(p,q) : p \geq p_0, q \geq q_0, (p,q) \neq (p_0,q_0)\}$ and $\{(p,q) :\ p \leq p_0, q \leq q_0, (p,q) \neq (p_0,q_0)\}$. Since A_Υ is the image of a single curve, hence connected, it is contained in one of these connected components. Suppose, without loss of generality, it is contained in $\{(p,q) :\ p \geq p_0, q \geq q_0, (p,q) \neq (p_0,q_0)\}$. It is clear that we can choose the curve $\alpha(t) = (\alpha_1(t), \alpha_2(t))$ so that both functions α_1, α_2 are non-decreasing, as Υ is monotone. Thus, we must have $\alpha_1(0) \geq \lim_{t\to-\infty} \alpha_1(t) \geq p_0$, $\alpha_2(0) \geq \lim_{t\to-\infty} \alpha_2(t) \geq q_0$. However, these inequalities imply that $\|\alpha(t)\| = \sqrt{\alpha_1(t)^2 + \alpha_2(t)^2}$ remains bounded as $t \to -\infty$. This contradicts the assumption that Υ was cursive, hence it must be maximally monotone, which proves the claim. □

We are now ready to prove Theorem 3.11.

Proof. By the definition of MEIP and Lemma 3.1, it is enough to show that if k is cursive, then so is λ. Let $\mathcal{A}_k$ be the set associated with k, and $\mathcal{A}_\lambda$ be the set associated with λ. It is clear that $(\tilde{\mathrm{u}}, \tilde{\mathrm{y}})$ is a steady-state of $\tilde{\Sigma}$ if and only if (u, y) is a steady-state of Σ, where these I/O pairs are related by the transformation T. Thus, $\mathcal{A}_\lambda$ is the image of $\mathcal{A}_k$ under the invertible linear map T. Since k is cursive, we let $\alpha : \mathbb{R} \to \mathbb{R}^2$ be a curve plotting $\mathcal{A}_k$. We define the curve $\beta(t) = T(\alpha(t))$. We claim that the curve β proves that $\mathcal{A}_\lambda$, and hence λ, is cursive. Indeed, it is clear that $\mathcal{A}_\lambda$ is the image of β. Furthermore, β is continuous as a composition of the continuous maps T and α. The third property in Definition 3.6 holds as $\lim_{|t|\to\infty} ||\beta(t)|| \geq \lim_{|t|\to\infty} \underline{\sigma}(T)||\alpha(t)|| = \infty$, where we use the fact that T is invertible, hence $\underline{\sigma}(T)$, the minimal singular value of T, is positive. Lastly, the fourth property in Definition 3.6 holds as $\beta(t) = \beta(s)$ if and only if $\alpha(t) = \alpha(s)$, as T is invertible. Thus, the set $\{t :\ \exists s \neq t, \beta(t) = \beta(s)\}$ is the same as the one for α, hence has measure zero. This completes the proof. □

Before moving to the network optimization framework, we wonder how common are cursive relations. Obviously, all stable linear systems have cursive steady-state I/O relations, as their steady-state I/O relations form a line inside $\mathbb{R}^2$. We can push this further, as demonstrated by the following proposition:

Proposition 3.7. *Consider the system Υ governed by the ODE $\dot{x} = -f(x) + g(x)u,\ y = h(x)$ for some C^1 smooth functions f, g and a continuous function h such that $g > 0$. Assume that the either f/g or h is strictly monotone ascending, and that either $\lim_{s\to\pm\infty} |h(s)| = \infty$ or $\lim_{s\to\pm\infty} |f(s)/g(s)| = \infty$. Then the system Υ has a cursive steady-state I/O relation.*

Proof. In steady-state, we have $\dot{x} = 0$, thus we have $f(\mathrm{x}) = g(\mathrm{x})\mathrm{u}$. Moreover, $\mathrm{y} = h(\mathrm{x})$ in steady-state. Thus the steady-state input-output relation

© Technion - Israel Institute of Technology, Elyachar Central Library

can be parameterized as $(f(\sigma)/g(\sigma), h(\sigma)$ for the parameter $\sigma \in \mathbb{R}$. Consider the curve $\alpha : \mathbb{R} \to \mathbb{R}^2$ defined by $\alpha(\sigma) = (f(\sigma)/g(\sigma), h(\sigma))$. Then the steady-state relation is the image of α, which is continuous. The norm of α is equal to $\sqrt{(f(\sigma)/g(\sigma))^2 + h(\sigma)^2}$, so the assumption on the limit shows that $\lim_{|t|\to\infty} ||\alpha(t)|| = \infty$. Lastly, due to strict monotonicity, the curve α is an injective map. Thus the steady-state input-output relation is cursive. □

Remark 3.11. *Note that the strict monotonicity assumption can easily be relaxed–it shows that the curve $\alpha(t) = (f(t)/g(t), h(t))$ is injective as one of its coordinates is an injective map, but in practice we may have a non-self-intersecting curve which can behave very wildly in each coordinate. Moreover, non-self-intersecting is a stronger requirement then needed, we only need that the "self-intersecting set" is of measure zero.*

As we showed that cursive relations appear for a wide class of systems, we can conclude the network optimization framework for EI-IOP(ρ, ν) agents by Theorem 3.10 and the network optimization framework for MEIP agents.

Theorem 3.12. *Consider the diffusively-coupled network $(\mathcal{G}, \Sigma, \Pi)$, and suppose that the agents Σ_i are EI-IOP(ρ_i, ν_i) with cursive steady-state I/O relations k_i, and that the controllers are MEIP with integral function Γ_e. Let $\mathcal{J} = \mathrm{diag}(T_1, T_2, \ldots, T_{|\mathcal{V}|})$ be a linear transformation, where T_i is chosen as in Theorem 3.10 to so that k_i becomes strictly monotone by applying T_i. Then the transformed network $(\mathcal{G}, \tilde{\Sigma}, \Pi)$ converges, and the the steady-state limits $(\tilde{\mathrm{u}}, \tilde{\mathrm{y}}, \zeta, \mu)$ are the minimizers of the following dual network optimization problems:*

TOPP		**TOFP**	
$\min_{\tilde{\mathrm{y}}, \zeta}$	$\Lambda^\star(\tilde{\mathrm{y}}) + \Gamma(\zeta)$	$\min_{\tilde{\mathrm{u}}, \mu}$	$\Lambda(\tilde{\mathrm{u}}) + \Gamma^\star(\tilde{\mu})$
s.t.	$\mathcal{E}^\top \tilde{\mathrm{y}} = \zeta$	*s.t.*	$\tilde{\mathrm{u}} = -\mathcal{E}\mu$

where $\Gamma(\zeta) = \sum_{e\in\mathbb{E}} \Gamma_e(\zeta_e)$, $\Lambda(\mathrm{u}) = \sum_{i\in\mathbb{V}} \Lambda_i(\mathrm{u}_i)$, and Λ_i is the integral function associated with the maximally monotone relation λ_i, obtained by applying T_i to k_i.

For the special cases in which the original EI-IOP(ρ, ν) agents have integral functions, we can use the discussion preceding Example 3.10, connecting the original and the augmented integral functions, to prescribe (TOPP) and (TOFP) in terms of (OPP) and (OFP), namely (TOPP) and (TOFP) can be viewed as regularized versions of (OPP) and (OFP), where quadratic terms are added both the the agents' integral functions and their duals, assuming these were defined for the original network. This is a generalization of Subsection 3.3.1 which prescribed the quadratic correction of (OPP) when the agents are EI-OP(ρ). The main difference between this approach and the one in Subsection 3.3.1 is that there the network optimization framework can always at least

© Technion - Israel Institute of Technology, Elyachar Central Library

be defined, and convexifying it leads to the passivizing transformation. In our case, the simultaneous input- and output-shortage of passivity can cause the network optimization framework to be undefined, forbidding us from trying to convexify it. In particular, we conclude this section by stating the main result of Subsection 3.3.1 and providing a proof using the methods we introduced.

Corollary 3.3. *Consider the diffusively-coupled network $(\mathcal{G}, \Sigma, \Pi)$, and suppose that the agents have cursive steady-state I/O relations k_i, and that the controllers are MEIP with integral function Γ_e. Let $\mathcal{J} = \operatorname{diag}(T_1, T_2, \ldots, T_{|\mathcal{V}|})$ be as in Theorem 3.12.*

i) Suppose that the agents Σ_i are EI-OP(ρ_i), and that the relations k_i^{-1} have integral functions $K_i^\star$. Then we can take

$$T_i = \begin{bmatrix} 1 & \beta_i \\ 0 & 1 \end{bmatrix},$$

for any $\beta_i > -\rho_i$, and the cost function of (TOPP) is $K^\star(\mathrm{y}) + \Gamma(\zeta) + \frac{1}{2}\mathrm{y}^\top \operatorname{diag}(\beta)\mathrm{y}$, where $K^\star(\mathrm{y}) = \sum_{i\in\mathbb{V}} K_i^\star(\mathrm{y}_i)$.

ii) Suppose that the agents Σ_i are EI-IP(ν_i), and that the relations k_i have integral functions K_i. Then we can take

$$T_i = \begin{bmatrix} 1 & 0 \\ \beta_i & 1 \end{bmatrix},$$

for any $\beta_i > -\rho_i$, and the cost function of (TOPP) is $K(\mathrm{u}) + \Gamma^\star(\mu) + \frac{1}{2}\mathrm{u}^\top \operatorname{diag}(\beta)\mathrm{u}$, where $K(\mathrm{y}) = \sum_{i\in\mathbb{V}} K_i(\mathrm{u}_i)$.

Proof. We only prove the first case, as the proof second case is completely analogous. Each agent is EI-OP(ρ_i), so that the associated PQI is given by $0 \le \xi\chi - \rho_i\chi^2$. We take any $\beta_i > -\rho_i$ and look for some T_i which transforms this PQI into $0 \le \xi\chi - (\rho_i+\beta_i)\chi^2$, which implies strict monotonicity. We build T_i according to Theorem 3.10, taking $(\xi_1, \chi_1) = (1, 0), (\xi_2, \chi_2) = (\rho_i, 1), (\xi_3, \chi_3) = (1, 0)$ and $(\xi_4, \chi_4) = (\rho_i + \beta_i, 1)$. We note that $(\xi_1 + \chi_1, \xi_2 + \chi_2) = (1 + \rho_i, 1)$ satisfies

$$\chi\xi - \rho_i\chi^2 = 1 + \rho_i - \rho_i = 1 \ge 0$$

meaning that $(\xi_1+\chi_1, \xi_2+\chi_2)$ satisfies the first PQI. Similarly, $(\xi_3+\chi_3, \xi_4+\chi_4)$ satisfies the second PQI. Thus, we can take T_i as:

$$T_i = \begin{bmatrix} \xi_3 & \xi_4 \\ \chi_3 & \chi_4 \end{bmatrix} \begin{bmatrix} \xi_1 & \xi_2 \\ \chi_1 & \chi_2 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & \rho_i + \beta_i \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & \rho_i \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & \beta_i \\ 0 & 1 \end{bmatrix},$$

which proves the first part. As for the second part, Table 3.1 implies that the steady-state relation λ_i of the augmented system is given by $\lambda_i^{-1}(\mathrm{y}_i) = k_i^{-1}(\mathrm{y}_i) + \beta_i\mathrm{y}_i$. Integrating this equation with respect to y_i gives that $\Lambda_i^\star(\mathrm{y}_i) = K_i^\star(\mathrm{y}_i) + \frac{1}{2}\beta_i\mathrm{y}_i^2$. Using $K^\star(\mathrm{y}) = \sum_{i\in\mathbb{V}} K_i^\star(\mathrm{y}_i)$ and $\Lambda^\star(\mathrm{y}) = \sum_{i\in\mathbb{V}} \Lambda_i^\star(\mathrm{y}_i)$ gives that $\Lambda^\star(\mathrm{y}) = K^\star(\mathrm{y}) + \frac{1}{2}\mathrm{y}^\top \operatorname{diag}(\beta)\mathrm{y}$, completing the proof. □

© Technion - Israel Institute of Technology, Elyachar Central Library

3.5 Case Studies

We present a total of three case studies. The first two deal with Examples 3.2 and 3.4, appearing in Section 3.2, and the third deals with a traffic model.

3.5.1 Unstable First Order Systems

As in Example 3.2, we consider a collection of $n = 100$ LTI agents, each modeled by the transfer function $G_i(s) = \frac{1}{s-1}$, or by the model $\dot{x}_i = x_i + u_i,\ y_i = x_i$. The agents are EI-OP($\rho$) with parameter $\rho = -1$. The steady-state input-output relation of the agents is given by $k_i(\mathrm{u}_i) = -\mathrm{u}_i$, or equivalently by $k_i^{-1}(\mathrm{y}_i) = -\mathrm{y}_i$. The integral function in this case is equal to $K_i^\star(\mathrm{y}) = -\frac{1}{2}\mathrm{y}_i^2$, which is indeed non-convex.

We consider a diffusively-coupled system of the form $(\mathcal{G}, \Sigma, \Pi)$ where the graph $\mathcal{G}$ is the complete graph on n agents, Σ are the agents, and the controllers are static gain of size 1. In that case, the closed-loop system can be written as $\dot{x} = (\mathrm{Id} - \mathcal{E}_\mathcal{G}\mathcal{E}_\mathcal{G}^\top)x$. The system is unstable, as the matrix $\mathrm{Id} - \mathcal{E}_\mathcal{G}\mathcal{E}_\mathcal{G}^\top$ has an eigenvalue 1 with eigenvector $\mathbb{1}_n$, implying it is not Hurwitz. The associated problem (OPP) reads:

$$\begin{aligned} \min \quad & -\frac{1}{2}\sum_{i\in\mathbb{V}} \mathrm{y}_i^2 + \frac{1}{2}\sum_{e\in\mathbb{E}} \zeta_e^2 \\ \text{s.t.} \quad & \zeta = \mathcal{E}_\mathcal{G}^\top \mathrm{y} \end{aligned}$$

which is non-convex. Thus we wish to regularize (OPP), where we note that $\bar{\rho}$, the average of the passivity indices, is equal to -1, meaning that we cannot apply the network-based regularization procedure. We consider both an agent-based regularization, and a hybrid regularization.

First, we consider agent-based regularization. It's obvious that the Tikhonov regularization term $\frac{1}{2}\sum_{i\in\mathbb{V}} \beta_i \mathrm{y}_i^2$ convexifies (OPP) whenever $\beta_i > 1 = -\rho_i$, as predicted by Theorem 3.3. Choosing $\beta_i = 1.1$, the regularized problem (ROPP) reads:

$$\begin{aligned} \min \quad & \frac{1}{20}\sum_{i\in\mathbb{V}} \mathrm{y}_i^2 + \frac{1}{2}\sum_{e\in\mathbb{E}} \zeta_e^2 \\ \text{s.t.} \quad & \zeta = \mathcal{E}_\mathcal{G}^\top \mathrm{y} \end{aligned}$$

which is minimized at $\mathrm{y} = \mathbf{0}$ and $\zeta = \mathbf{0}$. The corresponding output-feedback gives a closed-loop system governed by the equation $\dot{x} = (-0.1\mathrm{Id} - \mathcal{E}_\mathcal{G}\mathcal{E}_\mathcal{G}^\top)x$. Noting that the dynamics matrix is negative-definite, we conclude that the state $x(t)$ converges to $\mathbf{0}$ as $t \to \infty$, giving the steady-state output $\mathrm{y} = \mathbf{0}$. Thus the network optimization framework predicts a correct limit for the augmented closed-loop system. The trajectories of the augmented closed-loop system can be seen in Figure 3.10.

Now, we consider hybrid regularization. We consider five different cases here, which are summarized in the table below:

© Technion - Israel Institute of Technology, Elyachar Central Library

Scenario	Number of Self Regulating Agents	Size of α_i	Average of $\alpha_i + \rho_i$	Size of b
1	1	101	0.01	9703.97
2	1	201	1	382.41
3	10	10.1	0.01	81.09
4	10	20	1	3.35
5	20	8	0.6	0.73

Table 3.2: Comparison of hybrid regularization terms for networks with output-passive short agents.

It is evident from the table that although one self-regularizing agent is enough to regularize the entire network, the gains required for regularization might be very large. Reducing the size of the required gains can either be done by "over-regularizing" the self-regulating agent, i.e. give it a much larger constant α_i than needed, or by using a larger set of self-regulating agents. The table shows that even a small proportion of self-regulating agents, e.g. $10\% - 20\%$, is enough to significantly reduce the size of the required gains for regularization.

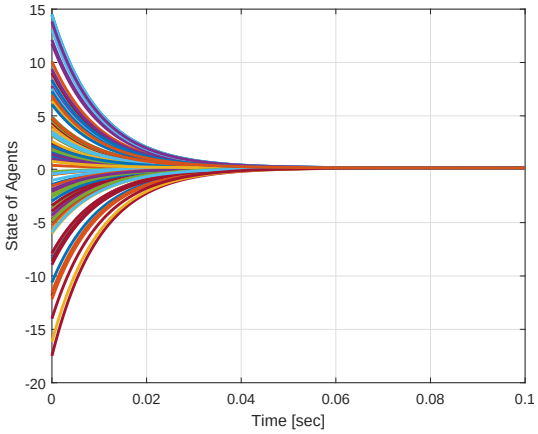
The trajectories of the augmented closed-loop system for scenarios 2 and 4 can be seen in Figure 3.10

3.5.2 Gradient Systems

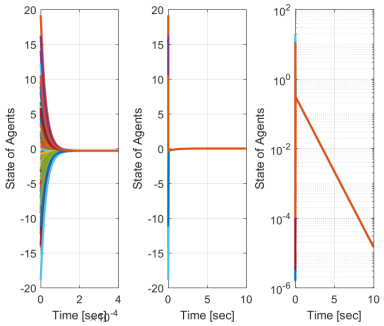
We return to Example 3.4, in which the system (3.7) was studied, for the function $U(x_i) = r_1(1 - \cos x_i) + \frac{1}{2} r_2 x_i^2$, with $r_1 = 2.5, r_2 = 0.1$. The system is EI-OP(ρ) with parameter $\rho = -2.4$. The steady-state input-output relation can be parameterized as $\mathrm{u} = r_1 \sin\sigma + r_2\sigma$, $\mathrm{y} = \sigma$ by the parameter $\sigma \in \mathbb{R}$. As can be seen in Figure 3.2, the relation is cursive, but non-monotone. We use the framework of Section 3.4 to monotonize the steady-state input-output relation, and show the resulting network-optimization framework predicts the correct limit of the augmented closed-loop networked system.

We apply Theorem 3.12. We note that the transformation $T = \left[\begin{smallmatrix} 1 & \kappa \\ 0 & 1 \end{smallmatrix}\right]$ regularizes the agent, so long that $\kappa \geq 2.4$. We choose $\kappa = 2.5$. The augmented agent $\tilde{\Sigma}_i$ has an input-output relation of the form $\mathrm{u} = r_1 \sin\sigma + r_2\sigma + 2.5\sigma = 2.5\sin\sigma + 2.6\sigma, \mathrm{y} = \sigma$, which is maximally monotone, as can be seen in Figure 3.11. The integral functions Λ_i,$\Lambda_i^\star$ for the augmented agent can be seen in Figure 3.11.

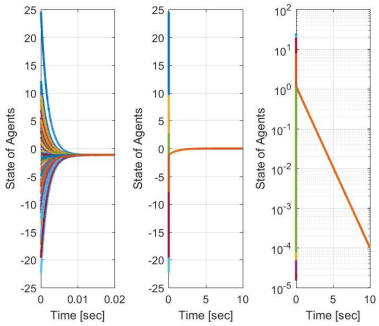
We consider the diffusively-coupled system $(\mathcal{G}, \tilde{\Sigma}, \Pi)$ with the augmented agents and original controllers, $\mu_e = \zeta_e$. As can be seen from Figure 3.2, the minimum of Λ_i is achieved at $\mathrm{y}_i = 0$. Moreover, the global minimum of $\Gamma(\zeta) = \frac{1}{2}\zeta^2$ is achieved at $\zeta = \mathbf{0}$, so the network optimization problem (OPP) for the augmented network is minimized at $\mathrm{y} = \mathbf{0}, \zeta = \mathbf{0}$. We run the augmented closed-loop system and plot the trajectories in Figure 3.12, which shows that the augmented closed-loop system converges to $\mathbf{0}$, as predicted.



(a) Agent-based regularization.



(b) Hybrid regularization, scenario 2.



(c) Hybrid regularization, scenario 4.

Figure 3.10: System trajectories for different regularization terms for a network of unstable first order systems.

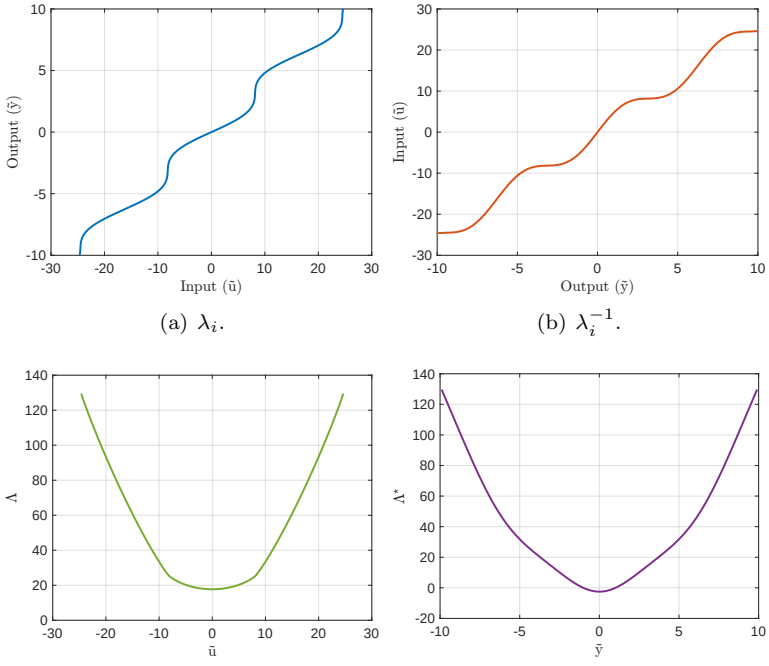
3.5.3 Traffic Model

Consider the traffic dynamics model proposed in [8], in which vehicles adjust their velocity x_i according to the equation $\dot{x}_i = \kappa_i(V_i(\Delta p) - x_i)$, where $\kappa_i > 0$ is a constant representing the sensitivity of the i-th driver, and

$$V_i(\Delta p) = V_i^0 + V_i^1 \sum_{j \sim i} \tanh(p_j - p_i), \tag{3.30}$$

is the adjustment, where V_i^0 are the preferred velocities, and V_i^1 are the "sensitivity coefficients". The parameters κ_i are heterogeneous and could be either positive or negative depending on the attentiveness of the individual driver. The case of $\kappa_i > 0$ was studied in [23], where it was shown that it can inhibit a clustering phenomenon. The case of $\kappa_i < 0$ corresponds to drowsy driving, or intoxicated driving.

© Technion - Israel Institute of Technology, Elyachar Central Library



(a) λ_i. (b) λ_i^{-1}.

(c) $\Lambda_i(y_i)$. (d) $\Lambda_i^\star(y_i)$.

Figure 3.11: Steady-state input-output relations and integral functions of the transformed system $\tilde{\Sigma}_i$.

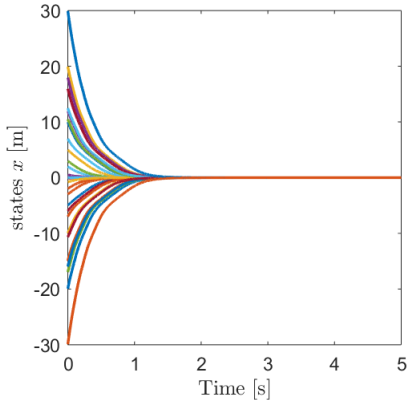


Figure 3.12: Trajectories of the augmented closed-loop system for a diffusively-coupled network of gradient systems.

Consider a case where only some agents know their own velocity (e.g., by a GNSS measurement). Thus, agents which have no GNSS reception cannot

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

implement the regularization of (OPP), or the self-feedback loops, resulting from the agent-based regularization protocol. Instead, we opt for the hybrid regularization protocol.

The model is a diffusively coupled network with the agents $\Sigma_i : \dot{x_i} = \kappa_i(-x_i + V_i^0 + V_i^1 u)$, $y_i = x_i$ and the controllers $\Pi_e : \dot{\eta}_e = \zeta_e$, $\mu_e = \tanh(\eta_e)$. The agents are EI-OP(ρ_i) if $V_i^1 \kappa_i > 0$, with $\rho_i = \kappa_i$, so $\kappa_i > 0$ corresponds to output-strict MEIP agents. We suppose that only agent i_0 has GNSS reception, implementing a correction term of the form $\alpha_{i_0} y_{i_0}^2 + \beta \zeta^\top \zeta$ to (OPP), giving us (HROPP). The new control law is $u(t) = -\alpha_{i_0} x_{i_0} \mathrm{e}_{i_0} - \beta \mathcal{E}_\mathcal{G} \mathcal{E}_\mathcal{G}^\top x - V(\Delta p)$ where $V(\Delta p) = [V_1(\Delta p), \cdots, V_n(\Delta p)]^\top$, and e_i is the i-th standard basis vector. Only the states x_{i_0} and $\mathcal{E}_\mathcal{G}^\top x$ are used in the control law, meaning that no agent but i_0 is required to know its velocity in a global frame of reference, but only positions and velocities relative to its neighbors.

To illustrate this, we consider a network of $n = 100$ agents, all connected to each other, with parameters κ_i randomly chosen either as -1 (w.p. $1/3$) and 1 (w.p. $2/3$). Moreover, the parameters V_i^0 were chosen as a Gaussian mixture model, with half of the samples having mean 20 and standard deviation 15, and half having mean 120 and standard deviation 15. Lastly, V_i^1 where chosen as $0.8\kappa_i$. In [23], it is shown that (OPP) in this case is given by

$$\min_{\mathrm{y},\zeta} \quad \sum_i \frac{1}{2V_i^1}(y_i - V_i^0)^2 + \sum_e |\zeta_e| \qquad \text{s.t.} \quad \zeta = \mathcal{E}_\mathcal{G}^\top y,$$

meaning that (HROPP) is given by:

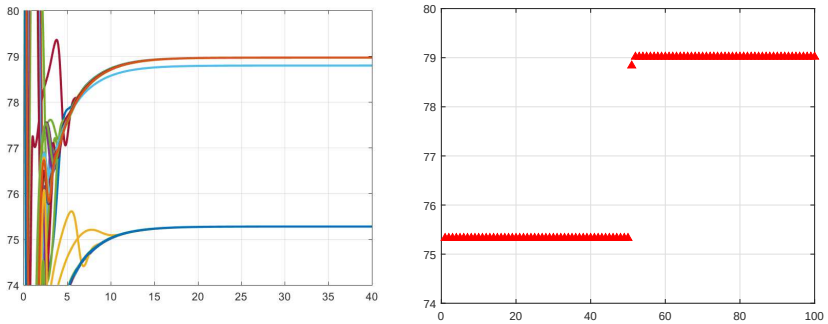
$$\begin{aligned} \min_{\mathrm{y},\zeta} \quad & \sum_i \frac{1}{2V_i^1}(y_i - V_i^0)^2 + \sum_e |\zeta_e| + \alpha_{i_0} y_{i_0}^2 + \beta \sum_e \zeta_e^2 \\ \text{s.t.} \quad & \zeta = \mathcal{E}_\mathcal{G}^\top y. \end{aligned}$$

It is evident that when $V_i^1 < 0$ (corresponding to $\kappa_i < 0$), the optimization problem (OPP) is non-convex, as it contains a negative quadratic term.

Here, the average $\frac{1}{n}\sum_i \rho_i = \frac{1}{n}\sum_i \kappa_i$ is positive, so we use the network-regularization method, choosing $\alpha_{i_0} = 0$, so that (HROPP) reduces to (NROPP). Choosing $\beta = (\mathbf{b} + \epsilon)\mathbb{1}$, we apply Theorems 3.8 and 3.6 to conclude that the system converges, and find its steady-state limit. We plot the trajectories of the system in Figure 3.13(a), as well as the minimizer of (NROPP) in Figure 3.13(b). One can see that the steady-state value of the system matches the forecast, namely the minimizer of (NROPP). It should be noted that we obtain a clustering phenomenon, as noted in [23]. However, the agents form much larger clusters than in [23]. This is due to the term $-\beta \mathcal{E}_\mathcal{G} \mathcal{E}_\mathcal{G}^\top x$ appearing in u, which not only passivizes the system, but also forces the trajectories closer to a consensus.

3.6 Conclusions

We considered the network optimization framework for equilibrium-independent input-output (ρ, ν)-passive agents. We have first shown that the framework,



(a) Vehicles' trajectories for network-only regularization.

(b) Asymptotic behaviour predicted by (NROPP).

Figure 3.13: Traffic control model.

unless augmented, may fail for equilibrium-independent passive-short agents in more than one way - the networked system might not converge, the integral functions might not be defined, and even when both of these obstructions do not happen, the framework might still predict a false value for the limit of the diffusively-coupled system. The failure of the system could be understood in terms of non-convexity of the agents' integral functions K,$K^\star$, or in terms of the non-monotonicity of the agents' steady-state relations k_i.

We first focused on networks with equilibrium-independent *output* ρ-passive agents. We showed that in this case, the problem (OPP) can still be defined, but the cost function is non-convex. We considered a regularization term for the problem (OPP), and showed that it results in an output-feedback for the agents, and that the augmented network system converges to the limit predicted by the regularized optimization problem. We considered three different possible regularization terms, agent-based, network-based, and hybrid, and explored when each can be used.

We then shifted our focus to networks with general equilibrium-independent input-output (ρ, ν)-passive agents. In this case, the integral function might not even be defined, so we exchange convexifying the integral function by monotonizing the steady-state input-output relation. We use a geometric-based argument to show that a monotonizing transformation can always be found, and that it induces an augmented agent which is passive with respect to all equilibria. As MEIP requires the steady-state relation to be maximally monotone, and not just monotone, we define the notion of cursive relations. We show that if the original, unaugmented agent has a cursive input-output steady-state relation, then the augmented agent is MEIP. Moreover, we show how the integral function of the agent is achieved from the original steady-state relation using the monotonizing transformation and integrating. By applying this procedure for each of the agents, we end up with a network of output-strictly MEIP agents

© Technion - Israel Institute of Technology, Elyachar Central Library

and MEIP controllers, for which the network optimization framework holds. We then presented the different regularization procedure for three case studies.

These regularization procedures allow us to extend the network optimization framework to a much larger class of networked systems. In turn, it will allow us to consider the applications appearing in Chapters 4, 5, 6 and 7 also for passive-short systems.

In the next chapter, we'll consider a first application of the network-optimization framework, when we'll use its convergence guarantees to solve a synthesis problem.

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

Chapter 4

A Network Optimization Framework for Controller Synthesis

This section is a review of [136, 142], and parts of [138]. We apply the network optimization framework to study the problem of synthesis, in which we are given a multi-agent system, in which the agents $\{\Sigma_i\}$ and interaction graph $\mathcal{G}$ are given, and some control goal is specified. Our goal is to find networked controllers $\{\Pi_e\}$ such that the diffusively-coupled system $(\mathcal{G}, \Sigma, \Pi)$ satisfies the control goal. We focus on final-value synthesis, in which the control goal is that some specified signal in the system (e.g., the agents' outputs) converges to a given desired steady-state value.

4.1 Introduction

As we saw, the study of multi-agent networks is a cornerstone of control research. An important problem in the study of multi-agent systems is that of controller synthesis - namely the construction of distributed controllers forcing the closed-loop system to converge to some desired output. This control goal encompasses many canonical problems including synchronization and formation control [92, 105,107,108]. It also encompasses the problem of clustering, in which the agents are divided into different groups (namely, clusters). The problem then tasks one to design a distributed control law forcing agents in the same cluster to synchronize, while agents in different clusters do not synchronize. The clustering problem is essential in fields like ecology [149], neuroscience [132], and biomimicry of swarms [109].

Our goal in this chapter is to describe a solution to the synthesis problem for multi-agent systems by applying the network optimization framework. Our control objective is to assure the convergence of the networked system to a desired

© Technion - Israel Institute of Technology, Elyachar Central Library

output or a desired relative output configuration - we term this desired output a *formation* of the system.[1] This is a crucial first step toward applications that will be used continuously throughout the later chapters of this thesis.

The rest of this chapter is organized as follows. The next section considers the general case of heterogeneous agents, and deals with the problem of final-value synthesis for the output and the relative-output configuration. The following section mainly deals with network having certain symmetries, culminating in the special case of synthesis on homogeneous networks. It is shown that in this case a clustering solution is achieved, and the notion of cluster synthesis is briefly explored.

4.2 Final-Value Synthesis for Multi-Agent Systems

Our goal is to design controllers $\{\Pi_e\}$ on the edges to achieve a desired steady-state output for the networked system. We can specify the demand in terms of the output $y(t)$, or in terms of the relative output $\mathcal{E}_{\mathcal{G}}^{\top} y(t) = \zeta(t)$. We first study the case of a desired output, and then adapt our results to the case of a desired relative output, or a desired *formation*. The final-value synthesis problem for the output vector is stated as follows:

Problem 4.1. *Let $\{\Sigma_i\}_{i\in\mathbb{V}}$ be MEICMP agents and let $\mathcal{G}$ be any graph on $|\mathbb{V}|$ nodes. Let $\mathrm{y}^\star \in (\mathbb{R}^d)^{|\mathbb{V}|}$ be some vector.*

i) Find a computationally feasible criterion assuring the existence of controllers $\{\Pi_e\}_{e\in\mathbb{E}}$, such that the output of the system $(\mathcal{G}, \Sigma, \Pi)$ has $\mathrm{y}^\star$ as a steady-state.

ii) In the case $\mathrm{y}^\star$ satisfies the criterion, find a construction for $\{\Pi_e\}_{e\in\mathbb{E}}$, that forces the system converge to $\mathrm{y}^\star$.

This section has five parts. Subsection 4.2.1 deals with solving part 1 of the Problem 4.1. Subsection 4.2.2 deals with solving the second part of the same problem. Subsection 4.2.3 deals with different control objectives $\mathrm{y}^\star$, namely by prescribing a procedure which uses a solution for some $\mathrm{y}_1^\star$ to find a solution for $\mathrm{y}_2^\star$ by augmenting the controller. Subsection 4.2.4 addresses outputs that do not satisfy the desired synthesis criteria. Finally, Subsection 4.2.5 adapts the results to the final-value synthesis problem for relative outputs, and briefly discusses other final-value synthesis problems.

For the rest of this section, we consider MIMO agents which are MEICMP, and, as before, denote the input-output steady-state relations of the nodes by k_i, and their integral functions by K_i. We choose the controllers to be output-strictly MEICMP, so we can discuss their input-output steady-state relations γ_e and their integral functions by Γ_e.

[1] This is not to be confused with the standard formation control problem which aims to control a team of agents to some desired spatial configuration [105]. Our use of the term formation in this context is more abstract.

© Technion - Israel Institute of Technology, Elyachar Central Library

4.2.1 Characterizing Forcible Steady-States

The result of Theorem 2.3 helps us predict the steady-state outputs of the closed-loop by solving the optimal potential problem (OPP), which we restate for the convenience of the reader:

$$\begin{aligned} \min_{\mathrm{y},\zeta} \quad & K^\star(\mathrm{y}) + \Gamma(\zeta) \\ \text{s.t.} \quad & \mathcal{E}_{\mathcal{G},d}^\top \mathrm{y} = \zeta. \end{aligned} \tag{4.1}$$

The outline to the solution of Problem 4.1 is given by studying the minimizers of the optimization problem (OPP). We recall Proposition 2.2, which says that $\mathrm{y}^\star$ is a steady-state of the closed-loop system if and only if the following "equation" holds:

$$\mathbf{0} \in k^{-1}(\mathrm{y}^\star) + \mathcal{E}_{\mathcal{G},d}\gamma(\mathcal{E}_{\mathcal{G},d}^\top \mathrm{y}^\star). \tag{4.2}$$

We conclude:

Corollary 4.1. *Let $\mathrm{y}^\star \in (\mathbb{R}^d)^{|\mathbb{V}|}$. One can choose output-strictly MEICMP controllers $\{\Pi_e\}_{e\in\mathbb{E}}$ so that $\mathrm{y}^\star$ is a steady state of the closed-loop system if and only if $k^{-1}(\mathrm{y}^\star) \cap \mathrm{Im}(\mathcal{E}_{\mathcal{G},d}) \neq \emptyset$.*

Proof. If $\mathrm{y}^\star$ is a steady state (for some choice of controllers), then (4.2) proves that $k^{-1}(\mathrm{y}^\star) \cap \mathrm{Im}(\mathcal{E}_{\mathcal{G},d}) \neq \emptyset$. Conversely, if $k^{-1}(\mathrm{y}^\star) \cap \mathrm{Im}(\mathcal{E}_{\mathcal{G},d}) \neq \emptyset$, then we can take some vector ξ such that $-\mathcal{E}_{\mathcal{G},d}\xi \in k^{-1}(\mathrm{y}^\star)$. If the MEICMP controllers Π_e are chosen so that $\gamma(\mathcal{E}_{\mathcal{G},d}^\top \mathrm{y}^\star) \ni -\xi$, then Proposition 4.1 implies that $\mathrm{y}^\star$ is a steady state of the closed-loop system, so it is enough to show that there are MEICMP controllers Π_e such that $-\xi \in \gamma(\mathcal{E}_{\mathcal{G},d}^\top \mathrm{y}^\star)$. There are many ways to choose these controllers, one of them being

$$\Pi_e : \begin{cases} \dot{\eta_e} = -\eta_e + \zeta_e - (\xi_e + \zeta_e^\star) \\ \mu_e = \eta_e \end{cases} . \tag{4.3}$$

□

Remark 4.1. *The chosen controllers have a special structure - these are linear controllers with constant exogenous inputs forcing the system to converge to $\mathrm{y}^\star$, but their dependence on $\mathrm{y}^\star$ is only through the constant $\xi_e + \zeta_e^\star$. This small change in the controller will force the entire system converge to a different point. We'll emphasize this point in Subsection 4.2.3.*

It is well-known that the set $\mathrm{Im}(\mathcal{E}_{\mathcal{G},d})$, called the cut-space of the graph $\mathcal{G}$, consists of all vectors $\mathrm{u} \in (\mathbb{R}^d)^{|\mathbb{V}|}$ such that $\sum_{i=1}^{|\mathbb{V}|} \mathrm{u}_i = 0$ [57]. Thus, the first part of Problem 4.1 is solved by the following result.

Corollary 4.2. *The vector $\mathrm{y} \in (\mathbb{R}^d)^{|\mathbb{V}|}$ is forcible as a steady-state if and only if $\mathbf{0} \in \sum_{i=1}^{|\mathbb{V}|} k_i^{-1}(\mathrm{y}_i)$.*

© Technion - Israel Institute of Technology, Elyachar Central Library

4.2.2 Forcing Global Asymptotic Convergence

We now solve part 2 of Problem 4.1, giving criteria for controllers to provide global asymptotic convergence and constructing controllers that satisfy these criteria. By Theorem 2.3, if we take output-strictly MEICMP controllers, then the closed-loop system converges to some $\hat{\mathrm{y}}$, so that $(\hat{\mathrm{y}}, \hat{\zeta} = \mathcal{E}_{\mathcal{G},d}^{\top}\hat{\mathrm{y}})$ form an optimal solution of (OPP).

Corollary 4.3. *If the chosen controllers are output-strictly MEICMP, and (OPP) has only one solution $(\hat{\mathrm{y}}, \hat{\zeta})$, then the closed-loop system globally asymptotically converges to $\hat{\mathrm{y}}$.*

The minimization of the function $K^\star(\mathrm{y}) + \Gamma(\zeta)$ appearing in (OPP) can be divided into two parts:

$$\begin{aligned}\min_{\mathcal{E}_{\mathcal{G},d}^{\top}\mathrm{y}=\zeta}[K^\star(\mathrm{y}) + \Gamma(\zeta)] &= \min_{\zeta\in\mathrm{Im}(\mathcal{E}_{\mathcal{G},d}^{\top})}\ \min_{\mathrm{y}:\mathcal{E}_{\mathcal{G},d}^{\top}\mathrm{y}=\zeta}[K^\star(\mathrm{y}) + \Gamma(\zeta)] \\ &= \min_{\zeta\in\mathrm{Im}(\mathcal{E}_{\mathcal{G},d}^{\top})}[\Gamma(\zeta) + \min_{\mathrm{y}:\mathcal{E}_{\mathcal{G},d}^{\top}\mathrm{y}=\zeta} K^\star(\mathrm{y})]. \qquad (4.4)\end{aligned}$$

Our goal is to have $(\mathrm{y}^\star, \zeta^\star = \mathcal{E}_{\mathcal{G},d}^{\top}\mathrm{y}^\star)$ as the sole minimizer of this problem. Thus we have two goals - the outer minimization problem needs to have $\zeta^\star$ as a sole minimizer, and the inner minimization problem needs to have $\mathrm{y}^\star$ as a sole minimizer. The main tool we employ is strict convexity. We note that if the systems $\{\Pi_e\}_{e\in\mathbb{E}}$ are output-strictly MEISCMP, then the input-output relation γ is strictly cyclically monotone, and therefore Γ is strictly convex. Let us first deal with the outer minimization problem in (4.4).

Definition 4.1. *The* minimal potential function *is a function $G := G_{\mathcal{G},K} : \mathrm{Im}(\mathcal{E}_{\mathcal{G},d}^{\top}) \to \mathbb{R}$, depending on the graph $\mathcal{G}$ and the agents' integral function, K, defined by*

$$G(\zeta) = \min\{K^\star(\mathrm{y})|\ \mathcal{E}_{\mathcal{G},d}^{\top}\mathrm{y} = \zeta\}.$$

Note that (4.4) can be rewritten as:

$$\min_{\mathcal{E}_{\mathcal{G},d}^{\top}\mathrm{y}=\zeta}[K^\star(\mathrm{y}) + \Gamma(\zeta)] = \min_{\zeta\in\mathrm{Im}(\mathcal{E}_{\mathcal{G},d}^{\top})}[\Gamma(\zeta) + G(\zeta)]. \qquad (4.5)$$

Proposition 4.1. *Suppose that* (4.2) *is satisfied by the pair $(\mathrm{y}^\star, \zeta^\star = \mathcal{E}_{\mathcal{G},d}^{\top}\mathrm{y}^\star)$. Suppose further that the function Γ_e is strictly convex in the neighborhood of $\zeta_e^\star$ for all $e \in \mathbb{E}$. Then $\zeta^\star$ is the unique minimizer of the optimization problem in* (4.5)*, i.e., $\zeta^\star$ is the unique minimizer of $\Gamma(\zeta) + G(\zeta)$.*

Proof. Because the function $K^\star$ is convex, the function G is also convex (see Appendix A). Thus $\Gamma(\zeta) + G(\zeta)$ is convex as a sum of convex functions, and it is strictly convex near $\zeta^\star$. Let M be the collection of $(G + \Gamma)$'s minima. It follows from Proposition 2.2 that $\zeta^\star \in M$. Furthermore, the set M is convex, since $G+\Gamma$ is a convex function. Finally, there is some small neighborhood $\mathcal{U}$ of $\zeta^\star$ such that $M \cap \mathcal{U}$ contains no more than one point, as $G+\Gamma$ is strictly convex

in a neighborhood of $\zeta^\star$. We claim that these facts imply that M contains the single point $\zeta^\star$, concluding the proof.

Indeed, suppose that there's some other $\zeta \in M$. By convexity, we have $\zeta_t = t\zeta + (1-t)\zeta^\star \in M$ for all $t \in (0,1)$, and in particular, for small $t > 0$. If $t > 0$ is small enough then $\zeta_t \in \mathcal{U}$, as $\mathcal{U}$ is open, meaning that $\zeta_t \in M \cap \mathcal{U}$ for $t > 0$ small. But this is impossible as $M \cap \mathcal{U}$ cannot contain more than one point. Thus $\zeta^\star$ is the unique minimizer of $G + \Gamma$. □

Now, we focus on the inner minimization problem of (4.4), namely the optimization problem defining $G(\zeta)$. We wish that $\mathrm{y}^\star$ would be the unique minimizer of $K^\star(\mathrm{y})$ on the set $\{\mathcal{E}_{\mathcal{G},d}^\top \mathrm{y} = \zeta^\star\} = \{\mathrm{y}^\star + \beta \otimes \mathbb{1} \,|\, \beta \in \mathbb{R}^d\}$. We consider $A : \mathbb{R}^d \to \mathbb{R}$ defined by $A(\beta) = K^\star(\mathrm{y}^\star + \beta \otimes \mathbb{1})$, and we wish that $\beta = \mathbf{0}$ will be the unique minimizer of A.

Minimizing A is the same as finding β such that $\mathbf{0} \in \partial A(\beta)$. By subdifferential calculus (see Appendix A and [121]), we have $\partial A(\beta) = \mathrm{Proj}_{\ker \mathcal{E}_{\mathcal{G},d}^\top} k^{-1}(\mathrm{y}^\star + \beta \otimes \mathbb{1})$, where we use $\{\beta \otimes \mathbb{1} : \ \beta \in \mathbb{R}^d\} = \ker \mathcal{E}_{\mathcal{G},d}^\top$. We know that the projection of a vector u on $\ker \mathcal{E}_{\mathcal{G},d}^\top$ is given by

$$\mathrm{Proj}_{\ker \mathcal{E}_{\mathcal{G},d}^\top} \mathrm{u} = \left(\frac{1}{|\mathbb{V}|} \sum_1^{|\mathbb{V}|} \mathrm{u}_i \right) \otimes \mathbb{1},$$

so we conclude that $\mathbf{0} \in \partial A(\beta)$ is equivalent to $\mathbf{0} \in \sum_1^{|\mathbb{V}|} k_i^{-1}(\mathrm{y}_i^\star + \beta \otimes \mathbb{1})$. Note that plugging $\beta = \mathbf{0}$ gives the exact same condition appearing in Corollary 4.2. Thus if $\mathrm{y}^\star$ satisfies the condition in Corollary 4.2, then it is a solution to the inner minimization problem of (4.4). We want to make sure that it is the only minimizer. By similar methods, we can prove the following result.

Proposition 4.2. *Consider the function $A(\beta) = K^\star(\mathrm{y}^\star + \beta \otimes \mathbb{1})$. If $\mathrm{y}^\star$ satisfies the condition in Corollary 4.2 and A is strictly convex near $\mathbf{0}$, then $\mathrm{y}^\star$ is the unique minimizer of $K^\star(\mathrm{y})$ on the set $\{\mathcal{E}_{\mathcal{G},d}^\top \mathrm{y} = \zeta^\star\}$.*

The proof is exactly the same as the proof of Proposition 4.1. We conclude the subsection with the main synthesis result.

Theorem 4.1 (Synthesis Criterion of MEICMP systems)**.** *Consider a networked system $(\mathcal{G}, \Sigma, \Pi)$, and let $\mathrm{y}^\star$ be the desired steady-state output. Suppose that $\{\Pi_e\}_{e \in \mathbb{E}}$ are output-strictly MEICMP controllers, and denote their input-output relations by γ_e, and the corresponding integral functions by Γ_e. Assume that the following conditions hold:*

i) the equation (4.2) is satisfied by the pair $(\mathrm{y}^\star, \zeta^\star = \mathcal{E}_{\mathcal{G},d}^\top \mathrm{y}^\star)$;

ii) for any $e \in \mathbb{E}$, the function $\Gamma_e^\star$ is strictly convex in a neighborhood of ζ_e;

iii) the function $A : \mathbb{R}^d \to \mathbb{R}$, defined by $A(\beta) = \sum_{i=1}^{|\mathbb{V}|} K_i^\star(y_i^\star + \beta \otimes \mathbb{1})$, is strictly convex near $\beta = \mathbf{0}$;

iv) the vector $\mathbf{0}$ is in the subdifferential set $\sum_{i=1}^{|\mathbb{V}|} k_i^{-1}(y_i^\star)$.

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

Then the output of the closed-loop system globally asymptotically converges to $\mathrm{y}^\star$*. Furthermore, if the agents are output-strictly MEICMP, we can relax our demand and require the controllers* $\{\Pi_e\}_{e\in\mathbb{E}}$ *to only be MEICMP.*

Proof. The MEICMP assumptions imply that the closed-loop system always converges to some solution of (OPP). The equation (4.4), together with conditions i)-iv) show that $(\mathrm{y}^\star, \zeta^\star = \mathcal{E}_{\mathcal{G},d}^\top \mathrm{y}^\star)$ are the unique minimizers of (OPP), implying that the system always converges to $\mathrm{y}^\star$. This completes the proof. □

Example 4.1. *Consider the controllers constructed in* (4.3)*:*

$$\Pi_e : \begin{cases} \dot{\eta_e} = -\eta_e + \zeta_e - (\xi_e + \zeta_e^\star) \\ \mu_e = \eta_e \end{cases},$$

for some $\{\xi_e\}_{e\in\mathbb{E}}$ *which are a function of* $\mathrm{y}^\star$*, and chosen so that condition i) of Theorem 4.1 is satisfied. In that case, we can compute and see that* $\gamma_e(\zeta_e) = \zeta_e - \xi_e - \zeta_e^\star$*, so that* $\Gamma_e(\zeta_e) = \frac{1}{2}||\zeta_e||^2 - \zeta_e^\top(\xi_e + \zeta_e^\star)$ *is a strictly convex function, yielding that condition ii) is satisfied. We note that the condition i) is equivalent in this case to* $k^{-1}(\mathrm{y}^\star) = \mathcal{E}_{\mathcal{G},d}\xi$*, meaning that* ξ *can be found in polynomial time (in the number of agents) by using the method of least squares, namely* $\xi = (\mathcal{E}_{\mathcal{G},d}^\top \mathcal{E}_{\mathcal{G},d})^\dagger \mathcal{E}_{\mathcal{G},d}^\top k^{-1}(\mathrm{y}^\star)$*, where* $\mathcal{E}_{\mathcal{G},d}^\top \mathcal{E}_{\mathcal{G},d}$ *is the tensor product of the edge Laplacian of the graph* $\mathcal{G}$ *with the identity matrix* Id_d *[174].*

Remark 4.2. *Note that the conditions iii) and iv) in Theorem 4.1 are controller independent, but conditions i) and ii) can always be satisfied by a correct choice of controller. This is a phenomenon similar to the one appearing in consensus protocols, in which agreement is achieved, but its convergence point is completely determined by the initial conditions of the agents and cannot be controlled.*

4.2.3 Changing the Objective and "Formation Reconfiguration"

In practical applications, we may want to change the desired output $\mathrm{y}^\star$ after some time. However, we wish to avoid a change in the controller design scheme. Note that in Example 4.1, we used the desired output $\mathrm{y}^\star$ to define the vector $\xi + \zeta^\star$. Barring this vector, the controller was independent of $\mathrm{y}^\star$. In this section, we propose a "Formation Reconfiguration" scheme, allowing one to solve the synthesis problem for arbitrary desired achievable output vector $\mathrm{y}^\star \in \mathbb{R}^{|\mathbb{E}|}$ using controller augmentation.

We wish to implement a mechanism similar to Example 4.1 for general controllers. We take a stacked controller of the form (1.2), and add a constant exogenous input $\omega = (\alpha, \beta)$,

$$\Pi_\omega^\# : \begin{cases} \dot{\eta} = \phi(\eta, \zeta - \alpha) \\ \mu = \psi(\eta, \zeta - \alpha) + \beta. \end{cases}$$

© Technion - Israel Institute of Technology, Elyachar Central Library

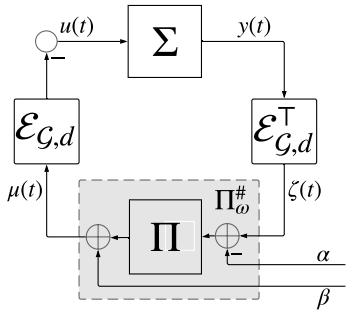


Figure 4.1: The formation reconfiguration scheme.

This design allows us to alter the controller by changing $\omega = (\alpha, \beta)$, yielding different steady-state outputs. We denote the steady-state output of the closed-loop system with the controller $\Pi^{\#}_{\omega}$ as y_0, i.e., $\Pi^{\#}_{\omega}$ solves the synthesis problem for y_0.

The following result implies that it is enough to solve the synthesis problem for a single output or relative output (e.g., consensus), applying the "formation reconfiguration" procedure to force any other forcible desired output.

Theorem 4.2 (Formation Reconfiguration)**.** *Consider a networked system $(\mathcal{G}, \Sigma, \Pi)$, and suppose that its output converges to y_0. Then there is a function $g : y \mapsto \omega$ such that for any desired achievable output $\mathrm{y}^\star$, satisfying conditions iii) and iv) of Theorem 4.1, if one defines $\alpha = \mathcal{E}^{\top}_{\mathcal{G},d}\mathrm{y}^\star - \mathcal{E}^{\top}_{\mathcal{G},d}\mathrm{y}_0$ and $\beta = g(\mathrm{y}^\star) - g(\mathrm{y}_0)$, then the output of the networked system $(\mathcal{G}, \Sigma, \Pi^{\#}_{\omega})$ converges to $y^\star$.*

The theorem can be understood using the standard approach of changing the steady-state behavior by changing the reference signal. The main point of Theorem 4.2 is that any steady-state limit can be achieved by appropriately choosing the reference signal. The controllers produced by the formation reconfiguration scheme are illustrated in Figure 4.1.

Proof. The steady-state input-output relation $\gamma^{\#}_{\omega}$ of $\Pi^{\#}_{\omega}$ can be computed from γ using the equation

$$\gamma^{\#}_{\omega}(\zeta) = \gamma(\zeta - \alpha) + \beta.$$

Given any achievable y, we know from condition iv) of Theorem 4.1 that $k^{-1}(\mathrm{y}) \cap \mathrm{Im}(\mathcal{E}_{\mathcal{G},d}) \neq \emptyset$, so we take some $\mu_{\mathrm{y}} \in (\mathbb{R}^d)^{|\mathbb{E}|}$ such that $-\mathcal{E}_{\mathcal{G},d}\mu_{\mathrm{y}} \in k^{-1}(\mathrm{y})$; we define $g(\mathrm{y}) = \mu_{\mathrm{y}}$.

Now, take some achievable $\mathrm{y}^\star$. We denote $\zeta_0 = \mathcal{E}^{\top}_{\mathcal{G},d}\mathrm{y}_0$, and $\zeta^\star = \mathcal{E}^{\top}_{\mathcal{G},d}\mathrm{y}^\star$, so

that $\alpha = \zeta^\star - \zeta_0$, and $\beta = \mu_{\mathrm{y}^\star} - \mu_{\mathrm{y}_0}$. Then,

$$\begin{aligned} k^{-1}(\mathrm{y}^\star) &= k^{-1}(\mathrm{y}_0) + [k^{-1}(\mathrm{y}^\star) - k^{-1}(\mathrm{y}_0)] \\ &= -\mathcal{E}_{\mathcal{G},d}\gamma(\zeta_0) - \mathcal{E}_{\mathcal{G},d}(\mu_{\mathrm{y}^\star} - \mu_{\mathrm{y}_0}) = -\mathcal{E}_{\mathcal{G},d}\left(\gamma(\zeta_0) - \mu_{\mathrm{y}_0} + \mu_{\mathrm{y}^\star}\right) \\ &= -\mathcal{E}_{\mathcal{G},d}(\gamma(\zeta_0) + \beta) = -\mathcal{E}_{\mathcal{G},d}(\gamma(\zeta_0) + \beta) = -\mathcal{E}_{\mathcal{G},d}\gamma_\omega^\#(\zeta_0 + \alpha) \\ &= -\mathcal{E}_{\mathcal{G},d}\gamma_\omega^\#(\zeta^\star), \end{aligned}$$

which proves our claim. □

4.2.4 Plant Augmentation and Leading Agents for Non-achievable Steady States

We saw in Subsection 4.2.1 that y can be forced as a steady-state of the system if and only if $\mathbf{0} \in \sum_{i\in\mathbb{V}} k_i^{-1}(\mathrm{y}_i)$. This can be troublesome in applications, in which a certain non-forcible steady-state can be desired for various reasons, e.g. cost minimization or efficiency maximization.

One method of coping with this problem is slightly augmenting the plant. This is done by introducing a constant external reference signal z to some of the nodes. In this direction, we consider a generalized notion of the nodal dynamical systems, which can be figuratively seen in Figure 4.2:

$$\Sigma_i' : \begin{cases} \dot{x_i} = f_i(x_i, u_i + \mathrm{z}_i, \mathrm{w}_i) \\ y_i = h_i(x_i, u_i + \mathrm{z}_i, \mathrm{w}_i), \end{cases} \tag{4.6}$$

Note that if a node is forced to have $\mathrm{z}_i = 0$, it is of the unaugmented form we studied earlier. We say that a node is a *follower* if we force it to have $\mathrm{z}_i = 0$, and we call it a *leader* otherwise. We focus on the case in which there is only one leading node, $i_0 \in \mathbb{V}$. Our interest in leading nodes can be summarized by the following definition.

Definition 4.2. *Let* $\mathrm{y} \in (\mathbb{R}^d)^{|\mathbb{V}|}$. *We say that the leading node* $i_0 \in \mathbb{V}$ *can* force y *if there is some constant vector* z_{i_0}, *such that the closed-loop system, with exogenous input* z_{i_0} *to the node* i_0 *and zero exogenous input for all nodes* $j \neq i_0$, *has* y *as a steady-state. We say that the leading node* $i_0 \in \mathbb{V}$ *is* omnipotent *if it can force any vector* $\mathrm{y} \in \mathrm{Im}(k)$.

Theorem 4.3. *Consider the network system* $(\mathcal{G}, \Sigma, \Pi)$ *and suppose all agents are MEICMP. Furthermore let* $i_0 \in \mathbb{V}$ *be the only leading node (i.e.,* $\mathrm{z}_i = 0$ *for all* $i \neq i_0$*). Then* i_0 *is omnipotent.*

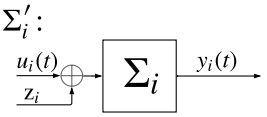


Figure 4.2: Agent augmentation overcoming non-achievable steady-states.

© Technion - Israel Institute of Technology, Elyachar Central Library

Proof. Recall that the steady-state input-output relations for the i-th node with zero exogenous input were denoted by k_i, and denote the steady-state input-output relation for the constant exogenous input z_{i_0} by $k_{i_0,\mathrm{z}_{i_0}}$. Then

$$k_{i_0,\mathrm{z}_{i_0}}(\mathrm{u}_{i_0}) = k_{i_0}(\mathrm{u}_{i_0} + \mathrm{z}_{i_0}),\ k^{-1}_{i_0,\mathrm{z}_{i_0}}(\mathrm{y}_{i_0}) = k^{-1}_{i_0}(\mathrm{y}_{i_0}) - \mathrm{z}_{i_0}.$$

Thus, we obtain that $i_0 \in \mathbb{V}$ can force $\mathrm{y} \in \mathbb{R}^d$ if there is some $\mathrm{z}_{i_0} \in \mathbb{R}^d$ such that

$$\mathbf{0} \in \sum_{i \neq i_0} k_i^{-1}(\mathrm{y}_i) + k^{-1}_{i_0,\mathrm{z}_{i_0}}(\mathrm{y}_{i_0}) = \sum_{i \in \mathbb{V}} k_i^{-1}(\mathrm{y}_i) - \mathrm{z}_{i_0}.$$

Hence, if we pick z_{i_0} to be some vector in $\sum_{i\in\mathbb{V}} k_i^{-1}(\mathrm{y}_i)$, then we get that indeed $\mathbf{0} \in \sum_{i\neq i_0} k_i^{-1}(\mathrm{y}_i) + k^{-1}_{i_0,\mathrm{z}_{i_0}}(\mathrm{y}_{i_0})$, allowing to force y as a steady-state. Thus i_0 is omnipotent. □

4.2.5 Final-Value Synthesis for Other Signals

Up to now, we have touched only on the final-value problem for the output vector, $y(t)$. One can consider an analogue to Problem 4.1 for a desired relative output vector $\zeta^\star$, a desired input vector $\mathrm{u}^\star$, and a desired controller output vector $\mu^\star$. This section briefly summarizes the small adaptations in the case of relative-output synthesis, and briefly discusses the other two synthesis problems.

The main difference between the synthesis problem for a desired output vector $\mathrm{y}^\star$ and the synthesis problem for a desired formation vector $\zeta^\star$ can be seen in the following theorem and corollary:

Theorem 4.4. *For every $\zeta \in \mathrm{Im}(\mathcal{E}^\top_{\mathcal{G},d})$, there exists a vector* y *such that $\mathcal{E}^\top_{\mathcal{G},d}\mathrm{y} = \zeta$ and $k^{-1}(y) \cap \mathrm{Im}(\mathcal{E}_{\mathcal{G},d}) \neq \emptyset$.*

Proof. We consider the function A defined before, which is the restriction of $K^\star$ on the set $Y = \{\mathrm{y} : \mathcal{E}^\top_{\mathcal{G},d}\mathrm{y} = \zeta\}$. As A is a convex function defined on an affine subspace, it must have a minimum at some point $\mathrm{y} \in Y$. Moreover, we know that the zero vector lies in $\partial A(\mathrm{y})$. In Appendix A, it is shown that $\partial A(y) = \mathrm{Proj}_{\ker \mathcal{E}^\top_{\mathcal{G},d}}(k^{-1}(\mathrm{y}))$. Thus, because we know that the zero vector is in the subgradient of A at y, we conclude that there is some vector $\mathrm{u} \in k^{-1}(\mathrm{y})$ such that $\mathrm{Proj}_{\ker \mathcal{E}^\top_{\mathcal{G},d}}(\mathrm{u}) = \mathbf{0}$, which is the same as $\mathrm{u} \in \ker(\mathcal{E}^\top_{\mathcal{G},d})^\perp = \mathrm{Im}(\mathcal{E}_{\mathcal{G},d})$. This completes the proof. □

Corollary 4.4. *For every $\zeta^\star \in \mathrm{Im}(\mathcal{E}^\top_{\mathcal{G},d})$, there exists some vectors* y, μ *such that both $\mathbf{0} \in k^{-1}(\mathrm{y}) + \mathcal{E}_{\mathcal{G},d}\mu$ and $\mathcal{E}^\top_{\mathcal{G},d}\mathrm{y} = \zeta^\star$ hold. In particular, if one chooses MEICMP controllers $\{\Pi_e\}$ such that $\gamma(\zeta^\star) = \mu$, then (4.2) is satisfied, implying that $(\mathrm{y}, \zeta^\star)$ is a minimizer of OPP, and that $\zeta^\star$ is a steady-state output of the system $(\mathcal{G}, \Sigma, \Pi)$.*

Proof. Take y to be the vector from Theorem 4.4. As $k^{-1}(\mathrm{y}) \cap \mathrm{Im}(\mathcal{E}_{\mathcal{G},d}) \neq \emptyset$, we conclude that there is some vector μ such that $\mathcal{E}_{\mathcal{G},d}(-\mu) \in k^{-1}(\mathrm{y})$. This is equivalent to $\mathbf{0} \in k^{-1}(\mathrm{y}) + \mathcal{E}_{\mathcal{G},d}\mu$. □

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

This solves the first part of the synthesis problem, showing that any formation vector $\zeta^\star \in \mathrm{Im}(\mathcal{E}_{\mathcal{G},d}^\top)$ can be forced as a steady-state. As for the second part, the discussion in Section 4.2 produces the following conclusion:

Remark 4.3. *Consider Theorem 4.1, where we only assume conditions i) and ii), i.e. we assume that:*

i) the equation (4.2) *is satisfied by the pair* $(\mathrm{y}^\star, \zeta^\star)$*;*

ii) for any $e \in \mathbb{E}$*, the function* $\Gamma_e^\star$ *is strictly convex in a neighborhood of* ζ_e*;.*

Then we get that the system $(\mathcal{G}, \Sigma, \Pi)$ *converges to some* $\hat{\mathrm{y}}$ *which satisfies* $\mathcal{E}_{\mathcal{G},d}^\top \hat{\mathrm{y}} = \mathcal{E}_{\mathcal{G},d}^\top \mathrm{y}^\star = \zeta^\star$*. We can take* $\mathrm{y}^\star$ *to be the vector* y *from Theorem 4.4, then we conclude that condition i) is satisfied. Thus, the only requirement for global asymptotical convergence to* $\zeta^\star$ *is strict convexity of* Γ *around it. It should be noted that the controllers from Example 4.1 always satisfy this requirement.*

The proof of the remark goes word-by-word as the proof of Theorem 4.1.

Remark 4.4. *In the general case, checking whether some function* $\Gamma : \mathbb{R}^{d|\mathbb{E}|} \to \mathbb{R}^{d|\mathbb{E}|}$ *is strictly convex (near* $\zeta^\star$*) might be a hard task - one can try and show that for all vectors* ζ_0, ζ_1, μ_0 *and any* $t \in \mathbb{R}$*, the function* $t \mapsto \mu_0^\top \gamma(\zeta_0 + t\zeta_1)$ *has no horizontal lines in its graph, which might not easily checkable.*

However, one should note that $\Gamma(\zeta) = \sum_e \Gamma_e(\zeta_e)$ *has the property that it is strictly convex (near* $\zeta^\star$*) if and only if* Γ_e *are strictly convex (near* $\zeta_e^\star$*) for all* $e \in \mathbb{E}$*. When the agents are SISO, this is easier to verify geometrically since the* $\Gamma_e : \mathbb{R} \to \mathbb{R}$ *are one-dimensional maps. In particular,* Γ_e *is strictly convex if its graph does not contain any straight lines, or equivalently,* γ_e *does not have any horizontal lines for all* $e \in \mathbb{E}$*. We emphasize a few special cases of importance.*

i) The relation $\gamma(\zeta) = \nabla\psi(\zeta)$ *defines a strictly convex function if and only if* ψ *is strictly convex.*

ii) The relation $\gamma(\zeta) = M\zeta + \mu_0$ *defines a strictly convex function if and only if* M *is a positive-definite matrix.*

iii) If γ *defines a monotone relation and it is given by a differentiable map* $\gamma(\zeta) = \phi(\zeta)$*, then it defines a strictly convex function near* $\zeta^\star$ *if the differential* $d\phi(\zeta^\star)$ *is a full-rank matrix. Note that it is possible that this condition is violated, but that we still have a strictly convex function - for example* $\gamma_e(\zeta_e) = \zeta_e^3$ *with* $\zeta^\star = \mathbf{0}$*.*

This concludes the theory for the synthesis problem for relative output vectors. One could also consider a synthesis problem with respect to the vector $\mathrm{u}^\star$, or the vector $\mu^\star$. For these cases, Proposition 2.2 is replaced with Proposition 2.3 as the tool used to design the controllers. In this case, unlike the synthesis for the output y, the equation in Proposition 2.2 can always be satisfied by a proper choice of $\gamma^{-1}(\mu^\star)$, for any value of $k(\mathrm{u}^\star) = k(-\mathcal{E}_{\mathcal{G},d}\mu^\star)$. One can also establish global asymptotic converge using strict convexity (of $\Gamma^\star$) as before.

We now exemplify the results of this chapter in case studies.

© Technion - Israel Institute of Technology, Elyachar Central Library

4.2.6 Case Studies

We consider two case studies regarding the final-value synthesis problem for the relative-output vector, and one case study regarding the final-value synthesis problem for the output vector.

4.2.6.1 Simple Integrators and Nonlinear Consensus

We now focus on the case in which our agents are single SISO integrators. They are governed by the equations $\dot{x_i} = u_i;\ y_i = x_i$. The input-output steady-state of each node and the corresponding integral function is given by:

$$k_i(\mathrm{u}_i) = \begin{cases} \mathbb{R}^N, & \mathrm{u}_i = 0 \\ \emptyset, & \mathrm{u}_i \neq 0 \end{cases},\ K_i(\mathrm{u}_i) = \begin{cases} 0, & \mathrm{u}_i = 0 \\ \infty, & \mathrm{u}_i \neq 0, \end{cases} \tag{4.7}$$

which has a dual function $K_i^\star(\mathrm{y}_i) = 0$. This simplifies the problem (OPP), as it reduces to optimizing $\Gamma(\zeta)$ over $\zeta \in \mathrm{Im}(\mathcal{E}_{\mathcal{G},d}^\top) = \mathrm{Im}(\mathcal{E}^\top)$. Equivalently, we can start from (4.2) and conclude that the equation at the minimum is just $\mathcal{E}\gamma(\zeta) = \mathbf{0}$.

Suppose we want to reach output agreement (i.e., the formation $\zeta^\star = \mathbf{0}$). We need $\gamma(\mathbf{0})$ to be in the kernel of $\mathcal{E}$. Thus a pick as $\gamma_k(\zeta_k) = \zeta_k \cdot \exp(\zeta_k^2)$ is viable. Furthermore, the corresponding convex function is $\Gamma_k(\zeta_k) = \frac{1}{2}\exp(\zeta_k^2)$, which is strictly convex. Thus the closed-loop system converges to a steady-state output-agreement. Implementing these controllers leads to the closed-loop system

$$\dot{x}_i = \sum_{j \sim i} (x_j(t) - x_i(t)) \cdot (\exp((x_j(t) - x_i(t))^2)),$$

which is a nonlinear coupling driving the system to consensus. Similarly, the choice $\gamma(\zeta) = \zeta$ with $\Gamma(\zeta) = \frac{1}{2}\|\zeta\|^2$, will lead to the well-known linear consensus protocol [92, 108], defined as:

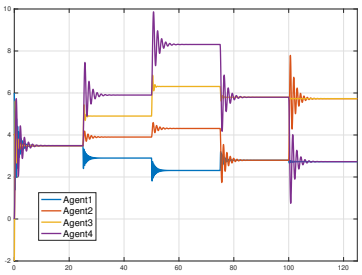
$$\dot{x}_i = \sum_{j \sim i} (x_j(t) - x_i(t)).$$

4.2.6.2 Formation Reconfiguration of Damped Oscillators

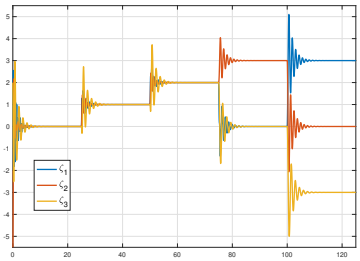
We consider a network of four damped SISO oscillators,

$$\Sigma_i \begin{cases} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} & = \begin{bmatrix} x_2 \\ -b_i x_2 - \omega_i^2 (x_1 - \mathrm{x}_i) + u \end{bmatrix}, \\ y & = x_1 \end{cases}$$

where x_i is the equilibrium point of the spring. The underlying graph was chosen to be a path graph (i.e., $\mathbb{V} = \{v_1, v_2, v_3, v_4\}, \mathbb{E} = \{\{v_1, v_2\}, \{v_2, v_3\}, \{v_3, v_4\}\}$). For the diagonal matrix W with ω_i on the diagonal, the input-output steady-state relation is given by $k(u) = W^{-2}u + \mathrm{x}$. This implies that $K^\star(y) =$



(a) The oscillators' outputs.



(b) The formation vector $\zeta(t) = \mathcal{E}^\top y(t)$ of the oscillators.

Figure 4.3: Formation control of damped oscillators.

$\frac{1}{2}y^\top W^2 y - y^\top W^2 \mathrm{x}$, and the minimization algorithm solving (4.2) can be solved by methods of quadratic programming. For this example, the values ω_i, b_i, and x_i were chosen randomly as

$$\omega = [15.54,\ 5.13,\ 7.89,\ 4.29](Hz),$$

$$b = [1.66,\ 1.22,\ 4.62,\ 1.23](1/sec),$$

$$\mathrm{x} = [3,\ -2,\ 1,\ 0](m).$$

For a consensus objective, $\zeta^\star = \mathbf{0}$, (4.2) reduces to $\mathcal{E}\gamma(\zeta^\star) = \mathbf{0}$. Solving for this $\zeta^\star$, we choose $\gamma(\zeta) = \tanh(\zeta)$. To implement the said input-output relation, we take the following SISO controller on each of the edges,

$$\begin{cases} \dot{\eta_k} = -\eta_k + \zeta_k \\ \mu_k = \tanh(\eta_k). \end{cases}$$

We then use the formation reconfiguration scheme to create an augmented controller. The desired formation was changed every 25 seconds as $\zeta^1 = [0,0,0]^\top$, $\zeta^2 = [1,1,1]^\top$, $\zeta^3 = [2,2,2]^\top$, $\zeta^4 = [0,3,0]^\top$, and $\zeta^5 = [3,0,-3]^\top$. The output $y(t)$ of the system can be seen in Figure 4.3(a) and relative outputs ζ in Figure 4.3(b). We can see that the agents do as their supposed to, converging to the desired formations.

4.2.6.3 Formation Reconfiguration of Damped MIMO Oscillators

We consider a network of four damped MIMO oscillators:

$$\Sigma_i \begin{cases} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \Omega_i x_2 \\ -D_i x_2 - \Omega_i^\top (x_1 - \mathrm{x}_i) + \Omega_i^{-1} u \end{bmatrix} \\ y = x_1 \end{cases}$$

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

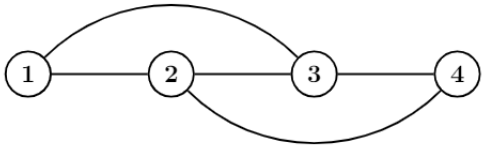


Figure 4.4: Network structure for synthesis of damped MIMO oscillators example.

where x_i is the equilibrium point of the oscillator, Ω_i is a matrix consisting of the self frequencies, and D_i is a damping matrix, which is positive-definite. The exact values of the matrices and x_i-s were randomly chosen. The underlying graph is given in Figure 4.4.

The steady-state input-output relation if Σ_i can be computed to be $k_i(u_i) = (\Omega_i\Omega_i^\top)^{-1}u + \mathrm{x}_i$, whose inverse is $k_i^{-1}(y) = (\Omega_i\Omega_i^\top)(y - \mathrm{x_i})$. This gives us the convex function $K_i^\star(y_i) = \frac{1}{2}y^\top\Omega_i\Omega_i^\top y - y^\top\Omega_i\Omega_i^\top\mathrm{x}_i$, which is strictly convex.

We solve the synthesis problem for $y^\star$ (see Chapter 4). Where the controllers are taken to be identical and equal to

$$\begin{cases}\dot{\eta_e} = -\eta_e + \zeta_e \\ \zeta_e = \psi(\eta_e).\end{cases}$$

The function ψ is given as

$$\psi(x) = \arcsin\left(\frac{\log^2\left(\frac{e^x+1}{2}\right)\mathrm{sgn}(x)}{\log^2\left(\frac{e^x+1}{2}\right)+1}\right),$$

where $\mathrm{sgn}(x)$ is the sign function. One can verify that $\psi(0) = 0$ and that ψ is a monotone ascending function. The associated integral function is given by:

$$\Gamma_e(\zeta_e) = \int_0^{\zeta_e}\arcsin\left(\frac{\log^2\left(\frac{e^x+1}{2}\right)\mathrm{sgn}(x)}{\log^2\left(\frac{e^x+1}{2}\right)+1}\right)dx.$$

We use the formation reconfiguration scheme (Theorem 4.2) to create an augmented controller, where we use the first node as a leading node (see Section 4.2). The control objective was changed every 30 seconds according to the following desired steady-states,

$$y^{\star 1} = [0,0,0,0,0,0,0,0]^\top, \qquad y^{\star 2} = [1,1,2,2,3,3,4,4]^\top,$$
$$y^{\star 3} = [1,2,3,4,5,6,7,8]^\top, \qquad y^{\star 4} = [-1,0,0,0,1,0,2,2]^\top,$$
$$y^{\star 5} = [2,2,2,2,2,2,-10,-10]^\top,$$

where the first two entries refer to the first agent, the next two refer to the second agent, and so on. The output of the system can be seen in Figure 4.5, exhibiting the positions of the agents $y(t)$. The blue line represents first coordinate, and the red one represents the second coordinate. We can see that the agents act as expected, converging to the desired formations.

© Technion - Israel Institute of Technology, Elyachar Central Library

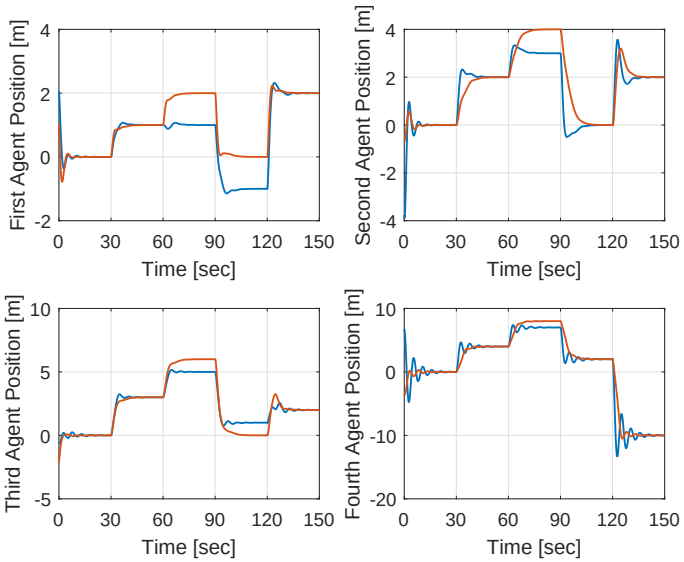


Figure 4.5: Formation control of damped MIMO oscillators.

4.3 Clustering in Symmetric Multi-Agent Systems

In this section, we treat a special case of the final-value synthesis problem, in which the control goal is for the agents to form clusters of prescribed sizes. Various methods have been used to study clustering, e.g. structural balance of the underlying graph [1], pinning control [111] and inter-cluster nonidentical inputs [59].

We approach the clustering problem using symmetry. The notion of symmetry is one of the cornerstones of mathematics and physics. It is used in control theory extensively for many different applications. Examples include designing observers [16], more efficient algorithms for model-predictive control [38], and bipedal locomotion [146]. In cooperative control, symmetry on the network level was used in [27, 28, 117] to study controllability and observability. However, these works discuss network symmetries preserving the agents' models, which can be different even if the agents are equivalent. Moreover, the current literature about symmetries in multi-agent systems deals with symmetries in the trajectories of the agents, although consensus and clustering only require symmetries on the steady-state level. In order to use the network optimization framework, we make either of the following two assumptions:

Assumption 4.1. *The agents Σ_i are output-strictly MEIP and the controllers Π_e are MEIP.*

Assumption 4.2. *The agents Σ_i are MEIP and the controllers Π_e are output-strictly MEIP.*

© Technion - Israel Institute of Technology, Elyachar Central Library

This rest of this section is as follows. First, we define the notion of static automorphisms for multi-agent systems. These are symmetries of the system on the steady-state level. We then show that the group of all static automorphisms acts on $\mathcal{G}$, and that this action can be used to understand the agents' clustering behavior. Lastly, we restrict ourselves to the case of statically-homogeneous networks, in which all agents and all controllers exhibit the same steady-state behavior, and consider the cluster synthesis problem, in which the agents are required to cluster in a prescribed manner. We begin with a brief overview about the role of symmetry in control and in multi-agent systems .

4.3.1 The Static Automorphism Group of a Multi-Agent System

As stated, symmetries have been used in the study of control laws for many systems [16, 38, 146]. In cooperative control, symmetries were used in the study of controllability and observability [27, 28, 117]. Namely, in [117], it is shown that if we have a weighted graph $\mathcal{G} = (\mathbb{V}, \mathbb{E}, \mathbb{W})$ and input nodes $S \subset \mathbb{V}$, then the controlled consensus system $\dot{x} = -L(\mathcal{G})x + Bu$, where $L(\mathcal{G})$ is the graph Laplacian and B is supported on S, is uncontrollable, as long as there exists a nontrivial graph automorphism $\psi \in \mathrm{Aut}(\mathcal{G})$ such that P_ψ commutes with $L(\mathcal{G})$ and $P_\psi B = B$. Later, [27] expended this idea to "Fractional Automorphism", using the inherent linearity of the system.

Pushing this idea a step further, we want to consider more general systems. A first step is the case of linear systems. If we try and mimic [27], then we require that the symmetry matrix P_ψ, which corresponds to some permutation, commutes with the dynamics matrix A of the entire system. This has a few drawbacks - The main one is that this is extremely model-dependent, i.e., different matrices A might yield different symmetries, even though the agents are equivalent. Specifically, on a two-vertex graph with one edge, where both agents have the same dynamics, but different realizations of the model, the graph automorphism exchanging the vertices is not a symmetry.

One possible direction to remove this problem is to try and use realization-independent models, like the transfer function for an LTI system. However, we take a different path, as we mostly care about the steady-state limit for clustering. We first define the notion of static equivalence between dynamical systems.

Definition 4.3. *Two dynamical systems Υ_1,Υ_2 are called* statically equivalent *if their steady-state input-output relations are identical.*

Example 4.2. *Consider the following dynamical systems:*

$$\Upsilon_1: \ y = u \qquad \Upsilon_2: \begin{cases} \dot{x} = -x + u, \\ y = x \end{cases}$$

$$\Upsilon_3: \begin{cases} \dot{x} = -10x + u, \\ y = 10x \end{cases} \qquad \Upsilon_4: \begin{cases} \dot{x} = -\tanh(x) + u, \\ y = \tanh(x) \end{cases}$$

$$\Upsilon_5: \begin{cases} \dot{x} = -x + \sinh(u), \\ y = \operatorname{arcsinh}(x) \end{cases} \qquad \Upsilon_6: \begin{cases} \dot{x} = -x + u, \\ y = 0.5(x + u) \end{cases}$$

These systems are vastly different from one another. One is memory-less, while the others are not. Some are LTI, and some are nonlinear. Of the nonlinear ones, one is input-affine nonlinear, while the other is not. All are output-strictly passive, but only Υ_6 is input-strictly passive. However, all of these systems have the steady-state input-output relation $k(\mathrm{u}) = \mathrm{u}$, meaning that they are statically equivalent.

Definition 4.4. *Let $(\mathcal{G}, \Sigma, \Pi)$ be any diffusively-coupled system. A* static automorphism *is a map $\psi : \mathbb{V} \to \mathbb{V}$ such that the following conditions hold:*

i) The map ψ is an automorphism of the graph $\mathcal{G}$.

ii) For any $i \in \mathbb{V}$, Σ_i and $\Sigma_{\psi(i)}$ are statically equivalent.

iii) For any $e \in \mathbb{E}$, Π_e and $\Pi_{\psi(e)}$ are statically equivalent.

iv) The map ψ preserves edge orientation.

We denote the collection of all static automorphisms of $(\mathcal{G}, \Sigma, \Pi)$ by $\mathrm{Aut}(\mathcal{G}, \Sigma, \Pi)$. Naturally, this is a subgroup of the group of automorphisms $\mathrm{Aut}(\mathcal{G})$ of the graph $\mathcal{G}$.

The name "static" automorphism hints at the existence of a "dynamic" automorphism. That would be an automorphism sending agents and controllers to agents and controllers having the same dynamics (e.g. that can be modeled using the same model). We shall not expand on that notion in this thesis.

Remark 4.5. *The requirement that the map ψ is orientation-preserving might appear unnatural - usually, the choice of edge orientation in a diffusively-coupled system is arbitrary. However, this is a key point in proving that these static automorphisms commute with $\mathcal{E}_{\mathcal{G},d}$, which in turn allows one to show that the steady-states of the system are invariant under these static automorphisms. This assumption can be traded by another assumption, namely that the controller's steady-state relations γ_e satisfy the relations $\gamma_e(-\mathrm{x}) = -\gamma_e(\mathrm{x})$ for all $e \in \mathbb{E}$ and all* x*. However, as we'll see later, this is enough to guarantee that the network converges to consensus, which prohibits a more refined clustering behavior.*

The remark urges us to consider the following assumption, or its invalidity:

© Technion - Israel Institute of Technology, Elyachar Central Library

Assumption 4.3. *The equality $\gamma_e(-x) = -\gamma_e(x)$ holds for every x and every $e \in \mathbb{E}$.*

Notation 4.1. *Each permutation $\psi : \mathbb{V} \to \mathbb{V}$ defines a linear map $\mathbb{R}^{|\mathbb{V}|} \to \mathbb{R}^{|\mathbb{V}|}$ by permuting the coordinates according to ψ. We denote the linear operator by P_ψ. If ψ is a graph automorphism for the graph $\mathcal{G} = (\mathbb{V}, \mathbb{E})$, then it gives rise to a permutation $\mathbb{E} \to \mathbb{E}$ on the edges. We denote the corresponding linear map $\mathbb{R}^{|\mathbb{E}|} \to \mathbb{R}^{|\mathbb{E}|}$ by Q_ψ. Namely, $(P_\psi)_{ij} = \delta_{\psi(i)j}$ and $(Q_\psi)_{ef} = \delta_{\psi(e)f}$ for $i, j \in \mathbb{V}$ and $e, f \in \mathbb{E}$.*

Proposition 4.3. *For any graph $\mathcal{G} = (\mathbb{V}, \mathbb{E})$, and for any static automorphism ψ, we have $P_\psi \mathcal{E}_{\mathcal{G},d} = \mathcal{E}_{\mathcal{G},d} Q_\psi$.*

Proof. It's enough to prove it in the case where $d = 1$, i.e. that the agents are SISO and $\mathcal{E}_{\mathcal{G},d} = \mathcal{E}$. Now, for every $i \in \mathbb{V}$ and $e \in \mathbb{E}$,

$$[P_\psi \mathcal{E}]_{ie} = \sum_{k \in \mathbb{V}} (P_\psi)_{ik} \mathcal{E}_{ke} = \sum_{k \in \mathbb{V}} \delta_{\psi(i)k} \mathcal{E}_{ke} = \mathcal{E}_{\psi(i),e}$$
$$[\mathcal{E} Q_\psi]_{ie} = \sum_{f \in \mathbb{E}} \mathcal{E}_{if} (Q_\psi)_{fe} = \sum_{f \in \mathbb{E}} \mathcal{E}_{if} \delta_{\psi(f)e} = \mathcal{E}_{i\psi^{-1}(e)}.$$

Thus, because $\psi(i) \in e$ if and only if $i \in \psi^{-1}(e)$, the entries of $\mathcal{E}$ are the same up to sign. Moreover, because ψ preserves edge orientations, then the signs are the same and the proof is complete. □

4.3.2 Steady-State Clustering in Multi-Agent Systems

We wish to build a connection between the group action of $\mathrm{Aut}(\mathcal{G})$ on $\mathcal{G}$ and the symmetries in the steady-state y of $(\mathcal{G}, \Sigma, \Pi)$. We start with the following proposition.

Proposition 4.4. *The function $F(\mathrm{y}) = K^\star(\mathrm{y}) + \Gamma(\mathcal{E}_{\mathcal{G},d}^\top \mathrm{y})$ is $\mathrm{Aut}(\mathcal{G}, \Sigma, \Pi)$-invariant. In other words, $F(P_\psi \mathrm{y}) = F(\mathrm{y})$ for any $\mathrm{y} \in \mathbb{R}^{|\mathbb{V}|}$ and $\psi \in \mathrm{Aut}(\mathcal{G}, \Sigma, \Pi)$.*

Proof. We first note that $K_i = K_{\psi(i)}$ and $\Gamma_e = \Gamma_{\psi(e)}$, as $k_i = k_{\psi(i)}$ and $\gamma_e = \gamma_{\psi(e)}$. Thus,

$$\begin{aligned} K(P_\psi \mathrm{y}) &= \sum_{i \in \mathbb{V}} K_i((P_\psi \mathrm{y})_i) = \sum_{i \in \mathbb{V}} K_i(\mathrm{y}_{\psi(i)}) = \sum_{i \in \mathbb{V}} K_{\psi(i)}(\mathrm{y}_{\psi(i)}) = \sum_{j \in \mathbb{V}} K_j((\mathrm{y})_j) \\ &= K(\mathrm{y}), \end{aligned}$$

where we use the switch $j = \psi(i)$ and the fact that $\psi : V \to V$ is a bijection. Similarly, due to Proposition 4.3, one has

$$\Gamma(\mathcal{E}_{\mathcal{G},d}^\top P_\psi \mathrm{y}) = \Gamma(Q_\psi \mathcal{E}_{\mathcal{G},d}^\top \mathrm{y}) = \sum_{e \in \mathbb{E}} \Gamma_e((Q_\psi \mathcal{E}_{\mathcal{G},d}^\top \mathrm{y})_e) = \sum_{e \in \mathbb{E}} \Gamma_e((Q_\psi \mathcal{E}_{\mathcal{G},d}^\top \mathrm{y})_e) =$$
$$\sum_{e \in \mathbb{E}} \Gamma_e((\mathcal{E}_{\mathcal{G},d}^\top \mathrm{y})_{\psi(e)}) = \sum_{e \in \mathbb{E}} \Gamma_{\psi(e)}((Q_\psi \mathcal{E}_{\mathcal{G},d}^\top \mathrm{y})_{\psi(e)}) = \sum_{k \in \mathbb{E}} \Gamma_k((\mathcal{E}_{\mathcal{G},d}^\top \mathrm{y})_k) = \Gamma(\mathcal{E}_{\mathcal{G},d}^\top \mathrm{y}),$$

where we switch $k = \psi(e)$. This completes the proof. □

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

Corollary 4.5. *Suppose that* $(\mathcal{G}, \Sigma, \Pi)$ *is a diffusively coupled network satisfying either Assumption 4.1 or 4.2. Then the set of steady-state outputs for the diffusively coupled network* $(\mathcal{G}, \Sigma, \Pi)$ *is* $\mathrm{Aut}(\mathcal{G}, \Sigma, \Pi)$*-invariant, i.e., it is preserved when applying* P_ψ*-s for* $\psi \in \mathrm{Aut}(\mathcal{G}, \Sigma, \Pi)$.

Proof. Immediate from Theorem 2.3 and Proposition 4.4. □

Up to now, we showed that if y is a possible steady-state output of the diffusively coupled network $(\mathcal{G}, \Sigma, \Pi)$, for some initial condition, then $P_\psi \mathrm{y}$ is also a possible steady-state output of the network, for some (maybe different) initial condition. We want to push the envelope and show that, actually, $P_\psi \mathrm{y} = \mathrm{y}$. Our main tool, as before, is strong convexity.

Theorem 4.5. *Consider the diffusively-coupled system* $(\mathcal{G}, \Sigma, \Pi)$*, and suppose that either Assumption 4.1 or Assumption 4.2 hold. Then for any steady-state* y *of the closed-loop and any static automorphism* $\psi \in \mathrm{Aut}(\mathcal{G}, \Sigma, \Pi)$*,* $P_\psi \mathrm{y} = \mathrm{y}$.

Proof. We recall that output-strictly MEIP systems have strictly monotone input-output steady-state relations [23], and that a convex function is strictly convex $\mathbb{R} \to \mathbb{R}$ if and only if its subdifferential is a strictly monotone relation [121]. Moreover, we recall that if F is a strictly convex function defined on the affine subspace $\{x \in \mathbb{R}^n : \ Ax = b\}$ for some matrix A and vector b, then it has a unique minimum [121].

Suppose that Assumption 4.1 holds. Then K_i are all strictly convex, and Γ_e are all convex. Thus the function $F(\mathrm{x}) = K^\star(\mathrm{x}) + \Gamma(\mathcal{E}_{\mathcal{G},d}^\top \mathrm{x})$ is strictly convex, meaning it has a unique minimum, which is y by Theorem 2.3. Thus, because $P_\psi \mathrm{y}$ is also a minimizer of F, we conclude that $P_\psi \mathrm{y} = \mathrm{y}$.

Alternatively, suppose that Assumption 4.2 holds. In that case, the functions K_i are convex and Γ_e are strictly convex. Thus F is strictly convex only in directions orthogonal to the consensus line $\mathrm{span}\{\mathbb{1}_{|\mathbb{V}|}\}$, meaning that there could be more than one minimizer. However, we note that for any $d \in \mathbb{R}$, the function F is strictly convex on the affine subspace $\mathcal{A}_c = \{x \in \mathbb{R}^{|\mathbb{V}|} | \mathbb{1}_{|\mathbb{V}|}^\top x = c\}$, meaning that F has a unique minimizer on each of these affine subspaces. Choose $c = \mathrm{y}^\top \mathbb{1}_{|\mathbb{V}|}$, so that $\mathrm{y} \in \mathcal{A}_c$. Because y is a minimizer of F on all of $\mathbb{R}^{|\mathbb{V}|}$, its also a minimizer on $\mathcal{A}_c$, making it the unique minimizer of F on $\mathcal{A}_c$. Noting that $P_\psi \mathrm{y}$ is also a minimizer of F, and that $\mathbb{1}_{|\mathbb{V}|}^\top \mathrm{y} = \mathbb{1}_{|\mathbb{V}|}^\top P_\psi \mathrm{y}$, we get that $\mathrm{y} = P_\psi \mathrm{y}$. □

The theorem shows that the system converges to a steady-state y invariant under static automorphisms. We want to restate it in a manner emphasizing the clustering that occurs. For that, we define the notion of exchangeability

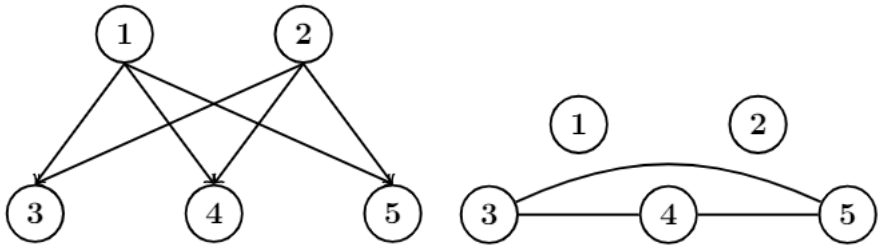
Definition 4.5. *We say that two agents* $i, j \in \mathbb{V}$ *are* exchangeable *if there exists a static automorphism* $\psi \in \mathrm{Aut}(\mathcal{G}, \Sigma, \Pi)$ *such that* $\psi(i) = j$*. We define the* exchangeability graph *of the diffusively-coupled system* $(\mathcal{G}, \Sigma, \Pi)$ *as the graph* $\mathcal{H} = \mathcal{H}(\mathcal{G}, \Sigma, \Pi) = (\mathbb{V}, \mathbb{E}_\mathcal{H})$*, where there is an edge* $\{i, j\} \in \mathbb{E}_\mathcal{H}$ *if* i *and* j *are exchangeable.*

Proposition 4.5. *The exchangeability graph* $\mathcal{H} = \mathcal{H}(\mathcal{G}, \Sigma, \Pi)$ *is a union of disjoint cliques.*

© Technion - Israel Institute of Technology, Elyachar Central Library

Proof. It's enough to show that if there is a path between vertices i, j, then there is an edge $\{i, j\}$. Let i, j be any two vertices, and suppose that there is a path $i = v_0, v_1, v_2, ..., v_{k-1}, v_k = j$ in $\mathcal{H}$. By definition, there are static automorphisms $\psi_0, ..., \psi_{k-1}$ such that $v_r = \psi_{r-1}(v_{r-1})$ for any $r = 1, 2, ..., k$. Because $\mathrm{Aut}(\mathcal{G}, \Sigma, \Pi)$ is a group, the composed map $\psi_{k-1}\psi_{k-2}\cdots\psi_1\psi_0$ is also a static automorphism, and naturally, it maps i to j. Thus the edge $\{i, j\}$ exists in the graph $\mathcal{H}$, completing the proof. □

Example 4.3. *Consider the graph $\mathcal{G}$ in Figure 4.6(a), where all the edges are oriented from $1, 2$ to 3.4.5, the nodes $1, 3, 4, 5$ are all LTI with transfer function $G(s) = \frac{1}{s+1}$, and node 2 is LTI with transfer function $G(s) = \frac{1}{2s+1}$. All edge controllers are static, having the form $\mu_e = \zeta_e$. We compute the exchangeability graph $\mathcal{H}$ of the diffusively coupled system. Suppose ψ is an automorphism of $\mathcal{G}$. Then ψ preserves the degree of each vertex. Thus, the sets $\{1, 2\}$ and $\{3, 4, 5\}$ are all invariant under ψ. Moreover, ψ cannot map 1 to 2, or vice versa, as the agents are not statically equivalent. Furthermore, the map ψ mapping $1 \to 1, 2 \to 2$, and $3 \to 4 \to 5 \to 3$ is a static automorphism of the diffusively-coupled system. Thus the exchangeability graph $\mathcal{H}$ contains the edges $\{3, 4\}, \{4, 5\}$ and $\{5, 3\}$. As we showed that agents 1 and 2 must remain invariant under static automorphism, no more edges exist in the exchangeability graph $\mathcal{H}$, so it is the union of three cliques - $\{1\}, \{2\}$ and $\{3, 4, 5\}$. The graph can be seen in Figure 4.6(b).*



(a) Underlying graph $\mathcal{G}$. (b) The exchangability graph $\mathcal{H}$.

Figure 4.6: A graph of the diffusively-coupled network in Example 4.3 and the corresponding exchangeability graph.

We can now restate Theorem 4.5 in a more profound way:

Theorem 4.6. *Consider the diffusively-coupled system $(\mathcal{G}, \Sigma, \Pi)$, and suppose that either Assumption 4.1 or Assumption 4.2 hold. Then the system converges to a clustering steady-state, with clusters corresponding to the connected components of the exchangeability graph $\mathcal{H}(\mathcal{G}, \Sigma, \Pi)$.*

Proof. The system converges to some steady-state y by Theorem 2.3. By Theorem 4.5, The steady-state y is invariant to all static automorphisms. If we

© Technion - Israel Institute of Technology, Elyachar Central Library

take any two vertices $\{i, j\}$ lying in the same connected component of the exchangeability graph $\mathcal{H}$, then by Proposition 4.5, the edge $\{i, j\}$ is in $\mathcal{H}$. Thus there is an automorphism ψ such that $\psi(i) = j$. Looking at the components of the equation $P_\psi \mathrm{y} = \mathrm{y}$ implies that $\mathrm{y}_i = \mathrm{y}_j$. In other words, we showed that the diffusively coupled system $(\mathcal{G}, \Sigma, \Pi)$ converges to a steady-state, and agents connected in the exchangeability graph $\mathcal{H}$ converge to the same limit. This completes the proof. □

4.3.3 Homogeneous Networks and Cluster Synthesis

In many practical examples, we are dealing with a diffusively coupled network in which the agents are identical. Examples include neural networks, platooning, coupled oscillators, and robotic swarms. Furthermore, in many practical scenarios we may desire to have all controllers in the system identical. This is the case where the agents have no identifiers like serial numbers. We can also try and use this frame to make clustering more robust - even if we use a wrong model for the controllers or the agents, we will still have clustering do to symmetry. It should be noted that designing controllers that force the system to cluster can be done by using the synthesis procedure appearing in the previous section, but there is no guarantee that the achieved edge controllers will be identical, or even statically equivalent. We note that the built scheme allows us to consider networks with identical agents/controllers by using static equivalent:

Definition 4.6. *A diffusively-coupled network is* statically homogeneous *if any two agents, and any two controllers, are statically equivalent.*

As seen in Example 4.2, statically homogeneous networks can include agents and controllers of many different kinds. Moreover, the notion of statically homogeneous networks allows us to study clustering using purely graph-theoretic and combinatorial methods. Indeed, we claim that static automorphisms for $(\mathcal{G}, \Sigma, \Pi)$ are just graph automorphisms of $\mathcal{G}$.

Proposition 4.6. *Let $(\mathcal{G}, \Sigma, \Pi)$ be any statically homogeneous diffusively coupled network. A map $\psi : \mathbb{V} \to \mathbb{V}$ is a static automorphism of the system if and only if $\psi \in \mathrm{Aut}(\mathcal{G})$.*

Proof. Follows from the definition of a static automorphism, and static equivalence of agents and controllers. □

Assumption 4.3 has special significance for statically homogeneous networks.

Theorem 4.7. *Suppose that $(\mathcal{G}, \Sigma, \Pi)$ is statically homogeneous, and either Assumption 4.1 or Assumption 4.2 holds. If Assumption 4.3 holds, then the network converges to consensus.*

Proof. Assumption 4.3 implies that $\gamma(\mathbf{0}) = \mathbf{0}$, meaning that Γ is minimized at $\mathbf{0}$. We note that by definition of statically homogeneous networks, all agents are statically equivalent, and they have the same integral function, i.e. $K_i = K_j$

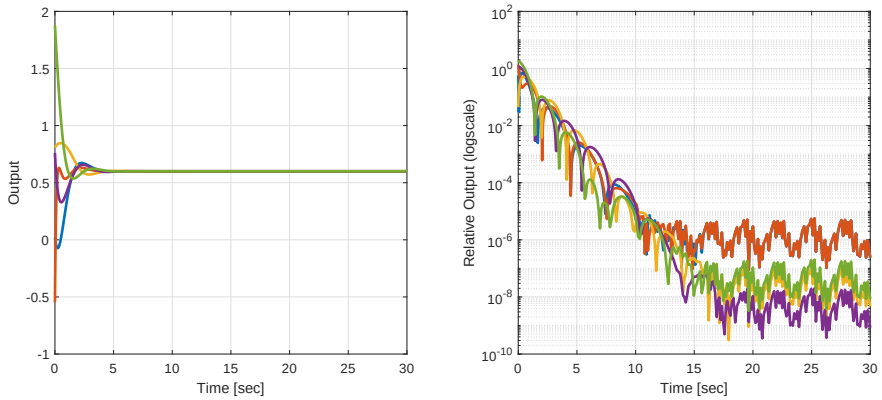


Figure 4.7: Trajectories of a statically homogeneous network with non-identical agents.

for all $i, j \in \mathbb{V}$. If we let β be the minimum of each of the integral functions K_i, then $\mathrm{y} = \beta \otimes \mathbb{1}_{|\mathbb{V}|}$ minimizes both $K(\mathrm{y})$ and $\Gamma(\mathcal{E}_{\mathcal{G},d}^{\top}\mathrm{y})$. Thus it minimizes (OPP), which is strictly convex in any direction orthogonal to the consensus line, meaning y is the unique minimizer, and completing the proof. □

Example 4.4. *We consider a cycle graph $\mathcal{G}$ on 5 nodes. The agents' models are given by $\Upsilon_2, \Upsilon_3, \Upsilon_4, \Upsilon_5, \Upsilon_6$ of Example 4.2, where we add an identical random* constant *exogenous input to all agents to avoid the mundane case of convergence to $\mathrm{y} = \mathbf{0}$. All the controllers on the edges are modeled as Υ_1 of the same example. Obviously, this is a statically homogeneous network with non-identical agents.Furthermore, the automorphism group $\mathrm{Aut}(G)$ can map any vertex in $\mathcal{G}$ to any other vertex, meaning that Theorem 4.6 implies that the system should converge to consensus. The output $y(t)$ and the relative output $\zeta(t)$ of the closed-loop system can be seen in Figure 4.7. It is evident that the system indeed converges to consensus, up to numerical errors due to limited precision.*

So long that Assumption 4.3 does not hold, so consensus is not forced, clustering in statically homogeneous networks can be understood in terms of the action of $\mathrm{Aut}(\mathcal{G})$ on the graph $\mathcal{G}$. One interesting problem that can benefit from this framework is cluster synthesis. Namely, given fixed identical (or statically equivalent) agents, how can one design the interaction graph $\mathcal{G}$ and identical controllers in order to achieve clustering with prescribed cluster sizes, at prescribed values. We give an example of a cluster synthesis problem below. However, it should be noted that the solution to these problems is not unique. For example, both the complete graph and cycle graph work when we want a single cluster, and there are many other solutions, such as Cayley graphs on finite groups [26].

Example 4.5. *We are given five agents, all are LTI with the TF $G(s) = \frac{1}{s+1}$. We wish to find a graph $\mathcal{G}$, and a collection of identical controllers, such that*

© Technion - Israel Institute of Technology, Elyachar Central Library

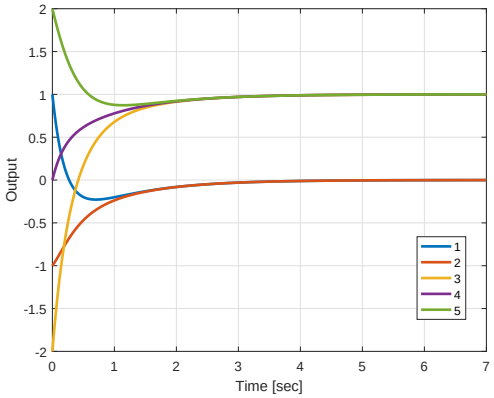


Figure 4.8: Cluster synthesis - The closed-loop system trajectories.

$(\mathcal{G}, \Sigma, \Pi)$ *converges to two clusters, one with two agents and one with three agents. The first cluster should be located at* $\mathrm{y}_i = 1$*, and the second at* $\mathrm{y}_i = 0$*.*

First, according to the discussion at Subsection 4.3.2, we want to find a graph $\mathcal{G}$*, having five vertices, so that vertices* $1, 2$ *are exchangeable, and vertices* $3, 4, 5$ *are exchangeable. We consider the graph again in Figure 4.6(a), where we orient the edges from* $1, 2$ *to* $3, 4, 5$*. Obviously, vertices* $1, 2$ *are exchangeable, and vertices* $3, 4, 5$ *are exchangeable as well (but not with* 1 *and* 2*). It can be shown that this is the graph having the minimal number of edges possessing this property.*

Now for the controller synthesis procedure. As we know from the Section 4.2, not all vectors are available as steady-state outputs of a diffusively-coupled network with prescribed agents. This can be fixed by adding an (identical) constant exogenous input to all agents, and the steady-state equation becomes $\mathrm{w} = k^{-1}(\mathrm{y}) + \mathcal{E}_{\mathcal{G},d}\gamma(\mathcal{E}_{\mathcal{G},d}^{\top}\mathrm{y})$ *(see Proposition 6.1). Writing this equation in coordinates, we get two equations, one for vertices in the* 1^{st} *cluster, having* $\mathrm{y}_i = 0$*, and another for vertices in the* 2^{nd} *cluster, having* $\mathrm{y}_i = 1$*:*

$$w = 0 - 3\gamma_1(1 - 0) = -3\gamma_1(1),$$
$$w = 1 + 2\gamma_1(1 - 0) = 1 + 2\gamma_1(1),$$

where γ_1 *is the steady-state input-output relation for each instance of the (identical) controller. We recall that we also need monotonicity, so* γ_1 *must be monotone. One possible solution to this set of equations is* $w = 0.6$ *and* $\gamma_1(\mathrm{x}) = -1.2 + \mathrm{x}$*, the latter realized by the controller* $\mu_e = -1.2 + \zeta_e$*. We simulate the closed-loop system with the prescribed agents and synthesized graph and controllers. The output of the system is available in Figure 4.8. It is evident that our solution indeed solves the cluster synthesis problem.*

Solving the general form of the cluster synthesis problem is an ongoing affair, and only partial results have been established. We'll present the current results, which revolve around asserting that agents form the desired clusters, without

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

worrying about the clusters locations. Namely, suppose one gives a list $r_1, \cdots, r_k$ of positive integers such that $|\mathbb{V}| = r_1 + \cdots + r_k$. We want to find a weakly connected[2] directed graph $\mathcal{G}$ such that the group action of $\mathrm{Aut}(\mathcal{G})$ on $\mathcal{G}$ forces clusters of sizes $r_1, \cdots, r_k$. This is another way of saying that the corresponding exchangeability graph $\mathcal{H}$ has cliques of sizes $r_1, \cdots, r_k$. The static connectivity requirement stems from demanding that the underlying communication graph is connected, which is desirable for many applications. We wish to understand how many edges a graph $\mathcal{G}$ can have, assuming that the group action of $\mathrm{Aut}(\mathcal{G})$ on $\mathcal{G}$ forces clusters of sizes $r_1, \cdots, r_k$.

Theorem 4.8. *Let $r_1, \cdots r_k$ be any positive integers such that $n = r_1 + \cdots + r_k$.*

i) Suppose that $\mathcal{G}$ is any weakly connected directed graph such that the action of $\mathrm{Aut}(\mathcal{G})$ on $\mathcal{G}$ forces clusters of sizes $r_1, \cdots, r_k$. Then $\mathcal{G}$ has at least m edges, where

$$m = \min_{\mathcal{T} \text{ tree on } k \text{ vertices}} \sum_{e \in \mathcal{T}, e=\{i,j\}} \frac{r_i r_j}{\gcd(r_i, r_j)}, \tag{4.8}$$

where $\gcd(a, b)$ the is the greatest common divisor of a and b.

ii) There exists a weakly connected directed graph $\mathcal{G}$ such that the action of $\mathrm{Aut}(\mathcal{G})$ on $\mathcal{G}$ forces clusters of sizes $r_1, \cdots, r_k$, and $\mathcal{G}$ has M edges, where

$$M = \min_{\mathcal{T} \text{ path on } k \text{ vertices}} \left(\sum_{\substack{e \in \mathcal{T}, \\ e=\{i,j\}}} \frac{r_i r_j}{\gcd(r_i, r_j)} \right) + \min_{i \in \mathbb{V}} r_i. \tag{4.9}$$

Proof. We start by proving the former claim. Consider a graph $\mathcal{G}$ such that the action of $\mathrm{Aut}(\mathcal{G})$ on $\mathcal{G}$ forces the desired clustering structure. Let $V_1, \cdots, V_k$ be the orbits of $\mathrm{Aut}(\mathcal{G})$ in $\mathcal{G}$, i.e., the different clusters in the exchangeability graph. For any two indices $i, j \in \{1, \cdots, k\}$, we consider the induced bi-partite subgraph $\mathcal{G}_{ij}$ on the vertices $V_i \cup V_j$.[3] We claim that if it is not empty, then $\mathcal{G}_{ij}$ has at least $\frac{r_i r_j}{\gcd(r_i, r_j)}$ edges.

Indeed, because V_j is invariant to $\mathrm{Aut}(\mathcal{G})$, for any $x \in V_i$ and any $\psi \in \mathrm{Aut}(\mathcal{G})$, x and $\psi(x)$ have the same number of edges from them to V_j. Thus, all vertices in V_i have the same $\mathcal{G}_{ij}$-degree. Similarly, all vertices in V_j have the same $\mathcal{G}_{ij}$-degree. Let d_i be the $\mathcal{G}_{ij}$-degree of vertices in V_i, and d_j be the $\mathcal{G}_{ij}$-degree of vertices in V_j. As the edges of $\mathcal{G}_{ij}$ are only between V_i and V_j, we conclude that the number of edges in $\mathcal{G}_{ij}$ is equal to $r_i d_i = r_j d_j$. Thus, the number r_j divides $r_i d_i$, and the number $\frac{r_j}{\gcd(r_i, r_j)}$ divides $\frac{r_i}{\gcd(r_i, r_j)} d_i$. However, the numbers $\frac{r_j}{\gcd(r_i, r_j)}, \frac{r_i}{\gcd(r_i, r_j)}$ are relatively prime, meaning that $\frac{r_j}{\gcd(r_i, r_j)}$ must divide d_i. In particular, $d_i \geq \frac{r_j}{\gcd(r_i, r_j)}$, and $\mathcal{G}_{ij}$ has at least $r_i d_i \geq \frac{r_i r_j}{\gcd(r_i, r_j)}$ edges.

[2] Recall that a directed graph is weakly connected if its unoriented counterpart is connected.

[3] In other words, only edges between V_i and V_j exist in the subgraph.

© Technion - Israel Institute of Technology, Elyachar Central Library

Now, we consider the condensed graph $\mathcal{G}'$. The vertices of $\mathcal{G}'$ are $\{v_1, \cdots, v_k\}$, and v_i is connected to v_j if and only if there is an edge between V_i and V_j. Obviously, since $\mathcal{G}$ is weakly connected, $\mathcal{G}'$ is also connected. Let $\mathcal{T}$ be a spanning tree for $\mathcal{G}'$. For each edge $e = (i, j)$ in $\mathcal{T}$, there is an edge between V_i and V_j, meaning that the graph $\mathcal{G}_{ij}$ contains at least $\frac{r_i r_j}{\gcd(r_i, r_j)}$ edges. Thus the graph $\mathcal{G}$ has at least $\sum_{e\in\mathcal{T}, e=\{i,j\}} \frac{r_i r_j}{\gcd(r_i, r_j)}$ edges, meaning that it has at least m edges.

We now move to the second part of the theorem. Suppose that $\mathcal{T}$ is a path on k vertices, and let $i_1, \cdots, i_k$ be the order of the vertices in the path. Moreover, let $e = \{i_j, i_{j+1}\}$ be an edge in $\mathcal{T}$. By reordering $r_1, \cdots, r_k$, we assume that $i_j = j$ for $j = 1, 2, \cdots, k$. We let i be the vertex at which r_i is minimized. For each $j \in \{1, \cdots, k\}$, we number the vertices in V_j as $v_1^j, \cdots, v_{r_j}^j$. Build the graph $\mathcal{G}$ in the following manner:

i) For every j, we consider the following edges between V_j and V_{j+1}. For $p = 1, \cdots, \frac{r_j r_{j+1}}{\gcd(r_j, r_{j+1})}$, add an edge from $v_{p \mod r_j}^j$ to $v_{p \mod r_{j+1}}^{j+1}$.

iii) For each $p = 1, \cdots, r_i$, connect v_p^i to $v_{(p+1) \mod r_i}^i$.

It's easy to see that the number of edges in $\mathcal{G}$ is

$$\left(\sum_{e\in\mathcal{T}, e=\{i,j\}} \frac{r_i r_j}{\gcd(r_i, r_j)} \right) + r_i.$$

We want to show that the orbits of the action of $\mathrm{Aut}(\mathcal{G})$ on $\mathcal{G}$ are exactly $V_1, \cdots, V_k$, and that G is weakly connected, concluding the proof.

First, we show that $\mathcal{G}$ is weakly connected. Indeed, we first note that the induced subgraph on V_i is weakly connected, as the following cycle eventually passes through all the nodes in the graph:

$$v_1^i \to v_2^i \to v_3^i \to \cdots$$

Now, by the way we built $\mathcal{G}$, any vertex v_p^i is connected all vertices $v_{p \mod r_l}^l$. Thus, for any two vertices $v_{p_1}^{j_1}$ and $v_{p_2}^{j_2}$, we can consider a path that starts from $v_{p_1}^{j_1}$, goes to $v_{p_1 \mod r_i}^i$, moves to $v_{p_2 \mod r_i}^i$ using the connectivity of induced subgraph on V_i, and continues to $v_{p_2}^{j_2}$. Thus $\mathcal{G}$ is weakly connected.

As for the orbits, we first consider the map ψ defined on the vertices of $\mathcal{G}$ by sending each vertex v_p^j to $v_{(p+1) \mod r_j}^j$. By definition of the graph $\mathcal{G}$, it's clear that ψ is a graph automorphism. Moreover, it's clear that repeatedly applying ψ can move v_p^j to any vertex in V_j. Thus the orbit of v_p^j contains the set V_j.

Conversely, we show that each V_i is invariant under $\mathrm{Aut}(\mathcal{G})$, proving that the orbits of the action of $\mathrm{Aut}(\mathcal{G})$ on $\mathcal{G}$ are exactly $V_1, \cdots V_k$. This is obvious if $k = 1$, as then $V_1 = \mathbb{V}$, so we assume that $k \geq 2$. In that case, either $i \neq 1$ or $i \neq k$ (or both). We assume without loss of generality that $i \neq k$, as the complementary case can be treated similarly. Graph automorphisms preserve all graph properties, and in particular, they preserve the out-degree of vertices. As all edges are oriented from V_j to V_{j+1} or from V_i to itself where $i \neq k$, the

© Technion - Israel Institute of Technology, Elyachar Central Library

vertices in V_k have an out-degree of 0, and they are the only ones with this property. Thus V_k must be invariant under $\mathrm{Aut}(\mathcal{G})$. The only vertices with edges to V_k are in V_{k-1}, meaning that V_{k-1} is also invariant under $\mathrm{Aut}(\mathcal{G})$. Iterating this argument, we conclude that $V_1, \cdots V_k$ must all be invariant under the action of $\mathrm{Aut}(\mathcal{G})$. Thus they are the orbits of the action, and the action of $\mathrm{Aut}(\mathcal{G})$ on $\mathcal{G}$ has orbits of size $r_1, \cdots, r_k$. This completes the proof. □

Remark 4.6. *One might ask why the lower bound considers all possible trees, while the upper bound only considers path graphs. The main reason for this distinction can be seen in the proof - one can build a general graph $\mathcal{G}$ where one uses a tree graph as a basis, instead of the path graph $1 \to 2 \to 3 \to \cdots \to k$. In that case, proving that each vertex v_p^i can reach all of V_i using graph automorphisms is easy, but it might be possible that its orbit is actually larger.*

Remark 4.7. *Note that the lower bound in the theorem can be found using Kruskal's Algorithm, or any other algorithm finding a minimal spanning tree in a graph [36], meaning it can be computed in polynomial time. However, the upper bound requires one to solve a specific case of a variant of the traveling salesman problem, which is known to be NP-hard [36].*

Theorem 4.8 deals with a general cluster structure, and some of the graphs it constructs can be seen in Example 4.6. We apply it to more specific cases in order to achieve concrete bounds on the number of edges needed for clustering in these cases.

Corollary 4.6. *Suppose that all cluster sizes $r_1, \cdots, r_k$ are equal, and bigger than 1. Then the minimal weakly connected directed graph $\mathcal{G}$ such that the action of $\mathrm{Aut}(\mathcal{G})$ on $\mathcal{G}$ forces clusters of size $r_1, \cdots, r_k$ has exactly n edges.*

Proof. Let r be the size of all clusters. We note that in this case, $\frac{r_i r_j}{\gcd(r_i, r_j)} = r$, and the number of clusters is $k = n/r$. Thus, if we build a graph $\mathcal{G}$ as in the proof of Theorem 4.8, then $\mathcal{G}$ has exactly n edges, as the summation over edges $e \in \mathcal{T}$ goes over $k-1$ edges. It remains to show that no such graph on less than n edges exists.

First, we know that any graph on n nodes and less than $n-1$ edges is not connected [15]. Thus, it suffices to show that no such graph on $n-1$ vertices exists. Indeed, let d_i be the out-degree of some arbitrary vertex in the i-th orbit, $i = 1, \cdots, k$. As the action of the automorphism group preserves the out-degree, it is the same for all vertices in the i-th orbit. The number of edges is the sum of the out-degree over all vertices, but it's equal to $rd_1 + \cdots + rd_k$. Thus the number of edges in $\mathcal{G}$ must be divisible in r. But $n = kr$, meaning that $n-1$ is not divisible by r unless $r = 1$. Thus there is no such graph on $n-1$ edges. □

Corollary 4.7. *Let $r_1, \cdots, r_k$ be positive integers such that $k \geq 2$ and that for every i, j, either r_i divides r_j or vice versa, and let $n = r_1 + \cdots + r_k$. Then the minimal weakly connected directed graph $\mathcal{G}$ such that the action $\mathrm{Aut}(\mathcal{G})$ on $\mathcal{G}$ forces clusters of sizes $r_1, \cdots, r_k$ has no more than n edges.*

© Technion - Israel Institute of Technology, Elyachar Central Library

Proof. We reorder the numbers $r_1, \cdots, r_k$ such that r_l divides r_j for $l \leq j$. We note that if r_l divides r_j, then $\frac{r_l r_j}{\gcd(r_l, r_j)} = \max(r_l, r_j) = r_l$. Thus, if we let $\mathcal{G}$ be the graph built in the proof of Theorem 4.8, it has the following number of edges:

$$\sum_{j=1}^{k-1} \frac{r_j r_{j+1}}{\gcd(r_j, r_{j+1})} + r_1 = \sum_{j=1}^{k-1} r_{j+1} + r_1 = \sum_{j=1} r_j = n.$$

□

Corollary 4.8. *Let $r_1, \cdots, r_k$ be positive integers such that $k \geq 2$ and $r_i \leq q$ for all i, , and let $n = r_1 + \cdots + r_k$. Then the minimal weakly connected directed graph $\mathcal{G}$ such that the action of $\mathrm{Aut}(\mathcal{G})$ on $\mathcal{G}$ forces clusters of sizes $r_1, \cdots, r_k$ has no more than $n + O(q^3)$ edges.*

Proof. Without loss of generality, we assume that the numbers r_i are ordered such that $r_1 \leq r_2 \leq \cdots r_k$. Let m_i be the number of clusters of size i for $i = 1, 2, \cdots, q$. As before, consider the graph $\mathcal{G}$ built in the proof of Theorem 4.8. We note that if $r_l = r_j$ then $\frac{r_l r_j}{\gcd(r_l, r_j)} = r_l$, and $\frac{r_l r_j}{\gcd(r_l, r_j)} \leq r_l r_j$ otherwise. Thus, the number of edges in $\mathcal{G}$ is given by:

$$\sum_{j=1}^{k-1} \frac{r_j r_{j+1}}{\gcd(r_j, r_{j+1})} + r_1 \leq \sum_{l \in \{1, \cdot, q\}, m_l \neq 0} (m_l - 1)l + \sum_{l=1}^{q-1} l(l-1) + r_1.$$

Indeed, for each $l \in \{1, \cdots, q\}$, if there's at least one cluster of size l, then there are $m_l - 1$ edges in the path $\mathcal{T} = 1 \to 2 \to \cdots \to k$ that touch two clusters of size l. The second term bounds the number of edges that appear between clusters of different sizes. We note that $n = \sum_{l=1}^{q} l m_l$, so the first term is bounded by n. As for the second term, we can bound $l(l-1)$ by l^2 and then use the formula $\sum_{l=1}^{q-1} l^2 = \frac{(q-1)q(2q-1)}{6}$. The formula first appears in Fibonacci's *Liber Abaci* from 1201, and can also be found in [143]. Lastly, the last term r_1 is bounded by q. This completes the proof. □

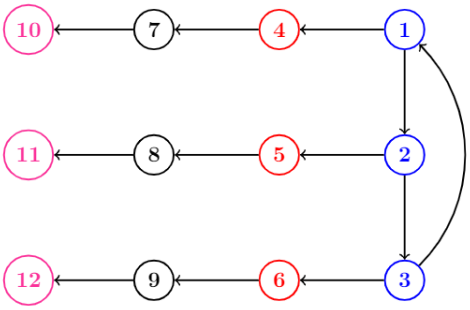
We now give examples of graphs constructed by Theorem 4.8, and show they indeed force a clustering structure on the agents.

Example 4.6. *We consider a collection of $n = 12$ identical agents, all of the form $\dot{x} = -x + u + \alpha, y = x$ where α is a log-uniform random variable between 0.1 and 1, identical for all agents. In all experiments described below, we considered identical controllers, equal to the static nonlinearity of the form*

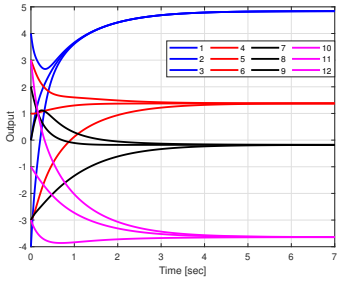
$$\mu = a_1 + a_2(\zeta + \sin(\zeta))$$

where a_1 was chosen as a Gaussian random variable with mean 0 and standard deviation 10, and a_2 was chosen as a log-uniform random variable between 0.1 and 10. We note that the agents are indeed output-strictly MEIP and the controllers are MEIP, so the conditions of Theorem 4.5 hold. Moreover, the network is homogeneous, so $\mathrm{Aut}(\mathcal{G}, \Sigma, \Pi) = \mathrm{Aut}(\mathcal{G})$. Thus, we can use the graphs constructed by Theorem 4.8 to force a clustering behavior.

© Technion - Israel Institute of Technology, Elyachar Central Library

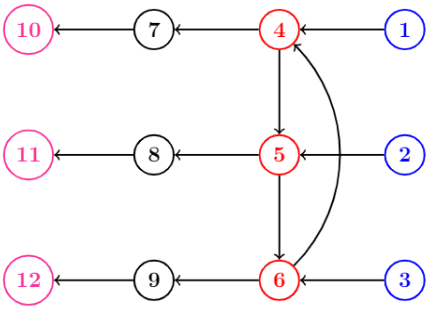


(a) Graph forcing cluster sizes $r_1 = r_2 = r_3 = r_4 = 3$. Nodes with the same color will be in the same cluster.

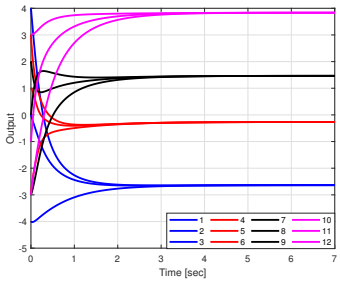


(b) Agent's trajectories for the closed-loop system. Colors correspond to node colors in the graph.

Figure 4.9: First example of graphs solving the cluster synthesis problem, as constructed in Theorem 4.8.



(a) Another graph forcing cluster sizes $r_1 = r_2 = r_3 = r_4 = 3$. Nodes with the same color will be in the same cluster.

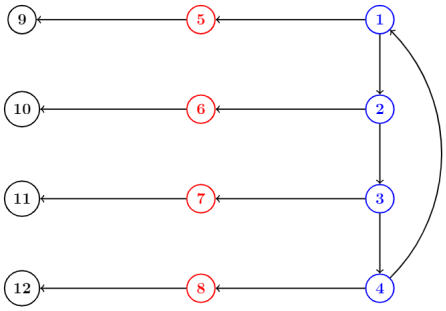


(b) Agent's trajectories for the closed-loop system. Colors correspond to node colors in the graph.

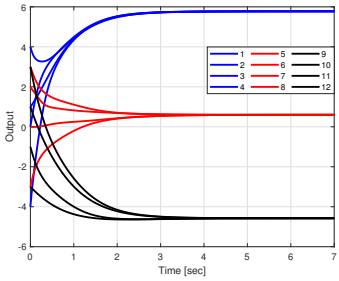
Figure 4.10: Second example of graphs solving the cluster synthesis problem, as constructed in Theorem 4.8.

We first consider the cluster synthesis problem with four equally-sized clusters, i.e., $r_1 = r_2 = r_3 = r_4 = 3$. One possible graph forcing these clusters, as constructed by Theorem 4.8, can be seen in Figure 4.9, along with the agent's trajectories for the corresponding closed-loop system. Another example of such graph can be seen in Figure 4.10, again with the agent's trajectories for the corresponding closed-loop system.

Secondly, we consider the cluster synthesis problem with three equally-sized clusters, i.e. $r_1 = r_2 = r_3 = 4$. One possible graph forcing these clusters, as constructed by Theorem 4.8, can be seen in Figure 4.11, along with the agent's

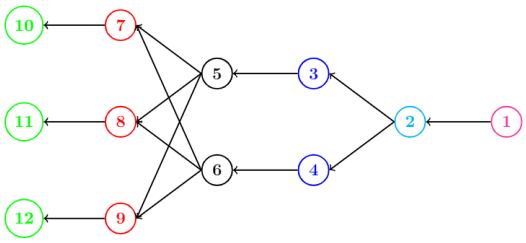


(a) Graph forcing cluster sizes $r_1 = r_2 = r_3 = 4$. Nodes with the same color will be in the same cluster.

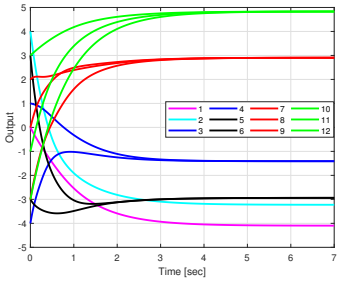


(b) Agent's trajectories for the closed-loop system. Colors correspond to node colors in the graph.

Figure 4.11: Third example of graphs solving the cluster synthesis problem, as constructed in Theorem 4.8.



(a) Graph forcing cluster sizes $r_1 = r_2 = 1$, $r_3 = r_4 = 2$, $r_5 = r_6 = 3$. Nodes with the same color will be in the same cluster.



(b) Agent's trajectories for the closed-loop system. Colors correspond to node colors in the graph.

Figure 4.12: Fourth example of graphs solving the cluster synthesis problem, as constructed in Theorem 4.8.

trajectories for the corresponding closed-loop system.

Lastly, we consider the cluster synthesis problem with six clusters of different sizes, defined by $r_1 = r_2 = 1$, $r_3 = r_4 = 2$, $r_5 = r_6 = 3$. *One possible graph forcing these clusters, as constructed by Theorem 4.8, can be seen in Figure 4.12, along with the agent's trajectories for the corresponding closed-loop system.*

4.4 Conclusions

In this chapter, we explored the problems of final-value synthesis and cluster synthesis. For the former, we presented a synthesis scheme for solving the

© Technion - Israel Institute of Technology, Elyachar Central Library

final-value synthesis problem, for which the closed-loop system globally asymptotically converges to the desired value. This was done using tools from convex optimization and strict convexity. Later we focused on clustering, which is a specific case of the final-value synthesis problem, in which we showed that certain symmetries in the network correspond to the formation of certain clusters in the steady-state output. We then presented a method to build these symmetric networks for the case of statically equivalent agents. Several examples were presented throughout the chapter demonstrating the problems and solution schemes.

In the next chapter, we'll solve a similar final-value synthesis problem, but using data-driven methods, with no knowledge on the specific models, or the steady-state input-output relations that the agents possess.

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

Chapter 5

Applications of the Network Optimization Framework in Data-Driven Control

This section is a review of [135]. We study the problem of practical final-value synthesis, in which one is given agents $\{\Sigma_i\}$, an interaction graph $\mathcal{G}$, a vector of desired relative positions $\zeta^\star \in \mathrm{Im}(\mathcal{E}^\top)$ and a tolerance level $\epsilon > 0$. Our goal is to find networked controllers $\{\Pi_e\}$ such that the diffusively-coupled system $(\mathcal{G}, \Sigma, \Pi)$ converges to a steady-state for which the relative positions vector is ϵ-close to $\zeta^\star$. This is an easier problem than the synthesis problem described in the previous section. However, we focus on the case in which there is no known model from the agents, but they are given as a blackbox, with no or a very limited model. We show how one can solve the problem in a data-driven manner, with either very few preliminary experiments, or an on-line iterative scheme.

5.1 Introduction

As we already saw, multi-agent systems have received extensive attention in the past years, due to their appearance in many fields of engineering, exact sciences and social sciences. The state-of-the-art approach to model-based control for multi-agent systems offers rigorous stability analysis, performance guarantees and systematic insights into the considered problem. However, with the growing complexity of systems, the modeling process is reaching its limits. Obtaining a reliable mathematical model of the agents can become a long and hard endeavor.

At the same time, modern technology allows for gathering and storing more and more data from systems and processes. Therefore, there has been an increasing interest in what is called *data-driven controller design*. There have been different approaches to data-driven controller design very generally (see,

© Technion - Israel Institute of Technology, Elyachar Central Library

e.g., [67,152]), and some approaches to multi-agent control from data more particularly [13, 21, 73]. Nonetheless, the centralized nature of the general design schemes, or the specificity of the agents' model and high measurement rate for the multi-agent approaches, prevents the application of many of the aforementioned approaches in various real-world scenarios.

In this chapter, we develop a data-driven controller approach for multi-agent systems that comes with rigorous theoretical analysis and stability guarantees for the closed loop, with almost no assumptions on the agents and very few measurements needed. The approach is based on the notion of high-gain controllers. Some ideas on high-gain approaches for cooperative control can be found in [164] and references therein. In [178], the authors provide a high-gain condition in the design of distributed $\mathcal{H}_\infty$ controllers for platoons with undirected topologies, while there are also many approaches for (adaptively) tuning the coupling weights, e.g. [171]. Our approach provides an upper bound on a high-gain controller on the basis of passivity measures. Passivity properties of the components can provide sufficient abstractions of their detailed dynamical models for guaranteed control. Such passivity properties, in turn, can be obtained from data as ongoing work shows (e.g., [95,123,125,153]).

This chapter generally studies the problem of controller synthesis for diffusively coupled systems. The control objective is to converge to an ϵ-neighborhood of a constant prescribed relative output vector. That is, for some tolerance $\epsilon > 0$, we aim to design controllers so that the steady-state relative output limit is ϵ-close to the prescribed values. The related problem of practical synchronization of multi-agent systems have been considered in [96], in which the plants were assumed to be known up to some bounded additive disturbance. However, a nominal model of the plant was needed to achieve the practical synchronization. It was also pursued in [76], in which strong coupling was used to drive agents to a neighborhood of a common trajectory, but again, a model for the agents was needed.

5.2 Problem Formulation and Verification of Passivity

We focus on the final-value synthesis problem for the relative-output vector of *SISO* agents. In this problem, the agents know the relative output $\zeta_e = y_i - y_j$ with respect to their neighbors, and the control goal is to converge to a steady-state with prescribed relative outputs $\zeta^\star$. Examples include the consensus problem, in which all outputs must agree, as well as relative-position based formation control of robots, in which the robots are required to organize themselves in a desired spatial structure [105].

More specifically, we are given a graph $\mathcal{G}$ and SISO agents Σ, and our goal is to design controllers Π so that the formation vector $\zeta(t)$ of the diffusively coupled network $(\mathcal{G}, \Sigma, \Pi)$ will converge to a desired, given steady-state vector $\zeta^\star$. One evident solution to the problem is to apply a (shifted) integrator as a

© Technion - Israel Institute of Technology, Elyachar Central Library

controller. However, this solution will not always work even when the agents are MEIP.

Example 5.1. *Consider the case of agents Σ_i with integrator dynamics, together with the controllers Π_e according to the previous idea, where we desire consensus (i.e., $\zeta^\star = \mathbf{0}$) over a connected graph $\mathcal{G}$,*

$$\Sigma_i : \begin{cases} \dot{x}_i = u_i \\ y_i = x_i \end{cases}, \qquad \Pi_e : \begin{cases} \dot{\eta}_e = \zeta_e \\ \mu_e = \eta_e \end{cases}.$$

The trajectories of the diffusively-coupled system can be understood by noting that the closed-loop system yields the second-order dynamics $\ddot{x} = -\mathcal{E}\mathcal{E}^\top x$, where $\mathcal{E}$ is the incidence matrix of the graph. Decomposing x using a basis of eigenvectors of the graph Laplacian $\mathcal{E}\mathcal{E}^\top$, which is a positive semi-definite matrix, we see that the trajectory of $x(t)$ oscillates around the consensus manifold $\{\mathrm{x} : \exists\lambda \in \mathbb{R},\ \mathrm{x} = \lambda\mathbb{1}_n\}$. Specifically, $x(t) - \frac{1}{n}\mathbb{1}_n^\top x(t) = \sum_{i=2}^n c_i \cos(\sqrt{\lambda_i}t + \varphi_i)v_i + +c_0 t\mathbb{1}$, where $\lambda_2, \ldots, \lambda_n > 0$ are the non-trivial eigenvalues of the graph Laplacian, $v_2, \ldots, v_n$ are corresponding unit-length eigenvectors, and c_i, φ_i are constants depending on the initial conditions $x(0), \eta(0)$. Thus $x(t) = y(t)$ does not converge anywhere, let alone to consensus. Moreover, the vector $\zeta(t) = \mathcal{E}^\top y(t) = \sum_{i=2}^n \lambda_i c_i \cos(\sqrt{\lambda_i}t + \varphi_i)v_i$ does not converge as $t \to \infty$. Thus the integrator controller does not solve the final-value synthesis problem for the relative output vector in this case.

Even if the integrator would solve this problem in general, we would like more freedom in choosing the controller. In practice, one might want to design the controller to satisfy extra requirements (like $\mathcal{H}_2$- or $\mathcal{H}_\infty$-norm minimization, or making sure that certain restrictions on the behavior of the system are not broken). We do not try and satisfy these more complex requirements, but instead show that a large class of controllers can be used to solve the practical final-value synthesis problem. In turn, this allows one to choose from a wide range of controllers, and try and satisfy additional desired properties. In the previous section, we saw an algorithm solving the relative position-based formation control problem with ease, as long as the agents are MEIP and a perfect model of each agent is known. This algorithm allows a vast amount of freedom in the choice of controllers. However, in practice we oftentimes have no exact model of our agents, or there is even no closed form mathematical model available at all.

To formalize the goals we aim at, we define the notion of *practical* final-value synthesis.

Problem 5.1. *Given a graph $\mathcal{G}$, agents Σ, a desired formation $\zeta^\star \in \mathrm{Im}(\mathcal{E}^\top)$, and an error margin ε, find a controller Π so that the relative output vector $\zeta(t)$ of the network $(\mathcal{G}, \Sigma, \Pi)$ converges to some ζ_0 such that $\|\zeta^\star - \zeta_0\| \leq \varepsilon$.*

By choosing suitable error margins ε, practical final-value synthesis (compared to final-value synthesis) comprises no restriction or real drawback in any

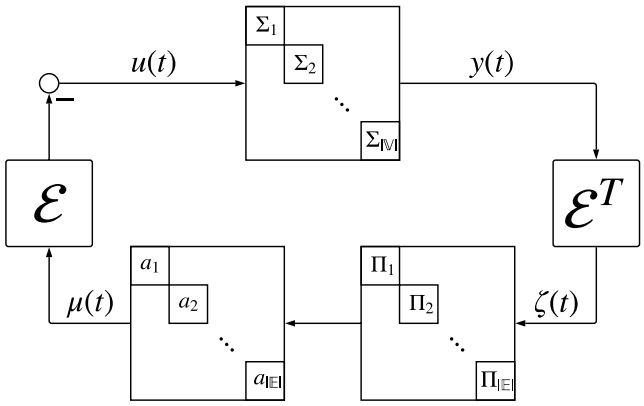


Figure 5.1: Block diagram of the diffusively-coupled network $(\mathcal{G}, \Sigma, \Pi, A)$.

application case. Therefore, solving the practical final-value synthesis problem constitutes an interesting problem especially for unknown dynamics of the agents. Thus, we strive to develop an algorithm solving this practical final-value synthesis problem without a model of the agents while still providing rigorous guarantees.

The underlying idea of our approach is amplifying the controller output. Consider the scenario depicted in Figure 5.1, where the graph $\mathcal{G}$, the agents Σ and the nominal controller Π are fixed, and the gain matrix A is a diagonal matrix $A = \mathrm{diag}(\{a_e\}_{e\in\mathbb{E}})$ with positive entries. We will show in the following that when the gains a_e become large enough, then the controller dynamics Π become much more emphasized than the agent dynamics Σ. By correctly choosing the nominal controller Π according to $\zeta^\star$, we can hence achieve arbitrarily close formations to $\zeta^\star$, as the effect of the agents on the closed-loop dynamics will be dampened. We denote the diffusively-coupled system in Figure 5.1 as the 4-tuple $(\mathcal{G}, \Sigma, \Pi, A)$, or as $(\mathcal{G}, \Sigma, \Pi, a)$ where a is the vector of diagonal entries of A. In case A has uniform gains, i.e., $A = \alpha I$, we'll denote the system as $(\mathcal{G}, \Sigma, \Pi, \alpha\mathbb{1}_n)$. In order to apply the network optimization framework of Theorems 1.2 and 2.3, we make the following assumption:

Assumption 5.1. *The agents $\{\Sigma_i\}_{i\in\mathbb{V}}$ are all MEIP, and the nominal controllers $\{\Pi_e\}_{e\in\mathbb{E}}$ are all output-strictly MEIP.*

Before expanding on the suggested controller design, we want to discuss Assumption 5.1. In practice, it might not be known whether an agent is MEIP. Therefore, we discuss how to either verify MEIP for the agents, or determine their shortage of passivity if they are not MEIP. We also discuss how to passivize the agents in the latter case.

First, in some occasions, we actually do have some model for the agents, which might be obscure or uncertain. For example, one might know that an

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

agent can be modeled by some gradient system, or some Euler-Lagrange system, but the exact model is unknown due to uncertainty on the characterizing parameters. In that case, we can use analytical results to check if the agents are MEIP. To exemplify this idea, we show how a very rough model can be used to prove that a system is MEIP.

Proposition 5.1. *Consider a control-affine SISO dynamical system of the form*

$$\dot{x} = -f(x) + g(x)u;\ y = h(x), \tag{5.1}$$

where we assume that $g(x) > 0$ *for all* x*, and that* $f/g : \mathbb{R} \to \mathbb{R}$ *is a continuous monotone ascending function, and that* h *is continuous strictly monotone ascending. Moreover, assume that either* $\lim_{|x|\to\infty} |f(x)/g(x)| = \infty$ *or* $\lim_{|x|\to\infty} |h(x)| = \infty$*. Then* (5.1) *is MEIP.*

The proof of the proposition will be delayed to the end of this subsection. See also Section 2.3 for a treatment on gradient systems with oscillatory terms. More generally, one can use an obscure model to give an estimate about equilibrium-independent passivity indices using similar ideas.

Another approach for verifying Assumption 5.1 is learning input-output passivity properties from trajectories. For LTI systems, the shortage of passivity can be asymptotically revealed by iteratively probing the agents and measuring the output signal as presented in [125] and extended, e.g., in [153]. More recently, it has been shown in [123] that even one input-output trajectory (with persistently exciting input) is sufficient to find the shortage of passivity of an LTI system. For nonlinear agents, one can apply approaches presented in [95, 124], under an assumption on Lipschitz continuity of the steady-state relation. However, for general non-linear systems, this is still a work in progress. It should be noted that for LTI systems, output-strict passivity directly implies output strict MEIP [23].

Using either approaches, we can either find that an agent is MEIP, or that it has some shortage of passivity, and we need to render the agent passive in order to apply the model-free formation control approaches presented in this chapter. We can use passivizing transformations on the non-passive agent in order to get a passive augmented agent. See Chapter 3 for more details.

With this discussion and relaxation of Assumption 5.1, we return to our solution of the practical final-value synthesis problem. Recall that we considered closed-loop systems of the form $(\mathcal{G}, \Sigma, \Pi, a)$, where a is a vector of edge gains. From here, the chapter diverges into two sections. The next section deals with theory and analysis for uniform edge gains. The following section deals with theory and analysis for the case of heterogeneous edge gains. However, before moving on, we repay our debt and prove Proposition 5.1. For the proof, we recall the notion of cursive relations, and specifically Lemma 3.1.

Proof. Consider an arbitrary steady-state of the system. As h is continuous and strictly monotone ascending, hence invertible, we must have $\dot{x} = \mathbf{0}$ for any steady-state input-output pair. Thus, we conclude that any steady-state

input-output pair can be written as $(f(\sigma)/g(\sigma), h(\sigma))$ for some $\sigma \in \mathbb{R}$. We first show passivity with respect to every steady-state, and then show that the steady-state input-output relation is maximally monotone. Take a steady-state $(f(\mathrm{x}_0)/g(\mathrm{x}_0), h(\mathrm{x}_0))$ of the system, and define $S(x) = \int_{\mathrm{x}_0}^{x} \frac{h(\sigma)-h(\mathrm{x}_0)}{g(\sigma)} d\sigma$. We claim that S is a storage function for the steady-state input-output pair $(f(\mathrm{x}_0)/g(\mathrm{x}_0), h(\mathrm{x}_0))$. Indeed, $S(x) \geq 0$, with equality only at x_0, immediately follows from strict monotonicity of h and the positivity of g. As for the inequality defining passivity, we have:

$$\begin{aligned}\frac{d}{dt}S(x) &= \frac{h(x)-h(\mathrm{x}_0)}{g(x)}(-f(x)+g(x)u) \\ &= (h(x)-h(\mathrm{x}_0))u - \frac{f(x)}{g(x)}(h(x)-h(\mathrm{x}_0)) \\ &= (h(x)-h(\mathrm{x}_0))\left(u - \frac{f(\mathrm{x}_0)}{g(\mathrm{x}_0)}\right) - \left(\frac{f(x)}{g(x)} - \frac{f(\mathrm{x}_0)}{g(\mathrm{x}_0)}\right)(h(x)-h(\mathrm{x}_0)),\end{aligned}$$

where the second term is non-positive as $\frac{f}{g}, h$ are monotone ascending, and the first term is $(y-h(\mathrm{x}_0))(u-\frac{f(\mathrm{x}_0)}{g(\mathrm{x}_0)})$. This proves that the system is indeed passive with respect to any steady-state input-output pair. As for maximal monotonicity of the steady-state relation, we recall that it can be parameterized as $(f(\sigma)/g(\sigma), h(\sigma))$ for $\sigma \in \mathbb{R}$. We claim that this relation is both monotone and cursive, which will show that the relation is maximal monotone. Monotonicity follows from the system being passive with respect to any steady-state. As for cursiveness, the map $\sigma \mapsto (f(\sigma)/g(\sigma), h(\sigma))$ is a curve whose image is the relation. Moreover, it's clear that the map is continuous, and also injective due to strict monotonicity of h. Lastly, we have

$$\lim_{|t|\to\infty} \left\| \left(\frac{f(t)}{g(t)}, h(t)\right) \right\| \geq \lim_{t\to\infty} \max\left\{ \left| \frac{f(t)}{g(t)} \right|, |h(t)| \right\} = \infty, \tag{5.2}$$

proving that the steady-state relation is also cursive, and completing the proof. □

Remark 5.1. *Following the proof of the proposition, we note that the passivity index with respect to the steady-state input-output pair* $(f(\mathrm{x}_0)/g(\mathrm{x}_0), h(\mathrm{x}_0))$ *is*

$$\rho = \inf_{x\in\mathbb{R}} \frac{\frac{f(x)}{g(x)} - \frac{f(\mathrm{x}_0)}{g(\mathrm{x}_0)}}{h(x)-h(\mathrm{x}_0)}.$$

This will help us apply some of the presented methods to control-affine systems for cases in which the actual value of the passivity index is needed, e.g. Chapter 7.

5.3 Uniform Gain Amplification Methods

We split this chapter into two parts. The first part deals with the theory, and the second part deals with the corresponding implementation of practical final-value

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

synthesis using uniform gains on the edges.

5.3.1 Theory

We wish to understand the effect of amplification on the steady-state of the closed-loop system. For the remainder of the section, we fix a graph $\mathcal{G}$, SISO agents Σ and controllers Π such that Assumption 5.1 holds. We consider the diffusively coupled system $(\mathcal{G}, \Sigma, \Pi, \alpha \mathbb{1}_n)$ in Figure 5.1, where the gains over all edges are identical and equal to $\alpha > 0$, and wish to understand the affect of α. We let K and Γ denote the sum of the integral functions of the agents and of the controllers, respectively. We first study the steady-states of this diffusively coupled system.

Lemma 5.1. *Under the assumptions above, the closed-loop system converges to a steady-state, and the steady-state vectors* y, ζ *of the closed-loop system are minimizers of the following optimization problem (OPP):*

$$\begin{aligned} \min_{\mathrm{y},\zeta} \quad & K^\star(\mathrm{y}) + \alpha\Gamma(\zeta) \\ s.t. \quad & \mathcal{E}^\top \mathrm{y} = \zeta. \end{aligned}$$

Proof. We define a new stacked controller, $\bar{\Pi} = \alpha\Pi$, by cascading the previous controller Π with the gain α. The resulting controller $\bar{\Pi}$ is again output-strictly MEIP, and we let $\bar{\gamma}, \bar{\Gamma}$ denote the corresponding steady-state input-output relation and integral function. Theorem 1.2 implies that the closed-loop system (with $\bar{\Pi}$) converges to minimizers of (OPP) for the system $(\mathcal{G}, \Sigma, \bar{\Pi})$. Moreover, we have $\bar{\gamma}(\zeta) = \alpha\gamma(\zeta)$ for any $\zeta \in \mathbb{R}^{|\mathbb{E}|}$. Integration thus yields $\bar{\Gamma} = \alpha\Gamma$, and writing (OPP) for the system $(\mathcal{G}, \Sigma, \bar{\Pi})$ reads:

$$\begin{aligned} \min_{\mathrm{y},\zeta} \quad & K^\star(\mathrm{y}) + \alpha\Gamma(\zeta) \\ s.t. \quad & \mathcal{E}^\top \mathrm{y} = \zeta. \end{aligned}$$

□

Our goal is to show that when $\alpha \gg 1$, the relative output vector ζ of the diffusively coupled system $(\mathcal{G}, \Sigma, \Pi, \alpha \mathbb{1}_n)$ globally asymptotically converges to an $\varepsilon = \varepsilon(\alpha)$-ball around the minimizer of Γ, and $\lim_{\alpha\to\infty} \varepsilon(\alpha) = 0$. Thus, if we design the controllers so that Γ is minimized at $\zeta^\star$, then $\alpha \gg 1$ provides a solution to the ε-practical final-value synthesis problem. Indeed, we can prove the following theorem:

Theorem 5.1. *Consider the closed-loop system* $(\mathcal{G}, \Sigma, \Pi, \alpha \mathbb{1}_n)$*, where we assume that the agents are MEIP and the controllers are output-strictly MEIP. Assume* Γ *has a unique minimizer in* $\mathrm{Im}(\mathcal{E}^\top)$*, denoted* ζ_1*. For any* $\varepsilon > 0$*, there exists some* $\alpha_0 > 0$*, such that for all* $\alpha > \alpha_0$ *and for all initial conditions, the closed-loop system converges to a vector* y *satisfying* $\|\mathcal{E}^\top \mathrm{y} - \zeta_1\| < \varepsilon$*.*

© Technion - Israel Institute of Technology, Elyachar Central Library

In order to prove the theorem, we study (OPP) for the diffusively coupled system $(\mathcal{G}, \Sigma, \Pi, \alpha \mathbb{1}_n)$, as described in Lemma 5.1. In order to do so, we need to prove a couple of lemmas. The first deals with lower bounds on the values of convex functions away from their minimizers.

Lemma 5.2. *Let U be a finite-dimensional vector space and let $f : U \to \mathbb{R}$ be a strictly convex function. Denote $x_0 \in U$ as the unique minimum of f. Then for any $\delta > 0$ there exists some $M > f(x_0)$ such that for any point $x \in U$, if $f(x) < M$ then $\|x - x_0\| < \delta$.*

Proof. We assume without loss of generality that $f(x_0) = 0$. Let ν be the minimum of f on the set $\{x \in U : \ ||x - x_0|| = \delta\}$, which is positive since x_0 is f's unique minimum and the set $\{x \in U : \|x - x_0\| = \delta\}$ is compact. We know that, for any $y \in U$, the difference quotient

$$\frac{f(x_0 + \lambda y) - f(x_0)}{\lambda},$$

is an increasing function of $\lambda > 0$ (see Appendix A). Manipulating this inequality implies that for any $x \in U$, if $||x|| \geq \delta$ then we have that $f(x) \geq \frac{||x||}{\delta}\nu$, and in particular $f(x) \geq \nu$ whenever $||x|| \geq \delta$. Thus, if $f(x) < \nu$ then we must have $||x - x_0|| < \delta$, so we can choose $M = \nu$ and complete the proof. □

The second lemma deals with minimizers of perturbed versions of convex functions on graphs.

Lemma 5.3. *Fix a graph $\mathcal{G} = (\mathbb{V}, \mathbb{E})$ and let $\mathcal{E}$ be its incidence matrix. Let $K : \mathbb{R}^{|\mathbb{V}|} \to \mathbb{R}$ be a convex function, and let $\Gamma : \mathbb{R}^{|\mathbb{E}|} \to \mathbb{R}$ be a strictly convex function having a unique minimum ζ_1 when restricted to the set $\mathrm{Im}(\mathcal{E}^\top)$. For any $\alpha > 0$, consider the function $F_\alpha(\mathrm{y}) = K^\star(\mathrm{y}) + \alpha\Gamma(\mathcal{E}^\top \mathrm{y})$. Then for any $\varepsilon > 0$, there exists some $\alpha_0 > 0$ such that if $\alpha > \alpha_0$ then all of F_α's minima, y, satisfy $\|\mathcal{E}^\top \mathrm{y} - \zeta_1\| < \varepsilon$.*

Proof. By subtracting constants from $K^\star$ and Γ, we may assume without loss of generality that $\min(K^\star) = \min(\Gamma) = 0$. Choose some $\mathrm{y}_0 \in \mathbb{R}^{|\mathbb{V}|}$ such that $\mathcal{E}^\top \mathrm{y}_0 = \zeta_1$ and let $m = K^\star(\mathrm{y}_0)$. Note that $F_\alpha(y_0) = m$, meaning that if y is any minimum of F_α, it must satisfy $F_\alpha(\mathrm{y}) \leq m$, and in particular $\Gamma(\mathcal{E}^\top \mathrm{y}) \leq \frac{m}{\alpha}$. Now, from Lemma 5.2 we know that there's some $M > 0$ such that if $\Gamma(\mathcal{E}^\top \mathrm{y}) < M$ then $\|\mathcal{E}^\top \mathrm{y} - \zeta_1\| < \varepsilon$. If we choose $\alpha_0 = \frac{m}{M}$, then whenever $\alpha > \alpha_0$ we have $\Gamma(\mathcal{E}^\top \mathrm{y}) < M$, implying $\|\mathcal{E}^\top \mathrm{y} - \zeta_1\| < \varepsilon$. This completes the proof of the lemma. □

We now connect the pieces and prove Theorem 5.1.

Proof. Lemma 5.1 implies that the closed-loop system always converges to a minimizer of (OPP)

$$\begin{aligned} \min_{\mathrm{y}, \zeta} \quad & K^\star(y) + \alpha\Gamma(\zeta) \\ s.t. \quad & \mathcal{E}^\top y = \zeta. \end{aligned}$$

© Technion - Israel Institute of Technology, Elyachar Central Library

Lemma 5.3 proves that there is some $\alpha_0 > 0$ such that if $\alpha > \alpha_0$ then all minimizers of (OPP) satisfy $\|\mathcal{E}^\top \mathrm{y} - \zeta_1\| < \varepsilon$. This proves the theorem. □

Remark 5.2. *The parameters ε and $\zeta^\star$ can be used to estimate the minimal gain α_0 solving the ε-practical final-value synthesis problem by following the proofs of Lemma 5.2 and Lemma 5.3. Namely, $\alpha_0 \leq \frac{m}{M}$ where M is the minimum of Γ on the set $\{\zeta \in \mathrm{Im}(\mathcal{E}^\top) : \ \|\zeta - \zeta^\star\| = \varepsilon\}$, and $m = K^\star(\mathrm{y}) - \min_y K^\star(\mathrm{y}) = K^\star(\mathrm{y}^\star) + K(\mathbf{0})$ where $\mathrm{y}^\star \in \mathbb{R}^{|\mathbb{V}|}$ is any vector satisfying $\mathcal{E}^\top \mathrm{y}^\star = \zeta^\star$.*

Corollary 5.1. *Let $(\mathcal{G}, \Sigma, \Pi, \alpha \mathbb{1}_n)$ satisfy Assumption 5.1 and let $(\mathcal{G}, \Sigma_{\mathrm{Int}}, \Pi)$ be a network with identical underlying graph and controllers, but single integrator dynamics for each agent. Denote the relative outputs of each system as $\zeta(t)$ and $\zeta_{\mathrm{Int}}(t)$ respectively. Then for any $\varepsilon > 0$, there exists an $\alpha_0 > 0$ such that if $\alpha \geq \alpha_0$, then the relative outputs $\zeta(t)$ and $\zeta_{\mathrm{Int}}(t)$ both converge to constant vectors $\zeta^\star$ and $\zeta^\star_{\mathrm{Int}}$ respectively, and satisfy $\|\zeta^\star - \zeta^\star_{\mathrm{Int}}\| \leq \varepsilon$.*

Proof. The agents Σ_{Int} are MEIP. Thus, by Theorem 1.2, we know that the diffusively-coupled system $(\mathcal{G}, \Sigma_{\mathrm{Int}}, \Pi)$ converges to a steady-state, and its steady-state output is a minimizer of the associated (OPP) problem. Note that the input-output relation of Σ_{Int}'s is given via $k^{-1}(\mathrm{y}) = \mathbf{0}$, meaning the integral function $K^\star$ is the zero function. Thus the associated problem (OPP) is the unconstrained minimization of $\Gamma(\mathcal{E}^\top \mathrm{y})$, meaning that the system $(\mathcal{G}, \Sigma_{\mathrm{Int}}, \Pi)$ converges, and its output converges to a minimizer of $\Gamma(\mathcal{E}^\top \mathrm{y})$, i.e., its relative output $\zeta(t)$ converges to the minimizer of Γ on $\mathrm{Im}(\mathcal{E}^\top)$. Applying Theorem 5.1 now completes the proof. □

Remark 5.3 (Almost Data-free control)**.** *Corollary 5.1 can be thought of as a synthesis procedure. Indeed, we can solve the synthesis problem as if the agents were simple integrators, and then amplify the controller output by a factor α. The corollary shows that for any $\varepsilon > 0$, there is a threshold $\alpha_0 > 0$ such that if $\alpha > \alpha_0$, then the closed-loop system converges to an ε-neighborhood of $\zeta^\star$. It is important to note that we only know that α_0 exists as long as the agents are MEIP. Computing an estimate on α_0, however, requires one to conduct a few experiments.*

There are a few possible approaches to try and eliminate this requirement. One can try an iterative scheme, in which the edge gains are updated between iterations. Gradient-descent and extremum-seeking approaches are discussed in the next section (see Algorithm 3), but both require to measure the system between iterations.

Another approach is to update the edge gains on a much slower time-scale than the dynamics of the system. This results in a two time-scale dynamical system, where the gains a_e of the system $(\mathcal{G}, \Sigma, \Pi, a)$ are updated slowly enough to allow the system to converge. Taking a_e as uniform gains of size α, and slowly increasing α, assures that eventually, $\alpha > \alpha_0$, so the system will converge ε-close to $\zeta^\star$. The only data we do need is whether or not the system has already converged to an ε-neighborhood of $\zeta^\star$, to know whether α should be updated or not. This requires no data on the trajectories themselves, nor information on

© Technion - Israel Institute of Technology, Elyachar Central Library

the specific steady-state limit. This results in an essentially data-free *solution of the practical final-value synthesis problem, in which the only data needed is whether or not the control goal has been achieved. Moreover, the algorithm is valid as long as the agents are known to be MEIP.*

5.3.2 Data-Driven Determination of Gains

In the previous subsection, we introduced a formula for a uniform gain α described by the ratio of m and M, that solves the practical final-value synthesis problem, where m and M are as defined in Remark 5.2. The parameter M depends on the integral function Γ of the controllers, evaluated on well-defined points, namely $\{\zeta \in \mathrm{Im}(\mathcal{E}^\top) : \|\zeta - \zeta^\star\| = \varepsilon\}$. Thus we can compute M exactly with no prior knowledge on the agents. This is not the case for the parameter m, which depends on the integral function of the agents. Without knowledge of any model of the agents, we need to obtain an estimate of m solely on the basis of input-output data from the agents.

From Remark 5.2 above, we know that $m = \sum_{i=1}^{n}(K_i^\star(\mathrm{y}_i^\star) + K_i(0)) = \sum_{i=1}^{n} m_i$ for some $\mathrm{y}^\star \in \mathbb{R}^n$ such that $\mathcal{E}^\top \mathrm{y}^\star = \zeta^\star$. Without any model of the agents, m cannot be computed directly, but we can yield an upper bound on m from measured input-output trajectories via the inverse relations k_i^{-1}, $i = 1, \ldots, n$.

Proposition 5.2. *Let* $(\mathrm{u}_i^\star, \mathrm{y}_i^\star)$, $(\mathrm{u}_{i,1}, \mathrm{y}_{i,1})$, $(\mathrm{u}_{i,2}, \mathrm{y}_{i,2})$, $\ldots$, $(\mathrm{u}_{i,r}, \mathrm{y}_{i,r})$ *and* $(0, \mathrm{y}_{i,0})$ *be steady-state input-output pairs for agent* i*, for some* $r \geq 0$*. Then:*

$$m_i \leq \mathrm{u}_{i,1}(\mathrm{y}_{i,1} - \mathrm{y}_{i,0}) + \cdots + \mathrm{u}_{i,r}(\mathrm{y}_{i,r} - \mathrm{y}_{i,r-1}) + \mathrm{u}_i^\star(\mathrm{y}^\star - \mathrm{y}_{i,r}).$$

Proof. We prove the claim by induction on the number of steady-state pairs, $r + 2$. First, consider the case $r = 0$ of two steady-state pairs. First, because $(0, \mathrm{y}_{i,0})$ is a steady-state pair, we know that $K_i(0) = -K_i^\star(\mathrm{y}_{i,0})$ by Fenchel duality. Similarly, $K_i(\mathrm{u}_i^\star) = \mathrm{u}_i^\star \mathrm{y}_i^\star - K_i^\star(\mathrm{y}_i^\star)$. Thus,

$$m_i = K_i^\star(\mathrm{y}_i^\star) + K_i(0) = K_i^\star(\mathrm{y}_i^\star) - K_i^\star(\mathrm{y}_{i,0}) \leq \mathrm{u}_i^\star(\mathrm{y}^\star - \mathrm{y}_{i,0}),$$

where we use the inequality $K_i^\star(b) - K_i^\star(c) \geq k_i^{-1}(c)(b - c)$ for $b = \mathrm{y}_{i,0}$ and $c = \mathrm{y}_i^\star$. Now, we move to the case $r \geq 1$. We write m_i as $(K_i^\star(\mathrm{y}_i^\star) - K_i^\star(\mathrm{y}_{i,r})) + (K_i^\star(\mathrm{y}_{i,r}) - K_i(0))$. The first element can be shown to be smaller or equal than $\mathrm{u}_i^\star(\mathrm{y}^\star - \mathrm{y}_{i,r})$ using the inequality $K_i^\star(b) - K_i^\star(c) \geq k_i^{-1}(c)(b - c)$ for $b = \mathrm{y}_{i,r}$ and $c = \mathrm{y}_i^\star$. The second element is smaller or equal than $\mathrm{u}_{i,1}(\mathrm{y}_{i,1} - \mathrm{y}_{i,0}) + \cdots + \mathrm{u}_{i,r}(\mathrm{y}_{i,r} - \mathrm{y}_{i,r-1})$ by induction hypothesis, as we use a total of $r+1$ steady-state input-output pairs. Thus, m_i is smaller or equal than the sum of the two bounds, which is equal to $\mathrm{u}_{i,1}(\mathrm{y}_{i,1} - \mathrm{y}_{i,0}) + \cdots + \mathrm{u}_{i,r}(\mathrm{y}_{i,r} - \mathrm{y}_{i,r-1}) + \mathrm{u}_i^\star(\mathrm{y}^\star - \mathrm{y}_{i,r})$. □

Remark 5.4. *If we only have two steady-state pairs,* $(\mathrm{u}_i^\star, \mathrm{y}_i^\star)$ *and* $(0, \mathrm{y}_{i,0})$*, the estimate on* m_i *becomes* $m_i \leq \mathrm{u}_i^\star(\mathrm{y}_i^\star - \mathrm{y}_{i,0})$*. Thus two steady-state pairs, corresponding to two measurements/experiments, are enough to yield a meaningful bound on* m_i*. However, it is important to note that more experiments yield better estimates of* m_i*, i.e., if* $r \geq 1$ *then the estimate in the theorem is better than the one above as long as* $(\mathrm{y}_{i,0}, \mathrm{y}_{i,1}, ..., \mathrm{y}_{i,r}, \mathrm{y}_i^\star)$ *is a monotone series.*

© Technion - Israel Institute of Technology, Elyachar Central Library

With Remark 5.4, we can hence compute an upper bound on m from the two steady-state pairs $(\mathrm{u}_i^\star, \mathrm{y}_i^\star)$ and $(0, \mathrm{y}_{i,0})$ of each agent. In the following, we present a method to actually obtain the required steady-state input-output pairs $(0, \mathrm{y}_{i,0})$ and $(\mathrm{u}_i^\star, \mathrm{y}_i^\star)$ of the agents.

We would like to estimate m_i from above by computing $\mathrm{y}_{i,0}$ and $\mathrm{u}_i^\star$. Designing experiments to measure these quantities are possible, but can require additional information on the plant, e.g. output-strict passivity. Instead, we opt for a different approach and try to estimate $\mathrm{y}_{i,0}$ and $\mathrm{u}_i^\star$ instead of computing them directly. This is described in Algorithm 1.

Algorithm 1 applies Remark 5.4 in order to bound m_i from above. It does so by using the monotonicity of the steady-state input-output relation to bound $\mathrm{u}_i^\star$ and $\mathrm{y}_{i,0}$ from above and below, as computing the exact values of $\mathrm{u}_i^\star$ and $\mathrm{y}_{i,0}$ might not be feasible only from experiments. It is important to note that the closed-loop experiments are done with output-strictly MEIP controller, which assure that the closed-loop system indeed converges. We prove the following:

Proposition 5.3. *The output $\mathbb{m}_i$ from Algorithm 1 is an upper bound on m_i.*

Proof. First, we show that the closed-loop experiments conducted by the algorithm indeed converge. The plant Σ_i is assumed to be passive with respect to any steady-state input-output pair it possesses. Moreover, the static controller $u = \beta_i(y - y_{\mathrm{ref}})$ is output-strictly passive with respect to any steady-state input-output pair it possesses. Thus it's enough to show that the closed-loop system has a steady-state, which will prove convergence as this is a feedback connection of a passive system with an output-strictly passive system. Indeed, a steady-state input-output pair $(\mathrm{u_i}, \mathrm{y_i})$ of the system needs to satisfy $\mathrm{u}_i \in k_i^{-1}(\mathrm{y}_i)$ and $\mathrm{u}_i = -\beta_i(\mathrm{y}_i - \mathrm{y}_{\mathrm{ref}})$. Thus it's enough to show that $-\beta_i(\mathrm{y_i} - \mathrm{y}_{\mathrm{ref}}) \in k^{-1}(\mathrm{y}_i)$ has a solution. This is equivalent to

$$0 \in k^{-1}(\mathrm{y}_i) + \beta_i(\mathrm{y}_i - \mathrm{y}_{\mathrm{ref}}) = \nabla\left(K_i^\star(\mathrm{y}_i) + \frac{\beta_i}{2}(\mathrm{y}_i - \mathrm{y}_{\mathrm{ref}})^2\right),$$

so y_i exists and is equal to the minimizer of $K_i^\star(\mathrm{y}_i) + \frac{\beta_i}{2}(\mathrm{y}_i - \mathrm{y}_{\mathrm{ref}})^2$. This shows that the closed-loop experiments converge. Thus the algorithm halts, and it remains to show that it outputs an upper-bound on m_i.

Using Remark 5.4, it's enough to show that $\mathrm{y}_{i,0} \in [\underline{\mathrm{y}_{i,0}}, \overline{\mathrm{y}_{i,0}}]$ and $\mathrm{u}_i^\star \in [\underline{\mathrm{u}_i^\star}, \overline{\mathrm{u}_i^\star}]$. To do so, we first claim that $U_1 \le \mathrm{u}_i^\star \le U_3$ and $Y_1 \le \overline{\mathrm{y}_{i,0}} \le Y_3$. We first show that $Y_1 \le \mathrm{y}_{i,0}$, by showing that $\mathrm{y}_{i,-} \le \mathrm{y}_{i,0}$. Indeed, because k_i is a monotone map, this is equivalent to saying that $\mathrm{u}_{i,-} \le 0$. By the structure of the second experiment, the steady-state input is close to -1, and in particular smaller than 0. The inequality $\mathrm{y}_{i,0} \le \mathrm{y}_{i,+}$ is proved similarly. We note that because $\mathrm{u}_{i,-} \approx -1$ and $\mathrm{u}_{i,+} \approx 1$, we have $\mathrm{u}_{i,-} \le \mathrm{u}_{i,+}$ and thus $\mathrm{y}_{i,-} \le \mathrm{y}_{i,+}$. as k_i is monotone.

Next, we prove that $U_1 \le \mathrm{u}_i^\star$. By monotonicity of k_i, this is equivalent to $Y_1 \le \mathrm{y}_i^\star$. Because $\mathrm{y}_{i,-} \le \mathrm{y}_{i,+}$, it's enough to show that either $\mathrm{y}_{i,-} \le \mathrm{y}_i^\star$ or $\mathrm{y}_{i,2} \le \mathrm{y}_i^\star$. If the first case is true, then the proof is complete. Otherwise, $\mathrm{y}_{i,-} > \mathrm{y}_i^\star$, so the algorithm finds $\mathrm{y}_{i,2}$ by running the closed-loop system in Figure 5.2

Algorithm 1 Estimating m_i for an MEIP Agent

Run the closed-loop system in Figure 5.2 with β_i small and $\mathrm{y}_{\mathrm{ref}} = \frac{1}{\beta_i}$
Wait for convergence, and measure the steady-state output $\mathrm{y}_{i,+}$ and the steady-state input $\mathrm{u}_{i,+}$
Run the closed-loop system in Figure 5.2 with β_i small and $\mathrm{y}_{\mathrm{ref}} = -\frac{1}{\beta_i}$
Wait for convergence, and measure the steady-state output $\mathrm{y}_{i,-}$ and the steady-state input $\mathrm{u}_{i,-}$
if $\mathrm{y}_{i,+} < \mathrm{y}_i^\star$ **then**
    Run the closed-loop system in Figure 5.2 with $\beta_i = 1$ and $\mathrm{y}_{\mathrm{ref}} \gg \mathrm{y}_i^\star$
    Wait for convergence, and measure the steady-state input $\mathrm{u}_{i,2}$ and output $\mathrm{y}_{i,2}$
else
    if $\mathrm{y}_{i,-} > \mathrm{y}_i^\star$ **then**
        Run the closed-loop system in Figure 5.2 with $\beta_i = 1$ and $\mathrm{y}_{\mathrm{ref}} \ll \mathrm{y}_i^\star$
        Wait for convergence, and measure the steady-state input $\mathrm{u}_{i,2}$ and output $\mathrm{y}_{i,2}$
    else
        Define $\mathrm{u}_{i,2} = \mathrm{u}_{i,+}$ and $\mathrm{y}_{i,2} = \mathrm{y}_{i,+}$
    end if
end if
Sort the array $\{\mathrm{u}_{i,-}, \mathrm{u}_{i,+}, \mathrm{u}_{i,2}\}$. Denote the result by $U = \{U_1, U_2, U_3\}$
Sort the array $\{\mathrm{y}_{i,-}, \mathrm{y}_{i,+}, \mathrm{y}_{i,2}\}$. Denote the result by $Y = \{Y_1, Y_2, Y_3\}$
if $U_2 > 0$ **then**
    Define $\underline{\mathrm{y}_{i,0}} = Y_1$ and $\overline{\mathrm{y}_{i,0}} = Y_2$
else
    if $U_2 < 0$ **then**
        Define $\underline{\mathrm{y}_{i,0}} = Y_2$ and $\overline{\mathrm{y}_{i,0}} = Y_3$
    else
        Define $\underline{\mathrm{y}_{i,0}} = Y_2$ and $\overline{\mathrm{y}_{i,0}} = Y_2$
    end if
end if
if $Y_2 > \mathrm{y}_i^\star$ **then**
    Define $\underline{\mathrm{u}_i^\star} = U_1$ and $\overline{\mathrm{u}_i^\star} = U_2$
else
    if $Y_2 < y_i^\star$ **then**
        Define $\underline{\mathrm{u}_i^\star} = U_2$ and $\overline{\mathrm{u}_i^\star} = U_3$
    else
        Define $\underline{\mathrm{u}_i^\star} = U_2$ and $\overline{\mathrm{u}_i^\star} = U_2$
    end if
end if
return $\mathbb{m}_i$ as the maximum over $\omega(\mathrm{y}_i^\star - \upsilon)$, where $\omega \in \{\underline{\mathrm{u}_i^\star}, \overline{\mathrm{u}_i^\star}\}$ and $\upsilon \in \{\underline{\mathrm{y}_{i,0}}, \overline{\mathrm{y}_{i,0}}\}$

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

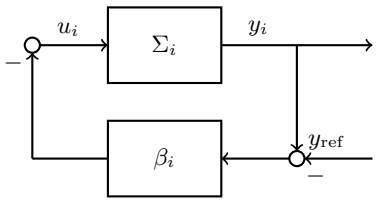


Figure 5.2: Experimental setup of the closed-loop experiment for estimating m_i as used in Algorithm 1.

with $\beta_i = 1$ and $\mathrm{y}_{\text{ref}} \ll y_i^\star$. The increased coupling strength implies that the steady-state output $\mathrm{y}_{i,2}$ should be close to y_{ref}, which is much smaller than $\mathrm{y}_i^\star$. Thus $\mathrm{y}_{i,2} < \mathrm{y}_i^\star$, which shows that $Y_1 \leq \mathrm{y}_1^\star$, or equivalently $U_1 \leq \mathrm{u}_1^\star$. The proof that $\mathrm{u}_1^\star \leq U_3$ is similar. This completes the proof of the proposition. □

Remark 5.5. *The second part of Algorithm 1, namely from Step 16 until the end, can be used to improve the estimates on* $\mathrm{u}_i^\star$ *and* $\mathrm{y}_{i,0}$*. Namely, we run another experiment, in which the agent converges to a steady-state input-output pair* $(\hat{\mathrm{u}}_i, \hat{\mathrm{y}}_i)$*. One then defines the array* $U = \{\underline{\mathrm{u}_i^\star}, \overline{\mathrm{u}_i^\star}, \hat{\mathrm{u}}_i\}$ *and* $Y = \{\underline{\mathrm{y}_{i,0}}, \overline{\mathrm{y}_{i,0}}, \hat{\mathrm{y}}_i\}$*, and applies this last part (line 16-36) once more. If* $\hat{\mathrm{u}}_i$ *is not between* $\underline{\mathrm{u}_i^\star}$ *and* $\overline{\mathrm{u}_i^\star}$*, (or equivalently,* $\hat{\mathrm{y}}_i$ *is not between* $\underline{\mathrm{y}_{i,0}}$ *and* $\overline{\mathrm{y}_{i,0}}$*), then we do not improve our estimate of* m_i*. Otherwise, we shrink the estimated intervals containing* $\mathrm{u}_i^\star$ *and* $\mathrm{y}_{i,0}$*, and thus improve our estimate of* m*. Doing this iteratively allows to exactly compute* $\mathrm{u}_i^\star$ *and* $\mathrm{y}_i^\star$*, which would be not advisable in practice due to the huge amount of necessary experiments.*

We saw that m_i can be bounded using no more than three experiments for general MEIP agents. However, we can improve on that if we somehow know that the agent is LTI. To be precise, we have the following proposition.

Proposition 5.4. *Suppose that the agent* Σ_i *is known to be both MEIP and LTI. Let* $(\tilde{\mathrm{u}}, \tilde{\mathrm{y}})$ *be any steady-state input-output pair for which either* $\tilde{\mathrm{u}} \neq 0$ *or* $\tilde{\mathrm{y}} \neq 0$.[1] *Then* $m_i = \frac{(\mathrm{y}_i^\star)^2 \tilde{\mathrm{u}}}{2\tilde{\mathrm{y}}}$*. Thus* m_i *can be calculated exactly using a single experiment.*

Proof. Indeed, [64] and Remark 2.9 show that in this case, k is a linear function, and the system state matrix is Hurwitz. Moreover, unless the transfer function of the agent is 0, k^{-1} is a linear function $k^{-1}(\mathrm{y}) = s\mathrm{y}$ for some $s > 0$. Thus $K^\star(\mathrm{y}) = \frac{s}{2}\mathrm{y}^2$. Now, $k^{-1}(0) = s \cdot 0 = 0$, so $(0,0)$ is a steady-state input-output pair, meaning that $\mathrm{y}_{i,0} = 0$. Moreover, we know that $\tilde{\mathrm{u}} = s\tilde{\mathrm{y}}$, and not both are zero, so we conclude that $\tilde{\mathrm{y}} \neq 0$, and that $s = \frac{\tilde{\mathrm{u}}}{\tilde{\mathrm{y}}}$. Thus, $K_i^\star(\mathrm{y}_{i,0}) = K_i(0) = 0$ and $K_i^\star(\mathrm{y}_i^\star) = \frac{s}{2}(\mathrm{y}_i^\star)^2$. This completes the proof, as $m_i = K_i^\star(\mathrm{y}_i^\star) - K_i^\star(\mathrm{y}_{i,0})$. □

[1] e.g., by running the system in Figure 5.2 with some $\beta > 0$ and $\mathrm{y}_{\text{ref}} \neq 0$

© Technion - Israel Institute of Technology, Elyachar Central Library

We conclude this chapter with Algorithm 2 for solving the practical final-value synthesis problem using the single-gain amplification scheme, which is applied to two case studies in Section 5.5

Algorithm 2 Synthesis Procedure for Practical Formation Control

1: Choose some output-strictly MEIP controllers Π_e such that the integral function Γ has a single minimizer $\zeta^\star$ when restricted to the set $\mathrm{Im}(\mathcal{E}^\top)$.
2: Choose some $\mathrm{y}^\star \in \mathbb{R}^n$ such that $\mathcal{E}^\top \mathrm{y}^\star = \zeta^\star$.
3: **for** $i = 1, ..., n$ **do**
4: Run Algorithm 1. Let $\mathtt{m}_i$ be the output
5: **end for**
6: Let $\mathtt{m} = \sum_{i=1}^n \mathtt{m}_i$
7: Compute $M = \min\{\zeta \in \mathrm{Im}(\mathcal{E}^\top) : \ \|\zeta - \zeta^\star\| = \varepsilon\}$
8: Compute $\alpha = \mathtt{m}/M$
9: **return** the controllers $\{\alpha\Pi_e\}_{e\in\mathbb{E}}$;

Remark 5.6. *Step 1 of the algorithm allows almost complete freedom of choice for the controllers. One possible choice are the static controllers $\mu_e = \zeta_e - \zeta_e^\star$. Moreover, if Π_e is any MEIP controller for each $e \in \mathbb{E}$, and $\gamma_e(\zeta_e) = 0$ has a unique solution for each $e \in \mathbb{E}$, then the "formation reconfiguration" scheme from Subsection 4.2.3 suggests a way to find the required controllers using mild augmentation.*

Remark 5.7. *The algorithm allows one to choose any vector $\mathrm{y}^\star$ such that $\mathcal{E}^\top \mathrm{y}^\star = \zeta^\star$. All possible choices lead to some gain α which assures a solution of the practical final-value synthesis problem, but some choices yield better results (i.e., smaller gains) than others. The optimal $\mathrm{y}^\star$, minimizing the estimate $\mathtt{m}$, can be found as the minimizer of the problem $\min\{K^\star(\mathrm{y}) : \ \mathcal{E}^\top \mathrm{y} = \zeta^\star\}$, which we cannot compute using data alone. One can use physical intuition to choose a vector $\mathrm{y}^\star$ which is relatively close to the actual minimizer, but the algorithm is still valid no matter which $\mathrm{y}^\star$ is chosen.*

5.4 Non-Uniform Gain Amplification Methods

Let us revisit Figure 5.1 and let $A = \mathrm{diag}(\{a_e\}_e \in \mathbb{E})$ with positive, but possibly distinct entries a_e. These additional degrees of freedom can be used, for example, to reduce the conservatism and retrieve a smaller norm of the adjustable gain vector a while still solving the practical final-value synthesis problem. It follows directly from Theorem 5.1 that there always exists a bounded vector a that solves the practical final-value synthesis problem. However, the question remains how the entries of a can be chosen based only on knowledge of input-output data and passivity properties.

Our idea here is to probe our diffusively coupled system for given gains a_e and adjust the gains according to the resulting steady-state output. By

© Technion - Israel Institute of Technology, Elyachar Central Library

iteratively performing experiments in this way, we strive to find controller gains that solve the practical final-value synthesis problem. This idea and approach is tightly connected to iterative learning control, where one iteratively applies and adjusts a controller to improve the performance of the closed-loop for a repetitive task [20]. Our approach here is based on passivity and network optimization with only requiring the possibility to perform iterative experiments.

One natural idea in this direction is to define a cost function that penalizes the distance of the resulting steady-state to the desired formation control goal and then apply a gradient descent approach, adjusting the gain a for each experiment. However, to obtain the gradient of $\|\mathcal{E}^\top \mathrm{y}(a) - \zeta^\star\|^2$ with respect to the vector a, where $\mathrm{y}(a)$ is the steady-state output of $(\mathcal{G}, \Sigma, \Pi, a)$, one requires knowledge of the inverse relations k_i^{-1} for all $i = 1, \dots, n$. With no model of the agents available, a direct gradient descent approach is hence infeasible. Moreover, it can be shown that even approximate- or stochastic-gradient descent approaches are very hard to implement distributedly, due to powers of weighted Laplacian matrices appearing the true value of the gradient. Therefore, we present in the following a simple iterative multi-gain control scheme without knowledge on the exact steepest descent direction.

We start off with an arbitrarily chosen gain vector a_0 with positive entries. Due to Assumption 5.1, the closed-loop converges to a steady state. According to the measured state, the idea is then to iteratively perform experiments and update the gain vector until we reach our control goal, i.e., practical final-value synthesis. The update formula can be summarized by

$$a_e^{(j+1)} = a_e^{(j)} + h v_e, \quad e \in \mathbb{E}, \tag{5.3}$$

where $h > 0$ is the step size and v, with entries v_e, $e = 1, \dots, |\mathbb{E}|$, is the update direction. We denote the e-th entry of $\mathcal{E}^\top \mathrm{y}$ as f_e and choose v in each iteration such that

$$v_e = \begin{cases} \frac{\mathrm{f}_e - \zeta_e^\star}{\gamma_e(\mathrm{f}_e)} & \gamma(\mathrm{f}_e) \neq 0 \\ 0 & \text{otherwise} \end{cases}, \tag{5.4}$$

for all $e = 1, \dots, |\mathbb{E}|$. If k^{-1} and γ are differentiable functions, then we claim that $F(a) = ||\mathcal{E}^\top \mathrm{y}(a) - \zeta^\star||^2$ decreases in the direction of v, i.e., $v^\top \nabla F(a) < 0$. This leads to a multi-gain distributed control scheme, using (5.3) with (5.4), which is summarized in Algorithm 3. This multi-gain distributed control scheme is guaranteed to solve the practical final-value synthesis problem after a finite number of iterations, which is summarized in the following theorem.

Theorem 5.2. *Suppose that the functions k^{-1}, γ are differentiable, and that there exists an agent $i_0 \in \mathbb{V}$ such that $\frac{dk_{i_0}^{-1}}{d\mathrm{y}_{i_0}} > 0$ for any point $\mathrm{y}_{i_0} \in \mathbb{R}$. Moreover, assume that $\frac{d\gamma_e}{d\zeta_e} > 0$ for any $e \in \mathbb{E}, \zeta_e \in \mathbb{R}$. Then $v^\top \nabla F(a) \leq 0$, with v, F as defined in Algorithm 3 (with equality if and only if $\mathcal{E}^\top \mathrm{y}(a) = \zeta^\star$). Furthermore, if the step size $h > 0$ is small enough, then the Algorithm 3 halts after finite time, providing a gain vector that solves the practical final-value synthesis problem.*

© Technion - Israel Institute of Technology, Elyachar Central Library

Algorithm 3 Practical Formation control with derivative-free optimization

1: Initialize $a^{(0)}$, e.g. with $\mathbb{1}_{|\mathbb{E}|}$.
2: Choose step size h and set $j = 0$.
3: **while** $F(a) = \|\mathcal{E}^\top \mathrm{y}(a) - \zeta^\star\|^2 > \varepsilon^2$ **do**
4: Apply $a^{(j)}$ to the closed loop.
5: Compute $v_e = \begin{cases} \frac{\mathrm{f}_e - \zeta^\star_e}{\gamma_e(\mathrm{f}_e)} & \gamma(\mathrm{f}_e) \neq 0 \\ 0 & \gamma(\mathrm{f}_e) = 0 \end{cases} \forall e.$
6: Update $a_e^{(j+1)} = a_e^{(j)} + h v_e$, $j = j + 1$.
7: **end while**
8: **return** a.

For streamlining reasons, we postpone the proof of the theorem to the end of the section. Algorithm 3 together with the theoretical results from Theorem 5.2 provide us with a very simple and distributed, iterative control scheme with theoretical guarantees. Note also, that the steady-states of the agents are independent of their initial condition. For each iteration, the agents can hence start from the position they converged to at the last iteration. This can be interpreted similarly to Remark 5.3, where gains are updated on a slower time scale than convergence of the agents. However, instead of only the information whether practical final-value synthesis is achieved, we need the actual difference $\mathcal{E}^\top \mathrm{y} - \zeta^\star$ that is achieved with the current controller in each iteration. In the special case of a proportional controllers $\mu_e = \zeta_e - (\zeta^\star)_e$, yielding $v_e = 1$, we retrieve the exact controller scheme proposed in Remark 5.3.

An alternative gradient-free control scheme is the extremum seeking framework as presented in [45]. Assuming that k^{-1} and γ are twice continuously differentiable, a step in the direction of steepest descent is approximated every $4|\mathbb{E}|$ steps (cf. [45, Theorem 1]). While the extremum seeking framework approximates the steepest descent (and the simple multi-gain approach only guarantees a descending direction), it also requires large amounts of experiments per approximated gradient step. Furthermore, the algorithm as presented in [45] cannot be computed in a purely distributed fashion. Therefore, the simple distributed control scheme in Algorithm 3 displays significant advantages in the present problem setup.

We now repay our debt and prove Theorem 5.2. We first need to prove a lemma:

Lemma 5.4. *Suppose that the assumptions of Theorem 5.2 hold, and let $C > 0$ be any constant. Define $A_1 = \{\mathrm{y} \in \mathbb{R}^n : \ \|\mathcal{E}^\top \mathrm{y} - \zeta^\star\| \leq C\}$ and $A_2 = \{\mathrm{y} \in \mathbb{R}^n : \sum_i k_i^{-1}(\mathrm{y}_i) = 0\}$. Then the set $A_1 \cap A_2$ is bounded.*

Proof. First, we note that the inequality $\|\mathcal{E}^\top \mathrm{y} - \zeta^\star\| \leq C$ implies that for any edge $\{i, j\} \in \mathbb{E}$, we have $|\mathrm{y}_i - \mathrm{y}_j| \leq C + ||\zeta^\star||$ by the triangle inequality. We let $\omega = (C + ||\zeta^\star||)\mathrm{diam}(\mathcal{G})$, where $\mathrm{diam}(\mathcal{G})$ is the diameter of the graph $\mathcal{G}$, so that if there exists some $i, j \in \mathbb{V}$ such that $|\mathrm{y}_i - \mathrm{y}_j| > \omega$ then $\mathrm{y} \notin A_1$. Moreover, let $\mathrm{z} = k(\mathbf{0})$. Then $\sum_i k_i^{-1}(\mathrm{z}_i) = 0$. Moreover, if $\mathrm{y} \in \mathbb{R}^n$ satisfies

© Technion - Israel Institute of Technology, Elyachar Central Library

$\forall i:\ \mathrm{y}_i > \mathrm{z}_i$, then $\mathrm{y} \notin A_2$. Indeed, for each i we have $k_i^{-1}(\mathrm{y}_i) \geq k_i^{-1}(\mathrm{z}_i)$, and $k_{i_0}^{-1}(\mathrm{z}_{i_0}) > k_{i_0}^{-1}(\mathrm{y}_{i_0})$, meaning that $\sum_i k_i^{-1}(\mathrm{y}_i) > \sum_i k_i^{-1}(\mathrm{z}_i) = 0$. Similarly, if $\forall i, \mathrm{z}_i > \mathrm{y}_i$ then $\mathrm{y} \notin A_2$. We now claim that for any $\mathrm{y} \in A_1 \cap A_2$ and any $i \in \mathbb{V}$, we have $C_1 < \mathrm{y}_i < C_2$, where $C_1 = \min_j \mathrm{z}_j - \omega - 1$ and $C_2 = \max_j \mathrm{z}_j + \omega + 1$. Indeed, take any $\mathrm{y} \in \mathbb{R}^n$, and suppose there exists $i \in \mathbb{V}$ such that $\mathrm{y}_i \geq C_2$. There are two possibilities.

i) There is some $k \in \mathbb{V}$ such that $\mathrm{y}_k < \max_j \mathrm{z}_j + 1$. Then $|\mathrm{y}_i - \mathrm{y}_k| > \omega$, implying that $\mathrm{y} \notin A_1$.

ii) For any $k \in \mathbb{V}$, $\mathrm{y}_k \geq \max_j \mathrm{z}_j + 1$, implying that $\mathrm{y} \notin A_2$.

Similarly, one shows that if there's some i such that $\mathrm{y}_i \leq C_1$, then $\mathrm{y} \notin A_1 \cap A_2$. This completes the proof. □

Proof of Theorem 5.2. Consider the solution $\mathrm{y}(a)$ of $\mathbf{0} = k^{-1}(\mathrm{y}) + \mathcal{E}\mathrm{diag}(a)\gamma(\mathcal{E}^\top \mathrm{y})$ as a function of a. Then $\mathrm{y}(a)$ is a differentiable function by the inverse function theorem, and it's differential is given by:

$$\frac{d\mathrm{y}}{da} = -X(\mathrm{y}(a))\mathcal{E}\mathrm{diag}(\gamma(\mathcal{E}^\top \mathrm{y}(a))), \tag{5.5}$$

where the matrix $X(\mathrm{y})$ is given by

$$X(\mathrm{y}) = [\mathrm{diag}(\nabla k^{-1}(\mathrm{y})) + \mathcal{E}\mathrm{diag}(\nabla\gamma(\mathcal{E}^\top \mathrm{y}))\mathcal{E}^\top]^{-1}. \tag{5.6}$$

We note that $X(\mathrm{y})$ is a positive-definite matrix for any $\mathrm{y} \in \mathbb{R}^n$, by Proposition 6.1. We conclude that the gradient of F is given by:

$$\nabla F(a) = -\mathrm{diag}(\gamma(\mathcal{E}^\top \mathrm{y}(a)))\mathcal{E}^\top X(\mathrm{y}(a))\mathcal{E}(\mathcal{E}^\top \mathrm{y}(a) - \zeta^\star). \tag{5.7}$$

We note that $v^\top \mathrm{diag}(\gamma(\mathcal{E}^\top \mathrm{y}(a)) = \mathcal{E}^\top \mathrm{y}(a) - \zeta^\star$, as $\gamma_e(x_e) = 0$ if and only if $\mathrm{f}_e = \zeta_e^\star$ by strict monotonicity. Thus,

$$v^\top \nabla F(a) = -(\mathcal{E}(\mathcal{E}^\top \mathrm{y}(a) - \zeta^\star))^\top X(\mathrm{y}(a))\mathcal{E}(\mathcal{E}^\top \mathrm{y}(a) - \zeta^\star) \tag{5.8}$$

which is ≤ 0 as $X(\mathrm{y}(a))$ is a positive-definite matrix.

Now, we claim that $v^\top \nabla F(a) = 0$ if and only if $\mathcal{E}^\top \mathrm{y}(a) = \zeta^\star$. Indeed, $\zeta^\star \in \mathrm{Im}(\mathcal{E}^\top)$, so we denote $\zeta^\star = \mathcal{E}^\top \mathrm{y}_0$ for some y_0. As $X(\mathrm{y}(a))$ is positive definite, (5.8) implies that $v^\top \nabla F(a) = 0$ if and only if $\mathcal{E}(\mathcal{E}^\top \mathrm{y}(a) - \zeta^\star) = \mathcal{E}\mathcal{E}^\top(\mathrm{y}(a) - \mathrm{y}_0)$ is the zero vector. The kernel of the Laplacian $\mathcal{E}\mathcal{E}^\top$ is the span of the all-one vector $\mathbb{1}_n$, so $\mathrm{y}(a) - \mathrm{y}_0 = \kappa\mathbb{1}_n$ for some $\kappa > 0$, hence $\mathcal{E}^\top \mathrm{y}(a) = \mathcal{E}^\top \mathrm{y}_0 = \zeta^\star$. This concludes the first part of the proof.

As for convergence, we know that if h is small enough, then $F(a^{(j+1)}) < F(a^{(j)})$, although the value of h so that $F(a^{(j+1)}) < F(a^{(j)})$ can depend on $a^{(j)}$ itself. However, it is obvious that if h is small enough, then for any j, we have $F(a^{(j)}) \leq F(a^{(0)})$. We let $C = \sqrt{F(a^{(0)})} = \|\mathcal{E}^\top \mathrm{y}(a^{(0)}) - \zeta^\star\|$, and consider the sets $A_1 = \{\mathrm{y}:\ \|\mathcal{E}^\top \mathrm{y} - \zeta^\star\| \leq C\}$ and $A_2 = \{\mathrm{y}:\ \sum_i k_i^{-1}(\mathrm{y}_i) = 0\}$.

For any j, we know that $\mathrm{y}(a^{(j)}) \in A_1$ by above, and that $\mathrm{y}(a^{(j)}) \in A_2$ by the steady-state equation $\mathbf{0} = k^{-1}(\mathrm{y}(a)) + \mathcal{E}\mathrm{diag}(a)\gamma(\mathcal{E}^\top \mathrm{y}(a))$. This shows that all

© Technion - Israel Institute of Technology, Elyachar Central Library

steady-state outputs achieved during the algorithm are in the set $\mathcal{D} = A_1 \cap A_2$, which is bounded by Lemma 5.4. The mapping sending a matrix to its minimal singular value is continuous, meaning that $\underline{\sigma}(X(\mathrm{y}))$ achieves a minimum on the set $\mathcal{D}$ at some point y_1, and the minimum is positive as $X(\mathrm{y}_1)$ is positive-definite. We denote the minimum value by $\underline{\sigma}(\mathcal{D})$.

Now, consider equation (5.8). We get that $v^\top \nabla F(a)$ is bounded by above $-\underline{\sigma}(\mathcal{D})||\mathcal{E}(\mathcal{E}^\top \mathrm{y}(a) - \zeta^\star)||^2$. In turn, we saw above that unless $\mathcal{E}^\top \mathrm{y}(a) = \zeta^\star$, $\mathcal{E}(\mathcal{E}^\top \mathrm{y}(a) - \zeta^\star) \neq \mathbf{0}$, meaning that $\|\mathcal{E}(\mathcal{E}^\top \mathrm{y}(a) - \zeta^\star)\| \geq \varsigma||\mathcal{E}^\top \mathrm{y}(a) - \zeta^\star||^2$, where ς is the minimal nonzero singular value of $\mathcal{E}$. Hence, at any time step j, $v^\top \nabla F(a^{(j)}) < -\underline{\sigma}(\mathcal{D})\varsigma F(a^{(j)})$. In turn we conclude that $F(a^{(j+1)}) = F(a^{(j)}) - h\underline{\sigma}(\mathcal{D})\varsigma F(a^{(j)}) + o(h) = (1 - h\underline{\sigma}(\mathcal{D})\varsigma)F(a^{(j)}) + o(h)$. Iterating this equation shows that eventually, $F(a^{(j)}) < \varepsilon$, completing the proof. □

5.5 Case Studies

We will present two extensive case studies. In each case, we apply the presented single-gain and multi-gain approaches. The first example considers a class of vehicles trying to coordinate their velocity in presence of drag and external forces. The second example studies the model-free cooperative control of a neural network.

5.5.1 Velocity Coordination in Vehicles with Drag and Exogenous Forces

Consider a collection of 5 one-dimensional robots, each modeled by a double integrator $G(s) = \frac{1}{s^2}$. The robots try to coordinate their velocity. Each of them has its own drag profile $f(\dot{p})$, which is unknown to the algorithm, but it is known that f is increasing and $f(0) = 0$. Moreover, each vehicle experiences external forces (e.g., wind, and being placed on a slope). The velocity of the vehicles is governed by the equation

$$\Sigma_i : \begin{cases} \dot{x_i} &= -f_i(x_i) + u_i + w_i \\ y_i &= x_i, \end{cases} \tag{5.9}$$

where x_i is the velocity of the i-th vehicle, f_i is its drag model, w_i are the exogenous forces acting on it, u_i is the control input, and y_i is the measurement. In the simulation, the drag models f_i are given by $c_d|x|x$, where the drag coefficient c_d is chosen randomly as a log-uniform value between 0.1 and 1. We assume that the vehicles are light, so the wind accelerates the vehicles by a non-negligible amount. Thus, the exogenous input w_i is randomly chosen between -2 and 2. We wish to achieve velocity consensus, with error no greater than $\epsilon = 1$. We consider a diffusive coupling of the system. We use the cycle graph $\mathcal{G} = \mathscr{C}_5$, and we choose proportional controllers $\zeta_e = \mu_e$ with gain equal to 1.

We apply the amplification scheme presented in Algorithm 2 and choose the consensus value $y_i^\star = 1.5_{m/sec}$ to use in the estimation algorithm. Note

© Technion - Israel Institute of Technology, Elyachar Central Library

that the plants are MEIP, but not output-strictly MEIP. We use Algorithm 1 to estimate the needed uniform gain α. In all cases, the first and second experiments are conducted with $\beta_i = 0.01$, and $y_{\text{ref}} = \pm 100$. Based on the results of the previous experiments, we run a third experiment on each of the agents for which this is required, this time with $\beta_i = 1$ and $y_{\text{ref}} = \pm 10$, where the sign is chosen according to Algorithm 1. The experimental results of the five plants are available in Figure 5.3.

We now estimate each m_i for $i = 1, ..., 5$ using Remark 5.4. For example, for agent #2 we get the three steady-state input-output pairs $(1.0167, -1.6748)$, $(-0.9586, -4.1426)$, and $(4.9341, 5.0659)$. Monotonicity implies that it has steady-states $(u_1^\star, y_1^\star) = (u_1^\star, 1.5)$ and $(0, y_{1,0})$ with $1.0168 \leq u_1^\star \leq 4.9341$ and $-4.1426 \leq y_{1,0} \leq -1.6748$. Thus we can estimate $m_2 \leq 4.9341 \cdot (1.5 - (-4.1426)) = 27.8412$. Similarly, we estimate m_1, m_3, m_4, m_5, and get $m_1 = 17.4568, m_3 = 2.1153, m_4 = 2.0410, m_5 = 13.0345$. Thus, their sum is $m = 62.4888$.

As for estimating M, we have $\Gamma_e(\zeta_e) = \zeta_e^2$, so $\Gamma(\zeta) = |\mathbb{E}|\|\zeta\|^2$. The minimum is at $\zeta = 0$, and by definition we have $M = \min_{\mathrm{x} \in \mathrm{Im}(\mathcal{E}^\top):\ \|\mathrm{x}-\mathrm{x}_0\|=\varepsilon} \Gamma(\mathrm{x}) = |\mathbb{E}|\varepsilon^2$. Thus, we get $\alpha = \frac{m}{M} = \frac{m}{5\varepsilon^2} = 12.47$. To verify the algorithm, we run the closed-loop system $(\mathcal{G}, \Sigma, \Pi, \alpha \otimes \mathbb{1})$ with the gain α we found. The results are available in Figure 5.4(a). One can see that the overall control goal is achieved - the agents converge to a steady-state which is ε-close to consensus. However, it should be noted that the agents actually converge to a much closer steady-state than predicted by the algorithm. Namely, the distance of the steady-state output from consensus is roughly 0.1, much smaller than 1. Thus, the algorithm probably overestimates the minimal gain α needed by at least one order of magnitude. One can mitigate this by using more experiments to better estimate m_i, as mentioned in Proposition 5.2 or in Remark 5.5. For comparison, running the algorithm with $\mathrm{y}^\star = \mathbf{0}$ gives $\alpha = 16.07$ and $\mathrm{y}_i^\star = 1.25_{m/sec}$ gives $\alpha = 12.40$.

As seen above, the main factor for α's size is the second agent, which contributes about 45% to the size of m. Thus we wish to reduce m_2's size by using additional experiment. Running a fourth experiment (just on agent #2) with $\beta_i = 1$ and $y_{\text{ref}} = 4.5$ gave the steady-state input-output pair $(2.1577, 2.3423)$. Estimating m_i using Remark 5.5 now gives $m_i = 11.96$, which in turn gives $\alpha = 9.33$. Thus one additional experiment on a single agent allowed us to reduce the value of α by about 25%.

Altogether we could show that Algorithm 2 manages to solve the practical velocity consensus problem of vehicles, affected by drag and exogenous inputs, without using any model for the agents, while conducting very few experiments for each agent. However, it overestimates the required coupling needed to achieve practical consensus, and thus has unnecessarily large energy consumption. The trajectories of the closed loop system with the new gain are available in Figure 5.4(b).

© Technion - Israel Institute of Technology, Elyachar Central Library

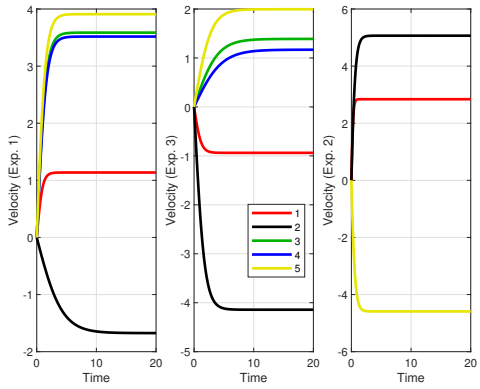


Figure 5.3: Results of the first set of experiments for the data-driven synthesis of a network of vehicles.

Let us now apply the iterative multi-gain control strategy. We start with $a^{(0)} = 0.1 \otimes 1_{|\mathbb{E}|}$, we choose the step size $h = 0.1$ and apply Algorithm 3. In fact, since $\zeta^\star = \mathbf{0}$ and $\zeta_e = \mu_e$, we receive $v = 1_{|\mathbb{E}|}$, which constitutes the special case where our iterative scheme yields the controller scheme proposed in Remark 5.3. The corresponding norm of the gain vector and the resulting ε in each iteration is illustrated in Figure 5.5. After 10 iterations, we already arrive at a vector, which solves the practical final-value synthesis problem with $\|a^{(10)}\| = 4.24$, while $\varepsilon = 0.99 < 1$. Note that the controller with the uniform gain had $\|a\| = \sqrt{|\mathbb{E}|} \cdot 12.47 = 27.8838$, so the iterative scheme beats it by a factor of 7 in terms of energy.

5.5.2 Clustering in Neural Networks

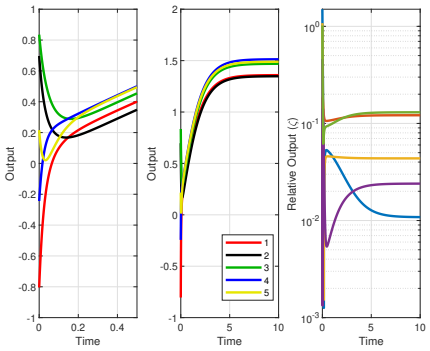
Consider a collection of $n = 40$ neurons, each modeled by the dynamical system

$$\begin{cases} \dot{x_i} = -\frac{1}{\tau_i} x_i + b_i u_i + w_i \\ y_i = \tanh(x_i), \end{cases} \tag{5.10}$$

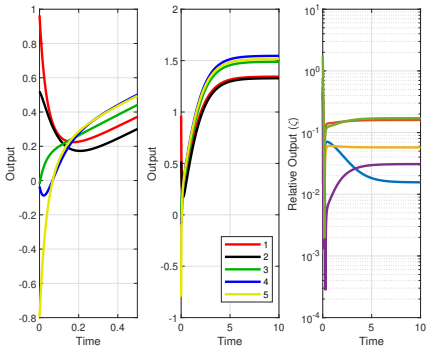
where x_i is the voltage of the i-th neuron, τ_i is its self-correlation time, b_i is the correlation coefficient, determining the susceptibility of the neuron to external inputs, and w_i is some (constant) exogenous input. The values of the parameters τ_i, b_i, w_i are unknown to the algorithm. However, it is known that $\tau_i, b_i > 0$. In the simulation, the numbers τ_i were chosen log-uniformly between 1 and 10, b_i were log-uniformly chosen between 0.3 and 3, and w_i uniformly chosen between -0.5 and 0.5.

We wish to design a two-cluster formation. Namely, we divide the $n = 40$ neurons into two groups of 20, and require that the neurons in each set will share the same steady-state output. Moreover, we require that the output of the first set will be smaller than the output of the second set by 1. We denote the desired formation by $\zeta^\star$. We allow an error margin of $\varepsilon = 0.2$. We use a

© Technion - Israel Institute of Technology, Elyachar Central Library



(a) The closed-loop network with uniform gain $\alpha = 12.47$. The two leftmost graphs plot the agents' trajectories over 0.5 seconds and over 10 seconds. The rightmost graphs plots the relative outputs.



(b) The closed-loop network with uniform gain $\alpha = 9.33$. The two leftmost graphs plot the agents' trajectories over 0.5 seconds and over 10 seconds. The rightmost graphs plots the relative outputs.

Figure 5.4: Results of data-driven control for a vehicle network.

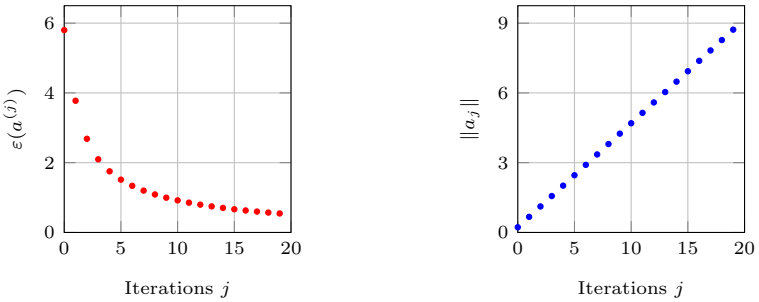
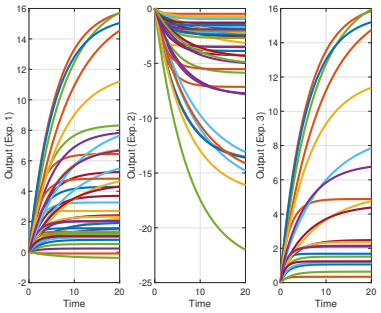


Figure 5.5: The norm of the gain vector and distance from desired formation over iterations j, when applying the iterative multi-gain control strategy for velocity coordination in vehicles.

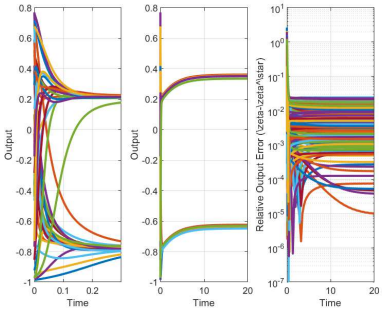
random Erdős-Rényi graph, with probability of each edge to appear is $p = 0.25$. We choose proportional controllers $\mu_e = \zeta_e - (\zeta^\star)_e$ with gain equal to 1.

We apply the amplification scheme presented in Algorithm 2 and choose the desired value $y^\star = [0, \cdots, 0, 1, \cdots, 1]^\top$. Note that the plants are MEIP, but not output-strictly MEIP. In all cases, the first and second experiments are conducted with $\beta_i = 0.01$, and $y_{\text{ref}} = \pm 100$. Based on the results of the previous experiments, we run a third experiment on each of the agents for which it is required, this time with $\beta_i = 1$ and $y_{\text{ref}} = \pm 2$, with sign chosen as in Algorithm 1. The experimental results are available in Figure 5.6(a).

We now estimate each m_i for $i = 1, ..., 40$ using Remark 5.4, and achieve $m = 61.1740$. As for estimating M, we have $\Gamma_e(\zeta_e) = \zeta_e^2$, so $\Gamma(\zeta) = \|\zeta\|^2$. The minimum is at $\zeta = \mathbf{0}$, and by definition we have $M = \min_{\mathrm{x} \in \mathrm{Im}(\mathcal{E}^\top):\ \|\mathrm{x} - \mathrm{x}_0\| = \varepsilon} \Gamma(\mathrm{x}) = |\mathbb{E}|\varepsilon^2$. Thus, we get $\alpha = \frac{m}{M} = \frac{m}{|\mathbb{E}|\varepsilon^2} = 8.8402$. To verify the algorithm, we run



(a) Results of first set of Experiments for the data-driven synthesis of a neural network.



(b) The closed-loop network with uniform gain $\alpha = 8.8402$. The two leftmost graphs plot the agents' trajectories over 0.3 seconds and over 20 seconds. The rightmost graph plots the error $\zeta(t) - \zeta^\star$.

Figure 5.6: Results of data-driven control for a neural network.

the closed-loop system $(\mathcal{G}, \Sigma, \Pi, \alpha\mathbb{1})$ with the gain α we found. The results are available in Figure 5.6(b). One can see that the overall control goal is achieved - the agents converge to a steady-state which is ε-close to the desired formation. However, as before, the agents actually converge to a much closer steady-state than predicted by the algorithm.

Applying the iterative multi-gain scheme, we start with $a^{(0)} = 1_{|\mathbb{E}|}$, we choose the step size $h = 1$ and apply Algorithm 3. In fact, since we consider proportional controllers with gain equal to 1 again, we receive $v = 1_{|\mathbb{E}|}$. The norm of the gain vector and the resulting ε in each iteration is illustrated in Figure 5.7. After 4 iterations, we already at a vector solving the specified practical final-value synthesis problem with $\|a^{(4)}\| = 65.76$, while $\varepsilon = 0.19 < 0.2$. The controller with the uniform gain had $\|a\| = \sqrt{|\mathbb{E}|} \cdot 8.8402 = 116.2747$, so the iterative scheme beats it by a factor of 2 in terms of energy.

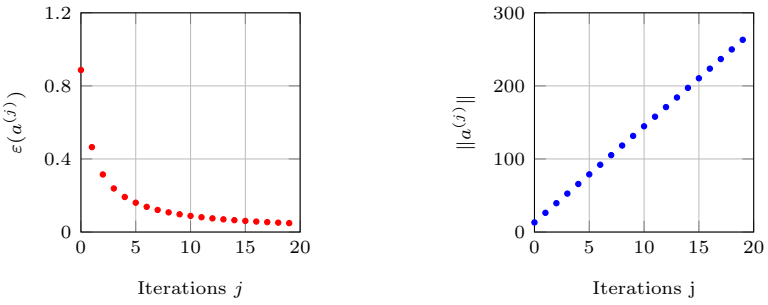


Figure 5.7: The norm of the gain vector and distance from desired formation over iterations j, when applying the iterative multi-gain control strategy for neural network clustering

© Technion - Israel Institute of Technology, Elyachar Central Library

5.6 Conclusions

We presented an approach for model-free practical final-value synthesis for diffusively coupled systems only on the premise of passivity of the agents, based on the network optimization framework. The presented approach led to two control schemes: with additional three experiments on the agents, we can bound the controller gain solving the practical final-value synthesis problem from above, or we can iteratively adapt the adjustable gain vector until practical final-value synthesis is reached. Both approaches are especially simple in their application, while still being scalable and providing theoretical guarantees. In the next section, we'll explore uncertainties in another fundamental aspect of multi-agent systems, focusing on the underlying graph instead of the agents' models, studying the network identification problem.

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

Chapter 6

Applications of the Network Optimization Framework in Network Identification

This section is a review of [137,141]. We apply the network optimization framework to study the problem of network identification, in which we are given a multi-agent system with known agents and controllers, but unknown interaction graph. Our goal is to find out what is the underlying interaction graph, where we are allowed to inject exogenous inputs into the network and measure the corresponding output.

6.1 Introduction

One of the most important aspects in multi-agent systems, both in theory and in practice, is the information-exchange layer, governing the interaction of agents with one another. Identifying the underlying network of a multi-agent system from measurements is of great importance in many applications. Examples include data mining and privacy in social networks [177], estimating brain connectivity using EEG in neuroscience [18, 127], and estimating influence between currencies using a history of their exchange rates in finance [102]. Other fields with similar problems include systems biology [54, 74], communication networks [29, 112], ecology [94, 159] and physics [14, 156].

A few different approaches for network identification were considered in the literature, including measuring naturally occurring excitations [172], using unmeasured intrinsic persistent excitations [33], probing the network with specifically design inputs [156], and using transfer entropy and other information-theoretic measures [163]. Seminal works dealing with network identification include [128] in which sparse enough topologies can be identified from a small number of observations, and [87, 88], providing exact reconstruction for tree-

© Technion - Israel Institute of Technology, Elyachar Central Library

like graphs. Other important works include [101], presenting a sieve method for solving the network identification problem for consensus-seeking networks, and [100], using a node-knockout method. More recent methods include spectral methods [89] and auto-regressive models [104]. The network identification theory for nonlinear agents, with possibly nonlinear interactions, is less developed. Some works linearize the dynamics of the system around a steady-state, and then use a sparse recovery technique to identify the connecting matrix [56, 81]. Other works use adaptive observers to try and find the network structure, while assuming the Lipschitzity of certain elements in the dynamics [22, 30]. However, these methods are not applicable when the dynamics are not Lipschitz, or when they cannot be linearized effectively. For example, the drag force exerted on many high-velocity moving objects is quadratic in their velocity. Another example includes finite-time consensus algorithms, which include a nonlinear network feedback term which is not Lipschitz [165]. We address this problem by providing a network identification scheme for a wide range of systems, including nonlinear ones, by probing them with specially constructed inputs.

We use the passivity-based network optimization scheme to solve the network identification problem for a large class of systems with nonlinear agents and controllers. We do so by injecting constant exogenous inputs, and tracking the output of the agents. By appropriately designing the exogenous inputs, we are able to differentiate the outputs of the closed-loop system associated to different underlying graphs. The key idea is that the steady-state outputs are solutions to network optimization problems and they are one-to-one dependent on the exogenous input. This dependency can be linearized, and the associated matrix can be found by looking at a finite number of inputs and outputs.

The rest of the chapter is organized as follows. In Section 6.2, we define the network model, and consider the related problem of differentiation between two multi-agent systems with identical agents and controllers, but different underlying graphs. In Section 6.3, we'll consider a solution to the network identification problem. Moreover, we'll study the network identification problem from a complexity theory point of view, and show that the resulting algorithm is optimal. We'll present relevant case studies at the end of each section.

6.2 Network Differentiation Using Constant Exogenous Inputs

The problem of network identification we aim to solve can be stated as follows. Given a multi-agent system $(\mathcal{G}, \Sigma, \Pi)$, in which models for Σ and Π are known, determine the underlying graph structure $\mathcal{G}$ from the network measurements and an appropriately designed exogenous input w. We start by defining the network model and formulating the problem.

© Technion - Israel Institute of Technology, Elyachar Central Library

6.2.1 Motivation and Problem Definition

Many works on network identification consider networks of consensus-seeking agents [100, 101],

$$\dot{x}_i = \sum_{i\sim j} \nu_{ij}(x_j - x_i) + B_i w_i, \tag{6.1}$$

where w_i is the controlled exogenous input for the i-th agent, and $\nu_{ij} = \nu_{ji}$ are the coupling coefficients. We consider a more general case of (possibly nonlinear) agents interacting over a modified protocol,

$$\dot{x}_i = f_i(x_i) + q_i(x_i) \sum_{i\sim j} \nu_{ij} g_{ij}(h_j(x_j) - h_i(x_i)) + B_i w_i, \tag{6.2}$$

where $x_i \in \mathbb{R}$, and $f_i, q_i, g_{ij}, h_i : \mathbb{R} \to \mathbb{R}$ are smooth functions.[1] Examples of systems governed by (6.2), for appropriate choice of functions f_i, q_i, g_{ij}, h_i, include traffic control models [8], neural networks [51], and the Kuramoto model for synchronizing oscillators [43]. We let f, q, g, h denote the stacked versions of f_i, q_i, g_{ij}, h_i.

In the model (6.1), the standard assumption is that only certain agents can be controlled using the exogenous input w_i (i.e., $B_i = 0$ is possible), and one can observe the outputs of only certain agents. For this chapter, we assume that the exogenous output w_i can be added to all agents, and that the output of all agents can be observed. For our case, we can assume without loss of generality that $B_i = 1$. We shall also denote the matrix of the coupling coefficients ν_{ij} as $N = \mathrm{diag}(\cdots, \nu_{ij}, \cdots)$.

We note that the system (6.2) is a diffusively-coupled network, where the agents and the controllers are given by

$$\Sigma_i : \begin{cases} \dot{x}_i &= f_i(x_i) + q_i(x_i)u_i + w_i \\ y_i &= h_i(x_i) \end{cases} \quad , \Pi_{ij} : \ \mu_{ij} = \nu_{ij} g_{ij}(\zeta_{ij}) \ , \tag{6.3}$$

and the network is connected using the diffusive coupling $\zeta = \mathcal{E}_{\mathcal{G}}^{\top} y$ and $u = -\mathcal{E}_{\mathcal{G}} \mu$. We would like to use the network-optimization framework to establish network identification results. We make the following assumptions on the agents and controllers, allowing us to use network optimization framework presented. With this model, we will often write the closed-loop as $(\mathcal{G}_\nu, \Sigma, g)$.

Assumption 6.1. *The systems Σ_i, for all $i \in \mathbb{V}$, are MEIP (See Proposition 5.1). Furthermore, the controllers Π_e, for all $e \in \mathbb{E}$, are output-strictly MEIP. Moreover, the weights ν_{ij} are all positive.*

Assumption 6.2. *The inverse of the steady-state input-output relation for each agent, $k_i^{-1}(\mathrm{y}_i)$, is a $\mathcal{C}^1$ function of y_i. Furthermore, the function $g_{ij}(\zeta_{ij})$ is a twice differentiable function of ζ_{ij}, and that the derivative $\frac{dg_{ij}}{d\zeta_{ij}} > 0$ for all $\zeta_{ij} \in \mathbb{R}$.*

[1] The functions g_{ij} are defined for all pairs, even those absent from the underlying graph. It is often assumed in multi-agent systems that each agent knows how to run a given protocol (i.e., consensus)

© Technion - Israel Institute of Technology, Elyachar Central Library

Assumption 6.2 implies that the integral function $K_i^\star$ associated with k_i^{-1} is smooth and $\nabla K_i^\star = k_i^{-1}$. Assumption 6.1 implies that g_{ij} is strictly monotone ascending. The stronger assumption on g_{ij}, namely Assumption 6.2, is made mainly to avoid heavy technical tools.

We will also consider the special case where the agents and controllers are described by LTI dynamics. For such systems, the input-output relation k_i for each agent is linear and strictly monotone, and so is the function g_{ij}. When Σ_i is an integrator, the input-output relation is given as $\{(0, \mathrm{y}) : \ \mathrm{y} \in \mathbb{R}\}$. In these cases, k_i^{-1} is a linear function over $\mathbb{R}$. In particular, $k_i^{-1}(\mathrm{x}_i) = a_i \mathrm{x}_i$ for some constant $a_i \geq 0$. We can then define the matrix $A = \mathrm{diag}(a_1, \ldots, a_n)$ such that $k^{-1}(\mathrm{x}) = A\mathrm{x}$. Similarly, we denote $g_{ij}(\mathrm{x}_{ij}) = b_{ij}\mathrm{x}_{ij}$, where $b_{ij} > 0$, and $B = \mathrm{diag}(\cdots, b_{ij}, \cdots) > 0$.

We can now formulate two fundamental problems that we will consider.

Problem 6.1 (Network Differentiation)**.** *Consider the network system (G_{ν_1}, Σ, g) of the form* (6.2) *satisfying Assumptions 6.1 and 6.2 with known steady-state input-output relations for the agents and controllers. Design the control inputs $w = w(t)$ so that it is possible to differentiate the network system $(\mathcal{G}_{\nu_1}, \Sigma, g)$ from the network system $(\mathcal{G}_{\nu_2}, \Sigma, g)$, when $\mathcal{G}_{\nu_1} \neq \mathcal{G}_{\nu_2}$.*

Problem 6.2 (Network Identification)**.** *Consider the network system $(\mathcal{G}_\nu, \Sigma, g)$ of the form* (6.2) *satisfying Assumptions 6.1 and 6.2 with known steady-state input-output relations for the agents and controllers, but unknown network structure $\mathcal{G}$ and coupling coefficients $\{\nu_e\}$. Design the control input $w = w(t)$ such that, together with the output measurements of the network, it is possible to reconstruct the graph $\mathcal{G}$ and the coupling coefficients $\{\nu_e\}$.*

The network optimization framework requires the exogenous inputs to be constant signals. Thus, we will consider a constant w, and denote it as w. We can write an equation connecting the steady-state output y to the constant exogenous input w.

Proposition 6.1. *Suppose that the system $(\mathcal{G}_\nu, \Sigma, g)$ is run with the constant exogenous input* w *. Suppose that the agents are MEIP and the controllers are output-strictly MEIP, or vice versa. If k is the steady-state input-output relation for the agents, then the output y of the system converges to a steady-state* y *satisfying the following equation,*

$$\mathrm{w} = k^{-1}(\mathrm{y}) + \mathcal{E}_\mathcal{G} N g(\mathcal{E}_\mathcal{G}^\top \mathrm{y}). \tag{6.4}$$

Proof. Immediately from Proposition 2.2 and Theorem 1.2, where we use $\gamma(\zeta) = Ng(\zeta)$ and $k^{-1}(\mathrm{y}) = \mathrm{u}+\mathrm{w}$. We note that this is an equation and not an inclusion due to Assumption 6.2. □

In practice, the value of w is known, and the value of y can be measured. We note that the steady-state output y depends not only on the steady-state input w, but also on the incidence matrix $\mathcal{E}_\mathcal{G}$ and the weights $N = \mathrm{diag}(\nu_{ij})$, as seen by equation (6.4). We wish to manipulate this relation to achieve network

© Technion - Israel Institute of Technology, Elyachar Central Library

differentiation and identification. For the remainder of this section, we assume that the graph is unweighted, i.e. that the weights ν_{ij} are all equal to 1 [100]. We note that if the values of ν_{ij} are known, then we can "absorb" them inside the functions g_{ij}, effectively yielding a diffusively-coupled network with an unweighted graph.

We denote the solution of (6.4) as $\mathrm{y}_{\mathcal{G}}$. Furthermore, we note that (6.4) is graph dependent, and that the steady-state output $\mathrm{y}_{\mathcal{G}}$ can be measured from the network (as the network converges to a steady-state by Proposition 6.1). The idea now is to choose the bias vectors w wisely so that different graphs will have different steady-state outputs.

Definition 6.1. *Consider a closed-loop system of the form* (6.2) *satisfying Assumptions 6.1 and 6.2, where the controllers g_{ij} have been determined for all possible pairs $\{i,j\}$. Let $\mathfrak{G}$ be a collection of graphs on n vertices. A vector $\mathrm{w} \in \mathbb{R}^n$ is called a* $\mathfrak{G}$-indication vector *if for any two graphs $\mathcal{G},\mathcal{H} \in \mathfrak{G}$ with $\mathcal{G} \neq \mathcal{H}$, the steady-state output of $(\mathcal{G},\Sigma,g)$ is different from the steady-state output of $(\mathcal{H},\Sigma,g)$. In other words, one has that $\mathrm{y}_{\mathcal{G}} \neq \mathrm{y}_{\mathcal{H}}$.*

Given an indication vector w, we can quantify how much it can differentiate between different graphs. We do so with the following definition.

Definition 6.2. *The* separation index *of* w*, denoted $\varepsilon = \varepsilon(\mathrm{w})$, is defined as the minimal distance between $\mathrm{y}_{\mathcal{G}}$ and $\mathrm{y}_{\mathcal{H}}$ where $\mathcal{G} \neq \mathcal{H}$, i.e., $\varepsilon = \min_{\mathcal{G}\neq\mathcal{H}} \|\mathrm{y}_{\mathcal{G}} - \mathrm{y}_{\mathcal{H}}\|$, where the minimization is over all graphs in $\mathfrak{G}$.*

Remark 6.1. *The separation index ε acts as a bound on the error we can tolerate when computing the steady-state output of the closed-loop system. This error can be comprised of both numerical errors, as well as errors arising from early termination of the system (i.e., before reaching steady-state). Indeed, suppose we want to differentiate between $\mathcal{G},\mathcal{H}$, where we know that $\|\mathrm{y}_{\mathcal{G}} - \mathrm{y}_{\mathcal{H}}\| \geq \varepsilon$. Suppose we have the terminal output* y *of the closed-loop system for either $\mathcal{G}$ or $\mathcal{H}$. By the triangle inequality, if $\|\mathrm{y} - \mathrm{y}_{\mathcal{G}}\| < 0.5\varepsilon$ then $\|\mathrm{y} - \mathrm{y}_{\mathcal{H}}\| \geq 0.5\varepsilon$ and vice versa. Thus, if we know that the sum of the numerical and early termination errors is less than 0.5ε, we can correctly choose the underlying graph by choosing which of $\mathrm{y}_{\mathcal{G}}$ and $\mathrm{y}_{\mathcal{H}}$ is closer to* y*.*

Our goal is to construct a $\mathfrak{G}$-indication vector, as we can use it to differentiate between any two diffusively-coupled systems of the form $\{(\mathcal{G},\Sigma,g)\}_{\mathcal{G}\in\mathfrak{G}}$. We will first focus on a solution for general networks, relying on randomization. We will later give additional construction methods for LTI systems using various algebraic methods. In both cases, we'll be able to give stronger results for LTI systems, but will require more analysis to do so. Assume for now that the agents and controllers are LTI. We can now restate (6.4) in a new manner, involving linear equations. The following result will be useful in the analysis.

Proposition 6.2. *If $A \neq 0$, then for any connected graph $\mathcal{G}$, the matrix $S = A + \mathcal{E}_{\mathcal{G}} N B \mathcal{E}_{\mathcal{G}}^{\top}$ is invertible.*

© Technion - Israel Institute of Technology, Elyachar Central Library

Proof. Note that $\mathcal{E}_{\mathcal{G}} N B \mathcal{E}_{\mathcal{G}}^{\top} \geq 0$ and that $A \geq 0$, implying that $S \geq 0$. Furthermore, the kernel of $\mathcal{E}_{\mathcal{G}} N B \mathcal{E}_{\mathcal{G}}^{\top}$ is equal to $\mathrm{span}(\mathbb{1})$. But we have that $\mathbb{1}^{\top} A \mathbb{1} = \sum_{i=1}^{|\mathbb{V}|} a_i > 0$, completing the proof. □

Proposition 6.2 gives an explicit form for (6.4) for the case $A \neq 0$ by inverting the matrix in question,

$$\mathrm{y}_{\mathcal{G}} = (A + \mathcal{E}_{\mathcal{G}} N B \mathcal{E}_{\mathcal{G}}^{\top})^{-1} \mathrm{w} = X_{\mathcal{G}, A \neq 0} \mathrm{w}.$$

If $A = 0$, however, we note that for any $\mathcal{G} \in \mathfrak{G}$, the linear operator $\mathcal{E}_{\mathcal{G}} N B \mathcal{E}_{\mathcal{G}}^{\top}$ preserves $\mathrm{Im}(\mathcal{E}_{\mathcal{G}}) = \mathbb{1}^{\perp}$, and moreover, it is invertible when restricted to it. Thus, we denote the restriction of $\mathcal{E}_{\mathcal{G}} N B \mathcal{E}_{\mathcal{G}}^{\top}$ on $\mathbb{1}^{\perp}$ by $Y_{\mathcal{G}}$, and obtain:

$$\mathrm{y}_{\mathcal{G}} = Y_{\mathcal{G}} \mathrm{Proj}_{\mathbb{1}^{\perp}} \mathrm{w} = X_{\mathcal{G}, A=0} \mathrm{w}.$$

These linear relations between the steady-state output $\mathrm{y}_{\mathcal{G}}$ and the constant exogenous input w give an equivalent, easier definition of indication vectors:

Corollary 6.1. *For LTI agents and controllers, and for a vector* $\mathrm{w} \in \mathbb{R}^n$*, the following statements hold:*

i) If $A \neq 0$, w *is a* $\mathfrak{G}$*-indication vector if and only if for any two different graphs* $\mathcal{G}, \mathcal{H} \in \mathfrak{G}$*, we have* $X_{\mathcal{G}, A \neq 0} \mathrm{w} \neq X_{\mathcal{H}, A \neq 0} \mathrm{w}$.

ii) If $A = 0$, w *is a* $\mathfrak{G}$*-indication vector if and only if any two different graphs* $\mathcal{G}, \mathcal{H} \in \mathfrak{G}$*, we have* $X_{\mathcal{G}, A=0} \mathrm{w} \neq X_{\mathcal{H}, A=0} \mathrm{w}$.

We will omit $A = 0$ and $A \neq 0$ and use $X_{\mathcal{G}}$ for notational simplicity. We first note the following interesting property of $X_{\mathcal{G}}$.

Proposition 6.3. *If* $\mathcal{G} \neq \mathcal{H}$ *then* $X_{\mathcal{G}} \neq X_{\mathcal{H}}$.

Proof. Suppose first that $A \neq 0$. We can reconstruct the weighted graph Laplacian $\mathcal{E}_{\mathcal{G}} N B \mathcal{E}_{\mathcal{G}}^{\top}$ from $X_{\mathcal{G}}$ using the relation $\mathcal{E}_{\mathcal{G}} N B \mathcal{E}_{\mathcal{G}}^{\top} = -A + X_{\mathcal{G}}^{-1}$, thus $X_{\mathcal{G}} = X_{\mathcal{H}}$ implies $\mathcal{G} = \mathcal{H}$, as $B, N > 0$. If $A = 0$, we note that $Y_{\mathcal{G}} = -X_{\mathcal{G}}^{-1}$ on the set $\mathbb{1}^{\perp}$. This determines the graph Laplacian, as it is the projection of the weighted graph Laplacian on $\mathrm{span}(\mathbb{1})^{\perp} = \ker(\mathcal{E}_{\mathcal{G}} N B \mathcal{E}_{\mathcal{G}}^{\top})^{\perp}$. □

Remark 6.2. *In the case of LTI agents and controllers, Corollary 6.1 implies that the relation between* $\mathrm{y}_{\mathcal{G}}$ *and* w *is linear. Thus, for any constant* $\beta > 0$*, the separation index satisfies* $\varepsilon(\beta \mathrm{w}) = \beta \varepsilon(\mathrm{w})$.

We now give a first method for constructing indication vectors, namely through randomization.

© Technion - Israel Institute of Technology, Elyachar Central Library

6.2.2 Constructing Indication Vectors Using Randomization

Our first approach is to construct the indication vectors via randomization. We claim that random vectors $\mathrm{w} \in \mathbb{R}^n$ are indication vectors with probability 1.

Theorem 6.1. *Let $\mathbb{P}$ be any absolutely continuous probability distribution on $\mathbb{R}^n$, and let $\mathfrak{G}$ be the collection of all graphs over n nodes. Then the following equation holds:*

$$\mathbb{P}(\mathrm{w} \text{ is a } \mathfrak{G}\text{-indication vector}) = 1.$$

In order to prove the Theorem, we need the following adaptation of the implicit function theorem:

Lemma 6.1. *Let $F : \mathbb{R}^n \to \mathbb{R}^m$ be a smooth function, and consider the set $Z = \{x : \ F(x) = \mathbf{0}\}$. Suppose that the differential $\nabla F(x)$ is not the zero matrix at any point $x \in Z$. Then Z is a zero-measure set.*

In order to streamline the proofs better, we delay the proof of the lemma to the end of the subsection. We now prove Theorem 6.1:

Proof. Recall that w is not an indication vector if and only if there are two graphs $\mathcal{G}_1, \mathcal{G}_2$ and a vector $\mathrm{y} \in \mathbb{R}^{|\mathbb{V}|}$ such that

$$k^{-1}(\mathrm{y}) + \mathcal{E}_{\mathcal{G}_i} N g(\mathcal{E}_{\mathcal{G}_i}^\top \mathrm{y}) = \mathrm{w}, \ i = 1, 2.$$

Subtracting one equation from the other gives

$$\mathcal{E}_{\mathcal{G}_1} N g(\mathcal{E}_{\mathcal{G}_1}^\top \mathrm{y}) - \mathcal{E}_{\mathcal{G}_2} N g(\mathcal{E}_{\mathcal{G}_2}^\top \mathrm{y}) = \mathbf{0}. \tag{6.5}$$

For each $\mathcal{G}_1, \mathcal{G}_2$, the collection of solutions to (6.5) forms a set, and note that w is an indication vector if and only if the solutions y are not in any of these sets. Define

$$F(y) = \mathcal{E}_{\mathcal{G}_1} N g(\mathcal{E}_{\mathcal{G}_1}^\top \mathrm{y}) - \mathcal{E}_{\mathcal{G}_2} N g(\mathcal{E}_{\mathcal{G}_2}^\top \mathrm{y}),$$

so that $F : \mathbb{R}^n \to \mathbb{R}^n$ is a smooth function. Its differential is given by

$$\nabla F(y) = \mathcal{E}_{\mathcal{G}_1} N \nabla g(\mathcal{E}_{\mathcal{G}_1}^\top \mathrm{y}) \mathcal{E}_{\mathcal{G}_1}^\top - \mathcal{E}_{\mathcal{G}_2} N \nabla g(\mathcal{E}_{\mathcal{G}_2}^\top \mathrm{y}) \mathcal{E}_{\mathcal{G}_2}^\top,$$

where $\nabla g = \mathrm{diag}(\frac{dg_{ij}}{d\zeta_{ij}})$ is the derivative of g. Because $\frac{dg_{ij}}{d\zeta_{ij}} > 0$ by Assumption 6.2, $\nabla F(y)$ is the difference of two weighted graph Laplacians, with underlying graphs $\mathcal{G}_1, \mathcal{G}_2$ and positive weights. Thus ∇F never vanishes, and Lemma 6.1 implies that the solutions of (6.5) form a zero measure set. Thus w is an indication vector if and only if the solutions y are not in the finite union of the zero measure sets defined by (6.5), i.e., a zero-measure set.

The mapping between w and y, $\mathrm{w} = k^{-1}(\mathrm{y}) + \mathcal{E}_{\mathcal{G}} N g(\mathcal{E}^\top \mathrm{y}) = G(\mathrm{y})$, is smooth and strictly cyclically monotone, meaning that it is the gradient of a strictly convex and smooth function. Thus, the inverse function $\mathrm{y} = G^{-1}(\mathrm{w})$ is a smooth and strictly cyclically convex function, as the gradient of the dual function,

© Technion - Israel Institute of Technology, Elyachar Central Library

which is also strictly convex and smooth (See Appendix A). This implies that G^{-1} is absolutely continuous [86], sending zero measure sets to zero measure sets. In turn, the set that y has to avoid (for w to be an indication vector) is zero-measure, meaning that the corresponding set that w has to avoid is also zero measure. But $\mathbb{P}$ is a absolutely continuous probability measure, and thus the probability of the zero-measure set that w has to avoid is zero. This completes the proof. □

Remark 6.3. *The proof above also works when we do not know that $\nu_{ij} = 1$ (i.e., that $N = \mathrm{Id}$), in the sense that for any two weighted graphs $\mathcal{G}_{1,\nu_1}, \mathcal{G}_{2,\nu_2}$,* w *differentiates them with probability* 1. *Indeed, the derivative of $F(y) = \mathcal{E}_{\mathcal{G}_1} N_1 g(\mathcal{E}_{\mathcal{G}_1}^\top \mathrm{y}) - \mathcal{E}_{\mathcal{G}_2} N_2 g(\mathcal{E}_{\mathcal{G}_2}^\top \mathrm{y})$, which is equal to $\nabla F(y) = \mathcal{E}_{\mathcal{G}_1} N_1 \nabla g(\mathcal{E}_{\mathcal{G}_1}^\top \mathrm{y}) \mathcal{E}_{\mathcal{G}_1}^\top - \mathcal{E}_{\mathcal{G}_2} N_2 \nabla g(\mathcal{E}_{\mathcal{G}_2}^\top \mathrm{y}) \mathcal{E}_{\mathcal{G}_2}^\top$, is still the difference of two weighted graph Laplacians, thus non-zero. However, we cannot use the union bound to show that* w*, with probability 1, differentiates any pair of weighted graphs, as the collection of weighted graphs is uncountable.*

Thus, we can find an indication vector by randomly sampling any absolutely continuous probability measure on $\mathbb{R}^n$, e.g. a random Gaussian vector. This method works as long as Assumptions 6.1 and 6.2 hold, but can produce stronger results when considering LTI agents and controllers. In particular, we can estimate the separation index of a randomly chosen vector.

Theorem 6.2. *Suppose the agents and controllers are LTI. Furthermore, suppose that* w *is sampled according to the standard Gaussian probability measure $\mathbb{P}$ on $\mathbb{R}^n$. Define $\beta = \min\{a_1, ..., a_n\}$ if $A \neq 0$, and $\beta = \min_{i,j}\{b_{ij}\}/\binom{n}{2}$ otherwise. Then for any $\delta > 0$, the separation index $\varepsilon = \varepsilon(\mathrm{w})$ satisfies $\delta \leq \varepsilon$ with probability $\geq 1 - 2^{n^2+1}(2\Phi(\delta/2\beta) - 1)$, where Φ is the cumulative distribution function of a standard Gaussian random variable.*

We first prove a lemma:

Lemma 6.2. *For any connected graphs $\mathcal{G}, \mathcal{H}$, $\bar{\sigma}(X_\mathcal{G} - X_\mathcal{H}) \leq C_{\mathcal{G},A} + C_{\mathcal{H},A}$ where we define*

$$C_{\mathcal{G},A} = \begin{cases} \frac{1}{\underline{\sigma}(A + \mathcal{E}_\mathcal{G} N B \mathcal{E}_\mathcal{G}^\top)} & A \neq 0 \\ \frac{1}{\underline{\sigma}(Y_\mathcal{G})} & A = 0 \end{cases}.$$

Proof. First, $\bar{\sigma}(X_\mathcal{G} - X_\mathcal{H}) \leq \bar{\sigma}(X_\mathcal{G}) + \bar{\sigma}(X_\mathcal{H})$ as $\bar{\sigma}(\cdot)$ is a norm on the space of matrices. The proof is now completed using the formula $\bar{\sigma}(C^{-1}) = \frac{1}{\underline{\sigma}(C)}$. □

We can now prove Theorem 6.2.

Proof. The distance between the steady-state output associated with $\mathcal{G}$ and the one associated with $\mathcal{H}$ is $\|(X_\mathcal{G} - X_\mathcal{H})\mathrm{w}\|^2$. We fix some $\mathcal{G}, \mathcal{H}$ and let $F = X_\mathcal{G} - X_\mathcal{H}$. We use the SVD decomposition to write $F^\top F = U^\top D U$ where U is an orthogonal matrix and $D = \mathrm{diag}(\sigma_1^2, ..., \sigma_n^2)$ is a diagonal matrix with entries

© Technion - Israel Institute of Technology, Elyachar Central Library

equal to the squares of the singular values of F. Using the fact that w and U^{-1}w both distribute according to $\mathbb{P}$, we see that:

$$\mathbb{P}(||F\mathrm{w}|| > \delta) = \mathbb{P}(||F\mathrm{w}||^2 > \delta^2) = \mathbb{P}(||FU^\top \mathrm{w}||^2 > \delta^2) =$$
$$\mathbb{P}(\mathrm{w}^\top UF^\top FU^\top \mathrm{w} > \delta^2) = \mathbb{P}(\mathrm{w}^\top D\mathrm{w} > \delta^2) = \mathbb{P}\Big(\sum_{i=1}^{n} \sigma_i^2 \mathrm{w}_i^2 > \delta^2\Big).$$

Now, we note that the entries w_i of w are all standard Gaussian random variables, and that they are independent. Thus, we can estimate:

$$\mathbb{P}\Big(\sum_{i=1}^{n} \sigma_i^2 \mathrm{w}_i^2 > \delta^2\Big) \geq \mathbb{P}\Big(|\mathrm{w}_1| \geq \frac{\delta}{\sigma_1}\Big) = 1 - 2\Phi\Big(-\frac{\delta}{\bar{\sigma}(F)}\Big)$$

Thus, we know that for any pair of graphs $\mathcal{G}, \mathcal{H}$, the chance that $||X_{\mathcal{G}}\mathrm{w} - X_{\mathcal{H}}\mathrm{w}|| > \delta$ is at most $1 - 2\Phi(-\frac{\delta}{\bar{\sigma}(X_{\mathcal{G}} - X_{\mathcal{H}})})$.

Now, we use Lemma 6.2 and the fact that Φ is monotone increasing to bound the probability that $||X_{\mathcal{G}}\mathrm{w} - X_{\mathcal{H}}\mathrm{w}|| > \delta$ from below by $1 - 2\Phi(-\frac{\delta}{C_{\mathcal{G},A} + C_{\mathcal{H},A}})$. We bound each $C_{\mathcal{G},A}$ using Lemma 6.2. First, if $A \neq 0$ then $\underline{\sigma}(A + \mathcal{E}_{\mathcal{G}} NB\mathcal{E}_{\mathcal{G}}^\top) \geq \min\{a_1, \cdots, a_n\}$, and otherwise $\underline{\sigma}(Y_{\mathcal{G}}) = \min_{i,j}\{\nu_{ij} b_{ij}\}\lambda_2(\mathcal{G}) \geq \min_{i,j}\{b_{ij}\}\binom{n}{2}^{-1}$ (See Appendix B), where we use $\nu_{i,j} = 1$. Because there are a total of $2^{\binom{n}{2}}$ possible graphs on n nodes, and a total of $(2^{\binom{n}{2}})^2 \leq 2^{n^2}$ of pairs $\mathcal{G}, \mathcal{H}$ to consider, we can use the union bound to conclude that the chance that $\varepsilon < \delta$ is no more than

$$\sum_{\mathcal{G},\mathcal{H}} \Big(1 - 2\Phi\Big(-\frac{\delta}{\bar{\sigma}(X_{\mathcal{G}} - X_{\mathcal{H}})}\Big)\Big) \leq 2^{n^2+1}\Big(\Phi\Big(\frac{\delta}{2\beta}\Big) - 1\Big).$$

□

Remark 6.4. *Theorem 6.2 and Remark 6.2 give a viable method for assuring that the distance between different steady-state outputs of the system (corresponding to different base graphs) is as large as desired. First, choose a desired degree of security p, which is the probability of the choice to be successful (say $p = .99$). Choose δ so that $p \leq 1 - 2^{n^2}(2\Phi(\delta/2\beta) - 1)$. Now choose* w *randomly according to a standard Gaussian distribution, and multiply it by $1/\delta$.*

To conclude this subsection, we repay our debt and prove Lemma 6.1:

Proof. We denote the coordinates of F by $F = (F_1, ..., F_m)$. We also let $Z_j = \{x : \ F_j(x) = 0\}$ for $j = 1, \ldots, m$, so that $Z \subseteq Z_j$. Let $x = (x_1, \ldots, x_n) \in Z$. We know that $\nabla F(x) \neq \mathbf{0}$, so there's some j such that $\nabla F_j(x) \neq \mathbf{0}$ in a small neighborhood of x. Thus, by the implicit function theorem, we can express one coordinate as a smooth function of the others. For ease of writing, assume without loss of generality that its the n-th coordinate, and let ϕ be the smooth function such that near x, the set Z_j is given by $y_n = \phi(y_1, ..., y_{n-1})$.

© Technion - Israel Institute of Technology, Elyachar Central Library

More precisely, we can find a small cube Q containing x such that the set $Z_j \cap Q$ is given by $y_n = \phi(y_1, ..., y_{n-1})$. If we attempt to compute the volume of $Z_j \cap Q$, we can do so using the integral $\int_Q \mathbb{1}_{y_n=\phi(y_1,...,y_{n-1})}(y)dx$, where $\mathbb{1}_E$ is the indicator function of E. This integral is obviously zero due to Fubini's theorem, and the fact that given $y_1, \ldots, y_{n-1}$, only one y_n can satisfy the equation $y_n = \phi(y_1, ..., y_{n-1})$. Thus the volume of $Z \cap Q$, which is smaller or equal to the volume of $Z_j \cap Q$ as $Z_j \subseteq Z$, nulls.

Up to now, we showed that if $x \in Z$ the there exists some small open set $Q = Q_x$ such that $Z \cap Q_x$ is of measure zero. If we let B_r denote the closed ball of radius r around the origin, then $Z \cap B_r$ is compact, and $\{Q_x : \ x \in Z \cap B_r\}$ is an open cover. Thus $Z \cap B_r$ is contained in the union of finitely many sets of the form $Z \cap Q_x$, each of them having zero measure, implying that $Z \cap B_r$ is zero measure. We note that Z is the countable union of the sets $Z \cap B_1, Z \cap B_2, Z \cap B_3, ...$, meaning that it is the countable union of zero measure sets, hence a zero measure set itself. □

Up to now, we considered a randomization-based method for constructing indication vectors. In the next subsection, we consider other, algebraic-based methods for constructing indication vectors

6.2.3 Constructing Indication Vectors Using Algebraic Methods

For this subsection, we assume that both agents and controllers are LTI. We first construct indication vectors by using computational bases. We can apply this method if the elements of A, B are rational. In this case, the elements of $X_{\mathcal{G}}$ are all rational. The idea is that we can reconstruct the entries of $X_{\mathcal{G}}$ from $X_{\mathcal{G}}\mathrm{w}$ if w is of the form $\mathrm{w} = [1, \mathcal{B}, \mathcal{B}^2, ..., \mathcal{B}^{n-1}]^\top$ for an integer $\mathcal{B}$ large enough. We can consider the following toy example:

Example 6.1. *Suppose that* $C = [\mathrm{a}, \mathrm{b}, \mathrm{c}]$ *is a vector with positive integer entries no greater than* 9. *Take* $\mathrm{w} = [1, 10, 100]^\top$. *Then* $C\mathrm{w} = \mathrm{a} + 10\mathrm{b} + 100\mathrm{c}$ *is a three-digit number, and we can reconstruct* C *by looking at the three digits individually* - a *is the units digit,* b *is the tens digit, and* c *is the hundreds digit.*

We can generalize this to a more general framework:

Theorem 6.3. *Suppose that* A, B *are rational, the denominators of all entries of the matrices* $\{X_{\mathcal{G}}\}_{\mathcal{G}\in\mathfrak{G}}$ *divide* D, *and that the numerator (in absolute value) is no larger than* N. *Let* $\mathcal{B}$ *be any integer larger than* $(2N+1)D$. *Then the vector* $\mathrm{w} = [1, \mathcal{B}, ..., \mathcal{B}^{n-1}]^\top$ *is a* $\mathfrak{G}$*-indication vector.*

Proof. Each element in the product $X_{\mathcal{G}}\mathrm{w}$ corresponds to a single row of $X_{\mathcal{G}}$ multiplied with w, so it's enough to reconstruct each row separately. We take a single row of $X_{\mathcal{G}}$ and mark it as $[\frac{p_1}{q_1}, \cdots, \frac{p_n}{q_n}]^\top$, where $|p_i| \leq N$ and q_i divides D. We let $\mathrm{R} = (X_{\mathcal{G}}\mathrm{w})_i$. Therefore,

$$\mathrm{R} = \begin{bmatrix}\frac{p_1}{q_1} & \cdots & \frac{p_n}{q_n}\end{bmatrix}\mathrm{w} = \frac{p_1}{q_1} + \frac{p_2}{q_2}\mathcal{B} + \cdots + \frac{p_n}{q_n}\mathcal{B}^{n-1}.$$

© Technion - Israel Institute of Technology, Elyachar Central Library

We can define $m_i = \frac{D}{q_1}$, which is an integer no larger than D, and multiply both sides of the equation by D to obtain

$$D\,\mathrm{R} = m_1 p_1 + m_2 p_2 \mathcal{B} + \cdots + m_n p_n \mathcal{B}^{n-1}.$$

Note that $m_i p_i$ is an integer lying between $-ND$ and ND. We can add $\sum_{i=0}^{n-1}(ND\mathcal{B}^i)$ to both sides of the equation, leading to

$$D\mathrm{R} + \sum_{i=0}^{n-1}(ND\mathcal{B}^i) = (m_1 p_1 + ND) + \cdots + (m_n p_n + ND)\mathcal{B}^{n-1}.$$

The left hand side is known, and the coefficients in the right hand side are integers between 0 and $2ND$. Thus, writing $D\mathrm{R} + \sum_{i=0}^{n-1}(ND\mathcal{B}^i)$ in base $\mathcal{B}$, the numbers $m_i p_i + ND$ can be computed by looking at the individual digits. Deducting $\sum_{i=0}^{n-1}(ND\mathcal{B}^i)$ and dividing the entries $\frac{p_i}{q_i}$ by D allows one to compute $X_{\mathcal{G}}$. In particular, because $\mathcal{G} \neq \mathcal{H}$ implies $X_{\mathcal{G}} \neq X_{\mathcal{H}}$, we get that w is an indication vector. □

Remark 6.5. *One can choose $\mathcal{B}$ to be either a power of 2 or a power of 10 in the above algorithm. In both cases, writing the number in base $\mathcal{B}$ becomes easy - in base 10, one just uses the decimal representation and batches digits together, and in base 2, one uses the binary representation and batches bits together.*

We now consider a slightly different method for constructing indication vectors. In the above, we assumed that the entries of $X_{\mathcal{G}}$ have a bound to show that $X_{\mathcal{G}}\mathrm{w} = X_{\mathcal{H}}\mathrm{w}$ must imply $X_{\mathcal{G}} = X_{\mathcal{H}}$, by writing both sides in an appropriate computational basis and comparing them. We can take it a step further, by using algebraic field theory (see Appendix E). As before, we motivate the idea by an example:

Example 6.2. *Let $C_1 = [\mathrm{a}_1, \mathrm{b}_1, \mathrm{c}_1]$ and $C_2 = [\mathrm{a}_2, \mathrm{b}_2, \mathrm{c}_2]$ be any two vectors with rational entries. Take $\mathrm{w} = [1, \pi, \pi^2]^\top$, and assume that $C_1\mathrm{w} = C_2\mathrm{w}$. We get the equation $(\mathrm{a}_1 - \mathrm{a}_2) + (\mathrm{b}_1 - \mathrm{b}_2)\pi + (\mathrm{c}_1 - \mathrm{c}_2)\pi^2 = 0$. We know that the number π is transcendental, meaning that for any polynomial $p(x)$ with rational coefficients, $p(\pi) = 0$ implies that $p \equiv 0$. Thus $\mathrm{a}_1 = \mathrm{a}_2, \mathrm{b}_1 = \mathrm{b}_2, \mathrm{c}_1 = \mathrm{c}_2$ and $C_1 = C_2$.*

Again, we can generalize this idea for a much larger class of systems:

Theorem 6.4. *Let $\mathbb{K} \supseteq \mathbb{F}$ be a field extension, where $\mathbb{F}$ is any field containing the rationals and $\mathbb{K} \subseteq \mathbb{R}$. Suppose that a_i, b_{ij} are in $\mathbb{F}$, and let $v_1, ..., v_n \in \mathbb{K}$ be linearly independent over $\mathbb{F}$. Then $\mathrm{w} = [v_1, ..., v_n]^\top$ is a $\mathfrak{G}$-indication vector.*

Proof. By construction, the entries of $X_{\mathcal{G}}$ lie in $\mathbb{F}$, as A, B have entries in $\mathbb{F}$ and $\mathcal{E}_{\mathcal{G}}$ has rational entries. However, by the definition of linear independence, we know that the map $\mathbb{F}^n \to \mathbb{K}$ sending the vector m to $\mathrm{m}^\top\mathrm{w}$ is one-to-one. In turn, the map $\mathbb{F}^{n\times n} \to \mathbb{K}^n$ sending the matrix M to $M\mathrm{w}$ is also one-to-one, as we can treat each row separately. Thus if $X_{\mathcal{G}}$ and $X_{\mathcal{H}}$ are different, so is their product with w. This proves that w is a $\mathfrak{G}$-indication vector. □

© Technion - Israel Institute of Technology, Elyachar Central Library

Corollary 6.2. *The vector* $\mathrm{w} = [1, \alpha, \alpha^2, ..., \alpha^{n-1}]^\top$ *is a* $\mathfrak{G}$*-indication vector, so long that* α *is transcendental and the entries of* A *are rational.*

The reader is referred to Appendix E for examples of transcendental numbers.

Corollary 6.3. *Suppose the entries of* A *are rational. Let* $p_1, ..., p_n$ *be any prime numbers. Then* $v = [1, \sqrt{p_1}, ..., \sqrt{p_{n-1}}]^\top$ *is a* $\mathfrak{G}$*-indication vector.*

Proof. The first corollary follows from the definition of a transcendental number. The second corollary follows from the square roots of the primes being linearly independent over $\mathbb{Q}$ (see Appendix E). □

We conclude this section with an example of network differentiation using randomly generated indication vectors.

6.2.3.1 Example of Network Differentiation

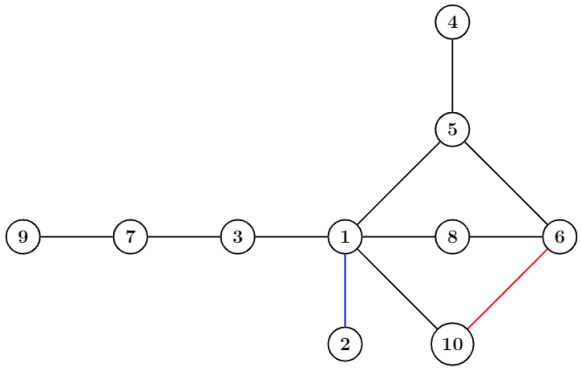
We consider a continuous neural network, as appearing in [130], on n neurons of one species. The governing ODE has the form,

$$\dot{V}_i = -\frac{1}{\tau_i} V_i + b_i \sum_{j \sim i} (\tanh(V_j) - \tanh(V_i)) + w_i, \tag{6.6}$$

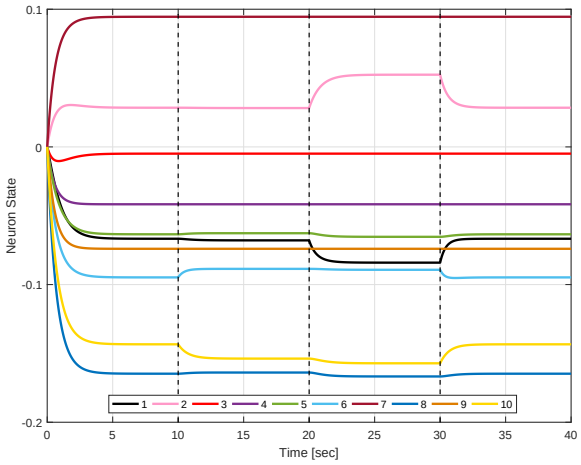
where V_i is the voltage on the i-th neuron, $\tau_i > 0$ is the self-correlation time of the neurons, b_i is a coupling coefficient, and the external input w_i is any other input current to neuron i. We run the system with $n = 10$ neurons. The correlation times were chosen log-uniformly between 0.5_{sec} and 1_{sec}, and we choose $b_i = 0.1$ for all i. In this case, the agents are MEIP and the controllers are output-strictly MEIP.

We choose an indication vector as in the proof of Proposition 6.1, and run the system with the original underlying graph, showing in Figure 6.1(a). The output of the system can be seen in Figure 6.1(b). We first run the system for 10 seconds, which are enough for convergence. After 10 seconds, the red edge in Figure 6.1(a) gets cut off. We can see that the output of agent #6 (in light blue) and agent #10 (in yellow) change meaningfully, so we are able to detect the change in the underlying graph. After ten more seconds, another edge gets removed from the graph, this time the blue one. We can see that again the outputs of two agents, #1 (in black) and #2 (in pink), are changed by a measurable amount, allowing to detect the second change in the underlying graph, which is now unconnected. Finally, after a total of 20 seconds, we reintroduce both of the removed edges, see that the system has returned to its original steady-state.

To conclude this section, we presented numerous ways to choose a constant exogenous input to differentiate between two systems with identical agents and controllers, but different underlying graphs. In the next section, we'll tackle the problem of reconstructing the (weighted) graph from measurements.



(a) The interaction graphs simulated in the case study. The red edge is cut after 10 seconds, and the blue edge is cut after 20 seconds.



(b) Trajectories of the neural network (6.6) with changes in the underlying network.

Figure 6.1: Network differentiation in neural networks.

6.3 Network Identification with Minimal Time Complexity

The approaches made in the previous section can be used to solve the network identification problem using the notion of look-up tables. Look-up tables are tables comprising of two columns, one called the key and the other called the value, that act like oracles and are designed to decrease runtime computations. The key is usually easy to come by, and the value is usually harder to find. Examples of look-up tables include mathematical tables, like logarithm tables

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

and sine tables. Other examples include phone books and other databases like hospital or police records.

Proposition 6.4. *Let $(\mathcal{G}, \Sigma, g)$ be a network system of the form* (6.2) *satisfying Assumptions 6.1 and 6.2. Then for any indication vector* w*, there exists an algorithm solving Problem 6.2 using only a single exogenous output, namely* w*.*

Proof. Let $\mathfrak{G}$ be the collection of all graphs on n vertices. We construct a $\mathfrak{G}$-indication vector w using Theorem 6.1. Before running the system, we build a lookup table with keys being graphs $\mathcal{H} \in \mathfrak{G}$, and values being the outputs $\mathrm{y}_{\mathcal{H}}$, which can be computed by (6.4). Now, run the closed-loop system with the input w. By definition of a $\mathfrak{G}$-indication vector, we know that the steady-state output y of closed-loop system completely classifies the underlying graph $\mathcal{G}$, i.e., different underlying graphs give rise to different steady-state outputs. We can now reconstruct the graph $\mathcal{G}$ by comparing y to the values of the look-up table, finding the graph $\mathcal{H}$ minimizing $\|\mathrm{y} - \mathrm{y}_{\mathcal{H}}\|$. Because w can differentiate the systems $(\mathcal{G}, \Sigma, g)$ and $(\mathcal{H}, \Sigma, g)$ if $\mathcal{G} \neq \mathcal{H}$, we must have that $\mathcal{G} = \mathcal{H}$. □

Remark 6.6. *In the proof above, we assumed that the closed-loop system is run until the output converges. However, in practice, both numerical errors and early termination errors give us a skewed value of the true terminal output of the closed-loop system, as was discussed in Remark 6.1. In the algorithm presented above, we can tolerate an error of up to* $0.5\varepsilon(\mathrm{w})$ *in the value of* y*.*

We should note that in order to implement the network detection scheme in the proof of Proposition 6.4, we need an observer with access to the look-up table, the output of all of the agents, and the input w. The size of the look-up table increases rapidly with the number of nodes if we don't assume anything about the underlying graph. One should note that the computation can be done offline, and that it can be completely parallelized - we are just comparing the entries of the table to the measured output. Furthermore, if we add additional assumptions on the graph (e.g., the underlying graph is a subgraph of some known graph), the size of the look-up table drops significantly. However, in the worst case, we are still dealing with super-exponential time complexity, implying this method is not useful in practice. Another approach tries to apply another method of constructing indication vectors, which also gives a method of reconstruction:

Theorem 6.5. *Let $(\mathcal{G}, \Sigma, g)$ be a network system of the form* (6.2) *satisfying Assumptions 6.1 and 6.2, consisting of LTI agents and controllers, so that the matrices A, B, N have rational entries. Suppose furthermore that the matrix N is known. Then there exists a distributed $O(n^{\omega_1})$ time algorithm solving Problem 6.2. It requires to run the system only once, with a specific constant exogenous input* w*.*

The number ω_1 is defined as the matrix inversion exponent for positive-definite matrices, and satisfies $2 \leq \omega_1 \leq \omega < 3$, where ω is the matrix multiplication exponent. See Appendix D for more on complexity of matrix multiplication and inversion.

© Technion - Israel Institute of Technology, Elyachar Central Library

Remark 6.7. *In the case of LTI agents and controllers, finding the graph $\mathcal{G}$ is roughly equivalent to finding $X_{\mathcal{G}}$. The distributive nature of the algorithm is manifested in the fact that the i-th row of $X_{\mathcal{G}}$ is computed solely from the steady-state output of the i-th agent.*

Proof. We pick an indication vector using the method of Theorem 6.3. The proof of Theorem 6.3 gives an easy way to reconstruct $X_{\mathcal{G}}$'s i-th row from the output of the i-th agent, taking $O(n)$ time. Doing this for all agents takes $O(n^2)$ times, and gives us $X_{\mathcal{G}}$. Afterward, we can reconstruct the graph Laplacian using the formula $\mathcal{E}_{\mathcal{G}} N B \mathcal{E}_{\mathcal{G}}^{\top} = -A - X_{\mathcal{G}}^{-1}$ in $O(n^{\omega_1})$ time, and then find the underlying graph by looking at the non-zero off-diagonal entries of it, completing the proof. □

This algorithm looks better, as it only takes a sub-cubic number of basic operations. However, in practice, it can be numerically unstable, as it deals with very large numbers, whose smaller digits can be inaccurate due to limited precision. This can be negated by using integer-based representation (instead of floating-point types), but will still be inefficient. We note that the algorithm consists of two parts. The first, in which we use the measurements to build the matrix $X_{\mathcal{G}}$, and the seconds, where we use it to reconstruct $\mathcal{G}$. The problematic part is the first, so we try and find a different approach to build the linear connection between the input and output. Moreover, even if we don't have LTI agents and controllers, we might still be able to linearize the relation and apply the algorithm. We start with LTI agents and controllers

6.3.1 An Algorithm for LTI Agents and Controllers

As before, suppose that our agents and controllers are LTI, i.e.,

$$\Sigma_i : \begin{cases} \dot{x}_i &= -\varsigma_i x_i + u_i + w_i \\ y_i &= \varrho_i x_i \end{cases}, \Pi_{ij} : \ \mu_{ij} = \nu_{ij} b_{ij} \zeta_{ij} \ , \tag{6.7}$$

for some $b_{ij}, \varrho_i > 0$ and $\varsigma_i \geq 0$. In this case, the steady-state input-output relations are $\gamma(\zeta) = B\zeta$ and $k^{-1}(\mathrm{y}) = A\mathrm{y}$, where $B = \mathrm{diag}(\cdots, b_{ij}, \cdots)$ and $A = \mathrm{diag}(\cdots, \varsigma_i/\varrho_i, \cdots)$. Thus, (6.4) takes the same form as before,

$$\mathrm{w} = A\mathrm{y} + \mathcal{E}_{\mathcal{G}} N B \mathcal{E}_{\mathcal{G}}^{\top} \mathrm{y}. \tag{6.8}$$

As before, the matrices A, B are known, but the matrices $\mathcal{E}_{\mathcal{G}}$ and N are not. We denote the unknown matrix $\mathcal{E}_{\mathcal{G}} N B \mathcal{E}_{\mathcal{G}}^{\top}$ by L, and the connecting matrix $A + \mathcal{E}_{\mathcal{G}} N B \mathcal{E}_{\mathcal{G}}^{\top}$ by M. We note that $M = A + L$, that $M = X_{\mathcal{G}}^{-1}$, and that L is a weighted Laplacian. Hence, we can reconstruct the graph $\mathcal{G}$ and the coupling coefficient matrix N by looking at the off-diagonal entries of L, or of M, as $A = M - L$ is diagonal. Indeed,

Proposition 6.5. *Suppose the matrix $M = A + \mathcal{E}_{\mathcal{G}} N B \mathcal{E}_{\mathcal{G}}^{\top}$ is known. Then the graph $\mathcal{G}$ and the coupling coefficients ν_{ij} can be exactly reconstructed. The graph $\mathcal{G}$ consists of all edges $e = \{i, j\}$ such that $L_{ij} \neq 0$, and the coupling coefficients are $\nu_{ij} = -\frac{M_{ij}}{b_{ij}}$.*

© Technion - Israel Institute of Technology, Elyachar Central Library

Proof. Directly follows from $b_{ij} > 0$, the fact that L is a weighted Laplacian with graph $\mathcal{G}$ and weights $b_{ij}\nu_{ij}$, and the fact that $M = A + L$, where A is a diagonal matrix. □

Our goal now is to reconstruct the matrix M only from measurements. The equation connecting the steady-state output and the constant exogenous input is $M\mathrm{y} = \mathrm{w}$, where both the vectors y and w are known. Thus, if we have a series of k measurements $\mathrm{y}_1, ..., \mathrm{y}_\mathrm{k}$ and $\mathrm{w}_1, ..., \mathrm{w}_\mathrm{k}$, with $k = n = |\mathbb{V}|$, such that $M\mathrm{y}_\mathrm{i} = \mathrm{w}_i$ for all i, and $\mathrm{y}_1, ..., \mathrm{y}_\mathrm{k}$ are linearly independent, we can reconstruct the matrix M. Namely, we have $M = WY^{-1}$ where Y is the matrix whose columns are $\mathrm{y}_1, ..., \mathrm{y}_\mathrm{n}$, and W is the matrix whose columns are $\mathrm{w}_1, ..., \mathrm{w}_\mathrm{n}$. Thus, we can solve the reconstruction problem by running the system n times with different exogenous inputs and measuring the achieved steady-states, as long as we can assure that the measured steady-state outputs will be linearly independent. We can easily enforce this by considering the properties of M as a linear operator.

Proposition 6.6. *Let $\kappa \neq 0$ be any constant, and consider the following collection of n vectors y_i:*

i) If $A \neq 0$, let $\{\mathrm{w}_i\}_{i=1}^{n}$ be any collection of linearly independent vectors. Let y_i be the solution of (6.8) with $\mathrm{w} = \mathrm{w}_i$.

ii) If $A = 0$, let $\{\mathrm{w}_i\}_{i=1}^{n-1}$ be any set of $n-1$ linearly independent vectors in the space orthogonal to $\mathbb{1}_n$. Let y_i be the solution of (6.8) with $\mathrm{w} = \mathrm{w}_i$, where $\mathrm{y}_i^\top \mathbb{1}_n = 0$ for $i = 1, \ldots, n-1$. Define $\mathrm{y}_n = \kappa \mathbb{1}_n$ with $\kappa \neq 0$.

Then the set $\{\mathrm{y}_1, ..., \mathrm{y}_n\}$ consists of linearly independent vectors.

Proof. Suppose first that $A \neq 0$. In that case, it is known that M is invertible by Proposition 6.2. Thus, M sends linearly independent sets of vectors to linearly independent sets of vectors. Now suppose that $A = 0$. The matrix $M = \mathcal{E}_\mathcal{G} N B \mathcal{E}_G^\top$ is a weighted Laplacian, meaning that the solution y to the equation $\mathrm{w} = M\mathrm{y}$ is not unique in $\mathbb{R}^n$, but only in $\mathrm{span}(\mathbb{1}_n)^\perp$, the space orthogonal to $\mathbb{1}_n$. Moreover, the matrix M preserves the space $\mathrm{span}(\mathbb{1}_n)^\perp$, and the restricted operator $M : \mathrm{span}(\mathbb{1}_n)^\perp \to \mathrm{span}(\mathbb{1}_n)^\perp$ is invertible. Thus, by the same reasoning as above, the vectors $\mathrm{y}_1, ..., \mathrm{y}_{\mathrm{n}-1}$ are linearly independent. Moreover, they are all orthogonal to y_n. implying that the vectors $\mathrm{y}_1, ..., \mathrm{y}_\mathrm{n}$ are linearly independent and completing the proof. □

Proposition 6.6 suggests an algorithm for reconstruction. In the case $A \neq 0$, we choose vectors w_i as in the theorem, run the system with them, measure the corresponding steady-state outputs, and then do a small computation involving matrix inversion, as described in the discussion prior to the proposition. In the case $A = 0$, we note that $M\mathbb{1}_\mathrm{n} = \mathbf{0}$, independent of the values of the weighted graph $\mathcal{G}_\nu$. This urges us to choose $\mathrm{y}_n = \mathbb{1}_n$ and $\mathrm{w}_n = \mathbf{0}$ and repeat the same procedure, this time running the system only $n-1$ times.

However, consider Proposition 6.6 in the case $A = 0$. There, the vectors $\mathrm{y}_1, ..., y_n$ are linearly independent, but the vectors $\mathrm{w}_1 = M\mathrm{y}_1, ..., \mathrm{w}_n = M\mathrm{y}_n$ are

© Technion - Israel Institute of Technology, Elyachar Central Library

not - the last vector is equal to zero. For reasons explained later (see Remark 6.10), we'll want the vectors $\mathrm{w}_1, ..., \mathrm{w}_\mathrm{n}$ to be linearly independent even in the case $A = 0$. To remedy the problem, we instead consider the matrix $M' = \mathcal{E}_\mathcal{G} N B \mathcal{E}_\mathcal{G}^\top + \frac{1}{n}\mathbb{1}_n\mathbb{1}_n^\top$ in the case $A = 0$. Moreover, we observe that $M'\mathbb{1}_n = \mathbb{1}_n$ and note that if y_i is in $\mathrm{span}(\mathbb{1}_n)^\perp$, then $M'\mathrm{y}_i = M\mathrm{y}_i$. Therefore, we choose vectors $\mathrm{w}_i, \mathrm{y}_i$ as in Proposition 6.6 for $i = 1, \ldots, n-1$, and also $\mathrm{y}_n = \mathrm{w}_n = \mathbb{1}_n$. Defining W, Y as above, we note that $M' = WY^{-1}$, and $M = M' - \frac{1}{n}\mathbb{1}_n\mathbb{1}_n^\top$. We will implement this scheme, in which the added term of the form $\frac{1}{n}\mathbb{1}_n\mathbb{1}_n^\top$ will be denoted by Q

Remark 6.8. *As we said, Proposition 6.6 gives a reconstruction scheme - choose any n (or $n-1$) linearly independent vectors in the proper space, run the system n (or $n-1$) times using them as inputs, measure the steady-state outputs, and then use the discussion preceding Proposition 6.6 to compute the graph $\mathcal{G}$ and the weights ν_{ij}. Instead of doing n (or $n-1$) separate experiments in which we run the system with one of the w_i-s, we can use the global asymptotic convergence of the system to use a switching signal. We use an exogenous input $w(t)$ whose value is changed every time the system reaches its steady-state, or ϵ-close to it. See also Remark 6.9 about declaring when a steady-state is reached.*

We conclude this subsection with an algorithm for network reconstruction for LTI agents and controllers, namely Algorithm 4.

Theorem 6.6. *Consider a diffusively coupled system $(\mathcal{G}_\nu, \Sigma, g)$. Suppose that the agents are MEIP, and that the static nonlinearities g_{ij} are output-strictly MEIP. Suppose furthermore that the agents and controllers are all LTI. Then*

i) Algorithm 4 outputs the correct graph and coupling coefficients.

ii) The time complexity of the Algorithm 4 is $O(n^\omega)$, where $\omega \leq 3$ is the matrix multiplication exponent (see Appendix D).

Proof. By Propositions 6.5 and 6.6, in order to prove the first claim, it's enough to show that the vectors w_i chosen are linearly independent, which is obvious.

As for complexity, we go over the different parts and estimate the time complexity required. The first part, before the for-loop, takes $O(n^2)$ time (just to initialize w_i for $i = 1, \cdots, \mathrm{NumOfRuns}$). The first for-loop takes $O(n^2)$ time as well, as each iteration takes $O(n)$ time just to store y_i in memory. The computation between the two for-loops takes $O(n^\omega)$ time. Lastly, the last for-loop also takes $O(n^2)$ time. Since $\omega \geq 2$, the result is obtained. □

Remark 6.9. *Algorithm 4 (as well as Algorithm 5 presented in the next subsection), like the algorithm presented in Theorem 6.5, runs the system with fixed exogenous inputs, and then measures the steady-state output of the closed-loop system. In practice, the exact steady-state output is achieved only asymptotically, which is both unfeasible to run, and forbids the switching input scheme of Remark 6.8. Therefore, we must stop the system and measure the output after a finite amount of time. We are therefore interested in understanding how long to run the system to assure sufficient proximity to the true steady-state output.*

© Technion - Israel Institute of Technology, Elyachar Central Library

Algorithm 4 Network Reconstruction Scheme for LTI agents and controllers

if $A = 0$ **then**
  Choose $\mathrm{w}_i = \mathrm{e}_i - \mathrm{e}_n$ for $i = 1, ..., n-1$.
  Put $\mathrm{y}_n = \mathbb{1}_n$ and $\mathrm{w}_n = \mathbb{1}_n$.
  Put NumOfRuns $= n - 1$.
  Put $Q = \frac{1}{n}\mathbb{1}_n\mathbb{1}_n^\top$.
else
  Choose $\mathrm{w}_i = \mathrm{e}_i$ for $i = 1, ..., n$.
  Put NumOfRuns $= n$ and $Q = 0$.
end if
for $i = 1$ to NumOfRuns **do**
  Update the value of $w(t)$ to w_i.
  Wait for the diffusively coupled network to converge (see Remark 6.9). Measure it's steady-state output and denote it as y_i.
end for
Define the matrix Y as the $n \times n$ matrix having $\mathrm{y}_1, ..., \mathrm{y}_\mathrm{n}$ as its columns.
Define the matrix W as the $n \times n$ matrix having $\mathrm{w}_1, ..., \mathrm{w}_\mathrm{n}$ as its columns.
Define $M' = WY^{-1}$.
Define $M = M' - Q$.
Define an empty graph $\mathcal{H}$ on n nodes.
for $i, j = 1$ to n such that $i \neq j$ **do**
  if $M_{i,j} \neq 0$ and $i \neq j$ **then**
    Add the edge $\{i, j\}$ to the graph $\mathcal{H}$.
    Define $p_{ij} = -\frac{M_{ij}}{b_{ij}}$
  end if
end for
Output the graph $\mathcal{H}$ and the coupling coefficients $\{p_{ij}\}$.

There are many ways to know the desired runtime of the system, or at least some approximation of it. One can use the storage function of the closed-loop system to estimate the distance from steady-state at each point in the run, terminating when the distance from steady-state is small enough. Another solution is to stop running the system when $\dot{y}$ (or $\dot{x}$) is small enough. Other ways to determine the runtime of the system include conducting computer-based simulations, or even intuition based on the physical time constants in the agents' dynamics.

Another method one can use is equilibrium-independent Lyapunov exponents. For LTI systems, if we have a steady-state x_{ss} for the agents, it corresponds to a quadratic storage function, namely of the form $S(x) = \sum_{i=1}^{n} q_i(x_i(t) - (\mathrm{x}_{ss})_i)^2$ for some positive numbers $q_1, ..., q_n$. We can consider the closed-loop system, for which we can write

$$\dot{S} \leq -\sum_{i=1}^{n} \rho_i \Big(h_i(x_i(t)) - h_i((x_{ss})_i) \Big)^2, \tag{6.9}$$

where $\rho_i > 0$ are the output-passivity parameters of the agents (see [23, Theorem 3.4]). The right-hand side is bounded by $-\min_i \left\{ \frac{q_i}{\rho_i \varrho_i} \right\} S(x)$. In other words, we can conclude from Lyapunov's theorem that the closed-loop system always converges to its steady-state exponentially fast with a known exponent, no matter what steady-state it has. More exactly, the storage function decays exponentially fast with exponent $\min_i \left\{ \frac{q_i}{\rho_i \varrho_i} \right\}$, meaning that by sensing the value of the storage function at the beginning of the run, we can compute a bound on how long we need to wait until we are ϵ-close to the steady-state, namely $\log(S(x(0))/\epsilon)$ divided by the exponent. This method can be generalized for other, nonlinear systems as well. Generically speaking, inequalities bounding the measurement function h_i from below using the storage function of the i-th agent can be used to achieve certain convergence rate guarantees, that in turn allow us to establish the needed runtime for the algorithm. We will return to this idea in Chapter 7, where it will be used for fault detection and isolation algorithms.

Remark 6.10. *After running the system, we end up with matrices W, Y and want to compute $M = WY^{-1}$. There is another way to do the same computation, which changes its time complexity from $O(n^\omega)$ to $O(n^{\omega_1})$. Indeed, we consider $M^{-1} = YW^{-1}$. If W is chosen smartly, then the product YW^{-1} can be computed in $O(n^2)$ time instead of $O(n^\omega)$ time. Indeed, this happens if W is diagonal, or more generally, if W is the product of at most $O(n)$ elementary matrices (see [66] for a definition). Indeed, if this is the case, then W^{-1} is also a product of at most $O(n)$ elementary matrices, and taking a product with an elementary matrix only takes $O(n)$ time. In this case, we want to invert the matrix $M^{-1} = YW^{-1}$, which it is positive-definite. Thus we can invert it using $O(n^{\omega_1})$ operations instead of $O(n^\omega)$ operations. We show below that W is the product of at most $O(n)$ elementary matrices.*

Proposition 6.7. *The matrix W constructed in Algorithm 4 (and 5) is the product of no more than $O(n)$ elementary matrices.*

Proof. This is clear for $A \neq 0$, as $W = I$, so we show it for $A = 0$. We run a Gaussian elimination procedure on the matrix W defined by the algorithm. Each row operation corresponds to a multiplication by an elementary matrix, so it suffices to show that the procedure halts after $O(n)$ steps. We'll show it halts after $2(n-1)+1$ steps. Indeed, we first consider the row operations of the form $R_n \to R_n + R_i$ for $i = 1, \cdots, n-1$, namely add row i to row n. These are $n-1$ total row operations, leaving all first $n-1$ rows unaltered, and changing the last row of the matrix to $[0, \cdots, 0, n]$. We now divide the n-th row by n, which is another row operation, altering the last row to $[0, \cdots, 0, 1]$. Lastly, we apply the row operations $R_i \to R_i - R_n$ for $i = 1, \cdots, n-1$. These operations nullify the only nonzero off-diagonal element in each row, achieving an identity matrix. Thus, by applying a total of $(n-1) + 1 + (n-1)$ row operations, we transformed the matrix W to the identity matrix. Thus W is the product of $2n-1$ row operations, completing the proof of the theorem. □

© Technion - Israel Institute of Technology, Elyachar Central Library

Remark 6.11. *We compare Algorithm 4 to the algorithm presented in Theorem 6.5. Both algorithms run roughly with the same time complexity, but Algorithm 4 has two advantages over the one presented in Theorem 6.5. Firstly, it does not assume that the coupling coefficients ν_{ij} are identical, or even known. Secondly, the algorithm in Theorem 6.5 can sometimes be difficult to implement due to the size of the numbers involved in it. In the presented algorithm, however, we trade this problem with inverting a matrix W, which is easily invertible, eliminating numerical instability. It should also be noted that the proposed algorithm is deterministic, unlike the one presented in Theorem 6.5.*

6.3.2 An Algorithm for General MEIP Agents and Controllers Using Linearization

In general, our agents and controllers might not be LTI. However, we can still try and apply the Algorithm 4 using linearization. As we'll see later, we will need another technical assumption, which is slight relaxation of Assumption 6.2:

Assumption 6.3. *For each i, j, k_i^{-1},g_{ij} are continuous monotone functions. Moreover, on any bounded interval, there are at most finitely many points at which k_i^{-1},g_{ij} are not twice differentiable. Lastly, the set of points on which $\frac{dg_{ij}}{d\zeta_{ij}} = 0$ is of measure zero.*

Heading toward linearization, we first run the system with some w_0 and get y_0. We can now linearize the equation $\mathrm{w} = k^{-1}(\mathrm{y}) + \mathcal{E}_{\mathcal{G}} N g(\mathcal{E}_{\mathcal{G}}^{\top}\mathrm{y})$ around y_0. If we input $\mathrm{w} = \mathrm{w}_0 + \delta\mathrm{w}$, then we obtain,

$$\mathrm{w} - k^{-1}(\mathrm{y}) = \mathcal{E}_{\mathcal{G}} N g(\mathcal{E}_{\mathcal{G}}^{\top}\mathrm{y}) \approx \mathcal{E}_{\mathcal{G}} N g(\mathcal{E}_{\mathcal{G}}^{\top}\mathrm{y}_0) + \mathcal{E}_{\mathcal{G}} N \nabla g(\mathcal{E}_{\mathcal{G}}^{\top}\mathrm{y}_0)\mathcal{E}_{\mathcal{G}}^{\top}\delta\mathrm{y}, \tag{6.10}$$

where y is the steady-state output of the network, and $\delta\mathrm{y} = \mathrm{y} - \mathrm{y}_0$. More precisely, we have the following result.

Proposition 6.8. *Suppose that the functions k^{-1}, g are twice differentiable at y_0. Then for any $\delta\mathrm{w}$ small enough, the equation*

$$\mathcal{E}_{\mathcal{G}} N \nabla g(\mathcal{E}_{\mathcal{G}}^{\top}\mathrm{y}_0)\mathcal{E}_{\mathcal{G}}^{\top}\delta\mathrm{y} = \delta\mathrm{w} - k^{-1}(\mathrm{y}) + k^{-1}(\mathrm{y}_0) + O(\|\delta\mathrm{y}\|^2) \tag{6.11}$$

holds, where $\mathrm{y} = \mathrm{y}_0 + \delta\mathrm{y}$.

Proof. Immediately follows from subtracting $\mathrm{w}_0 = k^{-1}(\mathrm{y}_0) + \mathcal{E}_{\mathcal{G}} N g(\mathcal{E}_{G}^{\top}\mathrm{y}_0)$ from $\mathrm{w} = k^{-1}(\mathrm{y}) + \mathcal{E}_{\mathcal{G}} N g(\mathcal{E}_{G}^{\top}\mathrm{y})$ and using Taylor expansion up to first order, where we note that the twice differentiability assumption implies that the error of the first order approximation is $O(\|\delta\mathrm{y}\|^2)$. □

As before, we inject n different signals into the system and measure n output vectors, $\delta\mathrm{y}_1, ..., \delta\mathrm{y}_n$. We can use (6.11) to estimate the value of the matrix $\mathcal{E}_{\mathcal{G}} N \nabla g(\mathcal{E}_{\mathcal{G}}^{\top}\mathrm{y}_0)\mathcal{E}_{\mathcal{G}}^{\top}$ applied on each of $\delta\mathrm{y}_1, ..., \delta\mathrm{y}_n$. As before, we can replace one of these vectors with $\mathbb{1}_n$, as we know that it lies in the kernel of the matrix. If we knew these vectors are linearly independent, we could use the same reconstruction method as in the linear case. Thus we strive to find a method in which $\delta\mathrm{y}_1, \cdots \delta\mathrm{y}_n$ are linearly independent.

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

Theorem 6.7. *Let $\mathbb{P}_{n,0}$ be any absolutely continuous probability measure on $\mathbb{R}^n$, and suppose that we sample w_0 according to $\mathbb{P}_{n,0}$. Let y_0 be a solution to the equation $\mathrm{w}_0 = k^{-1}(\mathrm{y}) + \mathcal{E}Ng(\mathcal{E}^\top \mathrm{y})$. Choose $\kappa \neq 0$ and define vectors $\delta\mathrm{y}_1, ..., \delta\mathrm{y}_{n-1}, \delta\mathrm{y}_n$ in the following way:*

i) If k^{-1} is differentiable at y_0 and $\nabla k^{-1}(\mathrm{y}_0) = \mathbf{0}$, choose $\delta\mathrm{w}_i = \kappa(\mathrm{e}_i - \mathrm{e}_n)$ for $i = 1, \cdots, n-1$. Define y_i as the solution to the equation $\mathrm{w}_0 + \delta\mathrm{w}_i = k^{-1}(\mathrm{y}_i) + \mathcal{E}Ng(\mathcal{E}^\top \mathrm{y}_i)$ and $\delta\mathrm{y}_i = (\mathrm{Id}_n - \frac{1}{n}\mathbb{1}_n\mathbb{1}_n^\top)(\mathrm{y}_i - \mathrm{y}_0)$, for $i = 1 \cdots, n-1$. Also, set $\delta\mathrm{y}_n = \kappa\mathbb{1}_n$.

ii) Otherwise, choose $\delta\mathrm{w}_i = \kappa\mathrm{e}_i$ for $i = 1, \cdots, n$. Define y_i as the solution to the equation $\mathrm{w}_0 + \delta\mathrm{w}_i = k^{-1}(\mathrm{y}_i) + \mathcal{E}Ng(\mathcal{E}^\top \mathrm{y}_i)$ and $\delta\mathrm{y}_i = \mathrm{y}_i - \mathrm{y}_0$.

Suppose Assumptions 6.1 and 6.3 hold. If κ is small enough, then the set $\mathcal{A} = \{\delta\mathrm{y}_1, ..., \delta\mathrm{y}_{n-1}, \delta\mathrm{y}_n\}$ is a basis for $\mathbb{R}^n$.

Before proving Theorem 6.7, we state and prove a lemma.

Lemma 6.3. *Suppose that the same assumptions as in Theorem 6.7 hold. Then for any $i \in \{1, ..., n\}$ and any number $\mathrm{x} \in \mathbb{R}$, the set of all $\mathrm{w} \in \mathbb{R}^n$ such that the solution y to $\mathrm{w} = k^{-1}(\mathrm{y}) + \mathcal{E}_G Ng(\mathcal{E}_\mathcal{G}^\top \mathrm{y})$ satisfies $\mathrm{y_i} = \mathrm{x}$ has measure zero.*

Proof. Consider the map $G : \mathbb{R}^n \to \mathbb{R}^n$ defined by $G(\mathrm{y}) = k^{-1}(\mathrm{y}) + \mathcal{E}_G Ng(\mathcal{E}_\mathcal{G}^\top \mathrm{y})$. The relevant set $\mathcal{S}$ is the image of $\mathcal{R} = \{\mathrm{y} \in \mathbb{R}^n : \mathrm{y_i} = \mathrm{x}\}$ under G. The assumption on k^{-1} implies that it is continuous and piece-wise smooth, hence locally Lipschitz. Thus, G is absolutely continuous, sending zero-measure sets to zero-measure sets. As $\mathcal{R}$ has measure zero, we conclude that $\mathcal{S}$ also has measure zero, which concludes the proof. □

Corollary 6.4. *Under the same assumptions as in Theorem 6.7, the function k^{-1} is twice differentiable at y_0 with probability 1.*

We can now prove Theorem 6.7.

Proof. The idea of the proof is to reduce the theorem to Proposition 6.6 using linearization. By Corollary 6.4, we know that the $\mathbb{P}_{n,0}$-probability that k^{-1} is not twice differentiable at y_0 is zero. Thus, we can assume this scenario does not happen. Under this assumption, we can write the following equation connecting $\delta\mathrm{y}_i$ and $\delta\mathrm{w}_i$,

$$\delta\mathrm{w}_i = \nabla k^{-1}(\mathrm{y}_i)\delta\mathrm{y}_i + \mathcal{E}_\mathcal{G} N \nabla g(\mathcal{E}_\mathcal{G}^\top \mathrm{y}_0)\mathcal{E}_\mathcal{G}^\top \delta\mathrm{y}_i + O(\|\delta\mathrm{y}_i\|^2). \tag{6.12}$$

It follows immediately in the case $\nabla k^{-1}(\mathrm{y}_0) \neq \mathbf{0}$, and uses $\mathcal{E}_\mathcal{G}^\top \mathbb{1}_n = \mathbf{0}$ in the case $\nabla k^{-1}(\mathrm{y}_0) = \mathbf{0}$. Because κ is small, and k^{-1} and g are twice differentiable at y_0 and $\mathcal{E}_\mathcal{G}^\top \mathrm{y}_0$, we can conclude that $\|\delta\mathrm{y}_i\| = O(\|\delta\mathrm{w}_i\|)$. Thus, we can rewrite (6.12) differently:

$$\delta\mathrm{w}_i - O(\|\delta\mathrm{w}_i\|^2) = \left(\nabla k^{-1}(\mathrm{y}_0) + \mathcal{E}_\mathcal{G} N \nabla g(\mathcal{E}_\mathcal{G}^\top \mathrm{y}_0)\mathcal{E}_\mathcal{G}^\top\right) \delta\mathrm{y}_i. \tag{6.13}$$

© Technion - Israel Institute of Technology, Elyachar Central Library

Let us first focus on the case $\nabla k^{-1}(\mathrm{y}_0) \neq \mathbf{0}$. The matrix $\nabla k^{-1}(\mathrm{y}_0) + \mathcal{E}_{\mathcal{G}} N \nabla g(\mathcal{E}_{\mathcal{G}}^\top \mathrm{y}_0)\mathcal{E}_{\mathcal{G}}^\top$ is invertible by Proposition 6.2, meaning that $\mathcal{A}$ is linearly independent if and only if the vectors on the left-hand side of (6.13) are linearly independent. However, these vectors are equal to $\kappa \mathrm{e}_i - \mathrm{z}_i$, for some vectors z_i satisfying $||\mathrm{z}_i|| = O(\kappa^2)$, making them linearly independent for κ small enough, meaning that $\mathcal{A}$ is a basis (with probability 1).

As for the case in which $\nabla k^{-1}(\mathrm{y}_0) = \mathbf{0}$, we note that $\mathcal{E}_{\mathcal{G}} N \nabla g(\mathcal{E}_{\mathcal{G}}^\top \mathrm{y}_0)\mathcal{E}_{\mathcal{G}}^\top$ preserves the space orthogonal to $\mathbb{1}_n$. Moreover, when restricted to that subspace, it is an invertible map. As $\delta\mathrm{y}_1, ..., \delta\mathrm{y}_{n-1}$ are orthogonal to $\delta\mathrm{y}_n = \kappa\mathbb{1}_n$, it's enough to show that the former are linearly independent. As the map $\mathcal{E}_{\mathcal{G}} N \nabla g(\mathcal{E}_{\mathcal{G}}^\top \mathrm{y}_0)\mathcal{E}_{\mathcal{G}}^\top$ is invertible on the space $\mathbb{1}_n^\perp$, this is the same as saying that the vectors on the left hand side of equation (6.13) are linearly independent. However, these vectors are of the form $\kappa(\mathrm{e}_i - \mathrm{e}_n) - O(\kappa^2)$, which are clearly linearly independent if κ is small enough. Thus $\mathcal{A}$ is a basis for $\mathbb{R}^n$ (with probability 1). This concludes the proof. □

Remark 6.12. *In the proof above, we used the fact that if k^{-1} and g are twice differentiable at y_0 and $\mathcal{E}_{\mathcal{G}}^\top \mathrm{y}_0$, then $\delta\mathrm{y}_i = O(\|\delta\mathrm{w}_i\|)$. In particular, the error rate in Proposition 6.8 is $O(\|\delta\mathrm{w}_i\|^2)$. Moreover, we note that Remark 6.8 still holds for nonlinear systems, so we may use switching inputs again.*

We wish to conclude this subsection with a proper description and analysis of the algorithm. The algorithm can be read in Algorithm 5.

Note 6.1. *We changed the query $M_{ij} \neq 0$ from the original algorithm to $M_{ij} < -\varepsilon$ in the augmented algorithm. This is much more robust to the inherent noise in M, arising not only from numerics, but also from the negligence of the unknown quadratic term in* (6.11).

It's clear that this time this is an approximate algorithm, as the quadratic error term will have an effect on the coupling coefficients. However, we can still use it to reconstruct the underlying graphs well, as will be shown later by examples. We can bound the error of algorithm, and determine its time complexity.

Theorem 6.8. *Consider a diffusively coupled system $(\mathcal{G}_\nu, \Sigma, g)$. Suppose that the agents are MEIP, and that the static nonlinearity g is output-strictly MEIP. Moreover, suppose that Assumptions 6.1, and 6.3 all hold. Then:*

i) Let M be the matrix calculated by Algorithm 5, and define

$$\mathcal{M} = \nabla k^{-1}(\mathrm{y}_0) + \mathcal{E}_{\mathcal{G}} N \nabla g(\mathcal{E}_{\mathcal{G}}^\top \mathrm{y}_0)\mathcal{E}_{\mathcal{G}}^\top .$$

Then for any $i, j \in \mathbb{V}$, we have

$$|M_{ij} - \mathcal{M}_{ij}| \leq O\bigg(\sqrt{n}\kappa\Big(1 + \max_{i,j}(\nu_{ij} d_{ij})\lambda_{max}(\mathcal{G})\Big)\bigg),$$

with probability 1, where $\lambda_{max}(\mathcal{G})$ is the maximal eigenvalue of the graph Laplacian of $\mathcal{G}$. Thus, Algorithm 5 outputs an approximation of the graph and coupling coefficients, with probability 1.

© Technion - Israel Institute of Technology, Elyachar Central Library

Algorithm 5 Network Reconstruction Scheme for MEIP agents and controllers

Randomly choose w_0 as a standard Gaussian vector. Change the value of $w(t)$ to w_0.
Wait for the diffusively coupled network to converge (see Remark 6.9). Measure it's steady-state output and denote it as y_0
if $\nabla k^{-1}(\mathrm{y}_0) = \mathbf{0}$ **then**
  Define $\delta \mathrm{w}_i = \kappa(\mathrm{e}_i - \mathrm{e}_n)$ for $i = 1, \cdots, n-1$.
  Put $\delta \mathrm{y}_n = \mathbb{1}_n$ and $\delta \mathrm{w}_n = \kappa \mathbb{1}_n$
  Put NumOfRuns $= n - 1$.
  Put $J = \mathrm{Id}_n - \frac{1}{n}\mathbb{1}_n\mathbb{1}_n^\top$
  Put $Q = \frac{1}{n}\mathbb{1}_n\mathbb{1}_n^\top$
else
  Define $\delta \mathrm{w}_i = \kappa \mathrm{e}_i$ for $i = 1, \cdots, n$.
  Put NumOfRuns $= n$.
  Put $J = \mathrm{Id}_n$
  Put $Q = 0$.
end if
for $i = 1$ to NumOfRuns **do**
  Change the value of $w(t)$ to $\mathrm{w}_0 + \delta \mathrm{w}_i$.
  Wait for the diffusively coupled network to converge (see Remark 6.9). Measure its steady-state output and denote it as y_i.
  Define $\delta \mathrm{y}_i = J(\mathrm{y}_i - \mathrm{y}_0)$.
end for
Define the matrix δY as the $n \times n$ matrix having $\delta \mathrm{y}_1, ..., \delta \mathrm{y}_n$ as its columns.
Define the matrix δW as the $n \times n$ matrix having $\delta \mathrm{w}_1, ..., \delta \mathrm{w}_n$ as its columns.
Compute $M' = \delta W \delta Y^{-1}$.
Compute $M = M' - Q$.
Define an empty graph $\mathcal{H}$ on n nodes.
for $i, j = 1$ to n **do**
  if $M_{i,j} < -\varepsilon$ and $i \neq j$ **then**
    Add the edge $\{i, j\}$ to the graph $\mathcal{H}$.
    Define $p_{ij} = -\frac{M_{ij}}{\frac{dg_{ij}}{d\zeta_{ij}}((y_0)_i - (y_0)_j)}$
  end if
end for
Output the graph $\mathcal{H}$ and the coupling coefficients $\{p_{ij}\}$.

ii) The time complexity of the Algorithm 5 is $O(n^\omega)$.

We first prove the following lemma:

Lemma 6.4. *Let δW be the matrix computed by Algorithm 5. Then the operator norm of δW^{-1} is bounded by $2/\kappa$.*

© Technion - Israel Institute of Technology, Elyachar Central Library

Proof. If $\nabla k^{-1}(\mathrm{y}_0) \neq \mathbf{0}$, then the matrix δW is equal to $\kappa \mathrm{Id}_n$, meaning that its inverse is $\kappa^{-1}\mathrm{Id}_n$, and the result is clear. If $\nabla k^{-1}(\mathrm{y}_0) = \mathbf{0}$, however, then δW is equal to κF, where F's columns are given by $\mathrm{e}_i - \mathrm{e}_n$ for $i = 1, \cdots, n-1$, and $\mathbb{1}_n$. Thus, the inverse of δW is equal to $\kappa^{-1}F^{-1}$, so it's enough to show that the operator norm of F^{-1} is bounded by 2. Consider the Gaussian elimination procedure applied to F, as described in the proof of Proposition 6.7. There, we used row operations to transform F to Id_n. The matrix F^{-1} can be computed by applying the same row operations on Id_n, in the same order. This yields a closed form for F^{-1}.

Indeed, we follow the same process, this time on Id_n. First, we apply the row operations $R_n \to R_n + R_j$ for $j = 1, \cdots, n-1$. This leaves all rows but the last unaltered, and the last row becomes $[1, \cdots, 1]$. Then, we apply the map $R_n \to \frac{1}{n}R_n$, dividing the last row by n. Lastly, we applied the row operations $R_j \to R_j - R_n$ for $j = 1, \cdots, n-1$, adding $\frac{1}{n}[1, \cdots, 1]$ to each of the rows but the last. Thus, the matrix F^{-1} is the sum of two matrices. The first is $\frac{1}{n}\xi\mathbb{1}_n^\top$, where $\xi = [-1, ..., -1, 1]^\top$. The second is the diagonal matrix $\mathcal{I} = \mathrm{Id}_n - \mathrm{e}_n\mathrm{e}_n^\top$, having all diagonal entries equal to 1, but the last, which is equal to 0. Thus, we have

$$\|F^{-1}\| \leq \frac{1}{n}\|\xi\mathbb{1}_n^\top\| + \|\mathcal{I}\| \leq \frac{1}{n}\|\mathbb{1}_n\|\|\xi\| + 1 = 1 + 1 = 2.$$

This completes the proof. □

We can now prove Theorem 6.8

Proof. The proof for the time complexity of the algorithm is identical to the proof of Theorem 6.6. Thus we focus on the error estimate.

By definition, the matrix M' satisfies $M'\delta\mathrm{y}_i = \delta\mathrm{w}_i$ for $i \in \mathbb{V}$. Using Corollary 6.4 and Remark 6.12, we conclude that for any $i \in \mathbb{V}$, $\|\delta\mathrm{w}_i - \mathcal{M}'\delta\mathrm{y}_i\| = O(\|\delta\mathrm{w}_i\|^2)$, where $\mathcal{M}' = \nabla k^{-1}(\mathrm{y}_0) + \mathcal{E}_{\mathcal{G}}N\nabla g(\mathcal{E}_{\mathcal{G}}^\top \mathrm{y}_0)\mathcal{E}_G^\top + Q\mathrm{y}_0$. Thus, we conclude that for any $i \in \mathbb{V}$, we have $\|(M' - \mathcal{M}')\delta\mathrm{y}_i\| \leq O(\max_k \|\delta\mathrm{w}_i\|^2) = O(\kappa^2)$. Hence the operator norm satisfies:

$$\|(M' - \mathcal{M}')\delta Y\| \leq \sqrt{\sum_{i\in\mathbb{V}} O(\kappa^2)^2} = O(\sqrt{n}\kappa^2).$$

Now, by submultiplicativity of the operator norm [66], we have that $\|M' - \mathcal{M}'\| \leq \|(M' - \mathcal{M}')\delta Y\|\|\delta Y^{-1}\|$. Thus, we have $\|M' - \mathcal{M}'\| \leq O\left(\sqrt{n}\kappa^2\|\delta Y^{-1}\|\right)$. We note that $M' - \mathcal{M}' = M - \mathcal{M}$, meaning that $\|M - \mathcal{M}\| \leq O\left(\sqrt{n}\kappa^2\|\delta Y^{-1}\|\right)$, implying the same inequality for all entries $|M_{ij} - \mathcal{M}_{ij}|$.

Now, we wish to estimate $\|\delta Y^{-1}\|$. We define $\delta\mathrm{v}_i = \mathcal{M}'\delta\mathrm{y}_i$ for $i = 1, \cdots, n$, so equation (6.13) reads $\delta\mathrm{v}_i = \delta\mathrm{w}_i - O(\kappa^2)$. We define the matrix δV as the matrix whose columns are $\delta\mathrm{v}_i$. Then $\delta V = \delta W - O(\kappa^2)$. In particular, by multiplying by δW^{-1} and using Lemma 6.4, we conclude that $\delta W^{-1}\delta V = \mathrm{Id}_n - O(\kappa)$, or equivalently, by taking inverses, $\delta V^{-1}\delta W = \mathrm{Id}_n + O(\kappa)$.

© Technion - Israel Institute of Technology, Elyachar Central Library

Now, we note that $\delta Y^{-1} = \mathcal{M}\delta V^{-1}$. Thus, by submultiplicativity of the operator norm, we conclude that:

$$\|\delta Y^{-1}\| \leq \|\mathcal{M}\| \cdot \|\delta V^{-1}\| \leq \|\mathcal{M}\| \cdot \|\delta V^{-1}\delta W\| \cdot \|\delta W^{-1}\|$$

We can now estimate each factor on its own. Lemma 6.4 implies that $\|\delta W^{-1}\| = O(\kappa^{-1})$. Moreover, $\delta W^{-1}\delta V = \mathrm{Id}_n - O(\kappa)$ implies that $\|\delta V^{-1}\delta W\| = O(1)$, namely $\|\delta V^{-1}\delta W\| = 1 + O(\kappa)$. Lastly, we can estimate the norm of M as following:

$$\begin{aligned}\|\mathcal{M}\| &\leq \|\nabla k^{-1}(\mathrm{y}_0)\| + \|\mathcal{E}_{\mathcal{G}} N \nabla g(\mathcal{E}^\top \mathrm{y}_0)\mathcal{E}_{\mathcal{G}}^\top\| + \|Q\| \\ &\leq O(1) + \max_{i,j}(\nu_{ij}d_{ij})\lambda_{\max}(\mathcal{G}) + O(1),\end{aligned}$$

where we use $Q = 0$ if $\nabla k^{-1}(\mathrm{y}_0) = \mathbf{0}$, and $Q = \frac{1}{n}\mathbb{1}_n\mathbb{1}_n^\top$ otherwise, implying $\|Q\| \leq \frac{1}{n}\|\mathbb{1}_n\|^2 = 1$. This completes the proof. □

Remark 6.13. *As with Algorithm 4, we can reduce the complexity of the computation of M' from $O(n^\omega)$ to $O(n^{\omega_1})$. As before, this is done by considering $(M')^{-1} = \delta W^{-1}\delta Y$. As the matrix δW is the product of $O(n)$ elementary matrices, we can compute $(M')^{-1}$ in $O(n^2)$ time by applying the corresponding row operations on δY. This time, unlike in Algorithm 4, the matrix $(M')^{-1}$ need not be positive-definite, or even symmetric, due to the quadratic error term in* (6.13)*. To remedy this problem, we symmetrize $(M')^{-1}$ by defining $(M')^{-1}_{sym} = \frac{1}{2}((M')^{-1} + ((M')^{-1})^\top)$, which is symmetric. Moreover, $(M')^{-1}_{sym}$ is close to the inverse of the positive definite-matrix $\nabla k^{-1}(\mathrm{y}_0) + \mathcal{E}_G N \nabla g(\mathcal{E}_{\mathcal{G}}^\top \mathrm{y}_0)\mathcal{E}_{\mathcal{G}}^\top + Q$, and eigenvalues are continuous functions. Thus, the eigenvalues of $(M')^{-1}_{sym}$ are positive, meaning it is a positive-definite matrix, so inverting it only costs $O(n^{\omega_1})$ time.*

Theorem 6.8 prescribes an estimate for the error in the elements of the matrix M. However, we want a clearer estimate on the error of the weighted graph computed by the algorithm. We conclude this chapter by relating between the estimate on $|M_{ij} - \mathcal{M}_{ij}|$, the estimates $\{p_{ij}\}$ on the weights $\{\nu_{ij}\}$, and the estimate $\mathcal{H}$ on the graph $\mathcal{G}$.

Proposition 6.9. *Suppose the same assumptions as in Theorem 6.8 hold. Suppose further that for any $i, j \in \mathbb{V}$, $|M_{ij} - \mathcal{M}_{ij}| \leq m$, and that $m \leq \frac{1}{4}\rho = \frac{1}{4}\min_{i,j}(\nu_{ij}d_{ij})$. If the number ε used in Algorithm 5 is equal to $2m$, then the graph $\mathcal{H}$ computed by the algorithm is identical to $\mathcal{G}$, and for all $\{i, j\} \in \mathbb{E}$, the difference of the computed weights p_{ij} from the true weights ν_{ij} is bounded by $d_{ij}^{-1}m$, where $d_{ij} = \frac{dg_{ij}}{d\zeta_{ij}}((y_0)_i - (y_0)_j)$.*

Proof. Suppose first that $\{i, j\} \notin \mathbb{E}$. Then $\mathcal{M}_{ij} = 0$, meaning that $M_{ij} \leq m < \varepsilon$. In particular, the algorithm does not add the edge $\{i, j\}$ to the graph $\mathcal{H}$, as required.

Now suppose that $\{i, j\} \in \mathbb{E}$. Then $\mathcal{M}_{ij} = -d_{ij}\nu_{ij}$. Thus $|M_{ij}| > d_{ij}\nu_{ij} - \varepsilon \geq d_{ij}\nu_{ij} - \frac{1}{2}d_{ij}\nu_{ij} = \frac{1}{2}d_{ij}\nu_{ij} \geq \varepsilon$. Thus the algorithm correctly chooses to add the

© Technion - Israel Institute of Technology, Elyachar Central Library

edge $\{i,j\}$ to $\mathcal{H}$. Moreover, $M_{ij} = -p_{ij}d_{ij}$ and $\mathcal{M}_{ij} = -\nu_{ij}d_{ij}$, meaning that the inequality $|M_{ij} - \mathcal{M}_{ij}| \leq m$ implies $|p_{ij} - \nu_{ij}| \leq d_{ij}^{-1}\varepsilon_0$. □

Theorem 6.8 and Proposition 6.9 show that Algorithm 5 is able to approximate the underlying weighted graph. Moreover, it shows that its time complexity is $O(n^\omega)$, where Remark 6.13 shows it can be reduced to $O(n^{\omega_1})$. In the next subsection, we'll discuss about the robustness of the algorithm, and in the following subsection, we'll ask ourselves whether a faster algorithm solving the network reconstruction problem exists.

6.4 Robustness and Practical Considerations

The algorithm presented in the previous subsection solves the network identification under some strong assumptions. First, the algorithm assumes the network is noiseless and disturbance-free, so the network converges to a constant steady-state. Second, the algorithm assumes that the measurements taken are perfect and are not subject to noise or disturbances. Lastly, the algorithm assumes that the exogenous input can be applied to all agents. This sub-subsection is dedicated to discuss all these points, and to give a brief comparison of the algorithm to other methods described in literature.

6.4.1 Robustness to Noise and Disturbances

We begin by studying how noise and/or disturbances affect the output of the diffusively-coupled network. Generally, if we make no passivity assumption on the network, then it might not converge in the presence of noise. One example of this phenomenon is the consensus protocol [107], in which noise does not disturb the asymptotical convergence to consensus (almost surely), but it does make the consensus value to be volatile. The consensus protocol can also be viewed as the diffusively coupled system with single-integrator agents and static gain controllers, with passive agents, and output-strictly passive controllers. However, we can still use passivity to obtain some form of noise- and disturbance-rejection:

Proposition 6.10. *Consider a diffusively-coupled system $(\mathcal{G}, \Sigma, g)$ with steady-state $(\mathrm{u}, \mathrm{y}, \zeta, \mu)$. Suppose that the agents are output-strictly passive with respect to $(\mathrm{u}_i, \mathrm{y}_i)$ with parameters $\rho_i > 0$, and that the controllers are passive with respect to (ζ_e, μ_e). We let S be the sum of the agents' storage functions, and denote $R = \mathrm{diag}(\rho_i) > 0$. Consider a parasitic exogenous input $d(t)$ to agents, so that the input is $u(t) = d(t) - \mathcal{E}_\mathcal{G}\mu(t)$. Assume that at any time, $||R^{-1/2}d(t)|| \leq \Delta$. Let $\mathcal{A} = \{x : \ \|R^{1/2}(h(x) - \mathrm{y})\| \leq \Delta\}$, and let $\Xi = \max_{x\in\mathcal{A}} S(x)$. Then for any initial conditions, there exists some T such that if $t > T$, then $\|y(t) - \mathrm{y}\| \leq \max_{x:\ S(x)\leq\Xi} \|h(x) - \mathrm{y}\|$*

Proof. Let $v(t) = -\mathcal{E}_\mathcal{G}\mu(t)$, so that the input is $u(t) = v(t) + d(t)$. By passivity,

we have $0 \leq \sum_{e\in\mathbb{E}}(\zeta_e - \zeta_e)(\mu_e - \mu_e)$, and

$$\frac{d}{dt}S(x) \leq \sum_{i\in\mathbb{V}}(-\rho_i\|y_i - \mathrm{y}_i\|^2 + (d_i + v_i - \mathrm{u}_i)(y_i - \mathrm{y}_i)).$$

Summing the equations and using the connections $v = -\mathcal{E}_{\mathcal{G}}\mu$, $\zeta = \mathcal{E}_{\mathcal{G}}y$ and their steady-state counterpart yields,

$$\begin{aligned}\frac{d}{dt}S(x) &= -(y-\mathrm{y})^\top R(y-\mathrm{y}) + d(t)^\top(y-\mathrm{y}) \\ &= -||R^{1/2}(y-\mathrm{y})||^2 + (R^{-1/2}d(t))^\top R^{1/2}(y-\mathrm{y}) \\ &\leq -||R^{1/2}(y-\mathrm{y})||^2 + \Delta\|R^{1/2}(y-\mathrm{y})\|\end{aligned} \tag{6.14}$$

We note that if $\|R^{1/2}(y(t)-\mathrm{y})\| \leq \Delta+\varepsilon$ does not hold, then the right-hand side is strictly negative, and is bounded from above by $-(\varepsilon+\Delta)\varepsilon$. If this happens indefinitely, then we will eventually have $S(x) < 0$, which cannot hold. Thus for some $T > 0$ we have $\|R^{1/2}(y(T)-\mathrm{y})\| \leq \Delta+\varepsilon$, so $x(T) \in \mathcal{A}$, and $S(x(T)) \leq \Xi$. If there's some $t > T$ such that $S(x(t)) > \Xi$, then $x(t) \notin \mathcal{A}$, and $\dot{S} < 0$. In particular, $S(x(t-\delta)) > S(x(t)) > \Xi$ for $\delta > 0$ small enough. Repeating this argument shows that $S(x(t_1)) > \Xi$ for all times $t_1 < t$, which is false as $S(x(T)) \leq \Xi$. This completes the proof. □

The proposition above shows that even in the presence of a disturbance or noise, the algorithm can sample the output y such that its not too far the true steady-state output. This will intertwine with Proposition 6.11, in which the effects of measurement errors will be accounted for.

Remark 6.14. *The proposition above does not distinguish between disturbances and random noises. In practice, the bound is approximately of the right order of magnitude for disturbances, but is a gross overestimate for noise. For example, consider the single agent of the form $\dot{x} = -x + u, y = x$, where u is chosen as a random white noise, bounded by C, and with variance σ^2. A similar proposition to the one above shows that the system converges to an output with $|y(t)| \leq C$ (as $\rho = 1$). However, writing $x(t)$ as a convolution integral and applying Itô calculus [106], e.g. the Itô isometry, shows that $\mathbb{E}(x(t)) = 0$ and that $\mathrm{Var}(x(t)) \leq \sigma^2/2$. Thus, Chebyshev's inequality gives a high-probability bound on where $x(t)$ can be, which is much better than the proposition bound if $\sigma \ll C$.*

We now shift our focus to measurement errors. Measurement errors give parasitic terms when defining the matrices δW, δY in Algorithm 5. We prove the following:

Proposition 6.11. *Suppose Algorithm 5 builds the matrix $\delta Y + \Delta Y$ instead of δY. If $||\Delta Y|| \ll ||\delta Y||$, then the algorithm outputs a matrix M whose distance from $\mathcal{M} = \nabla k^{-1}(\mathrm{y}_0) + \mathcal{E}_{\mathcal{G}}N\nabla g(\mathcal{E}_{\mathcal{G}}^\top \mathrm{y}_0)\mathcal{E}_{\mathcal{G}}^\top$ in the operator norm is bounded by $O\Big(\sqrt{n}\Big(1 + \max_{i,j}(\nu_{ij}d_{ij})\lambda_{\max}(\mathcal{G})\Big)\Big)\|\Delta Y \delta Y^{-1}\|$, plus the error term from Theorem 6.8.*

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

Proof. It's enough to bound $\|\delta W(\delta Y^{-1} - (\delta Y + \Delta Y)^{-1})\|$. Using the submultiplicativity of the operator norm, we can bound each factor on its own. The first factor can be bounded by $O(\kappa\sqrt{n})$, as δW can be written as $\kappa(\mathrm{Id}_n - \mathrm{e}_n\mathbb{1}_n^\top + \mathbb{1}_n\mathrm{e}_n^\top)$. As for the second factor, we can bound it as $\|\delta Y^{-1}\|\|\mathrm{Id}_n - (\mathrm{Id}_n + \Delta Y\delta Y^{-1})^{-1}\|$. The assumption $||\Delta Y|| \ll ||\delta Y||$ implies that $(\mathrm{Id}_n + \Delta Y\delta Y^{-1})^{-1} \approx \mathrm{Id}_n - \Delta Y\delta Y^{-1}$ up to a second order error, which in turn gives the desired bound, as $\|\delta Y^{-1}\| \leq O(\kappa^{-1}(1 + \max_{i,j}(\nu_{ij}d_{ij})\lambda_{\max}(\mathcal{G})))$, as shown in the proof of Theorem 6.8. □

Remark 6.15. *Proposition 6.11 gives a viable way to bound the algorithm error if a relative measure error is known. In some cases, we have an absolute error bound, e.g. Proposition 6.10. In that case, we can use it to bound the relative error, as* $\|\Delta Y\delta Y^{-1}\| \leq \|\delta Y^{-1}\|\|\Delta Y\|$*, and we can again use the same bound* $\|\delta Y^{-1}\| \leq O(\kappa^{-1}(1 + \max_{i,j}(\nu_{ij}d_{ij})\lambda_{\max}(\mathcal{G})))$.

6.4.2 Probing Inputs Supported on Subsets of Nodes

The previous sub-subsection shows that the algorithm is somewhat resistant to noise, either in the dynamics or the measurement. We now move to the last major assumption, namely that the exogenous input can be applied on all agents. There are two possible ways to try and relax the assumption. First, we can try and use compressed sensing methods, using the sparsity of the matrix M', which corresponds to a relatively low number of edges in the network $\mathcal{G}$, similarly to [56, 81]. Namely, the sparse matrix $M' \in \mathbb{R}^{n\times n}$ is recovered from $r < n$ equations of the form $\delta\mathrm{w}_i = M'\delta\mathrm{y}_i$ for $i = 1, \cdots, r$ with high probability. This is done by considering M' as a sparse vector of size n^2, such that each equation $\delta\mathrm{w}_i = M'\delta\mathrm{y}_i$ is translated to n one-dimensional constraints. One then solves the minimization problem minimizing the ℓ^1-norm of the vectorized matrix under these constraints. Under a certain assumption on the matrix M' (namely the *restricted isometry property*), a total of $O(\frac{|\mathbb{E}|}{n}\log(n^2/|\mathbb{E}|))$ samples are enough to achieve a reconstruction with high probability, as each sample gives n different constraints. See [9, 42] for more on compressed sensing.

Another approach is to still try and use n different measurements, with exogenous inputs supported only on ℓ nodes. Indeed, in order to reconstruct the matrix M', we need that the vectors $\delta\mathrm{y}_1, \cdots, \delta\mathrm{y}_n$ will span all of $\mathbb{R}^n$. If the steady-state equation (6.4) is inherently nonlinear, then even when the inputs are restricted to a subspace of dimension ℓ, the outputs can span all of $\mathbb{R}^n$. Abstractly, we can prove the following:

Proposition 6.12. *Let $F : \mathbb{R}^k \to \mathbb{R}^n$ be any function which is $\ell + 1$-times differentiable function at $x_0 \in \mathbb{R}^k$. Suppose that the dimension of the subspace spanned by all partial derivatives of F at x_0 up to order ℓ is r, and denote the number of all partial derivatives up to order ℓ by s. Let $\mathbb{P}$ be any probability measure on $\mathbb{R}^k$ which is supported on a small ball around x_0, and let $x_1, \cdots, x_s$ be i.i.d. samples according to $\mathbb{P}$. Then, with probability 1, the vectors $F(x_1) - F(x_0), \cdots, F(x_s) - F(x_0)$ span a subspace of dimension r inside $\mathbb{R}^n$. In particular, if $r = n$ then they span all of $\mathbb{R}^n$.*

© Technion - Israel Institute of Technology, Elyachar Central Library

Proof. Suppose that $\mathbb{P}$ is supported inside a ball around x_0 of radius $\kappa \ll 1$. By Taylor's theorem, we can write $\Phi_F = D_F W$ up to an error of order $O(\kappa^{\ell+1})$, where Φ_F is the matrix whose columns are $F(x_i) - F(x_0)$, D_F is the matrix whose columns are all the partial derivatives of F at x_0 up to order ℓ, and W is a square matrix consisting of polynomials of the entries of $x_i - x_0$ for $i \in \{1, \cdots, s\}$. We need to show that the rank of Φ_F is equal to n, where we know that the rank of D_F is equal to n. This follows immediately if we know that W is invertible, so we want to show that the probability that W is not invertible is 0. Indeed, consider the map $p : (\mathbb{R}^k)^s \to \mathbb{R}^n$ defined by $p(x_1, \cdots, x_s) = \det W$. This is a non-zero polynomial of $x_1, \cdots, x_s$, and W is invertible if and only if $p \neq 0$. However, it is known that the collection of zeros of a non-zero polynomial is a zero-measure set [25], and thus $p \neq 0$ with probability 1, which implies that W is invertible with probability 1. This concludes the proof. □

The proposition above can be applied for the map F mapping w to y according to (6.4). In many occasions, it's hard or impossible to compute the rank r defined in the theorem. However, it's possible to use the proposition in a more data-based fashion - take s random samples $\mathrm{w}_0 + \delta\mathrm{w}_i$ near w_0, and compute the s steady-state outputs $\mathrm{y}_0 + \delta\mathrm{y}_i$. The rank r can now be computed using $\delta\mathrm{y}_i$, and one can find the connecting matrix M' using the compressed sensing approach.

To conclude this subsection, we saw that the presented algorithm can be applied in real-world scenarios, in which noise and measuring errors can happen, and not all nodes are susceptible to controlled exogenous inputs. Other algorithms which use probing inputs, or similar methods, rely on linearizing the dynamics instead of the steady-state equation, or using higher-order terms in the Taylor approximation [56,81]. These methods thrive in dynamics-rich networks, but will oversample and work very slowly in fast converging networks. On the contrary, the presented algorithm will work poorly in dynamics-rich networks, but will operate well on fast converging networks, or in cases where sampling the system is expensive. Examples of such networks include, for example, networks of autonomous vehicles trying to coordinate their velocity for platooning. The network cannot have rich dynamics due to safety reasons, and the perturbations from the desired platooning velocity should be very small. Understanding these networks is essential for traffic management, and can form a first step in predicting traffic jams and accidents. Other application examples with similar conditions include multi-satellite arrays, UAVs, drones, and robots.

Note 6.2. *The network model and presented algorithm were introduced for the case of SISO agents and controllers. If the agents and controllers are MIMO, with q inputs and q outputs, the network model and algorithm still work with a slight modification of the assumptions. This is mainly because the network optimization framework holds also for MIMO agents, if we switch the MEIP assumption by MEICMP.*

© Technion - Israel Institute of Technology, Elyachar Central Library

6.4.3 Time Complexity Bounds for the Network Reconstruction Problem

In the previous subsections we presented an algorithm solving the network reconstruction problem in $O(n^{\omega_1})$ time using specifically constructed inputs. We ask ourselves if we can improve on that. We first need to discretize our problem in order to fit into the standard complexity theory framework.

Problem 6.3. *We are given some diffusively coupled system $(\mathcal{G}_\nu, \Sigma, g)$ where the agents Σ and the static controllers g are known. We are also given some integer $q > 0$, such that if the input to the network is a $\mathcal{C}^{q+1}$ signal, then the output is a $\mathcal{C}^q$ signal.*[2] *Our goal is to find the weighted graph $\mathcal{G}_\nu$ using measurements of the node outputs $y_i = h_i(x_i)$ and their derivatives $\frac{d^k}{dt^k} y_i$ up to order q. We are allowed to choose the exogenous input signal $w(t)$ as a $\mathcal{C}^{q+1}$ signal. Furthermore, accessing the measurements $y(t)$ or changing the function describing $w(t)$ can not be performed faster than at $\Delta t > 0$ second intervals. Moreover, the measured outputs $y(t)$ are accurate up to a relative order of magnitude no larger than ε*

After discretizing the problem, limiting the rate of measurement and change in input, we prove the following theorem.

Theorem 6.9. *Any (possibly randomized) algorithm solving Problem 6.3, estimating $\{\nu_{ij}\}$ with some finite error (with probability 1), must make $n-1$ measurements in the worst case. Moreover its worst-case complexity is at least $\Omega(n^{\omega_1})$.*

Corollary 6.5. *By Remark 6.13, Algorithm 5 is optimal in terms of computational time complexity*

Before proving the theorem, we need the following lemma:

Lemma 6.5. *Let $P \in \mathbb{R}^{(n-1)\times(n-1)}$ be a positive definite matrix, let $\mathcal{E}_{K_n}$ be the incidence matrix of the complete graph on n edges, and let $V \in \mathbb{R}^{(n-1)\times n}$ be any matrix such that $VV^\top = \mathrm{Id}_{n-1}$ and $V\mathbb{1}_n = \mathbf{0}$. Then there exists a positive-semi definite matrix $\mathcal{Q} \in \mathbb{R}^{n\times n}$ such that:*

i) There exists a positive-definite diagonal matrix N such that $\mathcal{Q} = \mathcal{E}_{K_n} N \mathcal{E}_{K_n}^\top$.

ii) The equality $P = V\mathcal{Q}V^\top$ holds.

Proof. We define $\mathcal{Q} = V^\top P V \in \mathbb{R}^{n\times n}$. The matrix $\mathcal{Q}$ is positive semi-definite as P is positive definite. Moreover, we have:

$$V\mathcal{Q}V^\top = VV^\top PVV^\top = \mathrm{Id}_{n-1} P \mathrm{Id}_{n-1} = P$$

which proves the second part. As for the first part, define the matrix N as follows - for each edge $e = \{i, j\}$ in K_n, we define the e-th diagonal entry of

[2] This is weaker than assuming that the functions f, q, g, h appearing in the dynamics are all smooth

© Technion - Israel Institute of Technology, Elyachar Central Library

N to be $-\mathcal{Q}_{ij} = -\mathcal{Q}_{ji}$. It is easy to check that the off-diagonal entries of $\mathcal{Q}$ are equal to the off-diagonal entries of $\mathcal{E}_{K_n} N \mathcal{E}_{K_n}^\top$, as the latter is a weighted Laplacian. As for the diagonal entries, $\mathbb{1}_n$ is in the nullspace of both $\mathcal{E}_{K_n} N \mathcal{E}_{K_n}^\top$ and $\mathcal{Q} = V^\top P V$. Thus the sum of the elements in each row of both matrices is zero, meaning that:

$$\mathcal{Q}_{ii} = -\sum_{j \neq i} \mathcal{Q}_{ij}, \ (\mathcal{E}_{K_n} N \mathcal{E}_{K_n}^\top)_{ii} = -\sum_{j \neq i} (\mathcal{E}_{K_n} N \mathcal{E}_{K_n}^\top)_{ij}.$$

Therefore the diagonal entries are also equal. This implies that $\mathcal{Q} = \mathcal{E}_{K_n} N \mathcal{E}_{K_n}^\top$ and completes the proof of the lemma. □

We now prove Theorem 6.9.

Proof. We first deal with a similar problem. We assume we have a single agent with m inputs and m outputs, evolving according to the equation $\dot{x} = -f(x) + w, \ y = h(x)$. We are again allowed to measure the output and its derivatives up to order q, or change the $\mathcal{C}^{q+1}$ function defining the input, no faster than once every Δt seconds. Moreover, all measurements are accurate up to a relative order of magnitude no larger than ε. Specifically, choose any positive-definite matrix $P \in \mathbb{R}^{m \times m}$ a large enough arbitrary scalar $\varrho > 0$. Consider the following single agent with m inputs and outputs,

$$\Sigma_P : \ \dot{x} = -\varrho P x + w, \ y = x.$$

We claim any algorithm computing P with some finite error (with probability 1) must take at least m measurements in the worst case, and that if the algorithm is deterministic, then its worst case complexity is at least $\Omega(m^{\omega_1})$. We will prove it below, but first show that it is enough to prove the theorem. Consider a network reconstruction problem with agents $\dot{x}_i = u_i$, static controllers $g_{ij}(x) = x$, and an underlying graph $\mathcal{G} = K_n$., where the coupling matrix N is unknown. The dynamics of the system can be written as $\dot{x} = -\mathcal{E}_{K_n} N \mathcal{E}_{K_n}^\top x + w$. We note that this system has two decoupled subsystems - one for the scalar $\mathbb{1}_n^\top x$, and one for the vector $\mathrm{Proj}_{\mathbb{1}_n^\perp} x$. We shall focus on the latter. More specifically, we consider the matrix $V \in \mathbb{R}^{(n-1) \times n}$ having the following vectors as rows:

$$v_k = \frac{1}{\sqrt{k^2 + k}} [\underbrace{1, \cdots, 1}_{k \text{ times}}, -k, \quad \underbrace{0, \cdots, 0}_{n-k-1 \text{ times}} \ ], \ k = 1, ..., n-1.$$

It's easy to check that $V^\top V = \mathrm{Id}_{n-1}$ and that $\mathbb{1}_n \in \ker(V)$. The vector $z = Vx$ satisfies the ODE $\dot{z} = -V \mathcal{E}_K N \mathcal{E}_K^\top V^\top z + V w$. By the lemma, we conclude that different choices of N yield all possible $(n-1) \times (n-1)$ positive-definite matrices, so we get a general system of the form Σ_P, where P can be any positive definite matrix, and reconstructing $P = V \mathcal{E}_K N \mathcal{E}_K^\top V^\top$ is equivalent to reconstructing N. This completes the proof of the theorem, as here $m = n - 1$.

Now, return to the system identification problem for the system Σ_P. Consider any measurement made by the algorithm. Let T_1 be the time of the

© Technion - Israel Institute of Technology, Elyachar Central Library

measurement, and let T_0 be the last time the function describing the input $w(t)$ was changed. We shift the times to assume that $T_0 = 0$, without loss of generality, and note that $T_1 \geq \Delta t$. The output $y(t)$ at any time $t \in [0, T_1]$ can be written as a convolution integral,

$$y(t) = \int_0^t e^{-\varrho\xi P} w(t-\xi) d\xi.$$

By assumption, w is continuously differentiable $q+1$ times, so by Lagrange's form of the remainder in Taylor's theorem, we write $w(t-\xi)$ as:

$$w(t-\xi) = \sum_{j=0}^{q} (-1)^j \frac{d^j w(t)}{dt^j} \frac{\xi^j}{j!} + (-1)^{q+1} \frac{d^{q+1} w(\tilde{t})}{dt^{q+1}} \frac{\xi^{q+1}}{(q+1)!}, \tag{6.15}$$

for some point $\tilde{t} \in [t-\xi, t]$. We plug this expression inside the integral describing $y(t)$. For the first q summands, we end up with integrals for the form $\int_{T_0}^t e^{-\varrho\xi P} \frac{d^j w(t)}{dt^j} \frac{\xi^j}{j!}$. The following formula appears in [180, Formula 2.321.2]:

$$\int x^n e^{cx} dx = e^{cx} \sum_{i=0}^{n} (-1)^{n-i} \frac{n!}{i! c^{n-i+1}} x^i, \tag{6.16}$$

which will be used to compute the said integrals.

The matrix P is positive-definite, so we can write it as $P = \sum_{k=1}^m \lambda_k \mathrm{v}_k \mathrm{v}_k^\top$, where $\lambda_k > 0$ are P's eigenvalues and v_k are its eigenvectors and $||\mathrm{v}_k|| = 1$. Then for any ξ, ϱ, we have that $e^{-\varrho\xi P} = \sum_{k=1}^m e^{-\varrho\lambda_k \xi} \mathrm{v}_k \mathrm{v}_k^\top$. Thus, we have that:

$$\begin{aligned}
&\int_T^t e^{-\varrho\xi P} \frac{d^j w(t)}{dt^j} \frac{\xi^j}{j!} d\xi = \sum_{k=1}^m \frac{\mathrm{v}_k \mathrm{v}_k^\top}{j!} \frac{d^j w(t)}{dt^j} \int_0^t e^{-\varrho\lambda_k \xi} \xi^j d\xi = \\
&\sum_{k=1}^m \frac{\mathrm{v}_k \mathrm{v}_k^\top}{j!} \frac{d^j w(t)}{dt^j} \left[e^{-\varrho\lambda_k \xi} \sum_{i=0}^{j} \frac{(-1)^{j-i} j!}{i! (-\varrho\lambda_k)^{j-i+1}} \xi^i \right]_{\xi=0}^t = \\
&\sum_{k=1}^m \frac{\mathrm{v}_k \mathrm{v}_k^\top}{j!} \frac{d^j w(t)}{dt^j} \left[- e^{-\varrho\lambda_k \xi} \sum_{i=0}^{j} \frac{j!}{i! (\varrho\lambda_k)^{j-i+1}} \xi^i \right]_{\xi=0}^t = \\
&\sum_{k=1}^m \mathrm{v}_k \mathrm{v}_k^\top \frac{d^j w(t)}{dt^j} \left[\frac{1}{(\varrho\lambda_k)^{j+1}} - e^{-\varrho\lambda_k t} \sum_{i=0}^{j} \frac{t^i}{i! (\varrho\lambda_k)^{j-i+1}} \right].
\end{aligned}$$

We can use functional calculus to compute the sums written above. For example,

$$\sum_{k=1}^m \frac{\mathrm{v}_k \mathrm{v}_k^\top}{(\varrho\lambda_k)^{j+1}} \frac{d^j w(t)}{dt^j} = \frac{1}{\varrho^{j+1}} P^{-(j+1)} \frac{d^j w(t)}{dt^j}.$$

Thus, we get that

$$\int_0^t \frac{\xi^j e^{-\varrho\xi P}}{j!} \frac{d^j w(t)}{dt^j} = \left[\frac{P^{-j-1}}{\varrho^{j+1}} - \sum_{i=0}^{j} \frac{t^i e^{-\varrho t P} P^{i-j-1}}{i! \varrho^{j-i+1}} \right] \frac{d^j w(t)}{dt^j}.$$

© Technion - Israel Institute of Technology, Elyachar Central Library

By (6.15), we get:

$$y(t) \approx \sum_{j=0}^{k}(-1)^j\left[\frac{P^{-j-1}}{\varrho^{j+1}} - \sum_{i=0}^{j}\frac{t^i e^{-\varrho tP}P^{i-j-1}}{i!\varrho^{j-i+1}}\right]\frac{d^j w(t)}{dt^j}, \tag{6.17}$$

with an error of the form $\int_0^t e^{-\varrho\xi P}\frac{d^{q+1}w(\tilde{t})}{dt^{q+1}}\frac{\xi^{q+1}}{(q+1)!}$. We claim that if ϱ is large enough, then $y(T_1) = \frac{1}{\varrho}P^{-1}w(T_1)$ up to a relative error of magnitude no larger than ε.

Indeed, we note that $T_1 \geq \Delta t$, so if $\varrho\underline{\sigma}(P) \gg \Delta t$, then

$$\frac{1}{\varrho}P^{-1} \approx \sum_{j=0}^{k}(-1)^j\left[\frac{P^{-j-1}}{\varrho^{j+1}} - \sum_{i=0}^{j}\frac{T_1^i e^{-\varrho tP}P^{i-j-1}}{i!\varrho^{j-i+1}}\right],$$

up to a relative error of magnitude no larger than $\varepsilon/2$. Indeed, for $j \neq 0$, the first element inside the outer sum behaves as $O(\frac{1}{\varrho^{j+1}\underline{\sigma}(P)^{j+1}})$, and the second element decreases exponentially fast with ϱ (for fixed P, T_1). Moreover, the error term in (6.17) can also be bounded similarly - we know that w is a $\mathcal{C}^{q+1}$ signal, meaning that $\frac{d^{q+1}w}{dt^{q+1}}$ is a continuous function on the compact interval $[0, T_1]$. Thus, the norm of the vector $\frac{d^{q+1}w(t)}{dt^{q+1}}$ is bounded by M for all $t \in [0, T_1]$. Then, the norm of the error term in (6.17) is bounded by:

$$\begin{aligned}
M\int_0^{T_1}\frac{\xi^{q+1}\|e^{-\varrho\xi P}\|}{(q+1)!}d\xi \leq & M\int_0^{T_1}\frac{\xi^{q+1}e^{-\varrho\underline{\sigma}(P)\xi}}{(q+1)!}d\xi \\
= & \frac{M}{(q+1)!}\left[e^{-\varrho\underline{\sigma}(P)\xi}\sum_{i=0}^{q+1}\frac{(-1)^{q+1-i}(q+1)!}{i!(-\varrho\underline{\sigma}(P))^{q+1-i+1}}\xi^i\right]_{\xi=0}^{T_1} \\
= & \left[-e^{-\varrho\underline{\sigma}(P)\xi}\sum_{i=0}^{q+1}\frac{M}{i!(\varrho\underline{\sigma}(P))^{q+1-i+1}}\xi^i\right]_{\xi=0}^{T_1} \\
= & \frac{M}{(\varrho\underline{\sigma}(P))^{q+2}} - e^{-\varrho\underline{\sigma}(P)T_1}\sum_{i=0}^{q+1}\frac{M}{i!(\varrho\underline{\sigma}(P))^{q+1-i+1}}T_1^i.
\end{aligned}$$

The first element is of order $O(\frac{1}{\varrho^{q+2}})$, and the second decays exponentially with ϱ (for fixed M, P, T_1). Thus, if ρ is large enough, then $y(T_1) = \frac{1}{\varrho}P^{-1}w(T_1)$, up to a relative error of order of magnitude no larger than ε. More specifically, this happens for any $\rho > \rho_0$, where ρ_0 is a threshold depending on the matrix P, the sample time $T_1 \geq \Delta t$, and the signal $w(t)$ (through M). As for derivatives of $y(t)$ at $t = T_1$, one can use the higher-order estimates of (6.17) together with the error estimate and $\dot{y} = -\varrho Py + w$ to conclude that $\frac{dy}{dt}(T_1) = -\varrho Py(T_1) + w(T_1) = \frac{1}{\varrho}P^{-1}\frac{dw}{dt}(T_1)$ up to a relative error of magnitude no larger than ε, provided that ϱ exceeds some threshold. Similarly, one can get that $\frac{d^j y}{dt^j} = -\varrho P\frac{d^{j-1}y}{dt^{j-1}} + \frac{d^{j-1}w}{dt^{j-1}}$ for all integers $0 \leq j \leq q$, allowing one to prove by induction that $\frac{d^j y}{dt^j}(T_1) =$

$\frac{1}{\varrho}P^{-1}\frac{d^j w}{dt^j}(T_1)$ up to a relative error of magnitude no larger than ε, provided that ϱ is large enough.

Until now, we saw that if ϱ is chosen sufficiently large, the results of all measurements made by the algorithm will be of the form $\mathrm{z}_i = \frac{1}{\varrho}P^{-1}\tau_i$, where τ_i is some vector depending on the input signal $w(t)$, and can be calculated exactly. We first assume that less than m measurement were made. Let $\mathrm{z}_1, ..., \mathrm{z}_r$ be the measurements corresponding to inputs $\tau_1, ..., \tau_r$, where $r < m$. We can find some nonzero vector $\tau_\star$ which is orthogonal to all of $\tau_1, ..., \tau_r$. It's clear now that the systems Σ_P and $\Sigma_{\mathcal{P}_\alpha}$ where $\mathcal{P}_\alpha = (P^{-1} + \alpha\tau_\star\tau_\star^\top)^{-1}$ will yield the same measurements, so we cannot differentiate them. Moreover, the error can be arbitrarily large for different values of α. Thus any (possibly randomized) algorithm solving the problem, estimating $\{\nu_{ij}\}$ up to some finite error with probability 1, should change the value of w at least $m-1$ times, and measure the output at least m times.

Now, we note that the relation between $\frac{1}{\varrho}\tau_i$ and the measured output z_i is linear at each measurement, with the connecting matrix being P^{-1} . Thus taking more than m measurements does not yield any additional data. In other words, the algorithm has measurements of P^{-1} times some m vectors, and it must return the value of P. Thus, the algorithm solves the matrix inversion problem for positive-definite matrices, and thus has complexity of $\Omega(m^{\omega_1})$. □

6.5 Case Studies

We present two case studies. One for first-order LTI agents and static gain controllers, and one for a neural network.

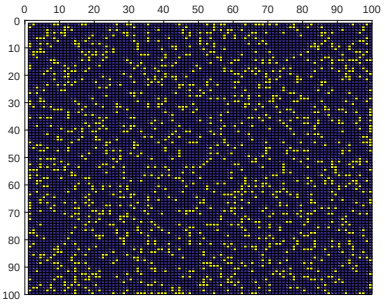
6.5.1 Linear Agents and Controllers

We consider a random Erdős-Rényi graph on $n = 100$ agents, where each edge exists with probability $p = 0.15$ independently from all other edges. Each agent i is LTI with transfer function $G_i(s) = \frac{1}{s+a_i}$, where a_i is chosen according to a log-uniform distribution between 1 and 100. Moreover, the controllers on each edge are static gains, chosen log-uniformly between 0.1 and 1. The unknown weights ν_e were chosen log-uniformly between 0.3 and 10.

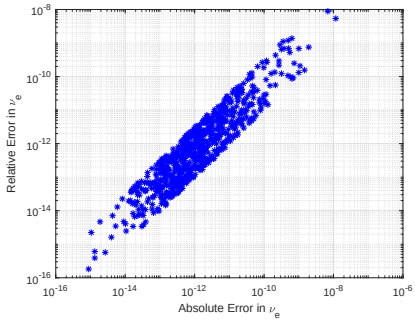
Algorithm 4 was run. Instead of waiting for convergence, the switching signal changed its value every 10 seconds. Moreover, instead of checking whether $M_{ij} \neq 0$, we checked whether $|M_{ij}| > 0.01 = \varepsilon$, dealing with numerical issues better. The adjacency matrix of the graph that was randomly chosen is available in Figure 6.2(a). The algorithm correctly identified all edges that exist in the graph $\mathcal{G}$, and Figure 6.2(b) shows the absolute and relative errors calculating the weights ν_e. The maximal absolute error is about 1.2×10^{-8}, and the maximal relative error is about 8.9×10^{-9}. It should be noted that we showed that, theoretically, Algorithm 4 should have no errors at all. The small errors in this simulation arise due to numerical errors, as well as using the output after 10 seconds for each switch, instead of taking the theoretical steady-state value. A

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library



(a) Adjacency matrix of the graph in the first case study. Yellow entries are equal to 1, and blue entries are equal to 0.



(b) Estimation errors of coupling strengths, as achieved by Algorithm 4.

Figure 6.2: Network reconstruction of an LTI network.

comparison of the errors achieved for different switching times is available in the table below:

Switching Time (in seconds)	Maximal Absolute Error	Maximal Relative Error
2.5	1.7176×10^{-4}	1.8141×10^{-4}
5	1.5342×10^{-7}	1.6204×10^{-7}
10	1.1621×10^{-8}	8.8682×10^{-9}
25	3.4505×10^{-9}	1.6210×10^{-9}

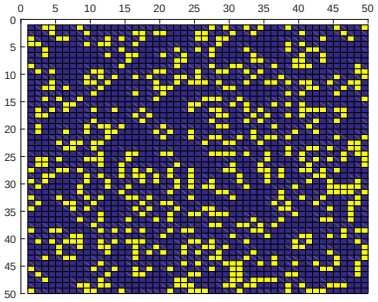
Table 6.1: Network detection algorithm performance for LTI systems

6.5.2 A Neural Network

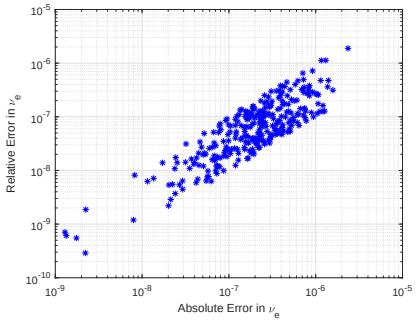
We consider a continuous neural network, as appearing in [130], on $n = 50$ neurons of one species. The governing ODE has the form,

$$\dot{V}_i = -\frac{1}{\tau_i} V_i + b_i \sum_{j \sim i} \nu_{i,j} (\tanh(V_j) - \tanh(V_i)) + \mathrm{w}_i, \tag{6.18}$$

where V_i is the voltage on the i-th neuron, $\tau_i > 0$ is the self-correlation time of the neurons, b_i is a self-coupling coefficient, $\nu_{i,j} = \nu_{j,i}$ are the coupling strengths between pairs of neurons, and the external input w_i is any other input current to neuron i. We run the system with 50 neurons, where the correlation times were chosen log-uniformly between 3_{sec} and 30_{sec}, and the self-coupling coefficient b_i were chosen log-uniformly between 1 and 5. We consider an unknown random Erdős-Rényi graph on $n = 50$ agents, where each edge exists with probability $p = 0.25$ independently from all other edges. Moreover, the unknown coupling coefficients $\nu_{i,j}$ were chosen log-uniformly between 1 and 10.



(a) Adjacency matrix of the graph in the second case study. Yellow entries are equal to 1, and blue entries are equal to 0.



(b) Estimation errors of coupling strengths, as achieved by Algorithm 5.

Figure 6.3: Network reconstruction of a neural network.

Algorithm 5 was run with $\kappa = 1 \times 10^{-3}$ and $\varepsilon = 0.01$. As with the previous case study, instead of waiting for convergence, the switching signal changed its value every 200 seconds. The adjacency matrix of the graph that was randomly chosen is available in Figure 6.3(a). The algorithm correctly identified all edges that exist in the graph $\mathcal{G}$, and Figure 6.3(b) shows the absolute and relative errors calculating the weights ν_e. The maximal absolute error is about 2.4×10^{-6}, and the highest relative error is about 1.9×10^{-6}. The algorithm was also run for different values of κ. A comparison of the errors achieved is available in the table below. We also ran the algorithm with $\kappa = 0.1$, which resulted in an erroneous reconstruction of the underlying graph - all existing edges were identified, but a few more non-existing edges were falsely declared as a part of the graph.

Value of κ	Maximal Absolute Error	Maximal Relative Error
1×10^{-2}	6.3376×10^{-6}	5.0690×10^{-6}
3×10^{-3}	4.8747×10^{-6}	3.8989×10^{-6}
1×10^{-3}	2.3563×10^{-6}	1.8846×10^{-6}
3×10^{-4}	2.2407×10^{-6}	6.4591×10^{-7}

Table 6.2: Network detection algorithm performance for neural networks

6.6 Conclusions

In this chapter, we explored the problem of network reconstruction. Namely, we studied two different problems - one regarding differentiation between networks with the same agents and controllers, but different subgraphs, and one

© Technion - Israel Institute of Technology, Elyachar Central Library

regarding reconstruction of the underlying network with no prior knowledge on it, but only on the agents and the controllers. This was done by injecting prescribed exogenous signals. For network differentiation, we prescribed various methods of finding such constant exogenous signals. For network identification, we presented a procedure operating on a networked system that allows for the reconstruction of the underlying network by injecting a prescribed switching signal, achieved for general maximally equilibrium-independent passive agents, allowing for detection of the underlying network in a very general case. The resulting algorithms for network identification had sub-cubic time complexity. We also presented a lower bound on the complexity of any algorithm solving the network reconstruction problem, proving that the presented algorithm is optimal in sense of time complexity. We demonstrated the results in a simulation, showing it can be applied for large networks.

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

Chapter 7

Applications of the Network Optimization Framework in Fault Detection and Isolation

This section is a review of [139]. We apply the network optimization framework to study the problem of network fault detection and isolation, in which the underlying graph can change throughout the run, modeling a communication fault or a cyber attack. Our goal is to design the network controllers to either detect the existence of a fault, or overcome it completely, depending on how serious the fault is. This is done using graph-theoretic guarantees, and requires very few assumptions beyond the standard ones used for the network optimization framework.

7.1 Introduction

As we saw in previous chapters, multi-agent systems have been widely studied in recent years, as they present both a variety of applications, as well as a deep theoretical framework [105, 107, 129]. One of the deepest concerns when considering applications of multi-agent systems is communication failures, which can drive the agents to act poorly, or fail their task altogether. These communication failures can either be accidental or planned by an adversary, and there is a need of detecting communication faults and dealing with them in real-time for the network to be secure.

The classical problem of fault detection can be dealt with limit checking, signal models or process-identification methods [70]. Other approaches for dealing with fault detection in multi-agent systems include more complex methods of limit or threshold checking [39, 114, 155], mixed $\mathcal{H}_\infty/\mathcal{H}_2$ synthesis [82], and

© Technion - Israel Institute of Technology, Elyachar Central Library

switched observer or sliding mode observer methods [91,103]. These usually require either a deep physical understanding of the system, or a state-space model. We aim to give a fault detection scheme for a wide range of systems relying on another concept widespread in multi-agent systems, namely passivity. Passivity was first used to address faults by [170] for control-affine systems, although only fault-tolerance is addressed, and no synthesis procedures are suggested. Other works tried to expand on that idea, although only for linear systems [31,41,79], or for specific applications, e.g. robotics [80, 97], teleoperation [98], and hydraulic systems [99].

We aim to use the network optimization framework in order to provide a network fault detection scheme. We also focus on solving adversarial games regarding communication faults. We strive to give graph-theoretic-based results, showing that fault detection and isolation (FDI) can be done for any MEIP multi-agent system, so long that the graph $\mathcal{G}$ satisfies certain conditions. We show that if the graph $\mathcal{G}$ is "connected enough," then we can solve the FDI problem. Namely, we show that if $\mathcal{G}$ is 2-connected, then detecting the existence of any number of faults is possible, and if $\mathcal{G}$ is r-connected with $r > 2$ then we can isolate $r - 2$ faults.

The rest of this chapter is as follows. Section 7.2 presents the problem formulation and the standing assumptions throughout the chapter. Section 7.3 presents the first technical tool used for building the fault detection schemes, namely edge-indication vectors, and shows how to construct them. Section 7.4 uses edge-indication vectors to design FDI schemes, as well as strategies for adversarial games, assuming the existence of a "convergence assertion protocol", a data-driven method of asserting that a given multi-agent system converges to a conjectured limit. Section 7.5 studies these convergence assertion protocols, prescribing two data-driven model-based approaches for such protocols. Lastly, we present two case studies demonstrating the constructed algorithms.

7.2 Problem Formulation and Assumptions

This section presents the problem we aim to solve, and states the assumptions we make to tackle it. We consider a diffusively-coupled network of the form $\mathcal{N}_{\mathcal{G}} = (\mathcal{G}, \{\Sigma_i\}_{i\in\mathbb{V}}, \{\Pi_e\}_{e\in\mathbb{E}_{\mathcal{G}}})$, where $\mathcal{G} = (\mathbb{V}, \mathbb{E}_{\mathcal{G}})$ is the interaction graph, Σ_i are the agents, and Π_e are the edge controllers. For any subgraph $\mathcal{H} = (\mathbb{V}, \mathbb{E}_{\mathcal{H}})$ of $\mathcal{G}$, we can consider another diffusively-coupled network $\mathcal{N}_{\mathcal{H}} = (\mathcal{H}, \{\Sigma_i\}_{i\in\mathbb{V}}, \{\Pi_e\}_{e\in\mathbb{E}_{\mathcal{H}}})$. We can think of $\mathcal{N}_{\mathcal{H}}$ as a faulty version of $\mathcal{N}_{\mathcal{G}}$, in which the edge controllers corresponding to the edges $\mathbb{E}_{\mathcal{G}} \setminus \mathbb{E}_{\mathcal{H}}$ have malfunctioned and stopped working. Edges can fault mid-run, but we assume that once an edge has malfunctioned, it remains faulty for the remainder to the run. If we let $\mathfrak{G}$ be the collection of all nonempty subgraphs of $\mathcal{G}$, then one can think of the closed-loop diffusively-coupled system as a switched system, where the switching signal $\varsigma : [0, \infty) \to \mathfrak{G}$ designates the functioning edges at each time instant. The assumption that faulty edges remain faulty throughout the run can be described using the switching signal ς. Namely, we require that the switching signal ς is *monotone de-*

© Technion - Israel Institute of Technology, Elyachar Central Library

scending, in the sense that for all times $t_1 < t_2$, $\varsigma(t_2)$ is a subgraph of $\varsigma(t_1)$. We'll denote the number of faulty edges at time t by $\hat{\varsigma}(t)$

Now, consider a collection of agents $\{\Sigma_i\}$ and a graph $\mathcal{G}$. Fix some constant vector $\mathrm{y}^\star \in \mathbb{R}^{|\mathbb{V}|}$. Our goal is to design a control scheme for which the closed-loop network will converge to the steady-state output $\mathrm{y}^\star$. In the absence of faults, we can solve the synthesis problem as in Theorem 4.1 and Theorem 4.3. However, designing our controllers while ignoring faults might prevent the system from achieving the control goal. We now formulate the problems of fault detection and isolation:

Problem 7.1 (Network Fault Detection)**.** *Let $\{\Sigma_i\}_{i\in\mathbb{V}}$ be a set of agents, $\mathcal{G}$ be a graph, and $\mathrm{y}^\star$ be any desired steady-state output. Find a synthesis for the edge controllers such that,*

i) if no faults occur, i.e., the switching signal is $\varsigma(t) = \mathcal{G}$, $\forall t$, then the closed-loop diffusively-coupled system converges to the steady-state output $\mathrm{y}^\star$;

ii) if faults do occur, the system declares that a fault was found. More precisely, this should happen for any monotone-descending switching signal ς such that for some time t, $\varsigma(t) \neq \mathcal{G}$.

Problem 7.2 (Network Fault Isolation)**.** *Let $\{\Sigma_i\}_{i\in\mathbb{V}}$ be a set of agents, $\mathcal{G}$ be a graph, and $\mathrm{y}^\star$ be any desired steady-state output. Given some $r < |\mathbb{E}_\mathcal{G}|$, find a synthesis for the edge controllers such that for any monotone-descending switching signal ς such that $\hat{\varsigma}(t) \leq r$, $\forall t$, the closed-loop system converges to the steady-state output $\mathrm{y}^\star$.*

We now state the assumptions used throughout the chapter, made in order to tackle the problem. For the remainder of the chapter, we fix the agents $\{\Sigma_i\}$, and make the following assumption:

Assumption 7.1. *The agent dynamics $\{\Sigma_i\}$ are MEIP, and the chosen controller dynamics $\{\Pi_e\}$ are output-strictly MEIP (or vice versa). Moreover, the relations k_i^{-1}, γ_e are $\mathcal{C}^1$ functions. Furthermore, the derivative $\frac{dk_i^{-1}}{d\mathrm{y}_i}$ is positive at any $\mathrm{y}_i \in \mathbb{R}$.*

The passivity assumption assures that all the systems $\mathcal{N}_\mathcal{H}$ will globally asymptotically converge to some limit. The added smoothness assumptions, together with the positive derivative assumption, are technical assumptions that are needed to apply tools from manifold theory that will be used later. In some cases, we'll need to sense the state of the system, including the state of the controllers. In some cases, the control model is such that the controller state has a physical meaning that can be measured even for non-connected agents. For example, in the traffic control model in [8], the state η_e is the relative position of the two vehicles. However, the controller state of some systems might not have a physical meaning. For example, consider a collection of robots trying to synchronize their positions, where the output $y(t)$ is the position of each robot and the edge controllers are some PI controllers. In that case, the controller state

$\eta(t)$ has no physical meaning, and thus cannot be defined for non-connected agents. Some of the techniques developed later require us to be able to sense the state of the system, which also contains the state of the controllers. Thus, we will sometimes make the following assumption:

Assumption 7.2. *The controllers Π_e are static nonlinearities given by the functions g_e, i.e., $\mu_e = g_e(\zeta_e)$. In this case, the steady-state relation γ_e is equal to the function g_e, and the closed-loop system is $\dot{x} = f(x, -\mathcal{E}_{\mathcal{G}} g(\mathcal{E}_{\mathcal{G}}^{\top} h(x)))$*

In one of the methods below, we'll want to have a clear relationship between the measurements $h_i(x_i)$ and the storage functions $S_i(x_i)$. To achieve this, we follow Proposition 5.1 and Remark 5.1 and assume that the agents are control-affine:

Assumption 7.3. *Assumption 7.2 holds, and the agents have the form $\dot{x}_i = -f_i(x_i) + q_i(x_i)u_i$; $y_i = h_i(x_i)$. In this case, the steady-state relation γ_e is equal to the function g_e, and the closed-loop system is governed by:*

$$\dot{x}_i = -f_i(x_i) + q_i(x_i) \sum_{e=\{i,j\}} g_e(h_j(x_j) - h_i(x_i)). \tag{7.1}$$

In the next section, we'll start heading toward a solution to the FDI problems. We do so by exhibiting a method for asymptotically differentiating between the nominal dynamical system $\mathcal{N}_{\mathcal{G}}$ and the faulty dynamical systems $\mathcal{N}_{\mathcal{H}}$, in a manner similar to Theorem 6.1. Later, we'll see how this asymptotic differentiation can induce a finite-time differentiation of the systems.

7.3 Asymptotic Differentiation Between Networks

In this section, we develop the notion of edge-indication vectors, which will be used for fault detection later. In Section 6.2, the notion of indication vectors was developed. These are constant exogenous inputs used to drive the closed-loop system, chosen appropriately to give different steady-state limits for systems with identical agents and controllers, but different underlying graphs. The idea of using constant exogenous inputs to drive the system into favorable steady-state outputs was also used in Section 6.3 to give a network reconstruction algorithm with optimal time complexity, although it considers sets of multiple constant exogenous inputs applied in succession. Here, we opt for a slightly different strategy.

In Chapter 6, the problem of network reconstruction was considered, in which we cannot affect the agents, controllers, or the underlying graph. In FDI, we are doing synthesis, so we can manipulate the controllers and (in most cases) the underlying network. For that reason, we opt for a slightly different idea, in which we add a constant exogenous signal to the *output of the controllers*, that is, we consider $u(t) = -\mathcal{E}_{\mathcal{G}}(\mu(t) + \mathrm{w})$. A system implementing this control

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

law is said to have the interaction protocol (Π, w). Analogously to the notion of indication vectors, we desire that networks with identical agents and controllers, but different underlying graphs, will be forced to converge to different steady-state outputs. This is because we can monitor the output y of the system and use it to detect changes in the underlying graph, i.e., network faults. For that, we first determine what the steady-state limit is for systems $(\mathcal{G}, \Sigma, (\Pi, \mathrm{w}))$.

Proposition 7.1. *Consider a diffusively-coupled system $\mathcal{N}_{\mathcal{H}} = (\mathcal{H}, \Sigma, \Pi)$ satisfying Assumption 7.1. Suppose that $\mathrm{w} \in \mathbb{R}^{|\mathbb{E}_{\mathcal{H}}|}$ is any constant signal added to the controller output, i.e., the loop is closed as $u(t) = -\mathcal{E}_{\mathcal{H}}(\mu(t) + \mathrm{w})$. Then y is a steady-state output of the closed-loop system if and only if*

$$k^{-1}(\mathrm{y}) + \mathcal{E}_{\mathcal{H}}\gamma(\mathcal{E}_{\mathcal{H}}^{\top}\mathrm{y}) = -\mathcal{E}_{\mathcal{H}}\mathrm{w}. \tag{7.2}$$

Proof. Follows from Proposition 2.2, as the new steady-state relation for the controllers is given as $\tilde{\gamma}(\zeta) = \gamma(\zeta) + \mathrm{w}$. □

In our case, the constant signal w will be in $\mathbb{R}^{|\mathbb{E}_{\mathcal{G}}|}$, as we determine the exogenous controller output on each edge of $\mathcal{G}$. If one then considers the system $\mathcal{N}_{\mathcal{H}}$ for some $\mathcal{H} \in \mathfrak{G}$, then the exogenous controller output will be different from w, as it will only have entries of w corresponding to edges in $\mathcal{H}$. To formulate this, take any graph $\mathcal{H} \in \mathfrak{G}$, and let $P_{\mathcal{H}}$ be the linear map $\mathbb{R}^{|\mathbb{E}_{\mathcal{G}}|} \to \mathbb{R}^{|\mathbb{E}_{\mathcal{H}}|}$ removing entries corresponding to edges absent from $\mathcal{H}$. In other words, this is a $\mathbb{R}^{|\mathbb{E}_{\mathcal{H}}| \times |\mathbb{E}_{\mathcal{G}}|}$ matrix with entries in $\{0, 1\}$, whose rows are the rows of the identity matrix $\mathrm{Id} \in \mathbb{R}^{|\mathbb{E}_{\mathcal{G}}| \times |\mathbb{E}_{\mathcal{G}}|}$ corresponding to the edges of $\mathcal{H}$.

We can now define the notion of edge-indication vectors.

Definition 7.1. *Let $(\mathcal{G}, \Sigma, \Pi)$ be a closed-loop system satisfying Assumption 7.1. Let $\mathrm{w} \in \mathbb{R}^{|\mathbb{E}_{\mathcal{G}}|}$ by any vector, and for any graph $\mathcal{H} \in \mathfrak{G}$, we denote the solution of (7.2) with underlying graph $\mathcal{H}$ and exogenous input $P_{\mathcal{H}}\mathrm{w}$ by $\mathrm{y}_{\mathcal{H}}$.*

i) The vector w is called a $(\mathfrak{G}, \mathcal{H})$-edge-indication vector if for any $\mathcal{H}' \in \mathfrak{G}$, if $\mathcal{H}' \neq \mathcal{H}$ then $\mathrm{y}_{\mathcal{H}} \neq \mathrm{y}_{\mathcal{H}'}$.

ii) The vector w is called a $\mathfrak{G}$-edge-indication vector if for any two graphs $\mathcal{H}_1 \neq \mathcal{H}_2$ in $\mathfrak{G}$, $\mathrm{y}_{\mathcal{H}_1} \neq \mathrm{y}_{\mathcal{H}_2}$.

Note 7.1. *An edge-indication vector is a bias chosen on each edge in $\mathcal{G}$. This bias can be programmed into the controllers and nodes, and need not be changed nor computed on-line. In this light, for any $\mathrm{w} \in \mathbb{R}^{|\mathbb{E}_{\mathcal{G}}|}$, (7.2) transforms into*

$$k^{-1}(\mathrm{y}) + \mathcal{E}_{\mathcal{H}}\gamma(\mathcal{E}_{\mathcal{H}}^{\top}\mathrm{y}) = -\mathcal{E}_{\mathcal{H}}P_{\mathcal{H}}\mathrm{w}, \tag{7.3}$$

We wish to find a $\mathfrak{G}$-edge-indication vector for given agents and controllers, or at least a $(\mathfrak{G}, \mathcal{G})$-edge-indication vector. As in Section 6, we use randomization. We claim that random vectors are $\mathfrak{G}$-edge-indication vectors with probability 1.

© Technion - Israel Institute of Technology, Elyachar Central Library

Theorem 7.1. *Let $\mathbb{P}$ be any absolutely continuous probability measure on $\mathbb{R}^{|\mathbb{E}_\mathcal{G}|}$. Let* w *be a vector sampled according to $\mathbb{P}$. Then*

$$\mathbb{P}(\text{w } \textit{is a } \mathfrak{G}\textit{-edge-indication vector}) = 1.$$

Proof. From the definition, w is not a $\mathfrak{G}$-edge-indication vector if and only if there are two graphs $\mathcal{G}_1, \mathcal{G}_2 \in \mathfrak{G}$ such that the same vector y solves equation (7.3) for both graphs. We show that for any $\mathcal{G}_1, \mathcal{G}_2 \in \mathfrak{G}$, the probability that the two equations share a solution is zero.

Let n be the number of vertices in $\mathcal{G}$. For each graph $\mathcal{H} \in \mathfrak{G}$, define a function $F_\mathcal{H} : \mathbb{R}^n \times \mathbb{R}^{|\mathbb{E}_\mathcal{G}|} \to \mathbb{R}^n$ by $F_\mathcal{H}(\mathrm{y}, \mathrm{w}) = k^{-1}(\mathrm{y}) + \mathcal{E}_\mathcal{H}\gamma(\mathcal{E}_\mathcal{H}^\top \mathrm{y}) + \mathcal{E}_\mathcal{H} P_\mathcal{H} \mathrm{w}$. The set of steady-state exogenous input and output pairs for the system $\mathcal{N}_\mathcal{H}$ is given by the set $\mathcal{A}_\mathcal{H} = \{(\mathrm{y}, \mathrm{w}) : \ F_\mathcal{H}(\mathrm{y}, \mathrm{w}) = \mathbf{0}\}$. We note that the differential $dF_\mathcal{H}$ always has rank n. Indeed, it can be written as $[\nabla k^{-1}(\mathrm{y}) + \mathcal{E}_\mathcal{H}\nabla\gamma(\mathcal{E}_\mathcal{H}^\top \mathrm{y})\mathcal{E}_\mathcal{H}^\top, \mathcal{E}_\mathcal{H} P_\mathcal{H}]$, where $\nabla\gamma(\mathcal{E}_\mathcal{H}^\top \mathrm{y}) \in \mathbb{R}^{|\mathcal{E}_\mathcal{H}| \times n}$ and we note that the first matrix, of size $n \times n$, is positive-definite by Assumption 7.1, hence invertible.

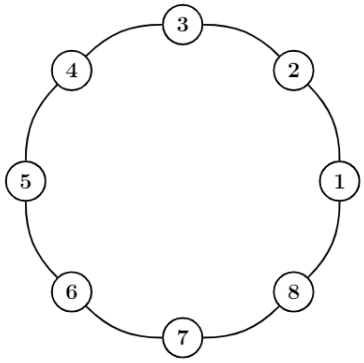
Thus, by the implicit function theorem, $\mathcal{A}_\mathcal{H}$ is a manifold of dimension $|\mathbb{E}_\mathcal{G}|$. Moreover, by Assumption 7.1, for any w there is only one corresponding steady-state output, meaning that $\mathbb{P}$ gives rise to an absolutely continuous probability measure on each manifold $\mathcal{A}_\mathcal{H}$. Thus, it's enough to show that for any $\mathcal{G}_1 \neq \mathcal{G}_2$, the intersection $\mathcal{A}_{\mathcal{G}_1} \cap \mathcal{A}_{\mathcal{G}_2}$ has dimension $\leq |\mathbb{E}_\mathcal{G}| - 1$.

To show this, we take any point $(\mathrm{y}, \mathrm{w}) \in \mathcal{A}_{\mathcal{G}_1} \cap \mathcal{A}_{\mathcal{G}_2}$. As both $\mathcal{A}_{\mathcal{G}_1}, \mathcal{A}_{\mathcal{G}_2}$ are of dimension $|\mathbb{E}_\mathcal{G}|$, it's enough to show that they do not have the same tangent space at (y, w). The tangent space of the manifold $\mathcal{A}_\mathcal{H}$ is given by the kernel of the differential $dF_\mathcal{H}(\mathrm{y}, \mathrm{w}) : \mathbb{R}^n \times \mathbb{R}^{\mathbb{E}_\mathcal{G}} \to \mathbb{R}^n$, so we show that if $\mathcal{G}_1 \neq \mathcal{G}_2$, the kernels $\ker dF_{\mathcal{G}_1}, \ker dF_{\mathcal{G}_2}$ are different at (y, w). As $\mathcal{G}_1 \neq \mathcal{G}_2$, we can find an edge existing in one of the graphs and not the other. Assume without loss of generality that the edge e exists in $\mathcal{G}_1$ but not in $\mathcal{G}_2$, and let $v = (\mathbf{0}, \mathbb{1}_e)$, where $\mathbb{1}_e$ is the vector in $\mathbb{R}^{\mathbb{E}_\mathcal{G}|}$ with all entries zero, except for the e-th entry, which is equal to 1. Then $v \in \ker dF_\mathcal{H}$ if and only if $\mathbb{1}_e \in \ker(\mathcal{E}_\mathcal{H} P_\mathcal{H})$. It's clear that $\mathbb{1}_e \notin \ker(\mathcal{E}_{\mathcal{G}_1} P_{\mathcal{G}_1})$, as $P_{\mathcal{G}_1}\mathbb{1}_e = \mathbb{1}_e$, and thus $\mathcal{E}_{\mathcal{G}_1} P_{\mathcal{G}_1}\mathbb{1}_e = \mathcal{E}_{\mathcal{G}_1}\mathbb{1}_e \neq \mathbf{0}$. Moreover, $\mathbb{1}_e \in \ker(\mathcal{E}_{\mathcal{G}_2} P_{\mathcal{G}_2})$, as $P_{\mathcal{G}_2}\mathbb{1}_e = \mathbf{0}$. Hence $\ker dF_{\mathcal{G}_1} \neq \ker dF_{\mathcal{G}_2}$ at (y, w). Thus $\mathcal{A}_{\mathcal{G}_1} \cap \mathcal{A}_{\mathcal{G}_2}$ is of dimension $\leq |\mathbb{E}_\mathcal{G}| - 1$, meaning that it is a zero-measure set inside both $\mathcal{A}_{\mathcal{G}_1}, \mathcal{A}_{\mathcal{G}_2}$. □

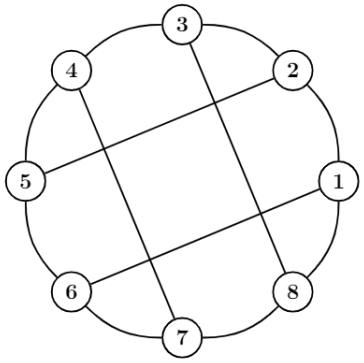
Theorem 7.1 presents a way to choose a $\mathfrak{G}$-edge-indication vector, but does not deal the control goal. One could satisfy the control goal by using Theorem 4.1 and Theorem 4.3 to solve the synthesis problem for the original graph $\mathcal{G}$, but we cannot assure we get an edge-indication vector. Note that any $\mathrm{w} \in \ker \mathcal{E}_\mathcal{G} P_\mathcal{G}$ gives a solution of (7.3) identical to the solution for $\mathrm{w} = \mathbf{0}$. Thus, choosing an exogenous control input in $\ker \mathcal{E}_\mathcal{G} P_\mathcal{G}$ does not change the steady-state output of the system $\mathcal{N}_\mathcal{G}$. However, it does change the steady-state output of all other systems $\mathcal{N}_\mathcal{H}$. This suggests to search for an edge-indication vector in $\ker \mathcal{E}_\mathcal{G} P_\mathcal{G}$. We show that this is possible if $\mathcal{G}$ is "sufficiently connected." We first explore this notion of sufficient connectivity.

Proposition 7.2 (Menger's Theorem [15])**.** *Let $\mathcal{G}$ be any connected graph. The following conditions are equivalent:*

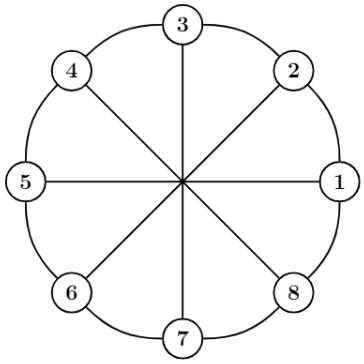
© Technion - Israel Institute of Technology, Elyachar Central Library



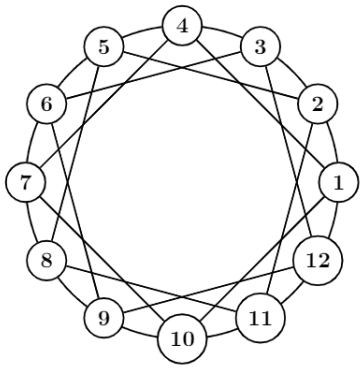
(a) A 2-connected graph on 8 nodes.



(b) A 3-connected graph on 8 nodes.



(c) Another 3-connected graph on 8 nodes.



(d) A 4-connected graph on 12 nodes.

Figure 7.1: Examples of r-connected graphs.

i) Between every two nodes there are r vertex-disjoint simple paths.

ii) For any $r-1$ vertices $v_1, \cdots, v_{r-1} \in \mathbb{V}$, the graph $\mathcal{G} - \{v_1, \cdots, v_{r-1}\}$ is connected.

Graphs satisfying either of these conditions are called r-connected graphs.

Examples of r-connected graphs can be seen in Figure 7.1. In general, there exists r-connected graphs on n nodes with $\lceil \frac{rn}{2} \rceil$ edges, but not with fewer [60]. We will take special interest in 2-connected graphs. Specifically, we can state the following theorem about edge-indication vectors in $\ker \mathcal{E}_{\mathcal{G}} P_{\mathcal{G}}$.

Theorem 7.2. *Let $\mathbb{P}$ be any absolutely continuous probability distribution on $\ker \mathcal{E}_{\mathcal{H}} P_{\mathcal{H}}$. Suppose that $\mathcal{H}$ is 2-connected. Suppose furthermore that* w *is a*

© Technion - Israel Institute of Technology, Elyachar Central Library

vector sampled according to $\mathbb{P}$*. Then*

$$\mathbb{P}(\mathrm{w} \textit{ is a } (\mathfrak{G}, \mathcal{H})\textit{-edge-indication vector}) = 1.$$

We first need to state and prove a lemma:

Lemma 7.1. *Let* $\mathcal{H}$ *be a 2-connected graph. Suppose we color the edges of* $\mathcal{H}$ *in two colors, red and blue. If not all edges have the same color, then there is a simple cycle in* $\mathcal{H}$ *with both red and blue edges.*

Proof. Suppose, heading toward contradiction, that any simple cycle in $\mathcal{H}$ is monochromatic. We claim that for each vertex x, all the edges touching x have the same color. Indeed, take any vertex x, and suppose that there are two neighbors v_1, v_2 of x such that the edge $\{x, v_1\}$ is blue and the edge $\{x, v_2\}$ is red. We note that $v_1 \to x \to v_2$ is a path from v_1 to v_2, meaning there is another path from v_1 to v_2 which does not pass through x . Adding both edges to the path yields a simple cycle with edges of both colors, as $\{x, v_1\}$ is blue and $\{x, v_2\}$ is red. Thus, each node touches edges of a single color.

Let $\mathbb{V}_{\mathrm{red}}$ be the set of nodes touching red edges, and $\mathbb{V}_{\mathrm{blue}}$ be the set of nodes touching blue edges. We know that $\mathbb{V}_{\mathrm{red}}$ and $\mathbb{V}_{\mathrm{blue}}$ do not intersect. Moreover, if we had an edge between $\mathbb{V}_{\mathrm{red}}$ and $\mathbb{V}_{\mathrm{blue}}$, it had a color. Assume, without loss of generality, it is blue. That would mean some vertex in $\mathbb{V}_{\mathrm{red}}$ would touch a blue edge, which is impossible. Thus there are no edges between $\mathbb{V}_{\mathrm{red}}$ and $\mathbb{V}_{\mathrm{blue}}$. By assumption, there is at least one edge of each color in the graph, meaning that both sets are nonempty. Thus we decomposed the set of vertices in $\mathcal{H}$ to two disjoint, disconnected sets. Thus $\mathcal{H}$ is disconnected, arriving at a contradiction and completing the proof. □

We can now prove Theorem 7.2.

Proof. We denote $m_1 = \dim\ker \mathcal{E}_{\mathcal{H}} P_{\mathcal{H}}$. The proof is similar to the proof of Theorem 7.1. We again define functions $F_{\mathcal{G}_1}$ for graphs $\mathcal{G}_1 \in \mathfrak{G}_n$ as $F_{\mathcal{G}_1}(\mathrm{y}, \mathrm{w}) = k^{-1}(\mathrm{y}) + \mathcal{E}_{\mathcal{G}_1}\gamma(\mathcal{E}_{\mathcal{G}_1}^\top \mathrm{y}) + \mathcal{E}_{\mathcal{G}_1} P_{\mathcal{G}_1} \mathrm{w}$, but this time we consider the function $F_{\mathcal{G}_1}$ as defined on the space $\ker \mathcal{E}_{\mathcal{H}} P_{\mathcal{H}} \subset \mathbb{R}^{|\mathbb{E}_{\mathcal{G}}|}$. As before, we define $\mathcal{A}_{\mathcal{G}_1} = \{(\mathrm{y}, \mathrm{w} : F_{\mathcal{G}_1}(\mathrm{y}, \mathrm{w}) = \mathbf{0}\}$. As before, we use the implicit function theorem to show that $\mathcal{A}_{\mathcal{G}_1}$ are all manifolds, but their dimension this time is $m_1 = \dim\ker \mathcal{E}_{\mathcal{H}} P_{\mathcal{H}}$. This time, we want to show that if $\mathcal{H} \neq \mathcal{G}_1$, then $\mathcal{A}_{\mathcal{H}} \cap \mathcal{A}_{\mathcal{G}_1}$ is an embedded sub-manifold of dimension $\leq m_1 - 1$, as we want to show that (with probability 1), the solutions (7.3) with graph $\mathcal{G}_1$ and graph $\mathcal{H}$ are different. As before, it's enough to show that if $(\mathrm{y}, \mathrm{w}) \in \mathcal{A}_{\mathcal{G}_1} \cap \mathcal{A}_{\mathcal{H}}$ then the kernels $\ker dF_{\mathcal{G}_1}$ and $\ker dF_{\mathcal{H}}$ are different at (y, w). We compute that for any graph $\mathcal{G}_1$,

$$dF_{\mathcal{G}_1} = [\nabla k^{-1}(\mathrm{y}) + \mathcal{E}_{\mathcal{G}_1} \nabla\gamma(\mathcal{E}_{\mathcal{G}_1}^\top \mathrm{y}) \mathcal{E}_{\mathcal{G}_1}^\top,\ (\mathcal{E}_{\mathcal{G}_1} P_{\mathcal{G}_1})|_{\ker \mathcal{E}_{\mathcal{H}} P_{\mathcal{H}}}],$$

where $\cdot|_{\ker \mathcal{E}_{\mathcal{H}} P_{\mathcal{H}}}$ is the restriction of the matrix to $\ker \mathcal{E}_{\mathcal{H}} P_{\mathcal{H}}$. Thus, if $\mathcal{G}_1$ is any graph in $\mathfrak{G}$ which is not a subgraph of $\mathcal{H}$, it contains an edge e absent from $\mathcal{H}$. Following the proof of Theorem 7.1 word-by-word, noting that $\mathbb{1}_e \in \ker \mathcal{E}_{\mathcal{H}} P_{\mathcal{H}}$,

© Technion - Israel Institute of Technology, Elyachar Central Library

we conclude that the $\ker dF_{\mathcal{G}_1}$ and $\ker dF_{\mathcal{H}}$ are different at (y, w). Thus we restrict ourselves to non-empty subgraphs $\mathcal{G}_1$ of $\mathcal{H}$.

For any collection E of edges in $\mathbb{E}_{\mathcal{H}}$, we consider $v = (\mathbf{0}, \mathbb{1}_E)$, where $\mathbb{1}_E$ is equal to $\sum_{e \in E} \mathbb{1}_e$. If E is a the set of edges of a cycle in $\mathcal{H}$, then the vector v lies in the kernel of $dF_{\mathcal{H}}$. We show that there is some cycle in $\mathcal{H}$ such that v does not lie in the kernel of $dF_{\mathcal{G}_1}$, completing the proof.

The graph $\mathcal{G}_1$ defines a coloring of the graph $\mathcal{H}$ - edges in $\mathcal{G}_1$ are colored in blue, whereas edges absent from $\mathcal{G}_1$ are colored in red. Because $\mathcal{G}_1$ is a non-empty proper subgraph of $\mathcal{H}$, this coloring contains both red and blue edges. By the lemma, there is a simple cycle in $\mathcal{H}$ that touches both red and blue edges. Let E be the set of the edges traversed by the cycle. We claim that $\mathcal{E}_{\mathcal{G}_1} P_{\mathcal{G}_1} \mathbb{1}_E \neq \mathbf{0}$, which will complete the proof of the theorem. Indeed, because the simple cycle contains both red and blue edges, we can find a vertex touching both a red edge in the cycle and a blue edge in the cycle. We let v be the vertex, and let e_1, e_2 be the corresponding blue and red edges. Recalling the cycle is simple, these are the only cycle edges touching v. However, by the coloring, we have that e_1 is in $\mathcal{G}_1$, but e_2 is not. Thus,

$$(\mathcal{E}_{\mathcal{G}_1} P_{\mathcal{G}_1} \mathbb{1}_E)_v = (\mathcal{E}_{\mathcal{G}_1})_{ve_1} (P_{\mathcal{G}_1})_{e_1 e_1} + (\mathcal{E}_{\mathcal{G}_1})_{ve_2} (P_{\mathcal{G}_1})_{e_2 e_2} = (\mathcal{E}_{\mathcal{G}_1})_{ve_1} = \pm 1 \neq 0,$$

and in particular, $\mathbb{1}_E \notin \ker \mathcal{E}_{\mathcal{G}_1} P_{\mathcal{G}_1}$. □

7.4 Network Fault Detection and Isolation

In this section, we consider two applications of the developed framework, namely fault detection and isolation, and defense strategies for adversarial games over networks. We first present a simple algorithm for network fault detection. Then, we'll discuss defense strategies for adversarial games over networks, which will require a bit more effort. Lastly, we exhibit a network fault isolation protocol, which will be a combination of the previous two algorithms. These algorithms will be used for case studies in Section 7.6. In order to apply the framework of edge-indication vectors, we need an algorithm which can improve the asymptotic differentiation we achieved in the previous section to an on-line differentiation scheme. Thus, we make the following assumption:

Assumption 7.4. *There exists an algorithm $\mathscr{A}$ which gets a model for a diffusively-coupled network $(\mathcal{G}, \Sigma, \Pi)$ and a conjectured limit $\mathrm{y}^\star$ as input, and takes measurements of the network in-run. The algorithm stops and declares "no" if and only if the network does not converge to $\mathrm{y}^\star$, and otherwise runs indefinitely. $\mathscr{A}$ is called a convergence assertion algorithm.*

For this section, we assume that such algorithm exists. We will discuss the existence of such algorithm in Section 7.5.

7.4.1 Fault Detection Over Networks

The problem of fault detection and isolation has concerned engineers for a long time. It deals with the detection of faults in a controlled plant, and overcom-

© Technion - Israel Institute of Technology, Elyachar Central Library

ing them by applying certain procedures. We focus on a communication fault scenario, in which the edges from the nominal underlying graph $\mathcal{G}$ may fault, resulting in a smaller underlying graph $\tilde{G}$, modeling communication faults between certain pairs of agents (see Problems 7.1 and 7.2). We focus on Problem 7.1 in this subsection.

To tackle the problem, we use the notion of edge-indication vectors from Section 7.3. Suppose we have MEIP agents $\{\Sigma_i\}$. We first take any output-strictly MEIP controllers $\{\Pi_e\}$ solving the classical synthesis problem, i.e., forcing the closed loop system to converge to $\mathrm{y}^\star$ (see Theorem 4.1 and Theorem 4.3). As we noted, if $\mathrm{w} \in \mathbb{R}^{|\mathbb{E}_\mathcal{G}|}$ lies in the kernel of $\mathcal{E}_\mathcal{G}$, then the solution of the following equations is the same:

$$k^{-1}(\mathrm{y}) + \mathcal{E}_\mathcal{G}\gamma(\mathcal{E}_\mathcal{G}^\top \mathrm{y}) = -\mathcal{E}_\mathcal{G} P_\mathcal{G} \mathrm{w}, \qquad k^{-1}(\mathrm{y}) + \mathcal{E}_\mathcal{G}\gamma(\mathcal{E}_\mathcal{G}^\top \mathrm{y}) = \mathbf{0}.$$

Thus, if w lies in $\ker(\mathcal{E}_\mathcal{G} P_\mathcal{G})$, running the interaction protocol (Π, w) does not change the steady-state output of the system. However, by Theorem 7.2, a random vector in $\ker(\mathcal{E}_\mathcal{G} P_\mathcal{G})$ gives a $(\mathfrak{G}, \mathcal{G})$-edge-indication vector, as long as $\mathcal{G}$ is 2-connected. In other words, if we use the interaction protocol (Π, w), where $\mathrm{w} \in \ker \mathcal{E}_\mathcal{G} P_\mathcal{G}$ is chosen randomly, then, with probability 1, all faulty systems converge to a steady-state output different from the steady-state output of the nominal faultless system, which is $\mathrm{y}^\star$. Applying the algorithm $\mathscr{A}$ allows an online, finite time distinction between the nominal faultless system and its faulty versions. We explicitly write the prescribed algorithm below:

Algorithm 6 Network Fault Detection in MEIP Multi-Agent Systems

1: Find a controller $\tilde{\Pi}$ solving the synthesis problem with graph $\mathcal{G}$, agents Σ, and control goal $\mathrm{y}^\star$ (see Theorem 4.1 and Theorem 4.3).
2: Find a basis $\{b_1, ..., b_\ell\}$ for the linear space $\ker \mathcal{E}_\mathcal{G} P_\mathcal{G}$
3: Pick a Gaussian vector $\alpha \in \mathbb{R}^\ell$ and define $\mathrm{w} = \sum_{i=1}^{\ell} \alpha_i b_i$
4: Define the interaction protocol as $(\tilde{\Pi}, \mathrm{w})$.
5: Run the system with the chosen interaction protocol.
6: Implement the algorithm $\mathscr{A}$ for the system $(\mathcal{G}, \Sigma, \Pi)$ with limit $\mathrm{y}^\star$. Declare a fault in the network if $\mathscr{A}$ declares that the system does not converge to the prescribed value.

Theorem 7.3 (Fault Detection)**.** *Suppose that n agents $\{\Sigma_i\}$ and a base graph $\mathcal{G}$ are given, and that $\{\Sigma_i\}$ satisfy Assumption 7.1. Suppose furthermore that the graph $\mathcal{G}$ is 2-connected. Then Algorithm 6 synthesizes an interaction protocol (Π, w) solving Problem 7.1, i.e., the algorithm satisfies the following properties:*

i) If no faults occur in the network, the output of the closed-loop system converges to $\mathrm{y}^\star$.

ii) The algorithm detects any number of network faults.

© Technion - Israel Institute of Technology, Elyachar Central Library

Proof. Follows from the discussion preceding Algorithm 6. Namely, Theorem 7.2 assures that w is a $(\mathfrak{G}, \mathcal{G})$-edge-indication vector, so long that $\mathcal{G}$ is 2-connected. In other words, the output of the closed-loop system with graph $\mathcal{G}$ converges to $\mathrm{y}^\star$, and for any graph $\mathcal{G} \neq \mathcal{H} \in \mathfrak{G}$, the output of the closed-loop system with graph $\mathcal{H}$ converges to a value different from $\mathrm{y}^\star$. It remains to show that the algorithm declares a fault if and only if a fault occurs. If no faults occur, then algorithm $\mathscr{A}$ does not declare a fault, so the same goes for Algorithm 6.On the contrary, suppose that any number of faults occurs in the network, and let $\mathcal{H}$ be the current underlying graph. The output of the closed loop system converges to a steady-state value $\mathrm{y} \neq \mathrm{y}^\star$, meaning that $\mathscr{A}$ eventually stops and declares a problem, and the existence of faults is detected. □

7.4.2 Multi-Agent Synthesis in the Presence of an Adversary

Consider the following 2-player game. Both players are given the same n SISO agents $\Sigma_1, \cdots, \Sigma_n$, the same graph $\mathcal{G}$ on n vertices and m edges, and the same vector $\mathrm{y}^\star \in \mathbb{R}^n$. There is also a server that can measure the state of the agents at certain intervals, and broadcast a single message to all agents once. The planner acts first, and designs a control scheme for the network and the server. The adversary acts second, removing at most r edges from $\mathcal{G}$. The system is then run. The planner wins if the closed-loop system converges to $\mathrm{y}^\star$, and the adversary wins otherwise. Our goal is to show that the planner can always win by using a strategy stemming from edge-indication vectors, assuming the agents are MEIP.

Namely, consider the following strategy. We consider all possible $\sum_{\ell=0}^{r} \binom{m}{\ell}$ underlying graphs. For each graph, the planner solves the synthesis problem as in Theorem 4.1 and Theorem 4.3. If the planner finds out the adversary changed the underlying graph to $\mathcal{H}$, he could notify the agents of that fact (through the server), and have them run the protocol solving the synthesis problem for $\mathcal{H}$. Thus the planner needs to find a way to identify the underlying graph after the adversary took action, without using the server's broadcast. This can be done by running the system with a $\mathfrak{G}$-edge-indication vector, and using the server to identify the network's steady-state. Namely, consider Algorithms 7, 9 and 8, detailing the synthesis procedure and in-run protocol for the planner. We can prove that they form a winning strategy for the planner.

Theorem 7.4. *Consider the game above. Algorithms 7, 9 and 8 describe a winning strategy for the planner. Moreover, if r is independent of n (i.e., $r = O(1)$), the synthesis algorithm has polynomial time complexity. Otherwise, the time complexity is $O(n^{cr})$ for some universal constant $c > 0$. Moreover, the size of the message broadcasted by the server is $O(r \log n)$.*

Proof. Suppose that the adversary changed the underlying graph to $\mathcal{H}$, which has entry j in Graphs. We first show that the server correctly identifies the graph. Assumption 7.4 assures that $\mathscr{A}$ never declare a fault if and only if the closed-loop system converges to the conjectured steady-state. We note that w

© Technion - Israel Institute of Technology, Elyachar Central Library

is a $\mathfrak{G}$-edge-indication vector by Theorem 7.1. Thus, the instance of convergence assertion protocol for SSLimits(j) never returns a fault, and the instances for other entries in SSLimits must eventually declare a fault. Thus the server correctly guesses the underlying graph. It then broadcasts the index j to the agents, allowing them to change the interaction protocol and run the solution of the synthesis problem with desired output $\mathrm{y}^\star$ and underlying graph $\mathcal{H}$. Thus the closed-loop system will converge to $\mathrm{y}^\star$, meaning that the planner wins.

We now move to time complexity. Note that $N = O(m^{r+1})$. The first for-loop has N iterations, and in each of them there are no more than $O(mn)$ actions done (where we save a graph in memory by its incidence matrix, which is of size $\leq m \times n$). Thus the first for-loop takes a total of $O(m^{r+2}n)$ time.

The second for-loop is a bit more complex. Solving the synthesis problem for $\mathcal{H}$ revolves around solving an equation of the form $\mathcal{E}_{\mathcal{H}}\mathrm{v} = \mathrm{v}_0$ for some known vector v_0 and an unknown v. We can solve this equation using least-squares, which takes no more than $O(\max\{m,n\}^3)$ time. As for finding the steady-state, it can be found by minimizing the equation (OPP), which takes a polynomial amount of time in n, m (e.g. via gradient descent). Recalling that $m \leq \binom{n}{2} = O(n^2)$, we conclude that if r is bounded, the total time used is polynomial in n. Moreover, if r is unbounded, the bottleneck is the first for-loop which takes $O(m^{r+2}n)$ time. Plugging $m \leq n^2$ gives a bound on the time complexity of the form $O(n^{2r+5})$. The complexity bound is now proven where we note that $n^5 = O(n^{2r})$.

Lastly, we deal with communication complexity. The message broadcasted by the server is a number between 1 and N. Thus, a total of $O(\log_2 N)$ bits are needed to transmit the message. Plugging in $N = O(m^{r+1})$ gives that the number of bits needed is $O(r \log_2 m) = O(r \log n)$. □

© Technion - Israel Institute of Technology, Elyachar Central Library

Algorithm 7 Planner Strategy in Adversarial Multi-Agent Synthesis Game with MEIP agents - Synthesis

Define $N = \sum_{\ell=0}^{r} \binom{m}{\ell}$. Let Graphs be an array with N entries, and let $j = 1$.
for $\ell = 0, \cdots, r$ **do**
  for $1 \leq i_1 < i_2 < \cdots < i_\ell \leq m$ **do**
    Insert the graph $\mathcal{H} = \mathcal{G} - \{e_{i_1}, \cdots, e_{i_\ell}\}$ to the j-th entry of Graphs. Advance j by 1.
  end for
end for
Define two arrays Controllers,SSLimits of length N.
Choose w as Gaussian random vector of length m.
for $j = 1, \cdots, N$ **do**
  Solve the synthesis problem for agents Σ and underlying graph Graphs(j) with affine controllers using Theorem 4.1 and Theorem 4.3. Insert the result into Controllers(j)
  Compute the steady-state limit of the closed-loop system with agents Σ, underlying graph Graphs(j), and interaction protocol (Π, w), where $\Pi_e : \mu_e = \zeta_e$. Insert the result into SSLimits(j)
end for

Algorithm 8 Planner Strategy - In-Run Protocol for Server

Define HasFaulted as an array of zeros of size N.
while HasFaulted has at least two null entries **do**
  Run N instances of the algorithm $\mathscr{A}$ simultaneously, with conjectured steady-states from SSLimits.
  for $j = 1$ to N **do**
    if The j-th instance declared "no" **then**
      Change the value of HasFaulted(j) to 1.
    end if
  end for
end while
Find an index j such that HasFaulted(j) $= 0$. Broadcast the message j to the agents.

Algorithm 9 Planner Strategy - In-Run Protocol for Agents

Run the interaction protocol (Π, w) where the controllers Π are the static nonlinearities $\mu_e = \zeta_e$.
When a message j is received, run the interaction protocol described by Controllers(j).

© Technion - Israel Institute of Technology, Elyachar Central Library

7.4.3 Network Fault Isolation in Multi-Agent Systems

We now deal with the problem of fault isolation, in which we wish to deal with faults occurring throughout the run. This time, unlike fault detection, we do not only want to identify the existence of faults, but also overcome them and allow the system to achieve its goal. This problem can be thought of as a tougher hybrid of the previous two problems - in Subsection 7.4.1, the faults could appear throughout the run, but we only needed to find out they exist. In Subsection 7.4.2, all of the faults occur before the system starts running, but we had to figure out a way to overcome them, not just identify them. Motivated by this view, we can offer a hybrid solution.

Ideally, the interaction protocol will have two disjoint phases - a first, "stable" phase in which the underlying graph is known and no extra faults have been found, and a second, "exploratory" phase in which extra faults have been found, and the current underlying graph is not yet known. The first phase can be solved by using the Fault Detection Algorithm 6, as long as the current underlying graph is 2-connected. The second state can be solved by the pre-broadcast stage of the planner strategy described in Algorithms 7, 9, and 8.

The main issue with this algorithm as it stands is what happens if the underlying graph changes again once we are in the exploratory phase. Suppose we entered the exploratory phase with underlying graph $\mathcal{H}_1$, but before identifying the graph as $\mathcal{H}_1$, it changed to $\mathcal{H}_2$. Recall that in the exploratory phase, we run an instance of $\mathscr{A}$ on all of the possible graphs simultaneously, continuing until no more than one instance has yet declared a fault. If the instance related to graph $\mathcal{H}_2$ has not declared a fault yet, then it will not display a fault from now on, unless another fault occurs before the exploratory phase is done. If the same instance has already declared a fault, then we have a problem - all other instances will eventually also declare a fault. There are two possibilities in this case.

The first option is that one instance will declare a fault last, meaning that we find a time in which all but one instances have declared a fault. In this case, we identify the graph as some $\mathcal{H}_3$. However, when we return to the stable phase and run the system with interaction protocol (Π, w) corresponding to $\mathcal{H}_3$, a fault will be declared and we'll return to the exploratory phase. Indeed, the vector w synthesized for the said interaction protocol is a $(\mathfrak{G}, \mathcal{H}_3)$-edge-indication vector, meaning that the de-facto steady-state limit (with graph $\mathcal{H}_2$) will be different than the conjectured steady-state limit (with graph $\mathcal{H}_3$), and $\mathscr{A}$ will declare a fault. The second option is that the last few instances of the convergence assertion protocol declare a fault together, reaching a stage in which all instances have declared a fault. We deal with this situation by restarting the exploratory phase. Thus, we get the synthesis algorithm and in-run protocol presented in Algorithms 10 and 11. We claim that these solve the fault isolation problem. We prove:

Theorem 7.5. *Let $\Sigma_1, \cdots, \Sigma_n$ be agents satisfying Assumption 7.1, and let $\mathcal{G}$ be a k-connected graph for $k \geq 3$ on n vertices and m edges. Then algorithms 10 and 11, run with $r = k - 2$, can detect and isolate up to r faults.*

© Technion - Israel Institute of Technology, Elyachar Central Library

Proof. We refer to steps 2 to 3 of Algorithm 11 as the stable phase of the algorithm, and to steps 4 to 13 as the exploratory phase. We claim it's enough to prove the following claims:

i) If we are in the stable phase, the graph $\mathcal{H} = \text{Graphs}(j)$ is the current underlying graph, and no more faults occur throughout the run, then the closed-loop system converges to $y^\star$.

ii) If we are in the stable phase, but the graph $\mathcal{H} = \text{Graphs}(j)$ is not the current underlying graph, then we will eventually move to the exploratory phase.

iii) Each instance of the exploratory phase eventually ends.

iv) If an instance of the exploratory phase is executed after the last fault of the run happened, it correctly identifies the current underlying graph.

We first explain why the theorem follows from these claims, and then show that they are true. Suppose a total of $\ell \leq r$ faults occur throughout the run. Let $T < \infty$ be the time at which the last fault occurs. We look at the phase of the system at times $t > T$. If we arrive at the stable phase with the correct graph, then the system converges to $y^\star$ (claim 7.4.3 If we start an exploratory phase, then it eventually ends and the stable phase starts with the correct graph (claims 7.4.3 and 7.4.3), implying the system converges to $y^\star$. If we are in the stable phase with a wrong graph, then we eventually leave it and start an exploratory phase (claim 7.4.3), which, as we saw, implies that the system converges to $y^\star$. Lastly, we could be in the middle of an instance of the exploratory phase. In that case, the instance eventually ends (claim 7.4.3), after which we either apply a new instance of the exploratory phase, or the stable phase (either with a correct or with an incorrect graph). In both cases, we saw that the system converges to $y^\star$. As these are all the possible cases, the system must converges to $y^\star$.

We now prove the claims. Note that if $\mathcal{H}$ can be yielded by removing no more than r edges from $\mathcal{G}$, then it is 2-connected. Indeed, if the removed edges are $e_1, ..., e_\ell$, choose a vertex v_i for each of them, so that $\mathcal{H}$ contains the graph $\mathcal{H}_1 = \mathcal{G} - \{v_1, \cdots, v_\ell\}$. For any vertex $v \neq v_1, \cdots, v_\ell$, because $\ell \leq r$ and $\mathcal{G}$ is $(r+2)$-connected, $\mathcal{H}_1 - \{v\} = \mathcal{G} - \{v_1, \cdots .v_\ell, v\}$ is connected. Thus $\mathcal{H}_1$, and hence $\mathcal{H}$, is 2-connected.

By Theorem 7.2, for each j, the vector w_j from the interaction protocol $\text{IP}(j)$ is a $(\mathfrak{G}, \text{Graphs}(j))$-edge-indication vector. As all graphs achieved by removing no more than r edges from $\mathcal{G}$ are non-empty, we conclude that for every j_1, j_2, if the system is run with interaction protocol $\text{IP}(j_1)$, the system with underlying graph $\text{Graphs}(j_1)$ will converge to a different value from the system with underlying graph $\text{Graphs}(j_2)$. We thus conclude by Assumption 7.4, that claims 7.4.3 and 7.4.3 are true.

Algorithm 10 Synthesis for Network Fault Isolation

1: Define $N = \sum_{\ell=0}^{r} \binom{m}{\ell}$. Let Graphs be an array with N entries, and let $j = 1$
2: **for** $\ell = 0, \cdots, r$ **do**
3: **for** $1 \leq i_1 < i_2 < \cdots < i_\ell < m$ **do**
4: Insert the graph $\mathcal{H} = \mathcal{G} - \{e_{i_1}, \cdots, e_{i_\ell}\}$ to the j-th entry of Graphs. Advance j by 1.
5: **end for**
6: **end for**
7: Define two arrays IP,SSLimits of length N.
8: Choose w as a Gaussian random vector of length m.
9: Choose controllers $\{\Pi_e\}_{e \in \mathbb{E}}$ satisfying Assumption 7.1.
10: **for** $j = 1, \cdots, N$ **do**
11: Run steps 1-4 of Algorithm 6. Insert the resulting interaction protocol into IP(j).
12: Compute the steady-state limit of the closed-loop system with the interaction protocol (Π, w). Insert the result into SSLimits(j)
13: **end for**

Algorithm 11 In-Run Protocol for Network Fault Isolation

1: Find the index j for which Graphs(j) $= \mathcal{G}$.
2: Command the agents to change their interaction protocol to the one described in IP(j). Define $\mathcal{H}$ = Graphs(j).
3: Run $\mathscr{A}$ for the closed-loop system with graph $\mathcal{H}$ and interaction protocol IP(j). Only if the algorithm declares a fault, continue to step 4.
4: Define HasFaulted as an array of zeros of size N.
5: Change the agents' interaction protocol to (Π, w).
6: **while** HasFaulted has at least two null entries **do**
7: Run N instances of the convergence assertion protocol $\mathscr{A}$ simultaneously, with conjectured steady-states from SSLimits.
8: **for** $j = 1$ to N **do**
9: **if** The j-th instance has declared a fault **then**
10: Change the value of HasFaulted(j) to 1.
11: **end if**
12: **end for**
13: **end while**
14: **if** HasFaulted has no entries equal to zero **then**
15: Go to step 4.
16: **end if**
17: Find an index j such that HasFaulted(j) $= 0$. Set $\mathcal{H}$ = Graphs(j). Go to step 2.

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

We now prove claims 7.4.3 and 7.4.3. By Theorem 7.1, the chosen vector w is a $\mathfrak{G}$-edge-indication vector. Thus, for any two different graphs Graphs(j_1) and Graphs(j_2), the steady-state output will be different. Thus, by Assumption 7.4, at least $N-1$ instances of $\mathscr{A}$ must eventually declare a fault (as there is only one true steady-state), even if the underlying graph changed while running this phase. This proves claim 7.4.3. Moreover, suppose that the last fault of the run happened before we started executing this instance of the exploratory phase. The true underlying graph appears in Graphs, as it is achieved from $\mathcal{G}$ by removing no more than r edges. If it the true underlying graph is equal to Graphs(j), then by Assumption 7.4, the j-th instance of the convergence assertion method will never declare a fault. Thus, the last remaining non-zero entry of HasFaulted is the j-th, meaning that we correctly identify the current underlying graph. This proves claim 7.4.3 and completes the proof of the theorem. □

Remark 7.1. *We can use a similar protocol to isolate more complex faults. We consider the collection of subgraphs $\mathcal{H}$ of $\mathcal{G}$ in which there is a set of vertices of size $\leq r$, so that each edge in $\mathcal{G}-\mathcal{H}$ touches at least one vertex in the set. This observation allows us to offer similar FDI algorithms for more complex types of faults. For example, we can consider a case in which each agent communicates with all other agents by a single transceiver, and if it faults, then all edges touching the corresponding vertex are removed from the graph. We can even use a hybrid fault model, in which faults correspond to certain subsets of edges touching a common vertex are removed from the graph. For example, suppose there are two distant groups of agents. Agents in the same group are close, and communicate using Bluetooth communication. Agents in different groups are farther, and communicate using Wi-Fi (or broadband cellular communication). When an agent's Bluetooth transceiver faults, all inter-group edges are removed, and when the Wi-Fi transceiver faults, all intra-group edges are removed.*

7.5 Online Assertion of Network Convergence

In the previous section, we used the notion of edge-indication vectors, together with Assumption 7.4, to suggest algorithms for FDI. The goal in this section is to propose algorithms $\mathscr{A}$ satisfying Assumption 7.4. This will be achieved by using convergence estimates, relying on passivity. First, we revisit a result from [23].

Proposition 7.3 ([23])**.** *Let $(\mathrm{u},\mathrm{y},\zeta,\mu)$ be a steady-state of $(\mathcal{G},\Sigma,\Pi)$. Suppose that the agents Σ_i are passive with respect to $(\mathrm{u_i},\mathrm{y_i})$ with passivity index $\rho_i \geq 0$, and that the controllers Π_e are passive with respect to $(\zeta_\mathrm{e},\mu_\mathrm{e})$, with passivity index $\nu_e \geq 0$. Let $S_i(x_i)$ and $W_e(\eta_e)$ be the agents' and the controllers' storage functions. Then $S(x,\eta)=\sum_{i\in\mathbb{V}} S_i(x_i)+\sum_{e\in\mathbb{E}} W_e(\eta_e)$ is a positive-definite $\mathcal{C}^1$-function, which nulls only at the steady-states (x,η) corresponding to $(\mathrm{u_i},\mathrm{y_I})$*

and (ζ_e, μ_e)*, and satisfying the inequality:*

$$\frac{dS}{dt} \leq -\sum_{i\in\mathbb{V}} \rho_i (y_i(t) - y_i)^2 - \sum_{e\in\mathbb{E}} \nu_e (\mu_e(t) - \mu_e)^2. \tag{7.4}$$

Proof. The proof follows immediately from S_i, W_e being positive-definite $\mathcal{C}^1$-functions nulling only at x_i, η_e, from summing the following inequalities:

$$\frac{dS_i}{dt} \leq (u_i(t) - u_i)(y_i(t) - y_i) - \rho_i (y_i(t) - y_i)^2$$
$$\frac{dW_e}{dt} \leq (\mu_e(t) - \mu_e)(\zeta_e(t) - \zeta_e) - \nu_e (\mu_e(t) - \mu_e)^2,$$

and using the equality $(u(t) - u)^\top (y(t) - y) = -(\mu(t) - \mu)^\top \mathcal{E}^\top (y(t) - y) = -(\mu(t) - \mu)(\zeta(t) - \zeta)$. □

The inequality (7.4) can be thought of as a way to check that the system is functioning properly. Indeed, we can monitor x, y, η, and μ, and check that the inequality holds. If it doesn't, there must have been a fault in the system. This idea has a few drawbacks, linked to one another. First, as we commented in Section 7.2, in some networks, the controller state $\eta_e(t)$ can be defined only for existing edges, so using $\eta(t)$ requires us to know the functioning edges, which is absurd. Thus, in some cases, we must use Assumption 7.2. Second, in practice, even if we have access to x, we cannot measure it continuously. Instead, we measure it at certain time intervals. One can adapt (7.4) to an equivalent integral form:

$$S(x(t_{k+1}), \eta(t_{k+1})) - S(x(t_k), \eta(t_k)) \leq -\int_{t_k}^{t_{k+1}} (\sum_{i\in\mathbb{V}} \rho_i \Delta y_i(t)^2 + \sum_{e\in\mathbb{E}} \nu_e \Delta\mu_e(t)^2) dt, \tag{7.5}$$

where $\Delta y_i = y_i(t) - y_i$ and $\Delta\mu_e = \mu_e(t) - \mu_e$. However, this gives rise to the third problem - unlike the function S, we can't assure that the functions $(y_i(t) - y_i)^2$ and $(\mu_e(t) - \mu_e)^2$ (or their sum) is monotone. Thus, we cannot correctly estimate the integral appearing on the right-hand side of the inequality.

We present two approaches to address this problem. First, we try and estimate the integral using high-rate sampling, by linearizing the right hand side of (7.5) and bounding the error. Second, we try to bound the right-hand side as a function of S, resulting in an inequality of the form $\dot{S} \leq -\mathcal{F}(S)$, which will give a convergence estimate.

7.5.1 Asserting Convergence Using High-Rate Sampling

Consider the inequality (7.5), and suppose $t_{k+1} - t_k = \Delta t_k$ is very small. Thus, the functions $y_i(t) - y_i$ and $\mu_e(t) - \mu_e$ are roughly constant in the time period used for the integral. More precisely, recalling that $y = h(x)$ and $\mu = \phi(\eta, \mathcal{E}^\top y)$, and assuming these functions are differentiable near $x(t_k), \eta(t_k)$, we expand the

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

right-hand side of (7.5) to a Taylor series,

$$\int_{t_k}^{t_{k+1}} \left(\sum_{i \in \mathbb{V}} \rho_i \Delta y_i(t)^2 + \sum_{e \in \mathbb{E}} \nu_e \Delta \mu_e(t)^2 \right) dt = \tag{7.6}$$
$$\left(\sum_{i \in \mathbb{V}} \rho_i \Delta y_i(t_k)^2 + \sum_{e \in \mathbb{E}} \nu_e \Delta \mu_e(t_k)^2 \right) \Delta t_k + O(\Delta t_k^2).$$

We wish to give a more explicit bound on the $O(\Delta t_k^2)$ term. We consider the following function G, defined on the interval $[t_k, t_{k+1}]$ by the formula

$$G(t) = \sum_i \rho_i (y_i(t) - \mathrm{y}_i)^2 + \sum_e (\mu_e(t) - \mu_e)^2. \tag{7.7}$$

The equation (7.6) is achieved from the approximation $G(t) = G(t_k) + O(|t - t_k|)$ which is true for differentiable functions. Using Lagrange's mean value theorem for $t \in [t_k, t_{k+1}]$, we find some point $s \in (t, t_{k+1})$ such that $G(t) = G(t_k) + \frac{dG}{dt}(s)(t - t_k)$. If we manage to bound the time derivative $\frac{dG}{dt}$ in the interval $[t_k, t_{k+1}]$, we'll be able to find a computational way to assert convergence. By the chain rule, the time derivative of G is given by

$$\frac{dG}{dt} = \sum_{i \in \mathbb{V}} \rho_i (y_i(t) - \mathrm{y}_i) \dot{y}_i + \sum_{e \in \mathbb{E}} \nu_e (\mu_e(t) - \mu_e) \dot{\mu}_e. \tag{7.8}$$

In order to compute the time derivative of y_i, μ_i, we recall that both are functions of x and η, namely $y = h(x)$ and $\mu = \phi(\eta, \mathcal{E}^\top y) = \phi(\eta, \mathcal{E}^\top h(x))$. Thus, we have that

$$\begin{cases} \dot{y} &= \nabla_x h(x(t)) \dot{x} \\ \dot{\mu} &= \nabla_\eta \phi(\eta(t), \zeta(t)) \dot{\eta} + \nabla_x \phi(\eta(t), \zeta(t)) \mathcal{E}^\top \nabla h(x(t)) \dot{x}, \end{cases} \tag{7.9}$$

where $\zeta(t) = \mathcal{E}^\top h(x(t))$, $\dot{x} = f(x, u) = f(x, -\mathcal{E}\phi(\eta, \zeta))$, and $\dot{\eta} = \psi(\eta, \zeta) = \psi(\eta, \mathcal{E}^\top h(x))$.

Thus we can write the time derivative of G as a continuous function of $x(t), \eta(t)$, as we plug the expressions for $\dot{y}, \dot{\mu}$ into (7.8). However, we do not know the value of $x(t), \eta(t)$ between measurements.

To tackle this problem, we provide a bound for dG/dt. We do this by noticing that we have some information on where $x(t), \eta(t)$ can lie. Namely, we have equation (7.4), showing that $S(x(t), \eta(t))$ is a monotone descending function. Thus, we know that $x(t), \eta(t)$ lies in the set $\mathbb{B} = \{(x, \eta) : \ S(x, \eta) \leq S(x(t_k), \eta(t_k))\}$. More precisely, we can show the following.

Proposition 7.4. *Assume the functions h_i, f_i, ϕ_e, ψ_e are all continuously differentiable. Then for any time $t \in [t_k, t_{k+1}]$, the following inequality holds:*

$$\left| \frac{dG}{dt}(t) \right| \leq (\rho_\star M_{\Delta y} M_{\dot{y}} + \nu_\star M_{\Delta \mu} M_{\dot{\mu}, x}) M_{\dot{x}} + \nu_\star M_{\Delta \mu} M_{\dot{\mu}, \eta} M_{\dot{\eta}},$$

where

$$
\begin{aligned}
M_{\dot{x}} &= \max_{(x,\eta)\in\mathbb{B}} \|f(x, -\mathcal{E}_{\mathcal{G}}\phi(\eta, \mathcal{E}_{\mathcal{G}}^{\top} h(x))\|,\\
M_{\dot{\eta}} &= \max_{(x,\eta)\in\mathbb{B}} \|\psi(\eta, \mathcal{E}_{\mathcal{G}}^{\top} h(x))\|,\\
M_{\delta y} &= \max_{(x,\eta)\in\mathbb{B}} \|h(x) - h(\mathrm{x})\|,\\
M_{\delta\mu} &= \max_{(x,\eta)\in\mathbb{B}} \|\psi((\eta, \mathcal{E}_{\mathcal{G}}^{\top} h(x)) - \mu\|,\\
M_{\dot{y}} &= \max_{(x,\eta)\in\mathbb{B}} \|\nabla_x h(x)\|,\\
M_{\dot{\mu},x} &= \max_{(x,\eta)\in\mathbb{B}} \|\nabla_{\zeta}\phi(\eta, \mathcal{E}_{\mathcal{G}}^{\top} h(x))\mathcal{E}_{\mathcal{G}}^{\top}\nabla_x h(x)\|,\\
M_{\dot{\mu},\eta} &= \max_{(x,\eta)\in\mathbb{B}} \|\nabla_{\eta}\phi(\eta, \mathcal{E}_{\mathcal{G}}^{\top} h(x))\|,
\end{aligned}
$$

$\rho_\star = \max_i \rho_i$, $\nu_e = \max_e \nu_e$, *and* $\mathbb{B} = \{(x,\eta) :\ S(x,\eta) \le S(x(t_k), \eta(t_k))\}$.

Proof. We fix some $t \in [t_k, t_{k+1}]$, so that $(x(t), \eta(t)) \in \mathbb{B}$. We use the expressions for $\dot{x}, \dot{\eta}, \dot{y}, \dot{\mu}$ found in (7.9). First, the conditions $\|\dot{x}\| \le M_{\dot{x}}$ and $\|\dot{\eta}\| \le M_{\dot{\eta}}$ are obvious. Equation (7.9) shows that $\|\dot{y}\| \le M_{\dot{y}} M_{\dot{x}}$ and $\|\dot{\mu}\| \le M_{\dot{\mu},x} M_{\dot{x}} + M_{\dot{\mu},\eta} M_{\dot{\eta}}$. Thus, by using Cauchy-Schwartz inequality on (7.8), we obtain $\left|\frac{dG}{dt}\right| \le \rho_\star M_{\delta y}\|\dot{y}\| + \nu_\star M_{\delta\mu}\|\dot{\mu}\|$, concluding the proof. □

Remark 7.2. *Suppose that Assumption 7.3 holds, so that the agents are given by* $\dot{x}_i = -f_i(x_i) + q_i(x_i)u_i; y_i = h_i(x_i)$ *and the controllers are given by* $\mu = g(\zeta)$. *In that case,* $\dot{x} = -f(x) + u = -f(x) - \mathcal{E}_{\mathcal{G}} g(\mathcal{E}_{\mathcal{G}}^{\top} x)$ *and* $\mu = g(\mathcal{E}^{\top} h(x))$, *so we can get a slightly more comprehensive bound by applying the same analysis. Namely,*

$$
\left|\frac{dG}{dt}(t)\right| \le (\rho_\star M_{\delta y} M_{\dot{y}} + \nu_\star M_{\delta\mu} M_{\dot{\mu}})(M_q M_u + M_f)
$$

where

$$
\begin{aligned}
M_u &= \max_{x\in\mathbb{B}} ||\mathcal{E}_{\mathcal{G}} g(\mathcal{E}_{\mathcal{G}}^{\top} h(x))||,\\
M_f &= \max_{x\in\mathbb{B}} ||f(x)||,\\
M_q &= \max_{x\in\mathbb{B}} \max_{i\in\mathbb{V}} |q_i(x)|,\\
M_{\delta y} &= \max_{x\in\mathbb{B}} ||h(x) - h(\mathrm{x})||,\\
M_{\delta\mu} &= \max_{x\in\mathbb{B}} ||g(\mathcal{E}_{\mathcal{G}}^{\top} h(x)) - g(\mathcal{E}_{\mathcal{G}}^{\top} h(\mathrm{x}))||,\\
M_{\dot{y}} &= \max_{x\in\mathbb{B}} ||\nabla h(x)||,\\
M_{\dot{\mu}} &= \max_{x\in\mathbb{B}} ||\nabla g(\mathcal{E}_{\mathcal{G}}^{\top} h(x))\mathcal{E}_{\mathcal{G}}^{\top}\nabla h(x)||,
\end{aligned}
$$

, $\rho_\star = \max_i \rho_i$, $\nu_e = \max_e \nu_e$, *and* $\mathbb{B} = \{x :\ S(x) \le S(x(t_k))\}$.

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

Corollary 7.1. *Fix any two times $t_k < t_{k+1}$, and consider the notation of Proposition 7.4. Then the following inequality holds:*

$$S(x(t_{k+1})) - S(x(t_k)) \leq -\left(\sum_i \rho_i \Delta y_i(t_k)^2 + \sum_{e\in\mathbb{E}} \nu_e \Delta\mu_e(t_k)^2\right)\Delta t_k + \frac{M}{2}\Delta t_k^2, \tag{7.10}$$

where

$$M = (\rho_\star M_{\Delta y} M_{\dot{y}} + \nu_\star M_{\Delta\mu} M_{\dot{\mu},x}) M_{\dot{x}} + \nu_\star M_{\Delta\mu} M_{\dot{\mu},\eta} M_{\dot{\eta}}.$$

Proof. Recall that $G(t) = \sum_{i\in\mathbb{V}} \rho_i (y_i(t) - \mathrm{y}_i)^2 + \sum_{e\in\mathbb{E}} \nu_e(\mu_e(t) - \mu_e)^2$. By Proposition 7.4, for every $t \in [t_k, t_{k+1}]$ we have $G(t) \leq G(t_{k+1}) + M|t - t_{k+1}|$. Thus (7.5) implies that $S(x(t_{k+1})) - S(x(t_k)) \leq G(t_{k+1})\Delta t_k + \frac{M}{2}\Delta t_k^2$. □

The proposes a mathematically-sound method for asserting convergence of the output $y(t)$ to y. One samples the $y(t)$, $x(t), \eta(t)$, and $\mu(t)$ at times $t_1, t_2, t_3, \ldots$. At every time instance t_{k+1}, one checks that the inequality (7.10) holds. We show that when $\Delta t_k \to 0$, this method asserts that the output of the system converges to the said value. In other words, assuming we sample the system at a high-enough rate, we can assert that it converges very closely to the supposed steady-state output. Indeed, we prove the following.

Proposition 7.5. *Let $t_1, t_2, \cdots$, be any monotone sequence of times such that $t_k \to \infty$, and suppose that the inequality (7.10) holds for any k. Then for any $\varepsilon > 0$, there are infinitely many $N > 0$ such that $\sum_{i\in\mathbb{V}} \rho_i \Delta y_i(t_N)^2 + \sum_{e\in\mathbb{E}} \nu_e \Delta\mu_e(t_N)^2 < \frac{M}{2}\Delta t_N + \varepsilon$. More precisely, for any two times $t_{N_1} \leq t_{N_2}$, if $t_{N_2} \geq t_{N_1} + \varepsilon^{-1} S(x(t_{N_1}), \eta(t_{N_1}))$, then there exists some $k \in \{N_1, N_1+1, \cdots N_2\}$ such that $\sum_{i\in\mathbb{V}} \rho_i \Delta y_i(t_k)^2 + \sum_{e\in\mathbb{E}} \nu_e \Delta\mu_e(t_k)^2 < \frac{M}{2}\Delta t_k + \varepsilon$.*

The proposition can be thought of as a close-convergence estimate. The left-hand side, viewed as a function of x, η, is a non-negative smooth function, which nulls only at the steady-state (x, η). Thus it is small only when $x(t), \eta(t)$ are close to (x, η), and because we know that $S(x(t), \eta(t))$ is monotone descending, once the trajectory arrives near (x, η), it must remain near (x, η). One might ask why "infinitely many times" is more useful in this case. Indeed, it does not add any more information if the time intervals Δt_k are taken as a constant (i.e., we sample the system at a constant rate). However, we can measure the system at an ever-increasing rate, at least theoretically. Taking $\Delta t_k \to 0$ (while still having $t_k \to \infty$, e.g. $t_k = 1/k$), we see that we must have $x(t) \to \mathrm{x}$ and $\eta(t) \to \eta$, meaning we can use the proposition to assert convergence. We now prove the proposition.

Proof. It's enough to show that for each $\varepsilon > 0$ and any $N_1 > 0$, there's some $N > N_1$ such that $\sum_{i\in\mathbb{V}} \rho_i \Delta y_i(t_N)^2 + \sum_{e\in\mathbb{E}} \nu_e \Delta\mu_e(t_N)^2 < \frac{M}{2}\Delta t_N + \varepsilon$. Indeed, suppose that this is not the case. Then for any $k > N_1$, the right-hand side of

(7.10) is upper-bounded by $-\varepsilon\Delta t_k$. Thus we can sum the telescopic series and show that for any $k > N_1$,

$$S(x(t_k)) - S(x(t_{N_1})) \leq -\sum_{j=N_1+1}^{k} \varepsilon\Delta t_j = -\varepsilon(t_k - t_{N_1}), \tag{7.11}$$

meaning that as $k \to \infty$, we get that $S(x(t_k), \eta(t_k)) \to -\infty$. This is absurd, as $S \geq 0$. Thus there must exist some $N > N_1$ such that $\sum_{i\in\mathbb{V}} \rho_i \Delta y_i(t_N)^2 + \sum_{e\in\mathbb{E}} \nu_e \Delta\mu_e(t_N)^2 < \frac{M}{2}\Delta t_N + \varepsilon$. The second part of the proposition follows from (7.11) and the demand that $S(x(t_k)) \geq 0$. □

Proposition 7.5 can be used for convergence assertion. We can consider the following scheme - begin at time t_0 and state x_0, η_0. We want to show that $S(x(t), \eta(t)) \to 0$. We instead show that $G(t)$, defined in (7.7), gets arbitrarily close to 0. As we said, this is enough as $G(t)$ is a $\mathcal{C}^1$ non-negative function of the state $x(t), \eta(t)$ that is only small when $x(t), \eta(t)$ is close to the steady-state $(\mathrm{x}, \boldsymbol{\eta})$. We prove:

Theorem 7.6. *Consider the algorithm $\mathscr{A}$, defined in the following form. Sample the system at times $t_1, t_2, \cdots$, and check whether the inequality* (7.10) *holds. If it does, continue, and if does not, the stop and declare "no." Then there exists a sequence $t_1, t_2, \cdots$, depending on the system and the initial conditions, such that $\mathscr{A}$ satisfies Assumption 7.4.*

Proof. By the discussion above, and the fact that $S(x(t), \eta(t))$ is a monotone descending function, it's enough to show that $\liminf_{k\to\infty} G(t_k) = 0$. We present the following method of choosing $t_1, t_2, \cdots$. We first choose $t_0 = 0$, an arbitrary $\delta_1 > 0$, compute M as in Proposition 7.4, and choose $\Delta_1 t = \frac{\delta_1}{M}$ and $\varepsilon = \frac{\delta_1}{2}$. Sample the system at rate $\Delta_1 t$ until time $t_{N_1} > t_0 + \varepsilon^{-1}(S(x_0, \eta_0))$. Now define $\delta_2 = \delta_1/2$ and repeat the process above. We claim that $\mathscr{A}$, with this choice of sample times, satisfies Assumption 7.4. If the diffusively-coupled network $(\mathcal{G}, \Sigma, \Pi)$ converges to $(\mathrm{x}, \boldsymbol{\eta})$, then Corollary 7.1 implies that the algorithm never stops, as required. It remains to show that if the algorithm never stops, then the system $(\mathcal{G}, \Sigma, \Pi)$ converges to the conjectured limit. Indeed, we first show at some point, $G(t) < \delta_1$. By choice of $\Delta_1 t$, if the inequality (7.10) holds at each time, then when we reach time t_{N_1}, we know that at some point, we had $G(t) \leq \frac{M}{2}\Delta t + \epsilon = \delta_1$. Reiterating shows that at some times $t_k^\star$, $G(t_k^\star) \leq \delta_k$, where $\delta_k = \frac{\delta_1}{2^k}$, so $\liminf_{k\to\infty} G(t_k) = 0$. □

The term "High-Rate Sampling" comes from the fact that if M is not updated when we re-iterate with smaller δ, then eventually, $t_{k+1} - t_k \to 0$, which is impractical in real-world cases. However, we note that the number M decreases as $S(x(t), \eta(t))$ decreases, as shown in Proposition 7.4. Thus, if M is updated between iterations, we might have $\Delta t \not\to 0$.

Remark 7.3. *There is a trade-off between the time-steps Δt and the time it takes to find a point in which $G(t) < \frac{M}{2}\Delta t_N + \varepsilon$, which is $t = \frac{S(x(0),\eta(0))}{\varepsilon}$. On one*

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

hand, we want larger time-steps (to avoid high-rate sampling) and shorter overall times, however, increasing both Δt and ε creates a worse eventual bound on $G(t)$. We can choose both by maximizing an appropriate cost function $C(\Delta t, \varepsilon)$, monotone in both Δt and ε, subject to $\frac{M}{2}\Delta t + \varepsilon = \delta_1, \varepsilon \geq 0, \Delta t \geq 0$. Choosing $C(\Delta t, \varepsilon)$ as linear is inadvisable, as the maximizing a linear function with linear constraints always leads to the optimizer being on the boundary, which means either $\Delta t = 0$ or $\varepsilon = 0$. The choice $\Delta t = \frac{\delta_1}{M}$ and $\varepsilon = \frac{\delta_1}{2}$ mentioned above corresponds to the geometric average cost function $C(\Delta t, \varepsilon) = \sqrt{\Delta t \varepsilon}$. Other choices of C can express practical constraints, e.g. relative apathy to large convergence times relative to high-rate sampling should result in a cost function penalizing small values of Δt more harshly than small values of ε.

7.5.2 Asserting Convergence Using Convergence Profiles

For this subsection, we now assume that Assumption 7.3 holds and that the agents are output-strictly MEIP, i.e., that $\rho_i > 0$. Consider (7.4) and suppose there is a non-negative monotone function $\mathcal{F}$ such that for any t, the right-hand side of (7.4) is bounded from above by $-\mathcal{F}(S)$. In that case, we get an estimate of the form $\dot{S} \leq -\mathcal{F}(S)$. This is a weaker estimate than (7.4), but it has a more appealing discrete-time form,

$$S(x(t_{k+1})) - S(x(t_k)) \leq -\int_{t_k}^{t_{k+1}} \mathcal{F}(S(x(t)))dt \leq -\mathcal{F}(S(x(t_{k+1}))) \cdot (t_{k+1} - t_k), \tag{7.12}$$

where we use the monotonicity of $\mathcal{F}$ and the fact that $S(x(t))$ is monotone non-ascending. Recalling that S is a sum of the functions $S_i(x_i)$, due to Assumption 7.3, we focus on the elements of the right-hand side of (7.4) corresponding to the agents, and neglect the ones corresponding to controllers. As the controllers are passive, we have $\nu_e \geq 0$, so removing the said term does not change the inequality's validity.

In order to find $\mathcal{F}$, it's natural to look for functions Ω_i satisfying $\Omega_i(S_i) \leq (y_i(t) - \mathrm{y}_i)^2$. We define the existence of the functions Ω_i properly in the following definition.

Definition 7.2. *Let $\Omega : [0, \infty) \to [0, \infty)$ be any function on the non-negative real numbers. We say that an autonomous system has the* convergence profile *(ρ, Ω) with respect to the steady-state (u, y) if there exists a $\mathcal{C}^1$ storage function $S(x)$ such that the following inequalities hold:*

i) $\frac{dS(x(t))}{dt} \leq (u(t) - \mathrm{u})(y(t) - \mathrm{y}) - \rho(y(t) - \mathrm{y})^2$,

ii) $\Omega(S(x(t))) \leq (y(t) - \mathrm{y})^2$.

Example 7.1. *Consider the SISO system Σ defined by $\dot{x} = -x + u$, $y = x$, and consider the steady-state input-output pair $(0, 0)$. The storage function $S(x(t)) = \frac{1}{2}x(t)^2$ satisfies*

$$\dot{S}(x(t)) = x(t)\dot{x}(t) = (u(t) - 0)(y(t) - 0) - (y(t) - 0)^2.$$

© Technion - Israel Institute of Technology, Elyachar Central Library

Thus Σ *has convergence profile* $(1, \Omega)$ *for* $\Omega(\theta) = \frac{1}{2}\theta$.

More generally, when considering an LTI system with no input-feedthrough, both functions $S(x)$ and $(y(t)-\mathrm{y})^2$ are quadratic in x. Thus there is a monotone linear function Ω such that the inequality $\Omega(S(x(t)) \leq (y(t) - \mathrm{y})^2$ holds. In particular, the function Ω exists in this case. We can show that the functions Ω exist for general cases.

Theorem 7.7. *Let* Σ *be the SISO system of the form* $\dot{x} = -f(x) + q(x)u$, $y = h(x)$. *Suppose* q *is a positive continuous function, that* f/q *is* $\mathcal{C}^1$ *and monotone ascending and that* h *is* $\mathcal{C}^1$ *and strictly monotone ascending. Let* $(\mathrm{u} = f(\mathrm{x})/q(\mathrm{x}), y = h(\mathrm{x}))$ *be any steady-state input-output pair of the system. Then*

i) using the storage function $S(x) = \int_{\mathrm{x}}^{x} \frac{h(\sigma)-h(\mathrm{x})}{q(\sigma)} d\sigma$, *the system* Σ *has the convergence profile* (ρ, Ω) *for a strictly ascending function* Ω *and* $\rho = \inf_x \frac{f(x)-f(\mathrm{x})}{h(x)-h(\mathrm{x})} \geq 0$;

ii) suppose there exists some $\alpha > 0$ *such that the limit* $\lim_{x\to\mathrm{x}} \frac{|h(x)-h(\mathrm{x})|}{|x-\mathrm{x}|^\alpha}$ *exists and is finite. Then the limit* $\lim_{\theta\to 0} \frac{\Omega(\theta)}{\theta^\beta}$ *exists and is finite, where* $\beta = \frac{2\alpha}{\alpha+1}$ *.In other words, if* h *behaves like a power law near* x, *then* Ω *behaves like a power law near* 0.

Proof. We build the function Ω in the following way. For every $\theta \geq 0$, we define the set $\mathbb{A}_\theta = \{x \in \mathbb{R} : \ (h(x) - h(\mathrm{x}))^2 \leq \theta\}$. We want that $x \in \mathbb{A}_\theta$ would imply that $\Omega(S(x)) \leq \theta$. Because h is continuous and monotone, it's clear that $\mathbb{A}_\theta$ is an interval containing x. Now, let ω be the function on $[0, \infty)$ defined as $\omega(\theta) = \sup_{x\in\mathbb{A}_\theta} S(x)$. We note that ω can take infinite values (e.g. when h is bounded, but S is not). However, we show that the restriction of ω on $\{\theta : \ \omega(\theta) < \infty\}$ is strictly monotone. If we show that this claim is true, then ω has an inverse function which is also strictly monotone. Define $\Omega = \omega^{-1}$ as the strictly monotone inverse function. By definition, for any $x \in \mathbb{R}$ we have that $x \in \mathbb{A}_\theta$ for $\theta = (h(x) - h(\mathrm{x}))^2$, so $S(x) \leq \omega(\theta)$. Thus $\Omega(S(x)) \leq \Omega(\omega(\theta)) = \theta = (h(x) - h(\mathrm{x}))^2$, concluding the first part of the proof.

We now prove that the restriction of ω on $\{\theta : \ \omega(\theta) < \infty\}$ is strictly monotone. It's clear that if $0 \leq \theta_1 < \theta_2$ then the interval $\{x : \ (h(x) - h(\mathrm{x}))^2 \leq \theta_1\} = \mathbb{A}_{\theta_1}$ is strictly contained in the interval $\{x : \ (h(x)-h(\mathrm{x}))^2 \leq \theta_2\} = \mathbb{A}_{\theta_2}$, as h is strictly monotone. Moreover, It's clear that S is strictly ascending in $[\mathrm{x}, \infty)$ and strictly descending in $(-\infty, \mathrm{x}]$, as the function $\frac{h(x)-h(\mathrm{x})}{g(x)}$ is positive on (x, ∞) and negative on $(-\infty, \mathrm{x})$. Thus we have $\omega(\theta_1) < \omega(\theta_2)$, unless $\omega(\theta_1) = \infty$, which is what we wanted to prove.

We now move to the second part of theorem, in which we show that if h behaves like a power law near x, then Ω behaves like a power law near zero. We use big-O notation (in the limit $x \to \mathrm{x}$). By assumption and strict monotonicity of h, we have:

$$h(x) - h(\mathrm{x}) = C\mathrm{sgn}(x - \mathrm{x})|x - \mathrm{x}|^\alpha + o(|x - \mathrm{x}|^\alpha), \tag{7.13}$$

© Technion - Israel Institute of Technology, Elyachar Central Library

for some constant $C > 0$, implying

$$(h(x) - h(\mathrm{x}))^2 = C^2|x - \mathrm{x}|^{2\alpha} + o(|x - \mathrm{x}|^{2\alpha}).$$

By definition, we conclude that for $\theta > 0$ small enough, $\mathbb{A}_\theta$ is an interval centered at x and has radius $\theta^{1/2\alpha}/C^{1/\alpha} + o(\theta^{1/2\alpha})$. We recall that $S(x) = \int_{\mathrm{x}}^{x} \frac{h(\sigma)-h(\mathrm{x})}{q(\sigma)} d\sigma$. We write $q(x) = q(\mathrm{x}) + o(1)$ as q is continuous, so (7.13) implies that $\frac{h(\sigma)-h(\mathrm{x})}{q(\sigma)} = \frac{1}{q(\mathrm{x})}(C\mathrm{sgn}(x - \mathrm{x})|x - \mathrm{x}|^{\alpha} + o(|x - \mathrm{x}|^{\alpha}))$. We conclude that $S(x) = \frac{C}{q(\mathrm{x})(\alpha+1)}|x - \mathrm{x}|^{\alpha+1} + o(|x - \mathrm{x}|^{\alpha+1})$. We can now compute $\omega(\theta)$ by definition, using our characterization of $\mathbb{A}_\theta$. We get:

$$\begin{aligned}\omega(\theta) = \max_{x \in \mathbb{A}_\theta} S(x) &= \max_{x \in \mathbb{A}_\theta} \left(\frac{C}{q(\mathrm{x})(\alpha+1)}|x - \mathrm{x}|^{\alpha+1} + o(|x - \mathrm{x}|^{\alpha+1}) \right) \\ &= \frac{C}{q(\mathrm{x})(\alpha+1)} \left(\frac{\theta^{\frac{1}{2\alpha}}}{C^{1/\alpha}} \right)^{\alpha+1} + o((\theta^{\frac{1}{2\alpha}})^{\alpha+1}) = (D + o(1))\theta^{\frac{\alpha+1}{2\alpha}}\end{aligned}$$

for $D = \frac{1}{q(\mathrm{x})(\alpha+1)C^{1/\alpha}} > 0$. Thus, the inverse function $\Omega(\theta)$ is given as $\Omega(\theta) = (D^{-\frac{2\alpha}{1+\alpha}} - o(1))\theta^{\frac{2\alpha}{1+\alpha}}$, as plugging this expression gives $\omega(\Omega(\theta)) = \theta$. This completes the proof. □

Example 7.2. *Consider a system with $g(x) = 1, h(x) = \sqrt[3]{x}$ and a steady state $\mathrm{u} = \mathrm{x} = \mathrm{y} = 0$. $h(x)$ behaves like a power law with power $\alpha = \frac{1}{3}$. Part 7.7 of Theorem 7.7 implies that Ω also behaves like a power law, with power $\beta = \frac{2\alpha}{\alpha+1} = \frac{1}{2}$. We exemplify the computation of Ω as done in the proof, and show it behaves like a power law with $\beta = \frac{1}{2}$, as forecasted by the theorem. Indeed, we have $S(x) = \int_0^x \sqrt[3]{\sigma} d\sigma = \frac{3}{4}x^{4/3}$, and $(h(x) - h(\mathrm{x}))^2 = x^{2/3}$. For every $\theta \geq 0$, we have $\mathbb{A}_\theta = \{x : \ x^{2/3} \leq \theta\} = [-\theta^{1.5}, \theta^{1.5}]$. Thus*

$$\omega(\theta) = \sup_{x \in \mathbb{A}_\theta} S(x) = \sup_{|x| \leq \theta^{1.5}} \frac{3}{4}x^{4/3} = \frac{3}{4}(\theta^{1.5})^{4/3} = \frac{3}{4}\theta^2,$$

implying that Ω, the inverse function of ω, is given by $\sqrt{\frac{4}{3}\theta}$, and one observes that actually $(h(x) - h(\mathrm{x}))^2 = \Omega(S(x))$.

Remark 7.4. *Theorem 7.7 gives a prescription to design the function Ω. However, some steps, namely the inversion of ω, are computationally hard. For example, if $h(x) = 1 - e^{-x}$ and $g(x) = 1$, then $\omega(\theta) = \log_e \frac{1}{1-\sqrt{\theta}} - \sqrt{\theta}$ for $\theta < 1$ and $\omega(\theta) = \infty$ for $\theta \geq 1$, which is almost impossible to invert analytically. To solve this problem, we can either precompute the different values of Ω numerically and store them in a table, or approximate them on-line using the bisection method. The strength of Theorem 7.7 is that it shows that a function Ω can always be found, implying this method is always applicable.*

Up until now, we managed to transform the equation (7.5) to the equation $\frac{dS}{dt} \leq \sum_i -\rho_i \Omega_i(S_i)$, for some non-negative monotone functions Ω_i. This is

© Technion - Israel Institute of Technology, Elyachar Central Library

closer to an inequality of the form $\dot{S} \leq -\mathcal{F}(S)$, but we still cannot use it without high-rate sampling, as we cannot assume that $S_i(x_i(t))$ are monotone decreasing. We want to transform the right hand side into a function of S. We note that $\Omega_i(\theta_i) = 0$ only at $\theta_i = 0$, as $S_i = 0$ happens only at x_i. We claim the following:

Proposition 7.6. *Let $\rho_1, ..., \rho_n$ be any positive numbers, and let $\Omega_1, ..., \Omega_n : [0, \infty) \to [0, \infty)$ be any $\mathcal{C}^1$ strictly monotone functions such that $\Omega_i(\theta_i) = 0$ only at $\theta_i = 0$. Suppose further that for any i there exists some $\beta_i > 0$ such that the limit $\lim_{\theta_i \to 0} \frac{\Omega_i(\theta_i)}{\theta_i^{\beta_i}}$ exists and is positive. Define $\Omega_\star : [0, \infty) \to [0, \infty)$ as $\Omega_\star(\theta) = \min_i \Omega_i(\theta)$. Then for every $D > 0$, there exists some constant $C > 0$ such that for all $D \geq \theta_1, \cdots, \theta_n \geq 0$, we have $\sum_{i=1}^n \rho_i \Omega_i(\theta_i) \geq C \cdot \Omega_\star(\sum_{i=1}^n \theta_i)$.*

Proof. Without loss of generality, we assume that $\Omega_i = \Omega_\star$ for all i. Indeed, we note that $\sum_{i=1}^n \rho_i \Omega_\star(\theta_i) \geq C\Omega_\star(\sum_{i=1}^n \theta_i)$ implies the desired inequality. We also assume that $\rho_i = 1$ for all i, as $\sum_i \Omega_i(\theta_i) \geq C \cdot \Omega_\star(\sum_i \theta_i)$ implies $\sum_i \rho_i \Omega_i(\theta_i) \geq C \min_i \rho_i \cdot \Omega_\star(\sum_i \theta_i)$. Define $F : [0, D]^n \backslash \{0\} \to \mathbb{R}$ as

$$F(\theta_1, \cdots, \theta_n) = \frac{\sum_{i=1}^n \Omega_\star(\theta_i)}{\Omega_\star(\sum_{i=1}^n \theta_i)},$$

where the claim is equivalent to F being bounded from below. For any $r > 0$, F is continuous on the compact set $[0, D]^n \backslash \{x : \ ||x|| > r\}$, so its minimum is obtained at some point. As F does not vanish on the set, the minimum is positive, so F is bounded from below on that set by a constant greater than zero. It remains to show that $\lim_{\theta_1, \cdots, \theta_n \to 0} \inf F(\theta_1, \cdots, \theta_n) > 0$. Let $\beta = \max_i \beta_i$, so that $\lim_{\theta \to 0} \frac{\Omega_\star(\theta)}{\theta^\beta} > 0$. Then

$$F(\theta_1, \cdots, \theta_n) = \frac{\sum_{i=1}^n \Omega_\star(\theta_i)}{\Omega_\star(\sum_{i=1}^n \theta_i)} = \frac{\sum_{i=1}^n \Omega_\star(\theta_i)}{(\sum_{i=1}^n \theta_i)^\beta} \cdot \frac{(\sum_{i=1}^n \theta_i)^\beta}{\Omega_\star(\sum_{i=1}^n \theta_i)}.$$

We want to bound both factors from below when $\theta_1, \cdots, \theta_n \to 0$. It's clear that the second factor is equal to $\frac{1}{\lim_{\theta \to 0} \frac{\Omega_\star(\theta)}{\theta^\beta}}$, which is a positive real number by assumption. As for the first factor, we can bound it as

$$\lim_{\theta_1 \cdots \theta_n \to 0} \frac{\sum_{i=1}^n \Omega_\star(\theta_i)}{(\sum_{i=1}^n \theta_i)^\beta} \geq \lim_{\theta_1 \cdots \theta_n \to 0} \frac{\Omega_\star(\max_i \theta_i)}{(n \max_i \theta_i)^\beta} > 0$$

as $\sum_{i=1}^n \theta_i \leq n \max_i \theta_i$ and $\sum_i \Omega_\star(\theta_i) \geq \Omega_\star(\max_i \theta_i)$ by monotonicity of $\Omega_\star$. This completes the proof. □

Corollary 7.2. *Let $S_1, ..., S_n$ be the storage functions of the agents, let $S = \sum_i S_i$, and let $\Omega_1, \cdots, \Omega_n$ be $\mathcal{C}^1$ strictly monotone functions such that $\Omega_i(\theta_i) = 0$ only at $\theta_i = 0$. Suppose that for any i there exists some $\beta_i > 0$ such that the limit $\lim_{\theta_i \to 0} \frac{\Omega_i(\theta_i)}{\theta_i^{\beta_i}}$ exists and is positive. Moreover, Suppose that $\dot{S} \leq \sum_i \rho_i \Omega_i(S_i)$. Then for every bounded set $B \subset \mathbb{R}^n$ there exists a constant $C > 0$ such that for any trajectory of the closed-loop system with initial condition in B, the inequality $\dot{S} \leq -C \cdot \Omega_\star(S)$ holds, where $\Omega_\star(\theta) = \min_i \Omega_i(\theta)$.*

© Technion - Israel Institute of Technology, Elyachar Central Library

Proof. Use $\theta_i = S_i$ and $D = S(x(0))$ in Proposition 7.6. □

Proposition 7.6 and Corollary 7.2 show that an inequality of the form (7.12) can be achieved, so long that the functions Ω_i from Theorem 7.7 "behave nicely" around 0, namely don't grow faster nor slower than a power law. This condition is very general, and only excludes pathologies as $\Omega(\theta) = \frac{1}{\log(1/\theta)}$, growing faster than any power law, and $\Omega(\theta) = \exp(-1/\theta^2)$, growing slower than any power law. Note that Theorem 7.7 shows that if h behaves like a power law near x, then so does Ω, so pathological functions $\Omega_\star$ can only come from pathological measurement functions h_i. We show it's enough to check the discretized equation (7.12) to assert convergence.

Proposition 7.7. *Let $\Omega_\star : [0,\infty) \to [0,\infty)$ be any continuous function such that $\Omega_\star(\theta) = 0$ only at $\theta = 0$. Let $\tilde{S}(t)$ be any time-dependent monotone decreasing function $\tilde{S} : [0,\infty) \to [0,\infty)$. Let $t_1, t_2, t_3, \cdots$ be any unbounded sequence of times such that $\liminf_{k\to\infty}(t_{k+1} - t_k) > 0$, and suppose that for every k, the inequality $\tilde{S}(t_{k+1}) - \tilde{S}(t_k) \leq -\Omega_\star(\tilde{S}(t_{k+1}))(t_{k+1} - t_k)$ holds. Then $\tilde{S}(t) \to 0$ as $t \to \infty$.*

Proof. By assumption $\tilde{S}(t_k)$ is monotone decreasing and bounded from below, as $\tilde{S}(t_k) \geq 0$. Thus it converges to some value, denoted $\tilde{S}_\infty$. Using $\tilde{S}(t_{k+1}) - \tilde{S}(t_k) \leq -\Omega_\star(\tilde{S}(t_{k+1}))(t_{k+1} - t_k)$ and taking $k \to \infty$ gives that $0 \leq -\Omega_\star(\tilde{S}_\infty)$. However, $\Omega_\star$ is non-negative, so we must have $\Omega_\star(\tilde{S}_\infty) = 0$, and thus $S_\infty = 0$, meaning that $\tilde{S}(t_k) \to 0$. By monotonicity of $\tilde{S}$, we conclude that $\tilde{S}(t) \to 0$ as $t \to \infty$. □

We want to use $\tilde{S}(t) = S(x(t))$. The results above suggest an algorithm for convergence assertion.

Algorithm 12 Convergence Assertion using Convergence Profile

Input: A diffusively-coupled system $(\mathcal{G}, \Sigma, \Pi)$, an initial condition $x(0)$ and a conjectured steady-state $\hat{\mathrm{x}}$.

1: Define $S_i(x_i) = \int_{\hat{\mathrm{x}}_i}^{x_i} \frac{h_i(\sigma_i) - h_i(\hat{\mathrm{x}}_i)}{q(\sigma_i)} d\sigma_i$
2: Let $S(x) = \sum_{i\in\mathbb{V}} S_i(x_i)$.
3: Use Theorem 7.7, Proposition 7.6 and Corollary 7.2 to find a function Ω such that $\dot{S} \leq -\Omega(S)$ for all times t, with initial condition $x(0)$.
4: Choose $\delta_0 = S(x(0))$ and $t_0 = 0$
5: **for** $k = 1, 2, 3, \cdots$ **do**
6: Define $\delta_k = \delta_{k-1}/2$.;
7: Define $M = \min_{x:\ S(x)\geq\delta_k} \Omega(S(x))$
8: Take some $t_k > t_{k-1} + \frac{S(x_0)}{M}$
9: Sample the system at time t_{k+1}.
10: **if** $S(x(t_{k+1})) - S(x(t_k)) \not\leq -\Omega(S(x(t_{k+1})))(t_{k+1} - t_k)$ **then**
11: Stop and return "no";
12: **end if**
13: **end for**

© Technion - Israel Institute of Technology, Elyachar Central Library

Theorem 7.8. *Algorithm 12, taking the system $(\mathcal{G}, \Sigma, \Pi)$, the initial state $x(0)$, and the conjectured steady-state $\hat{\mathrm{x}} = h^{-1}(\mathrm{y})$ as input, satisfies Assumption 7.4.*

Proof. We denote the true limit of the system $(\mathcal{G}, \Sigma, \Pi)$ by x. We first assume that the algorithm never stops, and show that $\hat{\mathrm{x}} = \mathrm{x}$. We show that $S(x(t_k)) \leq \delta_k$, which would suffice as $\delta_k \to 0$ and $S(t) \to 0$ implies that $x(t) \to \hat{\mathrm{x}}$, and thus $\mathrm{x} = \hat{\mathrm{x}}$. Suppose, heading toward contradiction, that $S(x(t_k)) \not\leq \delta_k$. Then $\Omega(S(x(t_k))) \geq M$, meaning that the right-hand side of the checked inequality is larger than $-S(x(t_k))$. Thus, if the inequality holds then $S(x(t_{k+1})) < 0$, which is absurd. Thus $S(x(t_k)) \leq \delta_k$, and $\hat{\mathrm{x}} = \mathrm{x}$. On the contrary, if the conjectured limit $\hat{\mathrm{x}}$ is the true limit of the network, then Theorem 7.7, Proposition 7.6 and Corollary 7.2 show that $S(x(t_{k+1})) - S(x(t_k)) \leq -\Omega(S(x(t_{k+1})))(x(t_{k+1}) - x(t_k))$ always holds, so the algorithm never stops, as expected. □

Remark 7.5. *Although we can prove convergence with this method using very seldom measurements, we should still sample the system at a reasonable rate. This is because we want to detect faults as soon as possible. If we sample the system in too large intervals, we won't be able to sense a fault until a large amount of time has passed.*

We conclude this section with a short discussion about the perks and drawbacks of the two presented convergence assertion methods. The convergence profile method allows the designer to sample the system at any desired rate, allowing one to prove convergence using very seldom measurements. Moreover, it gives certain rate of convergence guarantees before running the system. On the contrary, the high-rate sampling method can require a long time to assert convergence to a δ-ball around the desired steady-state, unless one is willing to increase the sampling rate, perhaps arbitrarily. However, it's main upshot over the convergence profile method is that we need not assume that Assumption 7.3 holds, and that the method is computationally easier, as one can avoid function inversion which is needed to compute the function Ω.

7.6 Case Studies

We consider two case studies. First, we apply our FDI scheme for a network of LTI systems. Second, we apply the FDI scheme for a network of velocity-coordinating vehicles.

7.6.1 Network FDI for LTI First Order Systems

We consider a network satisfying Assumptions 7.1 and 7.3, where the agents are first order systems of the form $G_i(s) = \frac{1}{\tau_i s + 1}$ with correlation times $\tau_i > 0$, the edge controllers are static gains of the form $\mu_e = b_e \zeta_e$, and the interaction graph $\mathcal{G}$ is as described by Figure 7.2(a). We note the graph $\mathcal{G}$ is 4-connected [19, Graph ID = 32659]. The parameters τ_i were chosen as log-uniformly between 0.1 and 10, and the parameters b_e were chosen log-uniformly between 0.1 and 10.

© Technion - Israel Institute of Technology, Elyachar Central Library

We wish to solve the synthesis problem, augmenting the control protocol so that the closed-loop system converges to $\mathrm{y}^\star = [0,0,0,1,1,2,2,2,2,2,2]^\top$, allowing up to 2 edges to fault. We run our FDI protocol, where we implement the profile-based convergence assertion scheme, and sample the system at 2_{Hz} (i.e., a modified version of Algorithm 12). We consider four scenarios. Each scenario 100 seconds long.

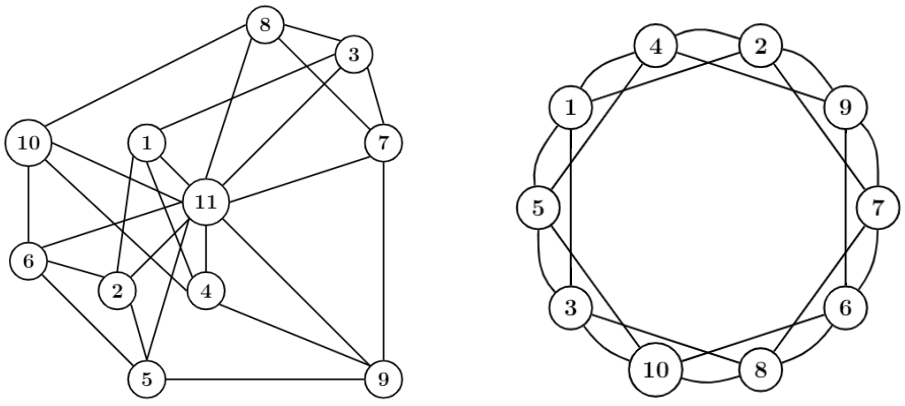
i) A faultless scenario

ii) At time $t = 20_{sec}$, the edge $\{3,8\}$ faults, and at time $t = 50_{sec}$, the edge $\{1,11\}$ faults.

iii) At time $t = 20_{sec}$, the edge $\{3,8\}$ faults, and at time $t = 21_{sec}$, the edge $\{1,11\}$ faults.

iv) At time $t = 1_{sec}$, the edge $\{3,8\}$ faults, and at time $t = 4_{sec}$, the edge $\{1,11\}$ faults.

The first scenario tests the nominal behavior of the protocol. The second tests its ability to handle single faults at a time. The third tests its ability to handle more than one fault at a time. The last tests its ability to deal with faults before the system converged. The results of the four scenarios are available in Figures 7.3(a), 7.3(b), 7.4(a), and 7.4(b). It can be seen that we achieve the control goal for all four scenarios. Moreover, in all scenarios and at all times, the state of the agents is not too far from the values found in $y^\star$, meaning that this protocol cannot harm the agents by demanding them to have very wild states. In the second and third scenario, the exploratory phases begins at the first measurement after the fault occurred. On the contrary, in the fourth scenario, it takes the exploratory phase begins only at $t = 3.5_{sec}$, 2.5 seconds after the



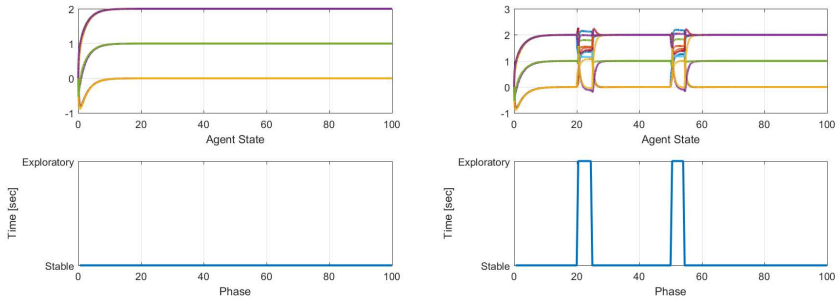
(a) Graph in LTI case study.

(b) Graph in vehicle case study.

Figure 7.2: Graphs for case studies.

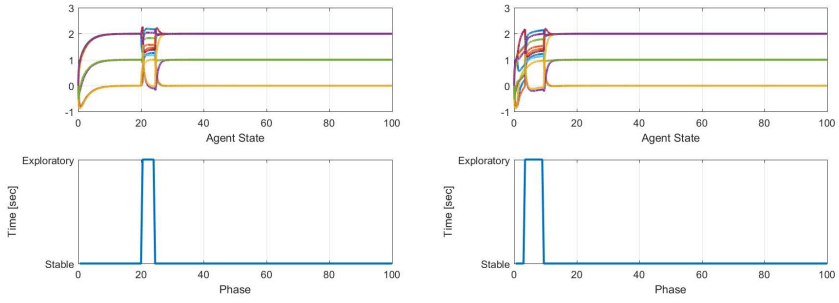
© Technion - Israel Institute of Technology, Elyachar Central Library

first fault. This is because the steady-states of the faulty and nominal closed-loop systems are relatively close, so it takes a little extra time to find that a fault exists. The same scenario was run with the high-rate sampling convergence assertion protocol as well. The faults were identified slightly quicker, but the sampling rate peaked at about 120_{Hz} in some cases.



(a) Results of first scenario. (b) Results of second scenario.

Figure 7.3: First set of scenarios in fault detection and isolation for LTI systems.



(a) Results of third scenario. (b) Results of fourth scenario.

Figure 7.4: Second set of scenarios in faulty detection and isolation for LTI systems.

7.6.2 Network FDI for Velocity-Coordinating Vehicles

We consider a diffusively-coupled network satisfying both Assumptions 7.1 and 7.3. The network consists of $n = 10$ vehicles trying to coordinate their velocity. Each agent is modeled as $\dot{x}_i = \kappa_i(-x_i + V_0^i + V_1^i u_i)$, where $\kappa_i > 0$ is an internal gain, V_0^i is the "preferred" velocity, and $V_1^i > 0$ is the "sensitivity" to other

© Technion - Israel Institute of Technology, Elyachar Central Library

vehicles [8, 23]. Unlike in [8], the planner tries to force the agents' velocities to a certain steady-state. The edge controllers are static nonlinearities given by sigmoid functions of the form $\mu_e = \tanh(\zeta_e)$. The interaction graph $\mathcal{G}$ is as described by Figure 7.2(b). We note that the graph $\mathcal{G}$ is 4-connected [19, Graph ID = 21063]. The gains κ_i were chosen as log-uniformly between 0.3 and 10, the sensitivities V_1^i were chosen log-uniformly between 0.1 and 1, and the preferred velocities V_0^i were chosen as normal with mean $\mu = 60_{km/h}$ and standard deviation $\sigma = 15_{km/h}$. The initial velocity of the agents was chosen to be Gaussian with mean $\mu = 70_{km/h}$ and standard deviation $\sigma = 20_{km/h}$. We wish to solve the synthesis problem, forcing the closed-loop system to converge to $\mathrm{y}^\star = [60, 70, 50, 60, 70, 50, 60, 70, 50, 60]^\top_{km/h}$, allowing up to 2 edges to fault. We run our FDI protocol, where we implement the profile-based convergence assertion scheme, and sample the system at 2_{Hz}(i.e., a modified version of Algorithm 12). We consider four different scenarios. Each scenario has length of 100 seconds.

i) A faultless scenario

ii) At time $t = 20_{sec}$, the edge $\{3, 5\}$ faults, and at time $t = 50_{sec}$, the edge $\{6, 9\}$ faults.

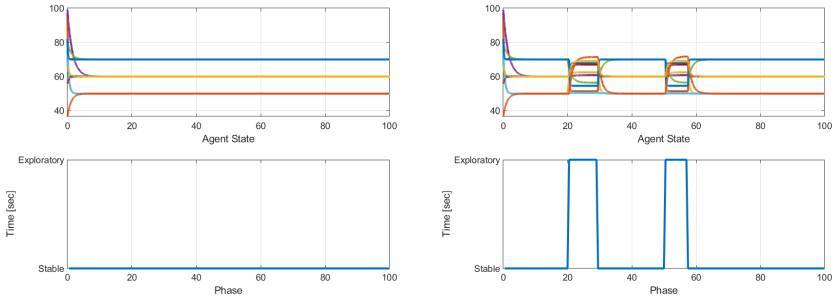
iii) At time $t = 20_{sec}$, the edge $\{3.5\}$ faults, and at time $t = 21_{sec}$, the edge $\{6, 9\}$ faults.

iv) At time $t = 1_{sec}$, the edge $\{3, 5\}$ faults, and at time $t = 4_{sec}$, the edge $\{6, 9\}$ faults.

The scenarios were chosen for similar reasons as in the previous case study. The results of the four scenarios are available in Figures 7.5(a), 7.5(b), 7.6(a), and 7.6(b). It can be seen that we achieve the control goal for all four scenarios. Moreover, in all scenarios and at all times, the velocities of the agents are not too far from the values found in $y^\star$, meaning that this protocol cannot harm the agents by demanding them to have very wild states. In the second and third scenario, the exploratory phases begins at the first measurement after the fault occurred. On the contrary, in the fourth scenario, it takes the exploratory phase begins only at $t = 2_{sec}$, a second after the first fault. As before, this is because the steady-states of the faulty and nominal closed-loop system are relatively close, meaning it takes a little extra time to find that a fault exists.

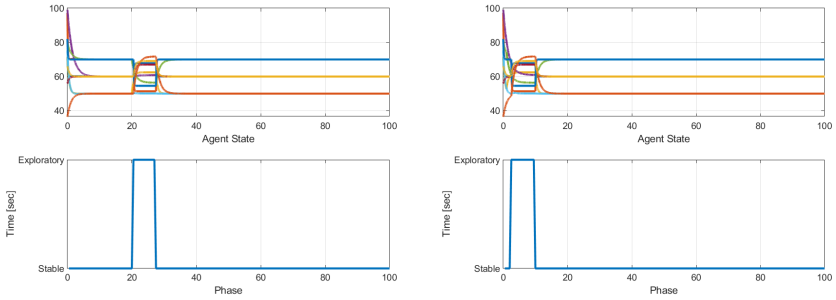
7.7 Conclusions

We considered multi-agent networks in which the agents are output-strictly MEIP and the controllers are MEIP. We considered a protocol in which the nominal controller output $\mu(t)$ is added an exogenous constant signal w. We showed that if w is chosen randomly, no matter what the underlying graph $\mathcal{G}$ is, then we can asymptotically differentiate between any two versions (faulty or faultless) of the system. We also showed that if w is chosen randomly in



(a) Results of first scenario. (b) Results of second scenario.

Figure 7.5: First set of scenarios in fault isolation for vehicle systems.



(a) Results of third scenario. (b) Results of fourth scenario.

Figure 7.6: Second set of scenarios in FDI for vehicle systems.

the correct subspace, we can asymptotically differentiate the faultless version of the system from its fault version, while also solving the synthesis problem for the faultless version, assuming $\mathcal{G}$ was connected enough. We then showed that this asymptotic differentiation allows a solution of the network detection and isolation problems for graphs $\mathcal{G}$ which are connected enough. We also discussed an adversarial problem in which an attacker tries to sabotage the underlying network. In order to developed said algorithms, we assumed the existence of "convergence assertion protocols", which are algorithms receiving a multi-agent network and a conjectured limit, and checking whether the multi-agent system converge to the conjectured limit using on-line measurements. We then studied two methods of constructing these convergence assertion protocols. Lastly, we demonstrated our protocols by case studies.

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

Chapter 8

Summary

In this chapter, we conclude the research about cooperative control and passivity presented in this thesis. We begin with a brief summary of the contributions, followed by suggestions for future research paths.

8.1 Conclusions

The research in this thesis addresses both theoretical and practical problems in the fields of multi-agent systems and cooperative control, both being fundamental tools in understanding natural distributed systems, as well as in designing novel engineered systems. Several examples of these application domains were presented throughout the thesis, including vehicle and traffic models, neural networks, and oscillators. The main tools driving both the theoretical advancements and the application domains were passivity theory and network optimization, due to the connection between network optimization and multi-agent systems, creating a dictionary between the fields.

The first part of the thesis extended the network optimization framework of [23] to a wider range of systems. In Chapter 2, we used the notion of cyclically monotone relations to extend the network optimization framework to MIMO systems, using the work of Rockafellar [119]. This was done by defining the appropriate system-theoretic property, namely maximal equilibrium-independent cyclically monotone passive systems, reformulating the steady-state equation for the diffusively-coupled system as an optimization problem, and using subdifferential calculus to prove duality between the two network optimization problems. We have also shown that two general classes of nonlinear systems are maximal equilibrium-independent cyclically-monotone passive. The following Chapter 3 focused instead on passive-short systems. We showed that the network optimization framework cannot hold for these systems unless augmented appropriately, and classified three possible reasons for the failure. We showed that for equilibrium-independent output passive-short systems, the network optimization problems can still be defined, although they are no longer convex. Con-

© Technion - Israel Institute of Technology, Elyachar Central Library

vexifying the network optimization problems led to a natural system-theoretic transformation, namely output feedback, which passivized the system and rendered the network optimization framework valid for this class of systems. In other words, in the context of the dictionary between network optimization and cooperative control, convexification corresponds to passivation. We then focused on general passive-short systems, for which the network optimization framework might not even be defined. In that case, the convexification problem is converted with a monotonization problem for the steady-state input-output relation. We showed that the steady-state relation of an equilibrium-independent general passive-short can always be monotonized using a linear transformation, and used the notion of elementary matrices to understand the induced augmentation on the plant, which happens to passivize it. In this more general context, monotonization of the steady-state input-output relation corresponds to passivizing the dynamical system.

The second part of the thesis applies the network optimization framework to solve classical problems in control for a wide range of (nonlinear) multi-agent systems. In Chapter 4, we used the network optimization framework to present a very general solution to the final-value synthesis problem, relying only on the passivity of the agents. We then studied the effects of network symmetries on the clustering of the steady-state, and showed that clusters can be understood in terms of the induced group action on the graph. Focusing on statically homogeneous networks, We presented a graph synthesis process forcing the steady-state to cluster in prescribed sizes, and demonstrated how to solve the cluster placement problem. In the following Chapter 5, we reconsidered the final-value synthesis problem, but sought for a data-driven solution scheme this time. The main idea used in the solution is considering a cascade of nominal controllers with adjustable positive gains. In that case, data can be used to reformulate the network optimization problems as robust optimization problems, which are easier to analyze. We showed that as the value of the gains increases to ∞, the closed-loop steady-state approaches the desired steady-state. We then showed two data-driven techniques to compute gains for which the closed-loop steady-state is ϵ-close to the desired steady-state, the first revolved around conducting experiments before connecting the agents to the network, and the second relied on an iterative adjustment scheme that was shown to converge. Both techniques have convergence and stability guarantees.

The remaining chapters took advantage of the reliance of the network optimization problems on the underlying graph. Chapter 6 studied the problem of network detection and the sub-problem of network differentiation. We first presented the notion of indication vectors, which are constant exogenous inputs forcing any two networks with identical agents and controllers but different underlying graphs to converge to different steady-state outputs, giving a solution to the network differentiation problem. We presented two approaches for constructing such indication vectors, the first using randomization, and the second using algebraic methods. We then moved on to network detection. We first presented a sub-cubic-time algorithm for identifying the network of a multi-agent system for networks with LTI agents and controllers. The algorithm relied on

© Technion - Israel Institute of Technology, Elyachar Central Library

linearity of the relation between constant exogenous inputs and steady-state outputs, for which the off-diagonal entries indicate the edges in the graph. The algorithm was then adapted to general non-linear MEIP networks using linearization, and an error estimate was given. We then showed that the algorithm is optimal in terms of time complexity, meaning any algorithm solving the network detection problem with positive probability and finite error cannot run asymptotically slower than the presented algorithm. The main leverage point in all proofs was the connection between constant exogenous inputs and steady-state outputs, which can be understood (and shown to behave nicely) using the network optimization framework.

Lastly, Chapter 7 dealt with the problem of network fault detection and isolation. The main tool was the notion of edge-indication vectors, which are a variant of indication vectors. The network optimization framework was coupled with manifold theory and graph theory to construct these edge-indication vectors in a manner which still allows to solve the synthesis problem, meaning that we can "asymptotically" identify the existence and type of network faults. We assumed the existence of "convergence assertion protocols", which are algorithms asserting that a diffusively-coupled system converges to a conjectured limit, and used them to give graph-theoretic solutions to the problems of fault detection and fault isolation, as well as an adversarial game. Lastly, we presented two model-based data-driven convergence assertion algorithms.

8.2 Outlook

The network optimization framework was significantly extended in Chapters 2 and 3 to include diffusively-coupled networks of *square* MIMO systems, output- and input-passive short systems, and *SISO* general passive-short systems. One major avenue for future research includes extending the range of the network optimization framework to a bigger range of systems. One possible extension is to non-square MIMO systems, which might be connected to restrictions of convex functions on subspaces through the subdifferential. Another interesting extension deals with MIMO general passive-short systems, which require a more delicate analogue of cursive relations. Another option is discrete-time systems. Other extensions to the theory include a better understanding of MEIP and MEICMP, which can be achieved by proving certain closure properties, as feedback- and parallel connection, or a variant of the Hill-Moylan lemma, as done in [144] for EIP.

Another theoretical aspect that follows from the research in this thesis is the study of the steady-state input-output relation of more general systems. Section 3.4 deals with monotonizing transformations, which are guaranteed to exist for any equilibrium-independent general passive-short system, or any equilibrium-independent finite $\mathcal{L}_2$-gain system. Monotone relations are geometrically simple, e.g. they can always be embedded inside a maximal monotone relation, which is, topologically, a simple line. These topological properties are preserved under linear maps (or homeomorphisms in general). This simple observation can

© Technion - Israel Institute of Technology, Elyachar Central Library

be used to prove that SISO equilibrium-independent general passive-short systems cannot have pitchfork bifurcations, or other extremely non-linear phenomena, meaning that very simple topological arguments can give strong system-theoretic results.

The applicability of the network optimization framework is also a possible avenue for research. For data-driven methods, one could consider harder problems (e.g. final-value synthesis for the output vector, or attack/fault detection). Another approach is to try and extend the existing work to include MIMO systems, as one of the approaches relies heavily on the agents being SISO. For clustering, one can research the cluster synthesis problem, giving an efficient synthesis procedure also for the controller. However, perhaps the most important obstacle to overcome heading toward application is the existence of delays in the loop. It can be shown that, apart from very pathological cases, systems with delay cannot be passive. These delays obviously do not change the constant steady-states of the closed-loop system, but can hinder convergence of the networked system, invalidating the network optimization framework. Some methods of dealing with delays in the repetitive control, preview control and loop shifting [37, 53, 58]. One recent method of dealing with delays in a networked setting is the wave variable transformation [4]. Extending the network optimization framework to include delays will probably exploit the wave variable transformation, but lies outside the scope of this thesis.

Chapter 7 deals with network fault detection and isolation using passivity. Natural continuations to this problem are other questions in secure systems. Examples include resilience to "spoofing", resilience to eavesdropping, and avoiding detection. Spoofing might be treated using similar ideas to the ones presented in Chapter 7, namely convergence assertion methods and using connectivity to detect the rebellious agent. Other approaches might include the use of *monotone control systems* [3] to try and outright ignore inputs from other agents which are illogical. As for eavesdropping, the network optimization framework can be used together with graph symmetries to show indistinguishability between some states of the network due to symmetry.

© Technion - Israel Institute of Technology, Elyachar Central Library

Appendix A

Convex Analysis and Optimization Theory

The concept of convexity is a cornerstone of modern optimization theory. As stated by R.T. Rockafellar, "...in fact, the great watershed in optimization isn't between linearity and nonlinearity, but convexity and nonconvexity" [120]. This appendix is dedicated to reviewing some basic notions in convex analysis, optimization theory, and related topics. See also [121] and [17] for more on convex optimization.

A.1 Convex Sets and Convex Functions

A set $C \subseteq \mathbb{R}^n$ is called *convex* if the line segment between any two points in C is also contained in C. In another words, if $x, y \in C$ and $t \in (0,1)$ then $tx + (1-t)y \in C$. Important examples of convex sets include the empty set, $\mathbb{R}^n$, a ball $\{x \in \mathbb{R}^n : \ ||x - x_0|| \leq r\}$ with respect to any norm, and half-spaces $\{x \in \mathbb{R}^n : a^\top x - b \leq 0\}$ for some $a \in \mathbb{R}^n, b \in \mathbb{R}$. A function $f : \mathbb{R}^n \to \mathbb{R}$ is called *convex* if the epi-graph $\{(x,h) \in \mathbb{R}^{n+1} : \ f(x) \leq h\}$ is a convex set. Equivalently, the following inequality holds for all $x, y \in \mathbb{R}^n$ and $t \in [0,1]$,

$$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y).$$

If the inequality is strict whenever $x \neq y$ and $t \neq 0, 1$, we say that f is *strictly convex*. Important examples of convex functions include affine functions $f(x) = a^\top x + b$, any norm $f(x) = ||x||$, quadratics $f(x) = x^\top Ax$ for $A \geq 0$, and indicator functions $I_C(x)$ whenever C is a convex set.

The notion of convexity is important in optimization theory, as it usually allows to solve optimization problems relatively easily using gradient descent or other, modern, optimization techniques, e.g. alternating direction method of multipliers (ADMM) [55]. Other techniques can also be used to solve most interesting convex optimization problems, e.g. cone programming and interior-point

© Technion - Israel Institute of Technology, Elyachar Central Library

methods. Most modern techniques for solving optimization problems with non-convex elements rely on some convexifying tool, allowing to restate the problem in a convex form using additional variables. To understand why convexity is an important factor in optimization, we consider the problem of finding the minimum of a function $f : \mathbb{R}^n \to \mathbb{R}$, which we assume to be twice differentiable. In order to show that $x_0 \in \mathbb{R}^n$ is an optimal solution of the problem, we need to show that for any $x \in \mathbb{R}^n$, $f(x) \geq f(x_0)$ holds. Obviously, this can be very tricky to check even if we have a perfect description of f, and impossible if we only partial information about it. One possible approach is to look for critical points, in which $\nabla f = \mathbf{0}$. If the gradient of f can be described simply enough, one can solve the equation and find all extremum points, checking them one-by-one to see which one is the global minimizer. The main problem with this approach is its computational complexity. Even if solving $\nabla f = \mathbf{0}$ can be done effortlessly, there still might be a huge, possibly even infinite number of critical points. For example, the function $f : \mathbb{R} \to \mathbb{R}$ defined by $f(x) = (\sin(x)+x)^2$ has a unique global minimum at $x_0 = 0$, but an infinite number of critical points given by the solutions of $\cos(x) = -1$. Thus there is no "local" method of even asserting that a given point $x_0 \in \mathbb{R}^n$ is the global minimum. However, if we assume that f is convex, the problem is much easier.

Theorem A.1. *Suppose that f is a convex function. If x_0 is a local minimizer of f, then it is also a global minimizer.*

Proof. By assumption, there exists some $\varepsilon > 0$ such that if $||x - x_0|| < \varepsilon$ then $f(x) \geq f(x_0)$. Let $y \in \mathbb{R}^n$ be any point, and consider $x = ty + (1-t)x_0$. By continuity, if $t > 0$ is small enough then $||x - x_0|| < \varepsilon$, meaning that $f(x) \geq f(x_0)$. Thus, by convexity,

$$f(x_0) \leq f(x) = f(ty + (1-t)x_0) \leq tf(y) + (1-t)f(x_0).$$

Shifting $(1-t)f(x_0)$ to the left-hand side and dividing by t proves that $f(y) \geq f(x_0)$. Because y was chosen arbitrarily, we conclude that x_0 is a global minimum. □

Thus, if f is twice differentiable, it's enough to check that $\nabla f(x_0) = \mathbf{0}$ and that the Hessian $Hf(x_0)$ is positive-semi definite to prove that x_0 is a global minimizer. However, it is known that for twice-differentiable convex functions, the Hessian is always positive semi-definite, so any critical point is a global minimum. Recalling that gradient descent always converges to a local minimum, we conclude that gradient descent always finds the global minimum when applied to a differentiable convex function. Before moving on to non-differentiable convex functions, we state and prove three lemmas that will be needed in the main text.

Lemma A.1. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function, and let $S : \mathbb{R}^n \to \mathbb{R}^d$ be a linear operator. We define the function $h : \mathrm{Im}(S) \to \mathbb{R}$ by*

$$h(x) = \min_{r:\ Sr=x} f(r),$$

© Technion - Israel Institute of Technology, Elyachar Central Library

assuming the minimum is always attained. Then:

i) h is a convex function.

ii) If f is strictly convex, then so it h.

Proof. Let $V = \mathrm{Im}(S)$. Take some $x, y \in V$ and $t \in [0,1]$. We want to show that $h(tx + (1-t)y) \leq th(x) + (1-t)h(y)$. We pick some $r_x, r_y \in \mathbb{R}^n$ such that $h(x) = f(r_x)$ and $h(y) = f(r_y)$, which exist by assumption. Then $S(tr_x + (1-t)r_y) = tr_x + (1-t)r_y$, meaning that $h(tx + (1-t)y) \leq f(tr_x + (1-t)r_y)$. On the other hand, by convexity,

$$f(tr_x + (1-t)r_y) \leq tf(r_x) + (1-t)f(r_y) = th(x) + (1-t)h(y).$$

Moreover, if f is strictly convex, $x \neq y$ and $t \neq 0, 1$ then we get a strict inequality, as $r_x \neq r_y$. Combining the two inequalities completes the proof. □

Lemma A.2. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function. Fix $x_0, y \in \mathbb{R}^n$ and define a function $g : (0, \infty) \to \mathbb{R}$ by $g(\lambda) = \frac{f(x_0+\lambda y)-f(x_0)}{\lambda}$. Then g is non-decreasing. Moreover, if f is strictly convex, then g is increasing.*

Proof. Take any two numbers $0 < \mu < \lambda$. We want to show that $g(\mu) \leq g(\lambda)$. The point $x_0 + \mu y$ can be written as a convex combination of x_0 and $x_0 + \lambda y$. Indeed,

$$x_0 + \mu y = \left(1 - \frac{\mu}{\lambda}\right) x_0 + \frac{\mu}{\lambda}(x_0 + \lambda y)$$

where $0 < \frac{\mu}{\lambda} < 1$ by choice of μ and λ. Thus, by convexity, □

$$f(x_0 + \mu y) \leq \left(1 - \frac{\mu}{\lambda}\right) f(x_0) + \frac{\mu}{\lambda} f(x_0 + \lambda y)$$

Dividing the equation by μ gives:

$$\frac{1}{\mu} f(x_0 + \mu y) - \frac{1}{\mu} f(x_0) \leq \frac{1}{\lambda} f(x_0 + \lambda y) - \frac{1}{\lambda}) f(x_0)$$

which reads $g(\mu) \leq g(\lambda)$. Moreover, if f is strictly convex, the first inequality is strict, implying that $g(\mu) < g(\lambda)$.

Lemma A.3. *Let $C \subseteq \mathbb{R}^n$ be a convex set and let $f : C \to \mathbb{R}$ be convex. Suppose that f achieves a minimum m in C, and let $M = \{x : \ f(x) = m\}$ be the set of f's minima. Then M is convex*

Proof. Let $x, y \in M$ and $t \in [0,1]$. We want to show that $z = tx + (1-t)y \in M$. Because $z \in C$, we have that $f(z) \geq m$. However, by convexity:

$$f(z) \leq tf(x) + (1-t)f(y) = tm + (1-t)m = m.$$

Thus $f(z) = m$ and $z \in M$. As x, y and t were chosen arbitrarily, we conclude that M is convex. □

© Technion - Israel Institute of Technology, Elyachar Central Library

A.2 Subdifferentials

In the previous section, we saw that gradient descent can be used to solve convex optimization problems with differentiable cost functions and constraints. However, some convex functions are not differentiable, e.g. the absolute value function $|x|$ or the Euclidean norm $||x||$. Actually, many problems in optimization theory model their cost functions and constraints as piece-wise linear functions, which are not everywhere differentiable. Moreover, their minimum is always at a point in which they are not differentiable. Thus, a treatment for non-differentiable convex functions is needed.

As a motivation, consider a convex function f which is twice differentiable. Because the Hessian $Hf(x)$ is positive semi-definite at any point, the first-order approximation $f(x_0)+\nabla f(x_0)(x-x_0)$ underestimates $f(x)$. Motivated by this, we make the following definition:

Definition A.1. *Let f be a convex function, and let $x_0 \in \mathbb{R}^n$. We say that $v \in \mathbb{R}^n$ is a* subdifferential *of f at x_0 if the inequality $f(x) \geq f(x_0)+v^\top(x-x_0)$ holds for any $x \in \mathbb{R}^n$. The* subgradient *or the* subdifferential set *of f at x_0, denoted $\partial f(x_0)$ is defined as the set of all subdifferentials of f at x_0.*

One can show some basic facts about subdifferentials. First, one can prove that $\partial f(x)$ is not the empty set, by using the separating hyperplane theorem [34]. Moreover, if f is differentiable at x then $\partial f(x) = \{\nabla f(x)\}$, and more precisely, f is differentiable at x if and only if $\partial f(x)$ contains only one point. Furthermore, the set $\partial f(x)$ is always convex and closed. Lastly, by definition, x_0 is a global minimum of f if and only if $\mathbf{0} \in \partial f(x_0)$. One can run the standard gradient descent algorithm, replacing the gradient with any element in the subgradient set, and get a simple algorithm for computing the minimizer of the convex function f. Note that, by definition, f is strictly convex if and only if ∂f contains no horizontal lines.

The subdifferential is related to the notion of duality for convex functions. The *Legendre transform*, or the *dual function*, of a convex function $f : \mathbb{R}^n \to \mathbb{R}$ is defined via $f^\star(y) = \sup_{y\in\mathbb{R}^n}\{y^\top x - f(x)\}$ [121]. By solving the supremum, it's possible to show that $\partial f^\star$ is the inverse relation of ∂f, i.e. $y \in \partial f(x)$ if and only if $x \in \partial f^\star(y)$. In particular, using the properties of the subgradient described above, duality maps differentiable convex functions to strictly convex functions, and vice versa.

Calculating the subgradient set can be cumbersome on some occasions. One method is to use the known subdifferential for one function to compute the subdifferential of another function. For example, if $\alpha > 0$ then $\partial(\alpha f)(x) = \alpha\partial f(x)$ follows immediately from the definition. It's also possible to show that if $f = f_1+\cdots+f_n$ then $\partial f(x) = \partial f(x_1)+\cdots+\partial f(x_n)$, and that if $h(x) = f(Ax+b)$, then $\partial h(x) = A^\top \partial f(Ax+b)$. We state and prove one specific result needed in the main text:

Proposition A.1. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function, and let $S : \mathbb{R}^n \to \mathbb{R}^d$ some linear transformation. Fix some $\zeta \in \mathrm{Im}(S)$, and define a map g :*

© Technion - Israel Institute of Technology, Elyachar Central Library

$\{y : Sy = \zeta\} \to \mathbb{R}$ by $g(x) = f(x)$. Then g is a convex function, and its subdifferential is given by $\partial g(x) = \mathrm{Proj}_{\mathrm{Im}(S)}(\partial f(x))$

Proof. Let $X = \{y : Sy = \zeta\}$, $V = \ker S$ and $U = \mathrm{Im}(S^\top)$. We know that $\mathrm{Im}(S^\top) = \ker(S)^\perp$, so we can identify $\mathbb{R}^n$ as a direct sum of U and V, giving a function $F : U \times V \to \mathbb{R}$ by $F(u, v) = f(u + v)$. In [121], it is proved that if $\chi : \mathbb{R} \to \mathbb{R}$ s convex, then $\partial\chi(t) = [\chi'_-(t), \chi'_+(t)]$, where $\chi'_\pm$ are the one-sided derivatives of χ. It's also proved in [121] that if $\rho : \mathbb{R}^n \to \mathbb{R}$ is convex, and we fix $w_0, w_1 \in \mathbb{R}^n$ and define $\chi(t) = \rho(w_0 + tw_1)$, then χ's one-sided derivatives are given by:

$$\chi'_+(t) = \max_{\alpha \in \partial\rho(w_0+tw_1)} \alpha^\top w_1, \quad \chi'_-(t) = \min_{\alpha \in \partial\rho(w_0+tw_1)} \alpha^\top w_1.$$

These facts, together with convexity of the subdifferential set, imply that $\partial\chi(t) = w_1^\top \partial\rho(w_0 + tw_1 s)$.

Now, by assumption, there's some y_0 such that $Sy_0 = \zeta$. Decompose y as $u_0 + v_0$ for some $u_0 \in \mathrm{Im}(S^\top)$ and $v_0 \in \ker S$. The set X is equal to $u_0 + V$. Thus the map g can be described as $g(v) = F(u_0, v)$. Take some $v_0, v_1 \in V$. Restricting g to the line $\{v_0 + tv_1 : t \in \mathbb{R}\}$ is identical to restricting F to the line $\{(u_0, v_0 + tv_1) : t \in \mathbb{R}\}$. Thus they yield the same differential sets at $t = 0$. By above, we get that: $v_1^\top \partial g(v_0) = v_1^\top \partial F(u_0, v_0) = v_1^\top \mathrm{Proj}_V(\partial F(u_0, v_0))$, implying that the sets $\partial g(v_0)$ and $\mathrm{Proj}_V(\partial F(u_0, v_0))$ look the same when hit by a linear functional on V. However, both sets are convex and closed, thus the separating hyperplane theorem [34] implies that they are equal. Recalling the definitions of V and F, we get that for any $x \in \mathrm{Im}(S)$, $\partial g(x) = \mathrm{Proj}_{\mathrm{Im}(S)}(\partial f(x))$. □

A.3 Rockafellar's Theorem and Cyclically Monotone Relations

In the previous section, we defined the notion of subdifferentials. In basic calculus, one can ask which functions are derivatives of some other function, which are exactly integrable functions. We ask an analogous basic question, namely which set-valued maps, or relations, are subdifferentials of convex functions.

It is known that in $\mathbb{R}$, the derivative of a differentiable convex function is monotone. Moreover, if $\phi : \mathbb{R} \to \mathbb{R}$ is a monotone function, then it is integrable, and one can easily show that $f(t) = \int_0^t \phi(s)ds$ is a convex integral function of ϕ, which answers the question for derivatives of differentiable convex functions. One can extend this idea to non-differentiable functions by considering the notion of *monotone relations*, which are set-valued maps k defined on $\mathbb{R}$, such that if $u_0, u_1 \in \mathbb{R}$ and $y_0 \in k(u_0)$, $y_1 \in k(u_1)$, then $(u_1 - u_0)(y_1 - y_0) \geq 0$. A maximal monotone relation is a monotone relation which is not contained in a larger monotone relation. One can easily show that the subgradient of a convex function $\mathbb{R} \to \mathbb{R}$ is maximally monotone, and that if ϕ is a maximally monotone relation (or set-valued function), then $f(t) = \int_0^t \max_{u \in k(s)}\{u\}ds$ is a convex

© Technion - Israel Institute of Technology, Elyachar Central Library

function whose subdifferential is equal to ϕ [121]. Thus, in one dimension, the subdifferentials of convex functions are exactly maximally monotone relations.

When considering the same problem in multiple dimensions, the solution becomes trickier. This is because the notion of monotonicity can be defined in multiple different ways in more than one dimension, and it's unclear which works best. In [119], Rockafellar introduced the notion of *cyclically monotone relations*:

Definition A.2 ([119])**.** *Let $d \geq 1$ be an integer, and consider a subset R of $\mathbb{R}^d \times \mathbb{R}^d$. We say that R is a* cyclically-monotonic *(CM) relation if for any $m \geq 1$ and any pairs $\{(u_j, y_j)\}_{j=1}^m \subseteq R$ of d-vectors, the following inequality holds:*

$$\sum_{i=1}^{m} y_i^\top (u_i - u_{i-1}) \geq 0, \tag{A.1}$$

where we use the convention that $u_0 = u_m$. We say that this relation is strictly cyclically-monotone *(SCM) if the inequality* (A.1) *is sharp whenever at least two u_i-s are distinct. We term the relation as maximal CM (or maximal SCM) if it is not strictly contained in a larger CM (SCM) relation.*

Remark A.1. *As was shown in the main text, cyclically monotone relations are always monotone relations, no matter the dimension, but the converse is not true. Indeed, one can show that the relation $\{(x, Jx) : \ x \in \mathbb{R}^2\}$, where $J = \left[\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right]$, is monotone but not cyclically monotone.*

Remark A.2. *The name "cyclic monotonicity" is derived from the cyclic shift operator $\sigma_m : \mathbb{R}^m \to \mathbb{R}^m$ defined by $\sigma_m([x_1, x_2, \cdots, x_m]^\top) = \sigma_m([x_2, \cdots, x_m, x_1]^\top)$. Indeed, if one defines $\mathbf{u} = [u_1^\top, \cdots, u_m^\top]^\top$ and $\mathbf{y} = [y_1^\top, \cdots, y_m^\top]^\top$, the inequality* (A.1) *can be written as $\mathbf{y}^\top((\mathrm{Id}_m - \sigma_m) \otimes \mathrm{Id}_d)\mathbf{u} \geq 0$ for all positive integers m. As we noted before, the case $m = 2$ coincides with the definition of monotonicity [121].*

The connection of cyclically monotone relations to convex functions is due to Rockafellar:

Theorem A.2 ([119])**.** *A relation $R \subseteq \mathbb{R}^N \times \mathbb{R}^N$ is CM if and only if it is contained in the subgradient of a convex function $\mathbb{R}^N \to \mathbb{R}^N$. Moreover, R is SCM if and only if it is contained in the subgradient of a strictly convex function. A relation R is maximally CM (or SCM) if and only if the inclusion above is equality, and in that case the convex function is unique up to an additive constant.*

Rockafellar's theorem allows to immediately show that the affine relation $k(u) = Su + v$ is CM if and only if S is positive semi-definite. It's possible to also show this fact without Rockafellar's theorem, using the Cholesky decomposition. We shall not go into details in this appendix.

© Technion - Israel Institute of Technology, Elyachar Central Library

Appendix B

Graph Theory and Algebraic Graph Theory

This appendix is devoted to presenting some basic concepts in graph theory [15] and algebraic graph theory [57].

As defined in the introduction, a graph $\mathcal{G}$ is a pair $\mathcal{G} = (\mathbb{V}, \mathbb{E})$ where $\mathbb{V}$ is the set of vertices and $\mathbb{E}$ is the set of edges. Each edge consists of two nodes $i \neq j$ from $\mathbb{V}$, called the ends of the edge, and we write $e = \{i, j\}$. We will orient the edge arbitrarily, say from i to j, and write $e = (i, j)$ or $e = i \to j$ in that case. The *out-degree* $d_{out}(i)$ of a node i is the number of edges leaving i, i.e. edges of the form $e = i \to j$ for some $j \in \mathbb{V}$. Similarly, the *in-degree* $d_{in}(i)$ of a node i is the number of edges entering i, and the *degree* $d(i)$ of the node is the sum of the out-degree and the in-degree, i.e. the number of edges touching the node. The number of edges in the graph $\mathcal{G}$ can be counted using the degrees of its nodes, as $|\mathbb{E}| = \sum_{i\in\mathbb{V}} d_{out}(i) = \sum_{i\in\mathbb{V}} d_{in}(i) = \frac{1}{2}\sum_{i\in\mathbb{V}} d(i)$.

One can consider the notion of inclusion between graphs. A graph $\mathcal{H} = (\mathbb{V}_\mathcal{H}, \mathbb{E}_\mathcal{H})$ is called a *subgraph* of $\mathcal{H} = (\mathbb{V}_\mathcal{G}, \mathbb{E}_\mathcal{G})$ if $\mathbb{V}_\mathcal{H} \subseteq \mathbb{V}_\mathcal{G}$ and $\mathbb{E}_\mathcal{H} \subseteq \mathbb{V}_\mathcal{H}$. The *induced subgraph* of $\mathcal{G}$ on a set of nodes $S \subseteq \mathbb{V}_\mathcal{G}$ is $\mathcal{H} = (S, \mathbb{E}_\mathcal{H})$, where the set $\mathbb{E}_\mathcal{H}$ consists of all edges of $\mathcal{G}$ with both ends are in S.

We can also discuss the connectivity of a graph. A *path* is a sequence $i_1, i_2, \cdots, i_k$ of nodes, such that any two consecutive nodes are connected by an edge. The length of a path is the number of edges it traverses. A path is said to be *simple* if no nodes appears more than once in it, except possibly for the first and last vertex. A *cycle* is a path which starts and ends at the same node. An unoriented graph is said to be *connected* if there exists a path between any two nodes. An oriented graph is said to be *weakly connected* if its unoriented version is connected, and said to be *strongly connected* if there exists a path between any two nodes that obeys the orientation. A *tree* is a connected graph which contains no cycles. A graph $\mathcal{G}$ on n nodes is a tree if and only if it is connected and it has exactly $n-1$ edges. Each connected graph has a *spanning tree*, which is a tree subgraph having the same set of nodes as $\mathcal{G}$.

© Technion - Israel Institute of Technology, Elyachar Central Library

If the unoriented graph $\mathcal{G}$ is not connected, we can define a relation on the nodes - i is connected to j if there exists a path between i and j. This is an equivalence relation, meaning that the set of nodes can be written as a disjoint union of sets, in which any two nodes are connected to one another. These sets are known as the *connected components* of the graph $\mathcal{G}$. The *distance* between two nodes i and j is defined to be the length of the shortest path between i and j, or ∞ if there is not path between them. The *diameter* $\mathrm{diam}(\mathcal{G})$ is defined as the maximal distance between two nodes in the graph $\mathcal{G}$.

Graph theory has benefited greatly throughout the years from linear algebra, as graphs can be neatly described using matrices in various ways. For example, the relation between nodes and edges in the graph $\mathcal{G} = (\mathbb{V}, \mathbb{E})$ can be read from the incidence matrix $\mathcal{E}_\mathcal{G}$, defined as a matrix of size $|\mathbb{V}| \times |\mathbb{E}|$ with elements $(\mathcal{E}_\mathcal{G})_{ie} = \begin{cases} -1 & \exists j, e = (i,j) \\ 1 & \exists j, e = (j,i) \\ 0 & \text{else} \end{cases}$. One can also consider the degree matrix $D = D_\mathcal{G}$, which is a diagonal matrix with entries equal to the degrees of the different nodes in $\mathcal{G}$, and the adjacency matrix $A_\mathcal{G}$, which is a $|\mathbb{V}| \times |\mathbb{V}|$ matrix defined as $(A_\mathcal{G})_{ij} = \begin{cases} 1 & \{i,j\} \in \mathbb{E} \\ 0 & \text{else} \end{cases}$. The Laplacian matrix $L_\mathcal{G}$ is defined as $D_\mathcal{G} - A_\mathcal{G} = \mathcal{E}_\mathcal{G}\mathcal{E}_\mathcal{G}^\top$. The Laplacian matrix can be shown to be positive semi-definite. If $\mathcal{G}$ is connected, then the eigenvalue 0 is simple, and has $\mathbb{1}$ as a corresponding eigenvector. Thus all other eigenvalues $\lambda_2 \leq \lambda_3 \leq \cdots \leq \lambda_n$ are strictly positive.

The Laplacian matrix $L_\mathcal{G}$ appears often in multi-agent systems and cooperative control due to the consensus protocol, defined as $\dot{x}_i = \sum_{j \sim i}(x_j - x_i)$, which can be concisely written as $\dot{x} = -L_\mathcal{G} x$ [107]. The consensus protocol forces the agents of a multi-agent system toward agreement, and the rate of convergence to agreement is dictated by the the second smallest eigenvalue λ_2 of the Laplacian $L_\mathcal{G}$. In discrete-time processes, like discrete-time agents or random walks on graphs, the Laplacian is usually replaced with the incidence matrix $A_\mathcal{G}$ or with its scaled version $D_\mathcal{G}^{-1} A_\mathcal{G}$, but we will not focus on this case in the thesis. The reader is referred to [107] or [84] and references therein for more on the subject. We conclude this appendix by noting that the second smallest eigenvalue of the graph Laplacian cannot be arbitrarily close to zero, as the following theorem suggests:

Theorem B.1 ([116])**.** *Let $\mathcal{G}$ be a connected graph on n nodes, and let λ_2 be the second smallest eigenvalue of the Laplacian. Then $\lambda_2 \geq \frac{1}{\binom{n}{2}}$.*

The reader is referred to [116] for a proof of the theorem.

© Technion - Israel Institute of Technology, Elyachar Central Library

Appendix C

Dynamical Systems, Passivity and Stability

This appendix is devoted to presenting some basic concepts in systems theory, namely stability, passivity, and related technical tools such as Lyapunov functions and Barbalat's lemma. It is based on [75], in which all omitted proofs are available. We focus on the case of continuous-time systems on $\mathbb{R}^n$, but most definitions and theorems can be adapted to discrete-time systems and more general (continuous or discrete) spaces.

C.1 Stability and Lyapunov Theory

A continuous-time control system is defined using two continuously differentiable functions $f : \mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}, h : \mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_y}$, where n_x, n_y, n_u are three positive integers. The threesome $(u(t), x(t), y(t))$ is called a trajectory of the control system, where $x : [0, \infty) \to \mathbb{R}^{n_x}$,$u : [0, \infty) \to \mathbb{R}^{n_u}$, and $y : [0, \infty) \to \mathbb{R}^{n_y}$ are functions, if the ODE $\dot{x}(t) = f(t, x(t), u(t))$ holds, and the relation $y(t) = h(t, x(t), u(t))$ also holds. Similarly, we can consider a dynamical system, parameterized by a function $f : \mathbb{R} \times \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$, for which a trajectory consists of a single function $x : [0, \infty) \to \mathbb{R}^{n_x}$ satisfying the ODE $\dot{x}(t) = f(t, x)$. In this thesis, we only consider time-invariant systems, in which the functions f and h do not explicitly depend on the time variable t.

One important notion used throughout the thesis is equilibria. A point x_0 is called an equilibrium of a dynamical system if the constant signal $x(t) \equiv \mathrm{x}_0$ is a trajectory of the dynamical system, or equivalently $f(\mathrm{x}_0) = 0$. For control systems, an equilibrium is a threesome $(\mathrm{u}_0, \mathrm{x}_0, \mathrm{y}_0)$ such that the corresponding constant signals form a trajectory of the control system. Again, this can be reformulated as $0 = f(\mathrm{x}_0, \mathrm{u}_0), \mathrm{y}_0 = h(\mathrm{x}_0, \mathrm{u}_0)$. An input-output steady-state pair is a pair $(\mathrm{u}_0, \mathrm{y}_0)$ such that there exists some signal $x(t)$ such that $(\mathrm{u}_0, x(t), \mathrm{y}_0)$ form a trajectory of the system. Steady-state input-output pairs are essentially a model-free version of equilibria, as they remove the dependency on the state

© Technion - Israel Institute of Technology, Elyachar Central Library

x of the system. Moreover, it allows to avoid a discussion about controllability and observability for nonlinear systems, as it does not require the state $x(t)$ to be constant as well.

One important notion regarding dynamical systems is the notion of stability:

Definition C.1 (Stability)**.** *Let $\dot{x} = f(x)$ be a dynamical systems, and let S be a set in $\mathbb{R}^{n_x}$.*

i) The set S is called stable (for the dynamical system $\dot{x} = f(x)$) if for any open set $V \supseteq S$ there exists an open set $W \supseteq S$ such that any trajectory $x(t)$ with $x(0) \in V$ satisfies $x(t) \in W, \forall t \in [0, \infty)$. If S is not stable, we'll say it's unstable.

ii) S is called locally asymptotically attractive (for the dynamical system $\dot{x} = f(x)$) if there is a an open set $V \supseteq S$ such that for any trajectory of the dynamical system with initial condition $x(0) \in V$, the distance $d(S, x) = \inf_{s \in S} \|x - s\|$ between S and $x(t)$ converges to 0 as $t \to \infty$. We say that S is globally asymptotically stable if $W = \mathbb{R}^{n_x}$. If S is a single point $S = \{\mathrm{x}_0\}$, we will also say that the dynamical system locally (globally) asymptotically converges to x_0.

One of the most useful ways to prove convergence is using Lyapunov theory, namely, Lyapunov's second method for stability:

Theorem C.1 (Lyapunov's Theorem)**.** *Consider a time-invariant dynamical system $\dot{x} = f(x)$ with an equilibrium x_0. Suppose there exists a continuously differentiable function $V : \mathbb{R}^{n_x} \to \mathbb{R}$, such that $V(x) > 0$ for all $x \neq \mathrm{x}_0$ and $V(\mathrm{x}_0) = 0$. Suppose that for any trajectory $x(t)$ of the system, $\frac{d}{dt}V(x(t)) = \nabla V(x(t))\dot{x}(t) \leq 0$.*

i) If the function V is radially unbounded, i.e. $\lim_{||x|| \to \infty} V(x) = \infty$, then any bounded set is stable.

ii) If $\frac{d}{dt}V(x(t)) < 0$ for any trajectory for which $x(t) \neq \mathrm{x}_0$, then the system globally asymptotically converges to x_0.

Lyapunov theory can also give local asymptotical convergence if the domain of V is changed to some arbitrary open set containing x_0. Moreover, Lyapunov theory can also be used to prove that almost all trajectories converge asymptotically to x_0. For example, one can show that $\frac{d}{dt}V(x(t))$ holds for all trajectories except for the equilibrium x_0, and a few unstable equilibria. Thus, if the initial conditions $x(0)$ of the trajectory $x(t)$ is not one of these unstable equilibria, is must converge to x_0. An example of this approach can be seen in [176].

Lyapunov theory and the related LeSalle's invariance theorem give very strong methods for proving that a time-invariant dynamical system locally (globally) asymptotically converges to some set. However, in some occasions, as we'll see below, we are forced to consider time-dependent systems, or time-dependent inequalities on the derivative of the function V. The main tool for proving convergence of said systems is Barbalat's lemma:

© Technion - Israel Institute of Technology, Elyachar Central Library

Theorem C.2 (Barbalat's Lemma)**.** *Let $\phi : \mathbb{R} \to [0, \infty)$ be any uniformly continuous function. Suppose that $\lim_{t\to\infty} \int_0^t \phi(s)ds$ exists and is finite. Then $\phi(t) \to 0$.*

For linear time-invariant dynamical systems, it's easy to characterize stability:

Proposition C.1. *Let $A \in \mathbb{R}^{n\times n}$ be any matrix. The dynamical system $\dot{x} = Ax$ globally asymptotically converges to 0 if and only all of A's eigenvalues have negative real part.*

C.2 Passivity

We now turn our attention to control systems. We consider a *square* control system, in which the input dimension n_u and output dimension n_y are equal.

Definition C.2 (Passivity)**.** *Consider a control system $\dot{x} = f(x, u), y = h(x, u)$, and let (u, y) be any steady-state input-output pair. We say that:*

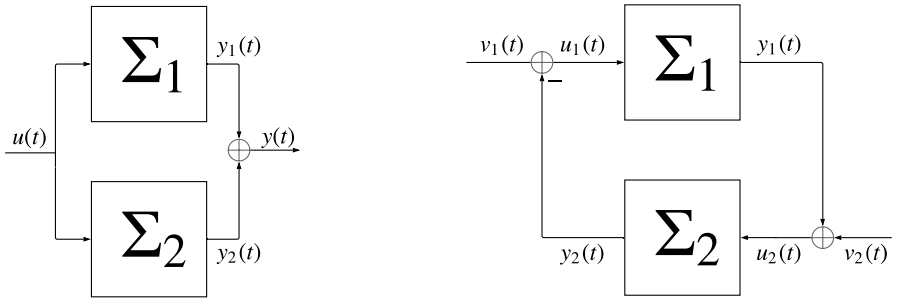
i) the system is passive with respect to (u, y) if there exists a positive semi-definite function continuously differentiable function $S(x)$ (called a storage function*) such that the inequality $\frac{d}{dt}S(x(t)) \leq (u(t) - \mathrm{u})^\top (y(t) - \mathrm{y})$ holds for any trajectory.*

ii) the system is output-strictly passive with respect to (u, y) if there exists a positive semi-definite function continuously differentiable function $S(x)$ and some $\rho > 0$ such that the inequality $\frac{d}{dt}S(x(t)) \leq (u(t) - \mathrm{u})^\top (y(t) - \mathrm{y}) - \rho(y(t) - \mathrm{y})^2$ holds for any trajectory.

iii) the system is input-strictly passive with respect to (u, y) if there exists a positive semi-definite function continuously differentiable function $S(x)$ and some $\nu > 0$ such that the inequality $\frac{d}{dt}S(x(t)) \leq (u(t) - \mathrm{u})^\top (y(t) - \mathrm{y}) - \nu(u(t) - \mathrm{u})^2$ holds for any trajectory.

The motivation for passivity theory stems from SISO electrical components. There, the input $u(t)$ is the voltage exerted on the system, and the output $y(t)$ is the current induced by the voltage. In this case, $u(t)y(t)$ is the power exerted on the electrical component. In that case, the storage function $S(x)$ can be thought of as a measure of the stored energy inside the component, and passivity then translates into the demand that the component does not generate energy. Strict passivity expands on that idea by prescribing rates at which the energy dissipates. This idea of passivity-based (i.e. energy-based) measures can be generalized to other nonlinear systems, and it is one of the cornerstones of nonlinear control. Another reason for its significance is that it implies stability:

Theorem C.3. *Let $\dot{x} = f(x, u), y = h(x, u)$ be a time-invariant control system, and assume f, h are locally Lipschitz. Let $(\mathrm{u}_0, \mathrm{x}_0, \mathrm{y}_0)$ be an equilibrium of the system.*



(a) Parallel interconnection.

(b) Feedback interconnection.

Figure C.1: Interconnections preserving passivity.

i) If the system is passive, then x_0 *is stable for the dynamics* $\dot{x} = f(x, \mathrm{u}_0)$.

ii) If the system is output-strictly passive, and the system $\dot{x} = f(x, \mathrm{u}_0)$ *is zero-state observable for* $(\mathrm{x}_0, \mathrm{y}_0)$[1]*, then the system* $\dot{x} = f(x, \mathrm{u}_0)$ *locally asymptotically converges to* x_0.

iii) If, along with the conditions of the previous case, the associated storage function is radially unbounded, then the system globally asymptotically converges to x_0.

One important property of stability is that, unlike stability, it is preserved under interconnection of systems:

Theorem C.4. *Let* $\Sigma_1 : u_1 \mapsto y_1$ *and* $\Sigma_2 : u_2 \mapsto y_2$ *be passive systems, where the dimension of* u_1, u_2, y_1, y_2 *is equal to* n.

i) The parallel interconnection, with input u *and output* y *defined by* $y = y_1 + y_2$*, where one takes* $u_1 = u_2 = u$*, as seen in Figure C.1(a), is passive.*

ii) The feedback interconnection, with input $v = [v_1, v_2]$ *and output* $y = [y_1, y_2]$*, where* $u_1 = v_1 - y_2$ *and* $u_2 = v_2 + y_1$*, as seen in Figure C.1(b), is passive.*

iii) Given a matrix $M \in \mathbb{R}^{n \times n}$*, the symmetric transformation of* Σ_1*,* $M^T \Sigma_1 M$*, is passive.*

Example C.1. *Consider the systems* $\dot{x} = A_1 x + u$ *and* $\dot{x} = A_2 x + u$ *for* $A_1 = \left[\begin{smallmatrix} -1 & 8 \\ 0 & -1 \end{smallmatrix}\right]$ *and* $A_2 = \left[\begin{smallmatrix} -1 & 0 \\ 8 & -1 \end{smallmatrix}\right]$*. Both systems are stable fixed input* $u = 0$*, as the eigenvalues of* A_1 *and* A_2 *are both equal to* -1*. However,* $A_1 + A_2 = \left[\begin{smallmatrix} -2 & 8 \\ 8 & -2 \end{smallmatrix}\right]$ *has eigenvalues* $\lambda_1 = -10$ *and* $\lambda_2 = 6 > 0$*, so the parallel interconnection,* $\dot{x} = (A_1 + A_2)x + u$*, is not stable with fixed input* $u = 0$.

[1] i.e., any trajectory $(\mathrm{u}_0, x(t), y(t))$ such that $y(t) = \mathrm{y}_0$ for all times must have $x(t) \equiv \mathrm{x}_0$.

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

Another important advantage that passivity has over stability is that it implies a connection between the size of the input and the size of the output through a *linear* bound:

Theorem C.5. *Let $\dot{x} = f(x,u), y = h(x,u)$ be a time-invariant control system, and assume f, h are locally Lipschitz. Suppose that $(0,0,0)$ is an equilibrium of the system. If the system is output-strictly passive with parameter $\rho > 0$, then it has finite $\mathcal{L}_2$ gain lower or equal than $\frac{1}{\rho}$.*[2]

If one focuses on linear systems, there are a few different (but related) approaches to characterize passivity. The first deals with the transfer function, and the second deals with the state-space representation. The first requires the notion of positive-real transfer functions.

Definition C.3. *Let $G(s)$ be an $n \times n$ proper transfer function matrix. Then $G(s)$ is called* positive real *if*

i) the poles of of $G(s)$ are in $\mathrm{Re}(s) \leq 0$.

ii) for all real ω such that $j\omega$ is not a pole of $G(s)$, $G(j\omega) + G(\jmath\omega)^\top$ is a positive semi-definite matrix.

iii) any pure imaginary pole $j\omega$ of $G(s)$ is a simple pole, and the residue matrix $\lim_{s \to j\omega}(s - j\omega)G(s)$ is positive semi-definite.

We say that $G(s)$ is strictly positive real if there's some $\varepsilon > 0$ such that $G(s-\varepsilon)$ is positive real.

Proposition C.2. *Consider a linear time-invariant system with transfer function $G(s)$. If $G(s)$ is positive-real, then the system is passive. Moreover, if $G(s)$ is strictly positive-real, then the system is strictly passive.*

Another, equivalent formulation of the property is given by the Kalman-Yakubovich-Popov lemma:

Theorem C.6 (Kalman-Yakubovich-Popov)**.** *Let $\dot{x} = Ax + Bu,\ y = Cx + Du$ be a minimal state-space realization of a linear time-invariant system Σ, and let $G(s)$ be the corresponding transfer function. Suppose there exists some $\varepsilon \geq 0$, and some matrices $P = P^\top > 0$, L and W such that the following conditions hold:*

$$\begin{aligned} PA + A^\top P &= -L^\top L - \varepsilon P \\ PB &= C^\top - L^\top W \\ W^\top W &= D + D^\top \end{aligned}$$

Then the transfer function $G(s)$ is positive real. If $\varepsilon > 0$, then the transfer function is strictly positive real.

[2] Recall that a system $\Sigma : u \mapsto y$ is said to have finite $\mathcal{L}_2$ gain if there exists some $\beta > 0$ such that $||y(t)|| \leq \beta ||u(t)||$ for all inputs $u(t)$ and outputs $y(t)$ with finite norm, where $||a(t)|| = \sqrt{\int_0^\infty |a(t)|^2 dt}$ indicates the power of the signal. The smallest parameter β is called the $\mathcal{L}_2$ gain of the system.

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

Appendix D

Complexity Theory for Matrix Multiplication and Inversion

Complexity theory is a field dealing with the amount of resources required to solve problems or to run certain algorithms [36]. It is one of the pinnacles of modern research in computer science and algorithmics, dealing with problems like $P = NP$, which asks whether its true that problems for which a solution can be efficiently verified, it can also be efficiently found. One important measure in complexity theory is time complexity, describing the running time of an algorithm. Usually, the time complexity of an algorithm is written using the big-O notation, using the size of the input as a parameter. The reason for this notation is the dependence of some basic operations on the hardware running the algorithm, e.g. multiplying two numbers is quicker on Intel Core i5 than on Pentium 4. The theory differentiates between *deterministic* algorithms and various kinds of *probabilistic* algorithms, e.g. algorithms which are correct with probability 1, algorithms which are correct with high probability (dependent on n), and algorithms which are correct only with probability $> 1/2$.

The time complexity of deterministic matrix multiplication is one of the most fundamental questions in complexity theory [36, 145]. The schoolbook matrix multiplication algorithm solves the problem of multiplying two $n \times n$ matrices in $O(n^3)$ time. For many years, it was believed that no faster algorithms exist. That changed in the late 1960s when Strassen released his seminal work on matrix multiplication [150]. In this work, he exhibited a matrix multiplication algorithm that uses a divide-and-conquer method, splitting each of the two matrices to four different $n/2 \times n/2$ parts. Then, instead of multiplying these matrices block-wise, the algorithm computes seven new products, and then uses matrix addition to compute the product instead. This simple algebraic trick gives a lower time complexity, namely $O(n^{\log_2 7}) \approx O(n^{2.807})$. When used in practice, this algorithm is a little less numerically stable then the classical al-

© Technion - Israel Institute of Technology, Elyachar Central Library

gorithm, but works faster when $n \gtrsim 100$, allowing its implementation for large matrices.

Over the next few years, algorithms with better asymptotic time complexity were found using more complex divide-and-conquer techniques. The current state-of-the-art algorithm is due to Le Gall [77], which is heavily-based on the Coppersmith-Winograd algorithm [35]. Its time complexity is about $O(n^{2.3728639})$. However, the constant in front of $n^{2.3728639}$ is extremely large, namely the algorithm is worse than even the schoolbook matrix multiplication algorithm on matrices that can be manipulated using modern-day hardware [69].

The essential time complexity of matrix multiplication is usually denoted $O(n^{\omega})$, where poly-logarithmic terms are neglected [145]. It is widely believed that $\omega = 2$, namely that $n \times n$ matrices can be multiplied in about $O(n^2 p(\log(n)))$ time for some polynomial p [145]. The current lower bound is due to Raz [118], which proved that in a specific computational model, matrix multiplication takes at least $\Omega(n^2 \log(n))$ time. It should be noted that faster *probabilistic* methods for asserting whether $AB = C$ for three matrices A, B, C are known, for example, Freivalds's algorithm for asserting matrix multiplication over the field of two elements, $\mathbb{Z}_2 = \{0, 1\}$ [52] . It get some integer parameters $k > 0$, randomly chooses k vectors $\xi_1, ...\xi_k \in \mathbb{Z}_2^n$, and checks whether $A(B\xi_i) = C\xi_i$ for all $i = 1, \ldots, k$. If the answer is "yes" for all occasions, then the algorithm declares that $AB = C$, and if there is an occasion for which $A(B\xi_i) \neq C\xi_i$, then the algorithm declares that $AB \neq C$. It's easy to verify that if $AB = C$, the algorithm always returns the correct answer, and if $AB \neq C$, then the algorithm returns the correct answer with probability $\geq 1 - 2^{-k}$. Moreover, the time complexity is $O(kn^2)$, which is quadratic in n. However, we focus on deterministic matrix multiplication, as, for the best of my knowledge, there is no known probabilistic matrix multiplication algorithm which is faster than the one found in Le Gall [77].

It is widely known that matrix inversion and matrix multiplication have the same time complexity [36,145]. In Section 6.3, we take special interest in inversion of positive-definite matrices. We suspect that the time complexity of this restricted problem is the same as the time complexity of general matrix inversion, but we did not manage to find any meaningful results in the literature in this context, nor prove it myself. We denote the time complexity of the chosen algorithm for inverting positive definite matrices by $O(n^{\omega_1})$, similarly neglecting poly-logarithmic terms. This allows us to distinguish between real-world applications (in which we use the classical algorithm or Strassen's algorithm) to theoretical complexity bounds (in which we can use Coppersmith-Winograd's or Le Gall's algorithms). Moreover, the results of Section 6.3 still hold even if inversion of positive definite matrices turns out to be easier than inversion of general matrices. However, it should be noted that the inequality $2 \leq \omega_1 \leq \omega$ holds. Indeed, $\omega_1 \leq \omega$ as any general matrix inversion algorithm can also be applied to positive-definite matrices. Moreover, $\omega_1 \geq 2$ as reading all of the input's entries requires n^2 time.

© Technion - Israel Institute of Technology, Elyachar Central Library

Appendix E

Tools From Algebraic Number Theory

This brief appendix explains the concepts of transcendentality and linear independence in number theory and field theory [44, 49], as used in Section 6.2.

Fields are algebraic structures standing in the basis of linear algebra, as they are the cornerstone of the definition of vector spaces and linear transformations. Examples include $\mathbb{Q}, \mathbb{R}, \mathbb{C}$ with regular addition and multiplication, as well as $\mathbb{Z}_p = \{0, 1, \cdots, p-1\}$ with addition and multiplication modulo p, where p is any prime number. A *field extension* $\mathbb{K} \supseteq \mathbb{F}$ consists of two fields $\mathbb{K}, \mathbb{F}$, having the same addition and multiplication, while $\mathbb{K}$ is a super-set of $\mathbb{F}$. In that case, one can consider $\mathbb{K}$ as a vector space over $\mathbb{F}$, meaning that we can use standard notions from linear algebra. Namely, we say that $x_1, \cdots, x_n \in \mathbb{K}$ are *linearly independent over* $\mathbb{F}$ if for any elements $f_1, \cdots, f_n \in \mathbb{F}$, if $f_1 x_1 + \cdots f_n x_n = 0$ then $f_1 = \cdots = f_n = 0$. One example of a non-trivial linearly independent set can be seen in the proposition below:

Proposition E.1. *Let $\sqrt{p_1}, \cdots, \sqrt{p_n}$ be a sequence of the square roots of any distinct prime numbers. Then they are linearly independent over $\mathbb{Q}$.*

The reader is referred to [12] for a proof.

Field extensions are richer than vector fields, as they also allow multiplication, and as a result, exponentiation. If $\mathbb{K} \supseteq \mathbb{F}$ is a field extension, an element $a \in \mathbb{K}$ is called *transcendental* over $\mathbb{F}$ is the set $\{1, a, a^2, a^3, \cdots\}$ is linearly independent over $\mathbb{F}$. Equivalently, if p is any polynomial with coefficients in $\mathbb{F}$, then $p(a) \neq 0$. One can define $\mathbb{F}(a)$ as the smallest sub-field of $\mathbb{K}$ containing both $\mathbb{F}$ and a. If $\mathbb{F} = \mathbb{Q}$, then transcendentality over $\mathbb{F}$ is often abbreviated to transcendentality. Number which are not transcendental are called *algebraic*. Proving that specific real numbers are transcendental, or even irrational, is a notoriously hard problem. For example, it is known that at least one of $e + \pi$ and $e\pi$ is irrational, but it is not known which, not to mention transcendentality. A list of known transcendental numbers can be found below:

© Technion - Israel Institute of Technology, Elyachar Central Library

Theorem E.1. *The following numbers are transcendental:*

i) e *and* π.

ii) If $x \neq 0$ *is algebraic, then* e^x *is transcendental.*

iii) If $x \neq 1$ *is positive and algebraic, then the natural logarithm* $\log(x)$ *is transcendental.*

iv) If $x \neq 0, 1$ *is algebraic and* y *is algebraic and irrational, then* x^y *is transcendental.*

v) If x *is non-zero and algebraic, then* $\cos(x), \sin(x), \tan(x)$ *are all transcendental.*

The reader is referred to [85] for proofs. This list is far from comprehensive, but it includes most important examples of transcendental numbers, and excludes examples which are very rarely found in actual systems, e.g. Liouville's constant $\sum_{n=0}^{\infty} 10^{-n!}$.

© Technion - Israel Institute of Technology, Elyachar Central Library

Bibliography

[1] C. Altafini. Consensus problems on networks with antagonistic interactions. *IEEE Transactions on Automatic Control*, 58(4):935–946, 2013.

[2] M. H. Amini, B. Nabi, and M. R. Haghifam. Load management using multi-agent systems in smart distribution network. In *Proc. 2013 IEEE Power Energy Society General Meeting*, pages 1–5, July 2013.

[3] D. Angeli and E. D. Sontag. Monotone control systems. *IEEE Transactions on Automatic Control*, 48(10):1684–1698, Oct 2003.

[4] P. J. Antsaklis, B. Goodwine, V. Gupta, M. J. McCourt, Y. Wang, P. Wu, M. Xia, H. Yu, and F. Zhu. Control of cyberphysical systems using passivity and dissipativity based methods. *European Journal of Control*, 19(5):379 – 388, 2013.

[5] M. Arcak. Passivity as a design tool for group coordination. *IEEE Transactions on Automatic Control*, 52(8):1380–1390, Aug. 2007.

[6] M. Arcak, C. Meissen, and A. Packard. *Networks of Dissipative Systems*. Springer International Publishing, 2016.

[7] H. Bai, M. Arcak, and J. Wen. *Cooperative Control Design: A Systematic, Passivity-Based Approach*. Communications and Control Engineering. Springer, 2011.

[8] M. Bando, K. Hasebe, A. Nakayama, A. Shibata, and Y. Sugiyama. Dynamical model of traffic congestion and numerical simulation. *Phys. Rev. E*, 51:1035–1042, Feb 1995.

[9] R. G. Baraniuk. Compressive sensing [lecture notes]. *IEEE Signal Processing Magazine*, 24(4):118–121, July 2007.

[10] M. Becherif, M. Y. Ayad, A. Henni, and A. Aboubou. Hybridization of solar panel and batteries for street lighting by passivity based control. In *Proc. 2010 IEEE International Energy Conference*, pages 664–669, Dec 2010.

[11] D. P. Bertsekas. *Network optimization: continuous and discrete models*. Athena Scientific Belmont, 1998.

© Technion - Israel Institute of Technology, Elyachar Central Library

[12] A. S. Besicovitch. On the linear independence of fractional powers of integers. *Journal of the London Mathematical Society*, s1-15(1):3–6, 1940.

[13] F. Bianchini, G. Fenu, G. Giordano, and F. A. Pellegrino. Model-free tuning of plants with parasitic dynamics. In *Proc. 2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 499–504, Dec 2017.

[14] S. Boccaletti, M. Ivanchenko, V. Latora, A. Pluchino, and A. Rapisarda. Detecting complex network modularity by dynamical clustering. *Phys. Rev. E*, 75:045102, Apr 2007.

[15] J. A. Bondy, U. S. R. Murty, et al. *Graph theory with applications*, volume 290. Macmillan London, 1976.

[16] S. Bonnabel, P. Martin, and P. Rouchon. Non-linear symmetry-preserving observers on Lie groups. *IEEE Transactions on Automatic Control*, 54(7):1709–1713, July 2009.

[17] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[18] S. L. Bressler and A. K. Seth. Wiener–Granger causality: A well established methodology. *NeuroImage*, 58(2):323 – 329, 2011.

[19] G. Brinkmann, K. Coolsaet, J. Goedgebeur, and H. Mélot. House of graphs: a database of interesting graphs. *Discrete Applied Mathematics*, 161(1-2):311–314, 2013.

[20] D. A. Bristow, M. Tharayil, and A. G. Alleyne. A survey of iterative learning control. *IEEE Control Systems Magazine*, 26(3):96–114, 2006.

[21] X. Bu, Z. Hou, and H. Zhang. Data-driven multiagent systems consensus tracking using model free adaptive control. *IEEE Transactions on Neural Networks and Learning Systems*, 29(5):1514–1524, May 2018.

[22] D. A. Burbano Lombana, R. Freeman, and K. M. Lynch. Distributed inference of the multiplex network topology of complex systems. *IEEE Transactions on Control of Network Systems*, pages 1–1, 2019.

[23] M. Bürger, D. Zelazo, and F. Allgöwer. Duality and network theory in passivity-based cooperative control. *Automatica*, 50(8):2051–2061, 2014.

[24] C. I. Byrnes, A. Isidori, and J. C. Willems. Passivity, feedback equivalence, and the global stabilization of minimum phase nonlinear systems. *IEEE Transactions on Automatic Control*, 36(11):1228–1240, 1991.

[25] R. Caron and T. Traynor. The zero set of a polynomial. Available on http://www1.uwindsor.ca/math/sites/uwindsor.ca.math/files/05-03.pdf, 2005.

© Technion - Israel Institute of Technology, Elyachar Central Library

[26] A. Cayley. Desiderata and suggestions: No. 2. the theory of groups: Graphical representation. *American Journal of Mathematics*, 1(2):174–176, 1878.

[27] A. Chapman and M. Mesbahi. On symmetry and controllability of multi-agent systems. In *Proc. 53rd IEEE Conference on Decision and Control*, pages 625–630, Dec 2014.

[28] A. Chapman and M. Mesbahi. State controllability, output controllability and stabilizability of networks: A symmetry perspective. In *Proc. 2015 54th IEEE Conference on Decision and Control (CDC)*, pages 4776–4781, Dec 2015.

[29] A. Chen, J. Cao, and T. Bu. Network tomography: Identifiability and fourier domain estimation. *IEEE Transactions on Signal Processing*, 58(12):6029–6039, Dec 2010.

[30] L. Chen, J. Lu, and C. K. Tse. Synchronization: An obstacle to identification of network topology. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 56(4):310–314, April 2009.

[31] W. Chen, S. X. Ding, A. Q. Khan, and M. Abid. Energy based fault detection for dissipative systems. In *Proc. Conference on Control and Fault-Tolerant Systems (SysTol)*, pages 517–521, Oct 2010.

[32] N. Chopra and M. W. Spong. *Advances in Robot Control: From Everyday Physics to Human-Like Movements*, chapter Passivity-Based Control of Multi-Agent Systems, pages 107–134. Springer, 2006.

[33] T. Chu and C. Glymour. Search for additive nonlinear time series causal models. *J. Mach. Learn. Res.*, 9:967–991, June 2008.

[34] J. B. Conway. *A course in functional analysis*, volume 96. Springer Science & Business Media, 2013.

[35] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *Journal of Symbolic Computation*, 9(3):251 – 280, 1990. Computational algebraic complexity editorial.

[36] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition, 2009.

[37] R. F. Curtain and H. Zwart. *An introduction to infinite-dimensional linear systems theory*, volume 21. Springer Science & Business Media, 2012.

[38] C. Danielson and F. Borrelli. Symmetric linear model predictive control. *IEEE Transactions on Automatic Control*, 60(5):1244–1259, May 2015.

© Technion - Israel Institute of Technology, Elyachar Central Library

[39] M. R. Davoodi, K. Khorasani, H. A. Talebi, and H. R. Momeni. Distributed fault detection and isolation filter design for a network of heterogeneous multiagent systems. *IEEE Transactions on Control Systems Technology*, 22(3):1061–1069, May 2014.

[40] C. De Persis and N. Monshizadeh. Bregman storage functions for microgrid control. *IEEE Transactions on Automatic Control*, 63(1):53–68, 2018.

[41] Q. Dinh Vu, R. Tan, and D. K. Y. Yau. On applying fault detectors against false data injection attacks in cyber-physical control systems. In *Proc. IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9, April 2016.

[42] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, April 2006.

[43] F. Dörfler and F. Bullo. Synchronization in complex networks of phase oscillators: A survey. *Automatica*, 50(6):1539 – 1564, 2014.

[44] D. Dummit and R. Foote. *Abstract Algebra*. Wiley, 2004.

[45] J. Feiling, A. Zeller, and C. Ebenbauer. Derivative-free optimization algorithms basen on non-commutative maps. *IEEE Control Systems Letters*, 2:743–748, 2018.

[46] G. B. Folland. *A course in abstract harmonic analysis*. Chapman and Hall/CRC, 2016.

[47] A. L. Fradkov. Passification of non-square linear systems and feedback Yakubovich-Kalman-Popov lemma. *European Journal of Control*, 9(6):577–586, 2003.

[48] A. L. Fradkov and D. J. Hill. Exponential feedback passivity and stabilizability of nonlinear systems. *Automatica*, 34(6):697–703, 1998.

[49] J. B. Fraleigh. *A first course in abstract algebra*. Pearson Education India, 2003.

[50] A. Franchi, P. R. Giordano, C. Secchi, H. I. Son, and H. H. Bulthoff. Passivity-based decentralized approach for the bilateral teleoperation of a group of uavs with switching topology. In *Proc. IEEE International Conference on Robotics and Automation*, pages 898–905, Piscataway, NJ, USA, 2011.

[51] A. Franci, L. Scardovi, and A. Chaillet. An input-output approach to the robust synchronization of dynamical systems with an application to the hindmarsh-rose neuronal model. In *Proc. 50th IEEE Conference on Decision and Control and European Control Conference*, pages 6504–6509, Dec 2011.

© Technion - Israel Institute of Technology, Elyachar Central Library

[52] R. Freivalds. Probabilistic machines can use less running time. In *Proc. IFIP congress*, volume 839, page 842, 1977.

[53] E. Fridman. *Introduction to time-delay systems: Analysis and control*. Springer, 2014.

[54] A. Fujita, J. R. Sato, H. M. Garay-Malpartida, R. Yamaguchi, S. Miyano, M. C. Sogayar, and C. E. Ferreira. Modeling gene expression regulatory networks with the sparse vector autoregressive model. *BMC Systems Biology*, 1(1):39, Aug 2007.

[55] D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1):17 – 40, 1976.

[56] T. S. Gardner, D. di Bernardo, D. Lorenz, and J. J. Collins. Inferring genetic networks and identifying compound mode of action via expression profiling. *Science*, 301(5629):102–105, 2003.

[57] C. Godsil and G. Royle. *Algebraic Graph Theory*. Graduate Texts in Mathematics. Springer New York, 2001.

[58] K. Gu, J. Chen, and V. L. Kharitonov. *Stability of time-delay systems*. Springer Science & Business Media, 2003.

[59] Y. Han, W. Lu, and T. Chen. Cluster consensus in discrete-time networks of multiagents with inter-cluster nonidentical inputs. *IEEE Transactions on Neural Networks and Learning Systems*, 24(4):566–578, April 2013.

[60] F. Harary. The maximum connectivity of a graph. *Proceedings of the National Academy of Sciences of the United States of America*, 48(7):1142–1146, 1962.

[61] R. Harvey and Z. Qu. Cooperative control and networked operation of passivity-short systems. In K. Vamvoudakis and S. S. Jagannathan, editors, *Control of Complex Systems: Theory and Applications*, pages 499–518. Elsevier, 2016.

[62] T. Hatanaka, N. Chopra, M. Fujita, and M. Spong. *Passivity-Based Control and Estimation in Networked Robotics*. Communications and Control Engineering. Springer International Publishing, 1 edition, 2015.

[63] A. Hatcher. *Algebraic Topology*. Cambridge University Press, 1st edition, 2001. p. 114–115.

[64] G. H. Hines, M. Arcak, and A. K. Packard. Equilibrium-independent passivity: A new definition and numerical certification. *Automatica*, 47(9):1949–1956, 2011.

© Technion - Israel Institute of Technology, Elyachar Central Library

[65] J. J. Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences*, 81(10):3088–3092, 1984.

[66] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, New York, NY, USA, 2nd edition, 2012.

[67] Z.-S. Hou and Z. Wang. From model-based control to data-driven control: Survey, classification and perspective. *Inf. Sci.*, 235:3–35, June 2013.

[68] P. Iñigo-Blasco, F. D. del Rio, M. C. Romero-Ternero, D. Cagigas-Muñiz, and S. Vicente-Diaz. Robotics software frameworks for multi-agent robotic systems development. *Robotics and Autonomous Systems*, 60(6):803 – 821, 2012.

[69] C. Iliopoulos. Worst-case complexity bounds on algorithms for computing the canonical structure of finite abelian groups and the hermite and smith normal forms of an integer matrix. *SIAM Journal on Computing*, 18(4):658–669, 1989.

[70] R. Isermann. *Fault-diagnosis systems: an introduction from fault detection to fault tolerance*. Springer Science & Business Media, 2006.

[71] T. Ishizaki, A. Ueda, and J. I. Imura. Convex gradient controller design for incrementally passive systems with quadratic storage functions. In *Proc. 2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 1721–1726, Dec 2016.

[72] A. Jain, M. Sharf, and D. Zelazo. Regulatization and feedback passivation in cooperative control of passivity-short systems: A network optimization perspective. *IEEE Control Systems Letters*, 2:731–736, 2018.

[73] H. Jiang and H. He. Data-driven distributed output consensus control for partially observable multiagent systems. *IEEE Transactions on Cybernetics*, pages 1–11, 2018.

[74] A. Julius, M. Zavlanos, S. Boyd, and G. J. Pappas. Genetic network identification using convex programming. *IET Systems Biology*, 3(3):155–166, May 2009.

[75] H. Khalil. *Nonlinear Systems*. Pearson Education. Prentice Hall, 2002.

[76] J. Kim, J. Yang, H. Shim, J. Kim, and J. H. Seo. Robustness of synchronization of heterogeneous agents by strong coupling and a large number of agents. *IEEE Transactions on Automatic Control*, 61(10):3096–3102, 2016.

[77] F. Le Gall. Powers of tensors and fast matrix multiplication. In *Proc. 39th International Symposium on Symbolic and Algebraic Computation*, ISSAC '14, pages 296–303, New York, NY, USA, 2014. ACM.

© Technion - Israel Institute of Technology, Elyachar Central Library

[78] D. Lee and M. W. Spong. Passive bilateral teleoperation with constant time delay. *IEEE Transactions on Robotics*, 22(2):269–281, April 2006.

[79] Q. Lei and J. Bao. Dissipativity based fault detection and diagnosis for linear systems. In *Proc. IEEE Conference on Control Applications (CCA)*, pages 133–138, Sep. 2015.

[80] Q. Lei, R. Wang, and J. Bao. Fault diagnosis based on dissipativity property. *Computers and Chemical Engineering*, 108:360 – 371, 2018.

[81] G. Li, X. Wu, J. Liu, J.-a. Lu, and C. Guo. Recovering network topologies via taylor expansion and compressive sensing. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 25(4):043102, 2015.

[82] Y. Li, H. Fang, J. Chen, and C. Yu. Distributed cooperative fault detection for multiagent systems: A mixed H_∞/H_2 optimization approach. *IEEE Transactions on Industrial Electronics*, 65(8):6468–6477, Aug 2018.

[83] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási. Controllability of complex networks. *Nature*, 473(7346):167–173, may 2011.

[84] L. Lovász. Random walks on graphs: A survey.

[85] K. Mahler. *Lectures on transcendental numbers*, volume 546. Springer, 2006.

[86] J. Malý. Absolutely continuous functions of several variables. *Journal of Mathematical Analysis and Applications*, 231(2):492 – 508, 1999.

[87] D. Materassi and G. Innocenti. Unveiling the connectivity structure of financial networks via high-frequency analysis. *Physica A: Statistical Mechanics and its Applications*, 388(18):3866–3878, 2009.

[88] D. Materassi and G. Innocenti. Topological identification in networks of dynamical systems. *IEEE Transactions on Automatic Control*, 55(8):1860–1871, Aug 2010.

[89] A. Mauroy and J. M. Hendrickx. Spectral identification of networks with inputs. In *Proc. 2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 469–474, Dec 2017.

[90] M. J. McCourt and P. J. Antsaklis. Connection between the passivity index and conic systems. *ISIS*, 9:009, 2009.

[91] P. P. Menon and C. Edwards. Robust fault estimation using relative information in linear multi-agent networks. *IEEE Transactions on Automatic Control*, 59(2):477–482, Feb 2014.

[92] M. Mesbahi and M. Egerstedt. *Graph Theoretic Methods in Multiagent Networks*. Princeton Series in Applied Mathematics. Princeton University Press, 2010.

© Technion - Israel Institute of Technology, Elyachar Central Library

[93] M. Mohammedi, O. Kraa, M. Becherif, A. Aboubou, M. Ayad, and M. Bahri. Fuzzy logic and passivity-based controller applied to electric vehicle using fuel cell and supercapacitors hybrid source. *Energy Procedia*, 50:619 – 626, 2014. Technologies and Materials for Renewable Energy, Environment and Sustainability (TMREES14 – EUMISD).

[94] P. Monestiez, J.-S. Bailly, P. Lagacherie, and M. Voltz. Geostatistical modelling of spatial processes on directed trees: Application to fluvisol extent. *Geoderma*, 128(3):179 – 191, 2005. Pedometrics 2003.

[95] J. M. Montenbruck and F. Allgöwer. Some problems arising in controller design from big data via input-output methods. In *Proc. 2016 IEEE 55th Annual Conference on Decision and Control (CDC)*, pages 6525–6530, 2016.

[96] J. M. Montenbruck, M. Bürger, and F. Allgöwer. Practical synchronization with diffusive couplings. *Automatica*, 53:235 – 243, 2015.

[97] L. Márton. Energetic approach to deal with faults in robot actuators. In *Proc. 20th Mediterranean Conference on Control Automation (MED)*, pages 85–90, July 2012.

[98] L. Márton and J. A. Esclusa. Energetic approach for actuator fault accommodation: Application to bilateral teleoperation. In *Proc. Conference on Control and Fault-Tolerant Systems (SysTol)*, pages 720–726, Oct 2013.

[99] L. Márton and D. Ossmann. Energetic approach for control surface disconnection fault detection in hydraulic aircraft actuators. *IFAC Proceedings Volumes*, 45(20):1149 – 1154, 2012. 8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes.

[100] M. Nabi-Abdolyousefi and M. Mesbahi. Network identification via node knockout. *IEEE Transactions on Automatic Control*, 57(12):3214–3219, Dec 2012.

[101] M. Nabi-Abdolyousefi and M. Mesbahi. A sieve method for consensus-type network tomography. In *Controllability, Identification, and Randomness in Distributed Systems*, chapter 3, pages 31–38. Springer Theses (Recognizing Outstanding Ph.D. Research), Springer, Cham, 2014.

[102] M. J. Naylor, L. C. Rose, and B. J. Moyle. Topology of foreign exchange markets using hierarchical structure methods. *Physica A: Statistical Mechanics and its Applications*, 382(1):199 – 208, 2007. Applications of Physics in Financial Analysis.

[103] Q. T. T. Nguyen, N. Messai, S. Martinez-Martinez, and N. Manamanni. A distributed fault detection observer-based approach for a network of multi-agent systems with switching topologies. In *Proc. 2017 11th Asian Control Conference (ASCC)*, pages 1513–1518, Dec 2017.

© Technion - Israel Institute of Technology, Elyachar Central Library

[104] E. Nozari, Y. Zhao, and J. Cortés. Network identification with latent nodes via auto-regressive models. *IEEE Transactions on Control of Network Systems*, PP(99):1–1, 2017.

[105] K.-K. Oh, M.-C. Park, and H.-S. Ahn. A survey of multi-agent formation control. *Automatica*, 53:424 – 440, 2015.

[106] B. Oksendal. *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media, 2013.

[107] R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, Jan 2007.

[108] R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533, Sept 2004.

[109] K. M. Passino. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems Magazine*, 22(3):52–67, June 2002.

[110] A. Pavlov and L. Marconi. Incremental passivity and output regulation. *Systems & Control Letters*, 57(5):400 – 409, 2008.

[111] J. Qin and C. Yu. Cluster consensus control of generic linear multi-agent systems under directed topology with acyclic partition. *Automatica*, 49(9):2898 – 2905, 2013.

[112] P. Qin, B. Dai, B. Huang, G. Xu, and K. Wu. A survey on network tomography with network coding. *IEEE Communications Surveys Tutorials*, 16(4):1981–1995, Fourthquarter 2014.

[113] Z. Qu and M. A. Simaan. Modularized design for cooperative control and plug-and-play operation of networked heterogeneous systems. *Automatica*, 50(9):2405 – 2414, 2014.

[114] Y. Quan, W. Chen, Z. Wu, and L. Peng. Distributed fault detection for second-order delayed multi-agent systems with adversaries. *IEEE Access*, 5:16478–16483, 2017.

[115] A. Quteishat, C. P. Lim, J. Tweedale, and L. C. Jain. A neural network-based multi-agent classifier system. *Neurocomputing*, 72(7):1639 – 1647, 2009. Advances in Machine Learning and Computational Intelligence.

[116] A. A. Rad, M. Jalili, and M. Hasler. A lower bound for algebraic connectivity based on the connection-graph-stability method. *Linear Algebra and its Applications*, 435(1):186 – 192, 2011.

© Technion - Israel Institute of Technology, Elyachar Central Library

[117] A. Rahmani and M. Mesbahi. Pulling the strings on agreement: Anchoring, controllability, and graph automorphisms. In *Proc. 2007 American Control Conference*, pages 2738–2743, July 2007.

[118] R. Raz. On the complexity of matrix product. *SIAM J. Comput.*, 32(5):1356–1369, May 2003.

[119] R. T. Rockafellar. Characterization of the subdifferentials of convex functions. *Pacific Journal of Mathematics*, 17(3):497—-510, 1966.

[120] R. T. Rockafellar. Lagrange multipliers and optimality. *SIAM Review*, 35(2):183–238, 1993.

[121] R. T. Rockafellar. *Convex Analysis*. Princeton Landmarks in Mathematics and Physics. Princeton University Press, 1997.

[122] R. T. Rockafellar. *Network Flows and Monotropic Optimization*. Athena Scientific, Belmont, Massachusetts, 1998.

[123] A. Romer, J. Berberich, J. Köhler, and F. Allgöwer. One-shot verification of dissipativity properties from input-output data. *IEEE Control Systems Letters*, 3:709–714, 2019.

[124] A. Romer, J. M. Montenbruck, and F. Allgöwer. Determining dissipation inequalities from input-output samples. In *Proc. 20th IFAC World Congress*, pages 7789–7794, 2017.

[125] A. Romer, J. M. Montenbruck, and F. Allgöwer. Sampling strategies for data-driven inference of passivity properties. In *Proc. IEEE 56th Conference on Decision and Control (CDC)*, pages 6389–6394, Melbourne, Australia, 2017.

[126] J. Sabater and C. Sierra. Reputation and social network analysis in multiagent systems. In *Proc. the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 1*, AAMAS '02, pages 475–482. ACM, 2002.

[127] V. Sakkalis. Review of advanced techniques for the estimation of brain connectivity measured with eeg/meg. *Computers in Biology and Medicine*, 41(12):1110 – 1117, 2011. Special Issue on Techniques for Measuring Brain Connectivity.

[128] B. M. Sanandaji, T. L. Vincent, and M. B. Wakin. Exact topology identification of large-scale interconnected dynamical systems from compressive observations. In *Proc. 2011 American Control Conference*, pages 649–656, June 2011.

[129] L. Scardovi, M. Arcak, and E. Sontag. Synchronization of interconnected systems with applications to biochemical networks: An input-output approach. *IEEE Transactions on Automatic Control*, 55(6):1367–1379, June 2010.

© Technion - Israel Institute of Technology, Elyachar Central Library

[130] L. Scardovi, M. Arcak, and E. D. Sontag. Synchronization of interconnected systems with an input-output approach. part ii: State-space result and application to biochemical networks. In *Proc. 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 615–620, Dec 2009.

[131] L. Scardovi and N. E. Leonard. Robustness of aggregation in networked dynamical systems. In *Proc. International Conference on Robot Communication and Coordination*, pages 1–6, Odense, Denmark, 2009.

[132] A. Schnitzler and J. Gross. Normal and pathological oscillatory communication in the brain. *Nature reviews. Neuroscience*, 6:285–96, 05 2005.

[133] R. Sepulchre, M. Janković, and P. V. Kokotović. *Constructive Nonlinear Control*. Springer-Verlag London, 1997.

[134] M. Sharf, A. Jain, and D. Zelazo. A geometric method for passivation and cooperative control of equilibrium-independent passivity-short systems. *arXiv preprint arXiv:1901.06512*, 2019. Submitted to IEEE Transactions on Automatic Control.

[135] M. Sharf, A. Koch, D. Zelazo, and F. Allgöwer. Model-free practical cooperative control for diffusively coupled systems. *arXiv preprint arXiv:1906.05204*, 2019. Submitted to IEEE Transactions on Automatic Control.

[136] M. Sharf and D. Zelazo. A network optimization approach to cooperative control synthesis. *IEEE Control Systems Letters*, 1(1):86–91, July 2017.

[137] M. Sharf and D. Zelazo. Network identification: A passivity and network optimization approach. In *Proc. 2018 IEEE Conference on Decision and Control (CDC)*, pages 2107–2113, Dec 2018.

[138] M. Sharf and D. Zelazo. Analysis and synthesis of mimo multi-agent systems using network optimization. *IEEE Transactions on Automatic Control*, 64(11):4512–4524, Nov. 2019.

[139] M. Sharf and D. Zelazo. A data-driven and model-based approach to fault detection and isolation in networked systems. *arXiv preprint arXiv:1908.03588*, 2019. Submitted to IEEE Transactions on Automatic Control.

[140] M. Sharf and D. Zelazo. Network feedback passivation of passivity-short multi-agent systems. *IEEE Control Systems Letters*, 3(3):607–612, July 2019.

[141] M. Sharf and D. Zelazo. Network identification for diffusively-coupled systems with minimal time complexity. *arXiv preprint arXiv:1903.04923*, 2019. Submitted to IEEE Transactions on the Control of Networked Systems.

© Technion - Israel Institute of Technology, Elyachar Central Library

[142] M. Sharf and D. Zelazo. Symmetry-induced clustering in multi-agent systems using network optimization and passivity. In *Proc. 2019 27th Mediterranean Conference on Control and Automation (MED)*, pages 19–24, July 2019.

[143] L. Sigler. *Fibonacci's Liber Abaci: a translation into modern English of Leonardo Pisano's book of calculation*. Springer Science & Business Media, 2003.

[144] J. W. Simpson-Porco. Equilibrium-independent dissipativity with quadratic supply rates. *IEEE Transactions on Automatic Control*, 64(4):1440–1455, April 2019.

[145] S. S. Skiena. *The Algorithm Design Manual*. Springer Publishing Company, Incorporated, 2nd edition, 2008.

[146] M. W. Spong and F. Bullo. Controlled symmetries and passive walking. *IEEE Transactions on Automatic Control*, 50(7):1025–1031, July 2005.

[147] M. W. Spong, J. K. Holm, and D. Lee. Passivity-based control of bipedal locomotion. *IEEE Robotics Automation Magazine*, 14(2):30–40, June 2007.

[148] G.-B. Stan and R. Sepulchre. Analysis of interconnected oscillators by dissipativity theory. *IEEE Transactions on Automatic Control*, 52(2):256–270, Feb. 2007.

[149] L. Stone, R. Olinky, B. Blasius, A. Huppert, and B. Cazelles. Complex synchronization phenomena in ecological systems. *AIP Conf. Proc.*, 622, 07 2002.

[150] V. Strassen. Gaussian elimination is not optimal. *Numer. Math.*, 13(4):354–356, Aug. 1969.

[151] J. Sztipanovits, X. Koutsoukos, G. Karsai, N. Kottenstette, P. Antsaklis, V. Gupta, B. Goodwine, J. Baras, and S. Wang. Toward a science of cyber–physical system integration. *Proceedings of the IEEE*, 100(1):29–44, Jan 2012.

[152] P. Tabuada, W. Ma, J. Grizzle, and A. D. Ames. Data-driven control for feedback linearizable single-input systems. In *Proc. 2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 6265–6270, Dec 2017.

[153] M. Tanemura and S.-I. Azuma. Efficient data-driven estimation of passivity properties. *IEEE Control Systems Letters*, 3:398–403, 2019.

[154] Y. Tang, Y. Hong, and P. Yi. Distributed optimization design based on passivity technique. In *Proc. 2016 12th IEEE International Conference on Control and Automation (ICCA)*, pages 732–737, June 2016.

© Technion - Israel Institute of Technology, Elyachar Central Library

[155] A. Teixeira, I. Shames, H. Sandberg, and K. H. Johansson. Distributed fault detection and isolation resilient to network model uncertainties. *IEEE Transactions on Cybernetics*, 44(11):2024–2037, Nov 2014.

[156] M. Timme. Revealing network connectivity from response dynamics. *Phys. Rev. Lett.*, 98:224101, May 2007.

[157] S. Trip, M. Bürger, and C. De Persis. An internal model approach to (optimal) frequency regulation in power grids with time-varying voltages. *Automatica*, 64:240–253, 2016.

[158] S. Trip and C. De Persis. Distributed optimal load frequency control with non-passive dynamics. *IEEE Transactions on Control of Network Systems*, 5(3):1232–1244, 2018.

[159] D. Urban and T. Keitt. Landscape connectivity: A graph-theoretic perspective. *Ecology*, 82(5):1205–1218, 2001.

[160] A. J. van der Schaft. *L2-Gain and Passivity Techniques in Nonlinear Control*. New York: Springer-Verlag, 2 edition, 1999.

[161] A. J. van der Schaft and D. Jeltsema. *Port-Hamiltonian Systems Theory: An Introductory Overview*. Now Publishers Inc, 2014.

[162] A. J. van der Schaft and B. M. Maschke. Port-hamiltonian systems on graphs. *SIAM Journal on Control and Optimization*, 51(2):906–937, 2013.

[163] R. Vicente, M. Wibral, M. Lindner, and G. Pipa. Transfer entropy—a model-free measure of effective connectivity for the neurosciences. *Journal of Computational Neuroscience*, 30(1):45–67, Feb 2011.

[164] D. D. Šiljak. Decentralized control and computations: Status and prospects. *A. Rev. Control*, 20, 1996.

[165] L. Wang and F. Xiao. Finite-time consensus problems for networks of dynamic agents. *IEEE Transactions on Automatic Control*, 55(4):950–955, April 2010.

[166] W.-J. Wang and J.-Y. Chen. Passivity-based sliding mode position control for induction motor drives. *IEEE Transactions on Energy Conversion*, 20(2):316–321, June 2005.

[167] M. Xia, P. J. Antsaklis, and V. Gupta. Passivity indices and passivation of systems with application to systems with input/output delay. In *Proc. IEEE Conference on Decision and Control (CDC)*, pages 783–788, Los Angeles, California, USA, December 15-17, 2014.

[168] M. Xia, P. J. Antsaklis, V. Gupta, and F. Zhu. Passivity and dissipativity analysis of a system and its approximation. *IEEE Transactions on Automatic Control*, 62(2):620–635, 2017.

© Technion - Israel Institute of Technology, Elyachar Central Library

[169] M. Xia, A. Rahnama, S. Wang, and P. J. Antsaklis. Control design using passivation for stability and performance. *IEEE Transactions on Automatic Control*, 63(9):2987–2993, 2018.

[170] H. Yang, V. Cocquempot, and B. Jiang. Fault tolerance analysis for switched systems via global passivity. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 55(12):1279–1283, Dec 2008.

[171] W. Yu, P. DeLellis, G. Chen, M. di Bernardo, and J. Kurths. Distributed adaptive control of synchronization in complex networks. *IEEE Transactions on Automatic Control*, 57:2153–2158, 2012.

[172] Y. Yuan, G.-B. Stan, S. Warnick, and J. Goncalves. Robust dynamical network structure reconstruction. *Automatica*, 47(6):1230 – 1235, 2011. Special Issue on Systems Biology.

[173] G. Zames. On the input-output stability of time-varying nonlinear feedback systems part one: Conditions derived using concepts of loop gain, conicity, and positivity. *IEEE Transactions on Automatic Control*, 11(2):228–238, April 1966.

[174] D. Zelazo and M. Mesbahi. Edge agreement: Graph-theoretic performance bounds and passivity analysis. *IEEE Transactions on Automatic Control*, 56(3):544–555, Mar. 2010.

[175] C. Zhao, U. Topcu, N. Li, and S. Low. Design and stability of load-side primary frequency control in power systems. *IEEE Transactions on Automatic Control*, 59(5):1177–1189, May 2014.

[176] S. Zhao and D. Zelazo. Bearing rigidity and almost global bearing-only formation stabilization. *IEEE Transactions on Automatic Control*, 61(5):1255–1268, May 2016.

[177] E. Zheleva, E. Terzi, and L. Getoor. Privacy in social networks. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 3(1):1–85, 2012.

[178] Y. Zheng, S. E. Li, K. Li, and W. Ren. Platooning of connected vehicles with undirected topologies: Robustness analysis and distributed H-infinity controller synthesis. *IEEE Transactions on Intelligent Transportation Systems*, 19:1353–1364, 2018.

[179] F. Zhu, M. Xia, and P. J. Antsaklis. Passivity analysis and passivation of feedback systems using passivity indices. In *Proc. 2014 American Control Conference*, pages 1833–1838, June 2014.

[180] D. Zwillinger, V. Moll, I. Gradshteyn, and I. Ryzhik. Indefinite integrals of elementary functions. In D. Zwillinger, V. Moll, I. Gradshteyn, and I. Ryzhik, editors, *Table of Integrals, Series, and Products (Eighth Edition)*, pages 63 – 247. Academic Press, Boston, 2014.

לא ידוע אילו זוגות של סוכנים מקיימים יחסי גומלין כאלה. אנחנו מציגים אלגוריתם לפתרון הבעיה בעזרת הזרקת אותות חיצוניים לסוכנים, המבוסס על לינאריזציה של הקשר בין קלט חיצוני לגבול המערכת במצב יציב, שבתורו מבוסס על המסגרת המבוססת על אופטימיזציה על רשתות. אנחנו מראים כי אלגוריתם זה פותר את הבעיה בצורה מקורבת, ונותנים חסם על השגיאה. אנחנו מראים גם כי האלגוריתם בעל מידת חסינות מסוימת לרעש, ומראים כי ניתן להפעילו גם במקרה בו לא ניתן להכניס אות חיצוני לכל הסוכנים. לבסוף, אנחנו גם מראים כי האלגוריתם שפותח הוא אופטימלי מבחינת סיבוכיות הזמן הנדרשת. אלגוריתם זה מאפשר למעשה למצוא את המבנה הטופולוגי של הרשת באמצעות תזוזות קטנות מאוד ממצב יציב נתון, כך שניתן להשתמש בו גם במקרים בהם ישנה דרישה חזקה לבטיחות, דוגמת רשתות של מכוניות אוטונומיות.

הפרק השביעי של התזה עוסק באפליקציה של המסגרת המבוססת על אופטימיזציה על רשתות לזיהוי והתגברות על תקלות תקשורת ברשתות. אנו חוזרים לבעיית הערך הסופי, אנו מניחים שחלק מהקשרים בין הסוכנים ברשת עלולים להישבר, בין אם עקב תקלת חומרה בסוכנים, או מתקפת סייבר עליהם. ראשית, אנו מראים כי ניתן להבדיל בין הגרסה התקינה של הרשת לבין הגרסאות התקולות שלה בצורה אסימפטוטית, תוך כדי פתרון בעיית הערך הסופי. הנ"ל נעשה באמצעות שימוש במסגרת המבוססת על אופטימיזציה על רשתות, כמו גם שימוש בתורת הגרפים. אנו מראים כיצד ניתן להפוך את ההבדלה האסימפטוטית להבדלה בזמן אמת על ידי שימוש באלגוריתמי "וידוא התכנסות", המקבלים מערכת מרובת סוכנים וגבול משוער ובודקים האם המערכת מתכנסת לגבול המשוער, ומשתמשים בהבדלה הזו כדי לבנות אלגוריתמים לזיהוי והתגברות על תקלות תקשורת בזמן אמת, תוך פתרון בעיית הערך הסופי בו-זמנית. לבסוף, אנו חוקרים את אלגוריתמי וידוא ההתכנסות, ומציעים שתי חלופות לבניית אלגוריתמים כאלו המתבססות על פסיביות ונתוני זמן-אמת.

הפרק השמיני והאחרון של התזה מסכם אותה ומציג כיווני מחקר לעתיד. בנוסף, התזה מצוידת בחמישה נספחים שונים בנושאי אנליזה קמורה, תורת הגרפים, תורת המערכות, תורת החישוביות ותורת המספרים האלגברית. שלושת הנספחים הראשונים מסכמים חומר מקדים הנדרש לתזה בתחומים הנזכרים מעלה, ושני הנספחים האחרונים מהווים הרחבות של נקודות ספציפיות בתזה.

© Technion - Israel Institute of Technology, Elyachar Central Library

© Technion - Israel Institute of Technology, Elyachar Central Library

הפרק הראשון של התזה מהווה מבוא לשאר הפרקים, וכולל סקר ספרות על רשתות מרובות-סוכנים ושימושיהן, כמו גם רקע נדרש באופטימיזציה על רשתות, רשתות בעלות צימוד דיפוזיבי, ושימושים בתורת הפסיביות עבור מערכות מרובות סוכנים. בפרק זה גם מוצג גם סיכום בראשי פרקים של הממצאים בכל אחד מפרקי התזה.

הפרק השני של התזה עוסק בהרחבת המסגרת המבוססת על אופטימיזציה על רשתות שהוצעה על ידי ברגר, זלזו ואלגוור לרשתות בהן הסוכנים והבקרים הם מערכות עם כניסות ויציאות מרובות. אנחנו משתמשים בתוצאה קלאסית של רוקפלר (Rockafellar) בנוגע ליחסים מונוטוניים-ציקלית מקסימליים (Maximal cyclically monotone) וקשרם לפונקציות קמורות כדי להציע הרחבה ל-MEIP המתאימה גם למערכות מרובות כניסות ויציאות. אנו מראים שההרחבה המוצעת, פסיביות מונוטונית-ציקלית לא-מבוססת שיווי-משקל מקסימלית (independent cyclically monotone passivity, MEICMP) Maximal equilibrium), היא אכן הרחבה של MEIP, ושהיא מאפשרת הרחבה של המסגרת המבוססת על אופטימיזציה על רשתות. לאחר מכן אנחנו מראים שתכונה זו מתקיימת עבור מחלקה רחבה של מערכות רבות, מה שמעיד על השימוש האפשרי הנרחב בתוצאה זו עבור מערכות בעלות כניסות ויציאות מרובות.

הפרק השלישי של התזה עוסק בהרחבת המסגרת המבוססת על אופטימיזציה על רשתות עבור מקרים בהם הסוכנים אינם פסיביים, אלא מערכות בעלות חוסר-פסיביות. ראשית, אנחנו מדגימים כיצד המסגרת המבוססת על אופטימיזציה על רשתות נכשלת עבור רשתות כאלו, ומאפיינים את הגורם לכישלון כחוסר קמירות או חוסר מוגדרות היטב של בעיות האופטימיזציה על רשתות במקרה זה. שנית, אנחנו מראים שאם בעיות אלו מוגדרות היטב אך אינן קמורות, ניתן להפוך את הבעיה לקמורה על ידי הוספת גורם ריבועי לפונקציית המחיר. אנחנו מראים כי גורם זה למעשה משרה גורם פידבק הגורם למערכת להפוך לפסיבית, מה שמאפשר לנסח ולהוכיח גרסה מותאמת של המסגרת המבוססת על אופטימיזציה על רשתות. אנחנו מראים שלוש דוגמאות לגורם הריבועי, המתאימות לשלושה סוגי פידבק שונים. לבסוף, אנחנו תוקפים את המקרה בו בעיות האופטימיזציה כלל אינן מוגדרות, ומראים שמצב זה נגרם מחוסר-מונוטוניות קיצונית של אוסף כל זוגות הקלט-פלט במצב יציב של הסוכנים. אנחנו מציעים באופן גיאומטרי העתקה ההופכת את האוסף הנ"ל למונוטוני, ומראים שהעתקה זו משרה טרנספורמציה של הסוכנים, ההופכת אותם לפסיביים, ומאפשרת לנסח ולהוכיח גרסה מותאמת של המסגרת המבוססת על אופטימיזציה על רשתות. דהיינו, המסגרת המבוססת על אופטימיזציה על רשתות ניתנת להרחבה למערכות בעלות חוסר-פסיביות, כל עוד היא מותאמת בצורה ראויה.

הפרק הרביעי של התזה עוסק בשימוש בתוצאת האנליזה העומדת בבסיס המסגרת המבוססת על אופטימיזציה על רשתות כדי לבצע סינתזה של בקרים. אנחנו מתמקדים בבעיית הערך הסופי, בה יש לסנתז בקרים מרושתים כך שהמערכת תתכנס לערך מבוקש. אנחנו משתמשים בשיטות המבוססות על חשבון תת-גרדיאנטים (subgradients) וקמירות ממש כדי לבנות את הבקרים הנ"ל. לאחר מכן, אנחנו חוקרים את ההשפעות של סימטריות ברשת על פתרון בעיית הערך הסופי. אנחנו מראים כי סימטריות ברשת גוזרות מבנה צבירי (clustering), בו הסוכנים מתכנסים לצבירים בגדלים שונים ומקומות שונים. אנחנו מראים כיצד ניתן לחזות את גודל ומבנה הצבירים לפי הסימטריות ברשת. לאחר מכן אנחנו מתרכזים במקרה בו כל הסוכנים מתנהגים בצורה זהה, ומראים כיצד ניתן לבנות רשת בה כל הבקרים זהים זה לזה, והיא מתכנסת למבנה צבירי נתון.

הפרק החמישי של התזה עוסק באפליקציה הראשונה של התוצאות, המתרכזת בתחום הבקרה מבוססת-נתונים. בתחום זה, המטרה היא לבנות מערכות בקרה מבלי להחזיק במודל של המערכת המבוקרת, או כאשר ישנו מודל מאוד מעורפל, אלא רק בסט נתונים הקשור למערכת. אנחנו חוקרים את בעיית הערך הסופי במקרה זה. אנחנו מראים כי בקר מדורג (cascaded) המורכב מבקר קבוע הנבחר בצורה מתאימה והגבר שניתן לשינוי פותר את הבעיה, כאשר ההוכחה מתבססת על המסגרת המבוססת על אופטימיזציה על רשתות. אנחנו מציגים שתי דרכים לבחירת ערך ההגבר בפועל מתוך נתונים, ומשווים בין הביצועים של שתי השיטות.

הפרק השישי של התזה עוסק באפליקציה נוספת, המתרכזת בבעיית זיהוי הרשת. בבעיה זו, נתונה רשת מרובת סוכנים בה ישנם מודלים מדויקים של הסוכנים ויחסי הגומלין בין הזוגות השונים, אך

© Technion - Israel Institute of Technology, Elyachar Central Library

תקציר עברי

בקרה מבוזרת ורשתות מרובות-סוכנים היוו כר פורה למחקר בשנים האחרונות, בו הושגה מסגרת תיאורטית רחבה לניתוח רשתות אלו, בד בבד עם מנעד רחב של שימושים בתחומים רבים, הכוללים רשתות ניורונים, ציי רכבים אוטונומיים, נחילי רובוטים, רשתות לייצור חשמל, ומערכי לוויינים. לאורך השנים, חוקרים ניסחו והציעו מסגרות רבות לניתוח אחיד של רשתות המורכבות ממערכות דינמיות. שני נושאים שצצו במסגרות רבות כאלו הם תורת הגרפים, המשמשת למידול יחסי הגומלין בין הסוכנים ברשת, ובקרה מבוססת-אנרגיה, הידועה גם בשם פסיביות. תורת הפסיביות מנסה להשליך מושגים מעולם הרשתות החשמליות למערכות דינמיות כלליות, ומתרכזת בקשר בין אנרגיה להספק. היא משמשת רבות לחקר מערכות לא לינאריות. תורת הפסיביות הוכנסה לראשונה לעולם הרשתות מרובות-הסוכנים ע"י ארצ'אק (Arcak) ב-2007, אך בנוסח הסטנדרטי שלה, המסתכל על פסיביות ביחס לשיווי משקל מסוים. נוסח זה מאפשר, למשל, להוכיח שהרשת מתכנסת למצב גבול מסוים, אך יש לדעת מראש את הגבול. כדי להתמודד עם בעיה זו, ולאפשר ניתוח של התנהגות הרשת מבלי ידע מוקדם על התנהגות זו, נוסחים שונים של פסיביות הוצעו. שניים מהם כוללים פסיביות אינקרמנטלית (Incremental Passivity), ופסיביות לא-תלוית-שיווי-משקל (Equilibrium independent passivity), המקוצרת לרוב כ-EIP.

ב-2014, ברגר, זלזו ואלגוור (Bürger, Zelazo & Allgöwer) הציעו נוסח חדש של פסיביות הנקרא פסיביות לא-תלוית-שיווי-משקל מקסימלית (Maximal equilibrium independent passivity, MEIP), בה מניחים שהמערכת פסיבית ביחס לכל שיווי משקל שיש לה, ושאוסף כל זוגות הקלט-פלט בשיווי משקל של המערכת מהווה יחס מונוטוני מקסימלי. תכונה זו מהווה הכללה של EIP, בה ההנחה היא שאוסף הזוגות הנ"ל הוא פונקציה מונוטונית, שהיא מקרה פרטי של יחס מונוטוני מקסימלי. ברגר, זלזו ואלגוור הראו כי רשת בעלת צימוד-דיפוזיבי (diffusively-coupled network) בה הסוכנים והבקרים הם MEIP מתכנסת כאשר הזמן מתבדר לאינסוף, וגבולה ניתן לחישוב על ידי פתרון שתי בעיות אופטימיזציה על רשתות דואליות זו לזו. בעיות אלו ידועות כ"בעיית הזרימה האופטימלית" (Optimal flow problem) ו"בעיית הפוטנציאל האופטימלי" (Optimal potential problem), והן נחקרו רבות במסגרת תורת האופטימיזציה על רשתות על ידי מתמטיקאים, מדעני מחשב, ומומחים לחקר ביצועים. קשר זה יצר למעשה מסגרת לאנליזה של מערכות מרובות סוכנים באמצעות אופטימיזציה על רשתות, ויצר מילון בסיסי בין שני התחומים. עם זאת, למסגרת זו ישנם כמה חסרונות. ראשית, המסגרת מניחה שהסוכנים והבקרים הם בעלי כניסה ויציאה בודדת (input single output, SISO), מה שמגביל את השימוש בה עבור מערכות אמיתיות עם כניסות יציאות מרובות (MIMO Single Multiple input multiple output,). שנית, המסגרת דורשת שכל הסוכנים יהיו פסיביים ביחס לכל מצב יציב שיש להם, מה שמדיר מערכות כמו גנרטורים ומערכות בעלות חוסר-פסיביות (shortage passivity) אחרות. לבסוף, התוצאה של ברגר, זלזו ואלגוור עוסקת באנליזה בלבד, דהיינו רק בגבול של רשת עם סוכנים ובקר נתונים. היא אינה נותנת שיטה לסינתזה של בקרים לפתרון בעיות בקרה. המחקר המוצג בתזה זו מתמודד עם כל שלוש הבעיות הללו, ומציג שימושים לפתרון בעיות שונות בבקרה מבוזרת של מערכות מרובות סוכנים, הכוללים בקרה מבוססת-נתונים, זיהוי טופולוגיית הרשת מתוך נתונים, וזיהוי והתגברות על תקלות תקשורת בזמן אמת.

המחקר בוצע בהנחיית פרופ' דניאל זלזו בפקולטה להנדסת אווירונוטיקה וחלל

אני מודה לקרן ישראל-גרמניה למחקר ופיתוח מדעי על התמיכה הכספית הנדיבה בהשתלמותי

© Technion - Israel Institute of Technology, Elyachar Central Library

שיטות מבוססות אופטימיזציה על רשתות בבקרה מבוזרת מבוססת-פסיביות

חיבור על מחקר

לשם מילוי חלקי של הדרישות לקבלת התואר דוקטור לפילוסופיה

מיאל שרף

הוגש לסנט הטכניון - מכון טכנולוגי לישראל

תשרי תש"ף, חיפה, אוקטובר 2019

© Technion - Israel Institute of Technology, Elyachar Central Library