Decentralized Signal Processing Systems With Conservation Principles

by

Tarek Aziz Lahlou

B.S., Electrical Engineering, George Mason University (2011)
S.M., EECS, Massachusetts Institute of Technology (2013)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2016

Author . . . .
Department of Electrical Engineering and Computer Science
May 18, 2016

Certified by . . . .
Alan V. Oppenheim
Ford Professor of Engineering
Thesis Supervisor

Certified by . . . .
Thomas A. Baran
Postdoctoral Research Affiliate
Thesis Supervisor

Accepted by . . . .
Leslie A. Kolodziejski
Chair, Department Committee on Graduate Students

Decentralized Signal Processing Systems With Conservation Principles

by

Tarek Aziz Lahlou

Submitted to the Department of Electrical Engineering and Computer Science
on May 18, 2016, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

In this thesis, a framework for designing fixed-point and optimization algorithms realized as asynchronous, distributed signal processing systems is developed with an emphasis on the system's stability, robustness, and variational properties. These systems are formed by connecting basic modules together via interconnecting networks. Several classes of systems are constructed using interconnecting networks that obey certain conservation principles where these principles specifically allow steady-state system variables to be interpreted as solutions to optimization problems in a generally non-convex class and provide local conditions on the individual modules to ensure that the variables tend to such solutions, including when the communication between modules is asynchronous and uncoordinated. A particular class of signal processing systems, referred to as scattering systems, is designed that can solve convex and non-convex optimization problems, and where convex problems do not require problem-specific tuning parameters. Connections between scattering systems and their gradient-based and proximal counterparts are also established. The primary contributions of this thesis broadly serve to assist with designing and implementing scattering systems, both by leveraging existing signal processing paradigms and by developing new results in signal processing theory. To demonstrate the utility of the framework, scattering algorithms implemented as web-services and decentralized processor networks are presented and used to solve problems related to optimum filter design, sparse signal recovery, supervised learning, and non-convex regression.

Thesis Supervisor: Alan V. Oppenheim
Title: Ford Professor of Engineering

Thesis Supervisor: Thomas A. Baran
Title: Postdoctoral Research Affiliate

Acknowledgments

I would first like to thank my thesis supervisor, Prof. Al Oppenheim, for his exceptional guidance over the past several years. Throughout my time at MIT, I learned many valuable lessons from working with Al on both research and teaching, and developed a deep admiration for his thoughtful perspectives, unconventional creativity, and ability to reduce complex ideas to their core principles. Al, thank you for the outstanding mentorship, both professional and personal, and your friendship. It has been a privilege and honor to work and teach alongside you.

Next, I'd like to thank Dr. Thomas Baran for the many roles he played throughout the progression of the thesis. Tom managed simultaneously to balance the responsibilities of a research collaborator, a teaching collaborator, a close friend, and a thesis supervisor. Tom, thank you for the countless life and technical discussions as well as being a terrific sounding board at all stages of the process. I look forward to many years of future collaboration.

The thesis benefited in many ways from my interactions with Prof. Pablo Parrilo, both in his role as an outstanding lecturer in several optimization theory courses as well as his role as a thesis committee member. Pablo, thank you for patiently working with me to identify the connections with existing optimization methods and in identifying the relevant issues to be addressed in interfacing with the optimization community.

The Digital Signal Processing Group (DSPG) includes in its membership some of the brightest and most creative minds I've had the good fortune to meet and collaborate with. I am delighted to have called DSPG my home for these past few years, and to have joined such an esteemed academic family. In alphabetical order, I'd like to acknowledge the following current and past DSPG members for helpful conversations regarding my thesis: Thomas Baran, Petros Boufounos, Sefa Demirtas, Sourav Dey, Dan Dudgeon, Yonina Eldar, Meir Feder, Xue Feng, Yuantao Gu, Steve Isabelle, Zahi Karam, Jane (Hsin-Yu) Lai, Jae Lim, Shay Maymon, Catherine Medlock, Martin McCormick, Pablo Martinez Nuevo, Bruce Musicus, Milutin Pajovic, Yugandi Ranaweera, Lesley Rice, Charlie Rohrs, Andrew Russel, Maya Said, Charles Sestok, Guolong Su, Wade Torres, Laura Von Bosau, Dennis Wei, Sally Wolfe, Greg Wornell, and Qing Zhuo. In particular, I'd like to thank Pablo and Guolong for always making time to discuss various ideas and for their help in strengthening many arguments.

In addition, there have been many researchers through which discussions have impacted the thesis in meaningful ways. In alphabetical order, I'd like to acknowledge the following people: Ganesh Ajjanagadde, Stephen Boyd, Yoram Bresler, Itthi Chatnuntawech, Gerald Cook, Luca Daniel, Bob Gallagher, Joseph Hibey, Steven Johnson, Anuran Makur, Jim McClellan, Jill Nelson, Lucas Nissenbaum, Yury Polyanskiy, Gal Shulkind, Martin Vetterli, George Verghese, Dave Vogan, Jim Ward, and Lizhong Zheng. In particular, I'd like to thank Anuran for helping me to connect the dots when I was unable to untangle myself from the details of various problems, and for providing technical insights at a truly deep level. The thesis also benefited from collaboration with Texas Instruments, both in the form of internships and regular meetings. In alphabetical order, I'd like to acknowledge the following TIers: Dennis Buss, Manar El-Chammas, Gene Frantz, Baher Haroun, Fernando Mujica, Arthur Redfern, and Charles Sestok.

There are many people who made my time at MIT both memorable and enjoyable, and I believe that contributions to the thesis come in various forms and are not necessarily all technical. In this spirit, I'd like to acknowledge some people whose support and friendship have made a big impact in my life over the past five years: Danica Chang, Bassam Dourassi, Erik Flading, Phillip Mai, Ryan Marks, Farrah Sabouni, Eric Schulte, Can Temel, and Brandt Williams. In particular, I'd like to thank Erik for his constant friendship and support through the best and worst of times. In addition, I'd like to acknowledge the staff at MITx and edX for the terrific opportunities in working with online and residential education. In particular, I'd like to thank Joe Martis and Isaac Chuang for their roles in this.

The few lines of acknowledgments and thanks are entirely insufficient to describe the contributions many people have had on the development of both myself and the thesis, and this sentiment is especially true with regard to my family. To my parents, Jaye and Mohammed: thank you for instilling in me the belief that I can accomplish anything I put effort into, and for your enduring love and support. To my sister, Rita: thank you for being a grounding source, terrific role model, and dear friend through every stage of life. To my wife, Chu: your constant patience, selflessness, and generous support never ceases to amaze me. I can't overstate the importance of your contributions, especially surrounding viable hardware concerns for embedded architectures and memory management. You've made these past few years the best of my life, thank you.

To my father, Mohammed.

Contents

List of Figures

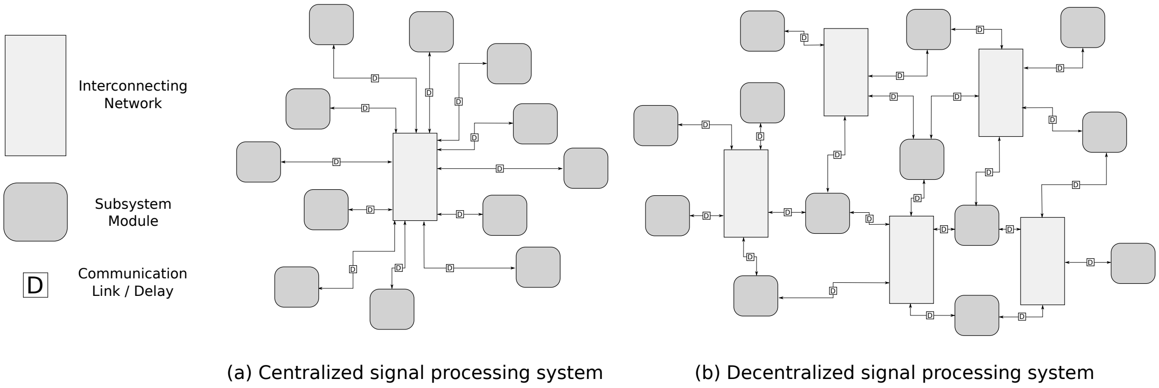
Chapter 1

Introduction

Natural phenomena closely resemble many problems appearing in signal processing and elsewhere, and in this sense, these problems have been solved already by the laws of physics, biology, etc. From the systems theory perspective, the physical laws governing natural systems often allow for decentralized and seemingly uncoordinated dynamics as the systems transition into equilibrium states satisfying extremal or variational principles. An important distinction between signal processing and natural systems lies in the ability of the former to modify or dismiss certain constraints when modeling the physical principles controlling the latter. This observation has motivated numerous philosophical viewpoints and mathematical frameworks, often leading to applications far removed from their nature-inspired origins. Examples of this are abundant: the role of biomimicry in pattern recognition and machine learning, the design of algorithms based upon sensory models of human perception, and the use of conservation principles in numerically solving differential and integral equations.

Analogies to the physical sciences, and to conservation laws in particular, have enabled researchers to determine the minimal set of features required by a class of systems so that a number of useful theorems then apply. For example, Tellegen's theorem allows for extremal properties to be easily identified in systems involving electrical, biological, chemical, and transport networks by first identifying the conserved quantities in each [4]. More broadly, physical reference systems have been used to make important connections between problems sharing fundamental structure across different fields and distinct application areas. In control theory, for example, open-loop conditions for closed-loop stability of certain feedback structures were developed by Zames in [1,5] by referencing equivalent representations of the feedback structure as an electrical network and by extending the physical notions of passivity and dissipativity to that setting. In economics, particular solutions to the Black-Scholes equation for valuating options in volatile financial markets were obtained by Edelstein and Govinder in [6] by making comparisons with common diffusion equations in materials science and by recognizing structural similarities in their conserved quantities. Despite a lack of economic or financial interpretation for these quantities, their utility in solving initial value problems remains. In the field of optimization theory, similarities between current-voltage pairs in ideal diode models and primal-dual complementarity conditions derived using Lagrangian duality principles were recognized by Dennis in [7,8] and then used to connect steady-state current and voltage distributions in a class of electrical networks with solutions to network programming problems. This approach was later extended by Chua in [9,10] to incorporate active circuit elements like operational amplifiers to address certain nonlinear optimization problems. In signal processing, conservation principles in physical and non-physical systems alike were unified within a cohesive framework developed by Baran in [11] where the basic idea is to properly organize a system's variables into a lossless connection of subsystems so that conserved quantities and their associated properties naturally emerge. This framework provides a common language where connections between classes of systems exhibiting conservation principles, including the example systems above, can be discussed.

A variety of large-scale signal processing problems can be formulated as nonlinear systems of equations and then solved using iterative techniques where any solution is considered as good as any other [12–14]. From the behavioral perspective of Willems [15,16], solutions to these problems correspond to the configurations of variables allowed by a system formed as a lossless interconnection of subsystem modules and, therefore, does not require the system variables to be explicitly labeled as inputs or outputs of the individual modules. When the behavioral model of a system is known to satisfy a variational principle, thereby providing the system variables and problem solutions interpretation as decision variables and extremal points of an associated optimization problem, the behavioral formulation provides a natural way to handle the mixture of input-output and constraint-based relationships that often form the optimality conditions. As such, the process of producing discrete-time algorithms



(a) Centralized signal processing system

(b) Decentralized signal processing system

Figure 1-1: Examples of two types of distributed signal processing systems. (a) An example of a centralized architecture. (b) An example of a decentralized architecture.

from behavioral models often involves discretizing differential or integral relationships and breaking implicit global constraints such as delay-free cycles or loops. Although the basic idea is straightforward, the behavioral viewpoint has been applied in signal processing contexts for the automated reduction of delay-free loops within linear signal-flow graphs [17] and the inversion of certain nonlinear and time-varying signal processing systems [18].

Procedures for arranging signal processing algorithms onto distributed processor networks have a rich history of blending tools from graph theory and linear algebra, including for the purpose of determining schedules and precedence relations as well as performing real-time adaptations [19–21]. A common strategy is to start with a non-distributed iteration and rearrange the computation so that it separates into concurrent subroutines that can run on independent processors that transfer data through communication or interconnecting networks [22, 23]. This strategy is limited by its ad hoc nature and requirement of a separable initial iteration. Figure 1-1 illustrates the difference between two types of distributed signal processing systems where the depicted interconnecting networks coordinate data exchanges between their adjacent subsystem modules. The centralized system in Figure 1-1(a) has a single interconnecting network that is connected to all subsystem modules while the decentralized system in Figure 1-1(b) has many interconnecting networks that independently coordinate communication between their connected subsystems.

In making efficient use of large-scale distributed signal processing systems to solve optimization and fixed-point problems, identifying conditions on the individual subsystem modules so that synchronous and asynchronous communication between the subsystems produces correct solutions becomes especially important as the storage, retrieval, and movement of data gets increasingly spread across networks prone to time-varying disruptions, congestion, and outages [24]. For example, the optimization algorithms designed by Dennis were obtained by discretizing models of electrical networks onto groups of processors where the interprocessor communication links mimicked the topology of the electrical network. The discretization of energy storage elements naturally brings forward questions of stability and robustness, and convergence arguments for Dennis' algorithms used their close relationship to the minimum heat principle and the associated conservation law, namely that the discretized current and voltage variables on each processor monotonically approach a limit point as a consequence of the sum of their products tending to zero. In numerically solving more general partial differential and integral equations, conservation principles have been used to stabilize a variety of successive approximation algorithms by perturbing the numerical solution on each iteration to conserve multiple quantities [25].

Identifying conserved quantities in physical and non-physical systems has provided a variety of insights into the dynamics and emergent properties of many algorithms spanning a broad range of applications. The examples discussed above suggest further opportunity in developing a framework in which to design algorithms realized as signal processing systems where the system's variables satisfy conservation principles and, therefore, enable the system to tackle complex problems in computing environments where the associated emergent properties have merit. The primary contributions of this thesis broadly serve to assist with developing such a framework in the context of solving fixed-point and optimization problems. To elaborate, this thesis proposes a modular strategy in which large-scale, distributed signal processing systems are formed by connecting basic modules together so that the system variables obey certain conservation principles. The conservation principles allow for interpretation of the system's steady-state variables as solutions to a broad class of convex and non-convex optimization problems and ensures the variables converge to such solutions when the modules communicate using uncoordinated and asynchronous protocols.

Toward this goal, the thesis is organized as follows: in Chapter 2, we review some background material on conservation principles and optimization theory that will be helpful in investigating properties of various organizations of a signal processing system. In Chapter 3, we develop an interconnective framework to handle large-scale systems with a focus on the separability and connectivity of a system's behavior, and we present a theorem to assist with recognizing correspondences between systems that are closely related to one another. Straightforward procedures for generating algorithms from behavioral models are developed by changing the coordinate system used to describe the model. Using the language developed in [11], two classes of signal processing systems are defined whose interconnecting networks satisfy certain conservation principles. These classes set the stage for Chapters 4 and 5 where the focus is on the stability and robustness of algorithms realized as signal processing systems in the first class and variational properties of systems in the other. In particular, systems belonging to the first class are shown in Chapter 4 to be robust to asynchronous communication protocols and can be used to solve a wide variety of fixed-point and constraint satisfaction problems. Connections between the second class of systems and optimality conditions for a broad class of convex and non-convex optimization problems are established in Chapter 5. Drawing upon the correspondences established in Chapter 3, stability properties belonging to the first class can take advantage of variational properties belonging to the second and vice versa, thereby providing direction in designing distributed and asynchronous optimization algorithms realized as signal processing systems where convergence for convex problems occurs without the use of problem-specific tuning parameters.

Although the primary focus of this thesis is the theoretical development of the contributions outlined above, the interconnective framework describes signal processing systems at an intermediary level of abstraction that can also be used to implement optimization algorithms leveraging a wide variety of computing resources. To demonstrate this, example algorithms for solving convex and non-convex optimization problems commonly appearing in signal processing applications are presented in Chapter 6 using centralized and decentralized structures. The results of several numerical experiments demonstrating the convergence of these algorithms implemented as web services and decentralized processor networks are also included.

Chapter 2

Background and conventions

The primary purpose of this chapter is to present background material on conservation principles and optimization theory using well-established concepts in the signal processing community and elsewhere. In particular, methods for handling implicit functions and relations are reviewed and then used to summarize existing behavioral and interconnective system models. With these tools in place, a framework is discussed wherein conservation principles can be identified and manipulated within signal processing systems. The chapter concludes with a brief overview of some key concepts in optimization theory, focusing specifically on aspects pertinent to signal processing applications. The intent of the presentation on optimization is not to provide a comprehensive literature review or list of references, but instead to indicate several interpretations of independent interest while establishing conventions and preliminaries to build upon in the subsequent chapters.

2.1 | Conventions

The non-negative and positive integers are respectively denoted by $\mathbb{N}_0 = \{0, 1, \dots\}$ and $\mathbb{N} = \{1, 2, \dots\}$. Vectors are written in boldface with individual entries denoted by subscripts, e.g. $\mathbf{v}_k$ is the k-th entry of $\mathbf{v}$. For vectors with parenthetical superscripts, subscripts refer to subvectors according to the partitioning scheme indicated by the superscript. Sequences of vectors are indicated using superscripts sans parenthesis, e.g. $\{\mathbf{v}^n \colon n \in \mathbb{N}_0\} = \{\mathbf{v}^0, \mathbf{v}^1, \dots\}$. The expectation and probability operators are respectively denoted by $\mathbb{E}$ and $\mathbb{P}$.

2.2 | Behaviors, relations, and graphs

Conventional methods of designing signal processing systems involve connecting basic modules together to form directed graphs, thereby establishing an overall system. For systems represented in this way, a separate and important issue pertains to whether computable algorithms exist that are consistent with the collective constraints imposed by the individual modules and the dependencies implied by the graph topology. As has been discussed in [26–28], functional relationships mapping overall system inputs to outputs are not generally guaranteed to exist, even if each module in the graph has a well-defined input-output form. Furthermore, when an overall mapping does exist, the form of the algorithm may not be directly evident from the assembled graph due to implicit global constraints such as delay-free cycles or loops [21]. These issues often require special consideration in determining the precedence relations used to schedule an execution order of the modules, and common remedies for some special cases, including iterative constraint satisfaction procedures [29,30] and nonlinear module transforms [31], attempt to break certain delay-free loops by invoking implicit mapping principles to produce valid input-output pairs.

In the context of solving a variety of large-scale signal processing problems, a fundamental challenge involves designing *distributed algorithms* where the processing instructions can be separated into concurrent subroutines and run on independent processors that communicate with only a small number of other processors. The design of these algorithms is often informed by the interplay between properties of the available computing resources and structure belonging to the problem itself [24]. For example, the kinds of communication protocols supported by the computing resources and the achievable time synchronization between connected processors often dictates the choice between *synchronous* and *asynchronous* protocols. Synchronous algorithms require the computing resources to progress as a group, and this level of coordination can hinder efficiency as the number of available processors becomes large [32]. Asynchronous algorithms, on the other hand, allow each processor to progress independently, thereby overcoming some of the drawbacks faced by synchronous algorithms in the presence of processor heterogeneity and interprocessor communication failure. Key research questions for enabling the widespread use of asynchronous, distributed algorithms for a class of problems include understanding and identifying sufficient conditions under which the results produced by asynchronous algorithms are correct. Important concerns also include the robustness and resilience of the asynchronous algorithms to faulty computing resources and various communication disturbances.

Motivated in part by the potential benefits afforded by the design of asynchronous, distributed algorithms realized as signal processing systems for certain problem classes and by the issues surrounding computability and precedence in cyclic graphs of computable modules, we proceed in this section to review several tools for capturing the action of a system or subsystem without solely relying on functional techniques. More broadly, the viewpoint in this thesis is that functions, which are the well-behaved subset of more general relations as can be identified using the so-called vertical line test, are not the critical issue in the design, analysis, and implementation of signal processing systems for a broad and important number of contexts and applications. The language we adopt to discuss these tools blends together terms developed by Willems in [15] and Polderman and Willems in [16], wherein their approach was to view equations as exclusion principles permitting only admissible configurations of the associated variables, and Zames in [1,5], wherein his approach was to focus on correspondences between constraints. These approaches have a long and rich history within the dynamical systems and control community, specifically in establishing fundamental results such as the small gain theorem for feedback control systems and without explicitly relying on Lyapunov or storage functions. These viewpoints guide the presentation style and interpretations used to develop stability conditions for distributed signal processing systems using uncoordinated and asynchronous communication protocols in Chapter 4.

Within a mathematical framework, if $T\colon \mathcal{U} \to \mathcal{V}$ denotes a function whose domain $\mathcal{U}$ and codomain $\mathcal{V}$ are vector spaces then the graph of T refers to the set of all ordered pairs $(\mathbf{u}, T(\mathbf{u}))$[1] consistent with T. In the field of signal processing, the term "graph" refers to different mathematical objects depending on context. To disambiguate, we henceforth use the term *behavior*, originally coined by Willems [15], in reference to functions and reserve

[1] In this thesis, the notation $(\mathbf{u}, \mathbf{v})$ is used as an in-line shorthand for the standard vector notation $\begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix}$.

the term *graph* to mean a collection of nodes and edges. Proceeding with this nomenclature, when T is bijective the behavior of the inverse function $T^{-1}\colon \mathcal{V} \to \mathcal{U}$ is related to the behavior of T in a straightforward way, i.e. the behavior of T^{-1} is the set of all ordered pairs $(T^{-1}(\mathbf{v}), \mathbf{v})$ consistent with T^{-1}. More formally, the behavior $\mathcal{B}_T$ of an invertible function T and the behavior $\mathcal{B}_{T^{-1}}$ of the inverse function T^{-1} are related according to

$$\begin{aligned}
\mathcal{B}_T &= \{(\mathbf{u},\mathbf{v}) \in \mathcal{U} \times \mathcal{V} \colon \mathbf{v} = T(\mathbf{u}),\, \mathbf{u} \in \mathrm{dom}T\} && (2.1)\\
&= \{(\mathbf{u},\mathbf{v}) \in \mathcal{U} \times \mathcal{V} \colon \mathbf{u} = T^{-1}(\mathbf{v}),\, \mathbf{v} \in \mathrm{dom}T^{-1}\} && (2.2)\\
&= \mathcal{B}_{T^{-1}} && (2.3)
\end{aligned}$$

where dom$T$ denotes the domain of T. Our primary use of behaviors moving forward is to model constraints, and to do so without distinguishing between input and output variables.

For prespecified behaviors $\mathcal{B}$, a *functional realization*, or *realization* for short, refers to any function T satisfying $\mathcal{B}_T = \mathcal{B}$. Functional realizations are generally not unique, and therefore do not uniquely determine how a particular input $\mathbf{u}$ generates $T(\mathbf{u})$, i.e. exhaustive input-output knowledge does not uniquely specify internal dynamics or relationships for many signal processing systems represented via operators. This point suggests evaluating notions of internal and external stability, complexity, and computability when selecting among candidate realizations, and is discussed further in Chapter 3. In this thesis, when the realization of a behavior exists for a specified configuration of the variables $\mathbf{u}$ and $\mathbf{v}$ as inputs and outputs, it is assumed to be computable, e.g. analytically, algorithmically, or through table lookup.

An important class of behaviors includes those generated by linear operators. With $A\colon \mathcal{U} \to \mathcal{V}$ denoting a linear map, the behavior $\mathcal{B}_A$ in (2.1) reduces to

$$\begin{aligned}
\mathcal{B}_A &= \{(\mathbf{u},\mathbf{v}) \in \mathcal{U} \times \mathcal{V} \colon A\mathbf{u} = \mathbf{v},\, \mathbf{u} \in \mathrm{dom}A\} && (2.4)\\
&= \mathrm{range}\left(\begin{bmatrix} I \\ A \end{bmatrix}\right) && (2.5)
\end{aligned}$$

where (2.5) holds if dom$A = \mathcal{U}$. The behavior $\mathcal{B}_A$ is itself a vector subspace of the parent

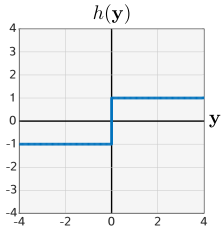
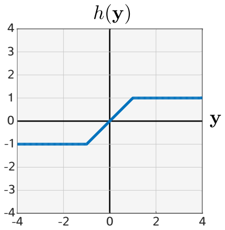
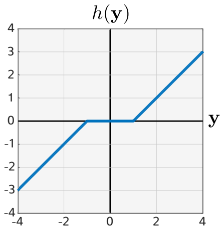
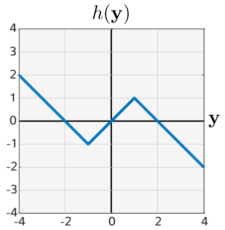
Linear map M	$M = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$M = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$	$M = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$	$M = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$
Transformed behavior				
Functional realization	n/a	$h(\mathbf{y}) = \begin{cases} \mathbf{y}, & \lvert\mathbf{y}\rvert \le 1 \\ \mathrm{sgn}(\mathbf{y}), & \lvert\mathbf{y}\rvert > 1 \end{cases}$	$h(\mathbf{y}) = \begin{cases} 0, & \lvert\mathbf{y}\rvert \le 1 \\ y - \mathrm{sgn}(\mathbf{y}), & \lvert\mathbf{y}\rvert > 1 \end{cases}$	$h(\mathbf{y}) = \begin{cases} \mathbf{y}, & \lvert\mathbf{y}\rvert \le 1 \\ 2\mathrm{sgn}(\mathbf{y}) - \mathbf{y}, & \lvert\mathbf{y}\rvert > 1 \end{cases}$

Figure 2-1: Four relations whose behaviors are related through invertible linear transformations. The first column corresponds to the original relation in (2.7) and the following three columns depict the functions whose behaviors are related to the first through the linear transformations listed.

space $\mathcal{U} \times \mathcal{V}$ with dimensionality equal to the dimension of $\mathcal{U}$. Consequently, behaviors of linear transformations can be completely characterized using standard vector space descriptions, e.g. as the null space or range space of an appropriately defined linear map.

A *relation* generalizes the notion of a function to allow for mapping objects T where the outcome $T(\mathbf{u})$ is multi-valued for at least one $\mathbf{u}$. Notationally, relations are described using the same behavioral definition as functions with the caveat that $\mathbf{v} = T(\mathbf{u})$ in (2.1) is interpreted to mean $\{\mathbf{v}\colon \mathbf{v} = T(\mathbf{u})\}$. Procedurally, for functions and relations, behaviors can be generated by collecting together the set of all input-output pairs that result from applying the mapping object to all elements in its domain and concatenating these pairs into vectors. The usual operations on functions extend to relations in a straightforward way, e.g. addition, scalar multiplication, composition, etc. In the coming chapters, we are interested in producing functional realizations of relations. The implicit function theorem is a standard tool in functional analysis that generates local (over open disks) realizations of a relation, provided the relation is appropriately differentiable over the local regions [33, Theorem 9.17]. In this thesis, however, the processing goal often involves solving fixed-point problems where the relations of interest are not differentiable at their fixed points. To circumvent this potential difficulty, we will instead focus on realizing the behavior of a relation after first transforming it by an invertible linear map. We motivate this approach with an example next.

An important principle that emerges as a consequence of the behavioral viewpoint taken in this thesis is that the relationship between two functions whose behaviors are related through an invertible linear transformation is often nonlinear from the perspective of the functions. Generalizing to include relations, let M and T respectively denote an invertible linear map and a generic relation. The transformation of $\mathcal{B}_T$ by M is then given by

$$M\mathcal{B}_T \quad = \quad \{M\mathbf{v} \colon \mathbf{v} \in \mathcal{B}_T\} \tag{2.6}$$

where the dimensions of M and T are such that (2.6) is well-defined. If $T = A$ is a linear map then any realization of $M\mathcal{B}_A$, if one exists, is also a linear map due to the closure of linear transformations under composition [34]. Our reasons for restricting attention to invertible transforms are many and will become clear in the context of system representation in Chapter 3 and asynchronous algorithms in Chapter 4. Another reason, illustrated next using a simple example, involves solving for a fixed-point of a relation using a realization of its transformed behavior. With $f \colon \mathbb{R} \to \mathbb{R}$ denoting the relation or multi-valued function corresponding to the subderivative of the absolute value function described by

$$\{f(\mathbf{x})\} \quad = \quad \begin{cases} \operatorname{sgn}(\mathbf{x}), & \mathbf{x} \neq 0 \\ [-1, 1], & \mathbf{x} = 0 \end{cases}, \tag{2.7}$$

the behavior $\mathcal{B}_f$ is a proper subset of $\mathbb{R}^2$ with fixed-points $(0, 0)$, $(1, 1)$, and $(-1, -1)$. Note that the implicit mapping theorem cannot be used to realize $\mathcal{B}_f$ over a region including all three fixed-points using either $\mathbf{x}$ or $f(\mathbf{x})$ as the independent variable. Figure 2-1 depicts the behaviors $\mathcal{B}_h = M\mathcal{B}_f$ for four invertible matrices M. Referring to the figure, the first column depicts the original relation (2.7) and the following three columns of the figure depict functional realizations of $\mathcal{B}_h$ associated with the matrices listed. Let $\mathbf{y}^\star$ denote a fixed-point of h, i.e. $h(\mathbf{y}^\star) = \mathbf{y}^\star$. A strategy for identifying fixed-points of f is to first solve for $\mathbf{y}^\star$ using functional techniques and then perform the transformation $M^{-1}(\mathbf{y}^\star, \mathbf{y}^\star)$. As we alluded to earlier, when this approach works it relies critically on M being invertible. For this particular example, fixed-points of f are not guaranteed to map to fixed-points of h and vice versa, so identifying a fixed-point of f by transforming points in the graph of h may require additional structure on M. This structure is developed for solving optimality conditions describing solutions to convex and non-convex optimization problems in Chapter 5.

2.3 | Interconnective and signal-flow representations

The representation of an overall system using directed graphs to couple together subsystem modules has a long history throughout branches of engineering. Within the signal processing community, many classical tools for synthesizing systems and analyzing their global properties are consistent with this approach. In this section, we contrast the widely-used signal-flow graph representation with the interconnective descriptions appearing, e.g., in [11,15,17]. For a complete introduction of signal-flow graphs, we refer readers to [35].

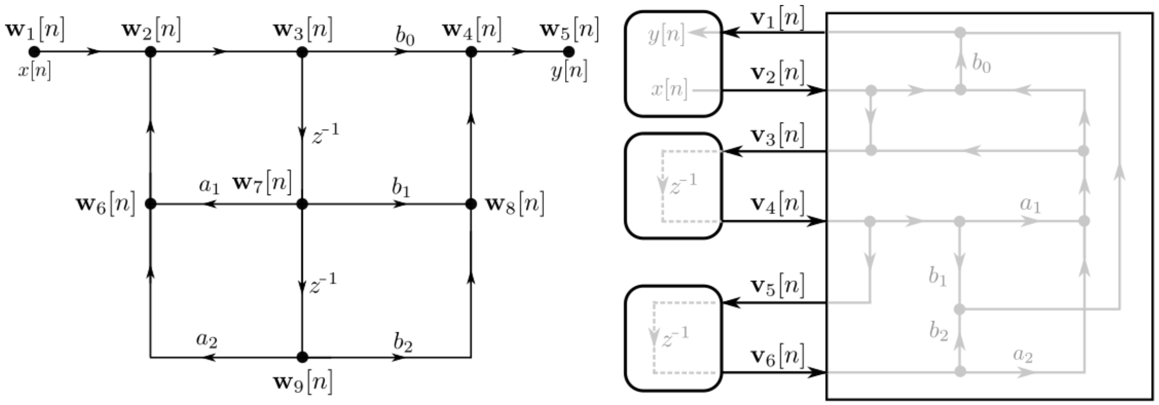
An important distinction emphasized in this thesis between signal-flow and interconnective graphs is the interchanged roles played by the graph's edges and nodes with respect to the signal and processing objects themselves. More specifically, for linear and nonlinear signal-flow graphs, the signal variables are defined as nodes and the processing instructions are assigned to edges as branch functions, whereas for interconnective graphs, the signal variables are defined as edges and the processing instructions are assigned to nodes. The duality between these representations, and importantly the perspective of the latter, forms the basis for what we refer to in the remainder of this thesis as the *interconnective viewpoint*; a generalization of which is the subject of Chapter 3. Although the basic distinction is straightforward, we will find the interconnective viewpoint beneficial in several contexts. This basic distinction between signal-flow and interconnective representations is illustrated next with the aid of an example, and is discussed in greater detail in Section 3.3.

As an example, the direct-form representation of a biquadratic filter or second-order section using signal-flow and interconnective graphical structures is depicted in Figure 2-2 for comparison. Referring to the signal-flow graph in Figure 2-2(a), the node variables correspond to the labeled signals $\mathbf{w}_i[n]$, for $i = 1, \dots, 9$, and the branches or directed edges indicate both pairwise dependencies between nodes as well as the their explicit functional relationships. The instantaneous value of a node variable, i.e. for a particular value of n, is computed by summing the values generated by applying the branch functions to the nodes incident neighbors, and nodes with no incident neighbors such as $\mathbf{w}_1[n]$ are designed overall system inputs. Procedurally, signal-flow graphs can be constructed from systems of equations by starting with a fully connected graph with each variable assigned a node, writing all pairwise branch functions, and then removing all branches corresponding to everywhere-zero branch functions.

In contrast to signal-flow graphs, interconnective graphical structures distinguish between two types of nodes. The first type of node is referred to as an interconnecting node and corresponds to a system's memoryless and linear relationships, and the second type is referred to as a constitutive subsystem or module and corresponds to generally nonlinear constraints that may contain memory and overall system inputs and outputs. Referring to the interconnective graph in Figure 2-2(b), the edges correspond to the labeled signals $\mathbf{v}_i[n]$, for $i = 1, \ldots, 6$, and the nodes indicate the computation which defines the signal values and are depicted using boxes. Observe that the interconnecting node is internally specified using a memoryless signal-flow representation and can be equivalently expressed as a matrix equation. More generally, the description of each processing node in an interconnective graph can be arbitrary since the interconnective graph of the overall system does not explicitly focus on the values of the variables internal to the nodes or the subcomputations performed within them. As demonstrated by this example, interconnective structures in this thesis will always be bipartite, meaning variable edges will only be defined only between processing nodes of opposite type. To reiterate, the key point of using an interconnective representation is to describe a behavioral model of a system as the coupling together of generally many modules using a lossless interconnecting system where the constraints imposed by each module can be described using either functions or relations.

For general interconnective descriptions of signal processing systems, we shall refer to the scalar-valued variables available for interconnection between the interconnecting nodes and constitutive modules as the *terminal variables* and collectively arrange them into a *terminal vector*. The behavior of a signal processing system, defined via the interconnective viewpoint, is then the set of all terminal vectors consistent with the constraints imposed by the overall system. Written more formally, the behavior of a system s with N terminal variables $\mathbf{v} = (\mathbf{v}_1, \ldots, \mathbf{v}_N)$ is the set $\mathcal{B}_s$ described colloquially by

$$\mathcal{B}_s \quad = \quad \{\mathbf{v} : \mathbf{v} \text{ complies with all constraints imposed by } s\}\,. \tag{2.8}$$



(a) Signal-flow graph

(b) Interconnective graph

Figure 2-2: An illustration used to underscore the fundamental quantities of interest in signal-flow and interconnective representations. In particular, a direct form biquadratic filter is portrayed using (a) a signal-flow graph and (b) an interconnective graph.

The interconnection of two systems is straightforward to characterize from the behavioral viewpoint. In particular, as the interconnection of two systems refers to a direct coupling of the terminal variables shared between the systems, the behavior of the interconnected system corresponds to the intersection of the behaviors of the individual systems over those variables that are shared. As a straightforward example, let s_1 and s_2 denote two systems that when interconnected share all of their terminal variables. The behavior of the interconnected system s_i is then

$$\begin{aligned} \mathcal{B}_{s_i} &= \mathcal{B}_{s_1} \cap \mathcal{B}_{s_2} && (2.9) \\ &= \{\mathbf{v} \colon \mathbf{v} \in \mathcal{B}_{s_1},\ \mathbf{v} \in \mathcal{B}_{s_2}\}. && (2.10) \end{aligned}$$

By induction, this definition extends to the interconnection of three or more systems in a straightforward way.

When both of the signal processing systems s_1 and s_2 correspond to linear and memoryless systems, respectively characterized by linear maps A_1 and A_2, then the behavior of the interconnected system $\mathcal{B}_{s_i} = \mathcal{B}_{A_1} \cap \mathcal{B}_{A_2}$ is a linear subspace. This point follows from the fact that the behavior of a linear map over a vector space is a vector space and that the intersection of vector spaces results in a vector space.

Utilizing the behavioral perspective of interconnective systems, many methods developed in this thesis do not require the terminal variables to be explicitly labeled as either inputs or outputs between the processing nodes. In this way, the interconnective viewpoint is a natural choice for handling signal processing systems with implicit or set-valued constraints, as may arise in signal-flow graphs with delay-free loops or implicitly defined modules.

2.4 | Conservation principles in signal processing systems

Conservation principles give rise to convenient properties in both physical and non-physical systems and are often used to analyze or emulate the steady-state and transient dynamics of a system. A characterization of conservation principles for signal processing systems was developed within a cohesive framework in [11] where the basic idea is to organize a system's variables in certain ways so that desirable properties emerge as a consequence; in particular, in response to the organization enforced by a system's memoryless and linear relationships. Using this framework, stability properties of a particular vehicle density control architecture were established in [36] by identifying a conservation principle inherent to the architecture itself. The class of results in [7, 11, 36] suggest further opportunity in utilizing this framework to establish robustness, stability, and variational properties of large-scale, distributed signal processing systems by appropriately designing their interconnecting networks. In this section, we review the core elements of this framework as they pertain to the class of signal processing systems considered in this thesis. In Chapter 3, we return to this framework to make connections between behavioral and interconnective system models sharing intrinsic similarities that may otherwise be difficult to connect together.

2.4.1 | Organized variable spaces

The main goal in this section is to establish the terminology necessary to describe conservation principles as quadratic forms q satisfying $q(\mathbf{v}) = 0$ for all $\mathbf{v}$ in a so-called conservative set, and to use this language to define a conservative signal processing system. In building to the formal definition of conservation in signal processing systems, we first introduce the concept of an *organization* for an even-dimensional, real inner product space $(\mathcal{V}, \langle\cdot,\cdot,\rangle_{\mathcal{V}})$.

The idea behind an organization is straightforward and involves decomposing $\mathcal{V}$ in two ways so that the action of a quadratic form on $\mathcal{V}$ belonging to a particular class can be expressed in agreement with both decompositions. To this end, let $q\colon \mathcal{V} \to \mathbb{R}$ denote a real quadratic form written using the inner product on $\mathcal{V}$ as

$$q(\mathbf{v}) \quad = \quad \langle C\mathbf{v}, \mathbf{v}\rangle_{\mathcal{V}}\,, \qquad\qquad \mathbf{v} \in \mathcal{V} \tag{2.11}$$

where $C\colon \mathcal{V} \to \mathcal{V}$ is a symmetric linear map referred to as a *correspondence map* and is required to be invertible and possess a spectral decomposition[2] with an equal number of positive and negative eigenvalues. A correspondence map satisfying this spectral property is referred to as being *balanced.* The quadratic form q, or equivalently the correspondence map C, serves as the primary tool by which subsets of $\mathcal{V}$ may be identified as being conservative. In the broader scope of linear algebra, a correspondence map paired with an inner product space $(\mathcal{V}, \langle\cdot,\cdot,\rangle_{\mathcal{V}})$ is referred to as an indefinite inner product space and is primarily associated with the study of matrix polynomials, Ricatti equations, and symmetric differential and difference equations [37].

The first decomposition used to define an organization of $(\mathcal{V}, \langle\cdot,\cdot,\rangle_{\mathcal{V}})$ is called a *partition decomposition* and consists of K even-dimensional, linear subspaces $\{\mathcal{V}_k\colon \mathcal{V}_k \subseteq \mathcal{V}, 1 \le k \le K\}$ that uniquely decompose $\mathcal{V}$ according to the direct sum

$$\mathcal{V} \quad = \quad \mathcal{V}_1 \oplus \cdots \oplus \mathcal{V}_K \tag{2.12}$$

and linearly separate the quadratic form q into the sum of K terms as

$$q\left(\mathbf{v}^{(1)} + \cdots + \mathbf{v}^{(K)}\right) \quad = \quad \sum_{k=1}^{K} \left\langle C_k \mathbf{v}^{(k)}, \mathbf{v}^{(k)} \right\rangle_{\mathcal{V}}, \qquad \mathbf{v}^{(k)} \in \mathcal{V}_k, \quad k = 1, \ldots, K \tag{2.13}$$

where each $C_k\colon \mathcal{V}_k \to \mathcal{V}_k$ is a correspondence map for $\mathcal{V}_k$, i.e. is linear, invertible on $\mathcal{V}_k$, symmetric, and balanced. In Chapter 3, we generalize the definition of a partition decomposition presented here for the purpose of system representation and do so without reference

[2] Since C is symmetric without loss of generality, the eigenvalue decomposition, singular value decomposition, and Schur decomposition are equal to within sign changes of the associated decomposition vectors.

to quadratic forms. The link between these settings and the role played by the quadratic form in establishing this link is discussed in Section 3.4.

The second decomposition used to define an organization of $(\mathcal{V}, \langle\cdot,\cdot,\rangle_{\mathcal{V}})$ is called a *conjugate decomposition* and consists of two linear subspaces $\mathcal{V}_A$ and $\mathcal{V}_B$ that uniquely decompose $\mathcal{V}$ according to

$$\mathcal{V} = \mathcal{V}_A \oplus \mathcal{V}_B \tag{2.14}$$

and further allow the quadratic form q to be expressed through an inner product on a lower dimensional inner product space $(\mathcal{U}, \langle\cdot,\cdot\rangle_{\mathcal{U}})$. Specifically, a conjugate decomposition requires the existence of an inner product space $(\mathcal{U}, \langle\cdot,\cdot\rangle_{\mathcal{U}})$ as well as two linear, invertible maps $M_A \colon \mathcal{V}_A \to \mathcal{U}$ and $M_B \colon \mathcal{V}_B \to \mathcal{U}$ such that

$$q(\mathbf{v}^{(A)} + \mathbf{v}^{(B)}) \ = \left\langle M_A\mathbf{v}^{(A)}, M_B\mathbf{v}^{(B)}\right\rangle_{\mathcal{U}}, \qquad \mathbf{v}^{(A)} \in \mathcal{V}_A, \mathbf{v}^{(B)} \in \mathcal{V}_B. \tag{2.15}$$

Note that the invertibility required of M_A and M_B is between $\mathcal{U}$ and the conjugate subspaces $\mathcal{V}_A$ and $\mathcal{V}_B$, not between $\mathcal{U}$ and $\mathcal{V}$ directly. When these objects are well-defined, the inner product space $(\mathcal{U}, \langle\cdot,\cdot\rangle_{\mathcal{U}})$ is called a *comparison space* and the linear maps M_A and M_B are called *conjugate maps*. The role played by the vectors $M_A\mathbf{v}^{(A)}$ and $M_B\mathbf{v}^{(B)}$ is reminiscent of conjugate effort and flow variables that are associated with describing classical notions of power conservation in physical systems.

With formal conditions for partition and conjugate decompositions in place, we are now prepared to define an organization. Indeed, an *organization* $\mathcal{O}$ of the inner product space $(\mathcal{V}, \langle\cdot,\cdot,\rangle_{\mathcal{V}})$ is defined as a triple

$$\mathcal{O} \ \triangleq \ (C, \mathcal{D}_p, \mathcal{D}_c) \tag{2.16}$$

consisting of a correspondence map C, a partition decomposition $\mathcal{D}_p = \{\mathcal{V}_k, 1 \le k \le K\}$, and a conjugate decomposition $\mathcal{D}_c = \{\mathcal{V}_A, \mathcal{V}_B\}$ where the partition decomposition subspaces are uniquely decomposed by the conjugate decomposition subspaces according to

$$\mathcal{V}_k \;=\; (\mathcal{V}_k \cap \mathcal{V}_A) \oplus (\mathcal{V}_k \cap \mathcal{V}_B)\,, \qquad k = 1, \ldots, K \tag{2.17}$$

and the conjugate decomposition subspaces are uniquely decomposed by the partition decomposition subspaces according to

$$\mathcal{V}_A \;=\; \bigoplus_{k=1}^{K} (\mathcal{V}_k \cap \mathcal{V}_A) \tag{2.18}$$

$$\mathcal{V}_B \;=\; \bigoplus_{k=1}^{K} (\mathcal{V}_k \cap \mathcal{V}_B)\,. \tag{2.19}$$

Putting this all together, the collection of $(\mathcal{V}, \langle \cdot, \cdot, \rangle_{\mathcal{V}})$ and an organization $\mathcal{O}$ is referred to as an *organized variable space*. For a particular organized variable space, we shall say that a set $\mathcal{S} \subseteq \mathcal{V}$ is *conservative* or satisfies conservation principle q provided that

$$q(\mathbf{v}) \;=\; 0, \qquad\qquad \mathbf{v} \in \mathcal{S}. \tag{2.20}$$

In the language of quadratic forms, a vector space $\mathcal{S}$ satisfying (2.20) is referred to as an isotropic subspace. Similarly, in the language of indefinite inner product spaces, a set $\mathcal{S}$ satisfying (2.20) is referred to as a neutral set. Concerning signal processing, the definition of a conservation principle in this section is consistent with [11] in that it does not explicitly pertain to a signal processing system, but rather to arbitrary collections of vectors allowing application in more general settings.

The interconnective system representation introduced in Section 2.3 provides a reasonable definition of a conservative system through (2.20) where the set $\mathcal{S}$ corresponds to the system's behavior, provided $\mathcal{S}$ is appropriately contained within an even-dimensional, real vector space. For reasons of tractability, we proceed entirely focused on conservative systems where $\mathcal{S}$ is a subspace describing the behavior of a system's linear and memoryless constraints, previously referred to as the interconnecting network. From the discussion surrounding (2.9), the interconnection of these constraints with constitutive modules only further restricts the behavior of the overall system, thus, any system whose linear interconnection behavior is conservative remains conservative independent of the constitutive modules coupled to it. Indeed, this restriction allows for a broad class of systems to be identified as conservative while relying primarily on linear and quadratic analysis techniques.

2.4.2 | Linear transformations of conservative sets

As was mentioned previously, we are primarily interested in realizing signal processing systems after linearly transforming their behaviors. In this subsection, we discuss the relationship between an organization for which a set is conservative and an organization for which the transformed set is conservative. To state this relationship, we rely on straightforward variable substitutions and properties of correspondence maps. Specifically, let M denote an invertible linear map and let $\mathcal{S}$ be a conservative set in $(\mathcal{V}, \langle \cdot, \cdot \rangle_{\mathcal{V}})$ with respect to the organization $\mathcal{O} = (C, \mathcal{D}_p, \mathcal{D}_c)$. The set $M\mathcal{S}$, contained in the same inner product space, is then conservative with respect to the organization $\mathcal{O}' = (C', \mathcal{D}_p', \mathcal{D}_c')$ where

$$\begin{aligned} C' &= M^T C M && (2.21)\\ \mathcal{D}_p' &= M^{-1}\mathcal{D}_p && (2.22)\\ \mathcal{D}_c' &= M^{-1}\mathcal{D}_c. && (2.23) \end{aligned}$$

The identity used to establish (2.21)-(2.23) as well as verify that the unprimed organization for S and the primed organization for $M\mathcal{S}$ do indeed satisfy the requirements of an organization is

$$\langle C\mathbf{v}, \mathbf{v}\rangle_{\mathcal{V}} = 0, \quad \mathbf{v} \in M\mathcal{S}, \quad \Longleftrightarrow \quad \langle CM\mathbf{v}, M\mathbf{v}\rangle_{\mathcal{V}} = 0, \quad \mathbf{v} \in \mathcal{S}. \qquad (2.24)$$

The validity of C' as a bona fide correspondence map follows from several observations. First, the composition of invertible linear maps results in an invertible linear map, and the form of C' implies it preserves the symmetry of C. It remains to be seen that C' is balanced, i.e. has equal numbers of positive and negative eigenvalues. This fact follows by direct application of Sylvester's law of inertia [37], which states that the number of positive and negative eigenvalues of any symmetric matrix A is invariant to transformations of the form $B^T AB$ for any invertible matrix B. The conjugate maps M'_A and M'_B associated with the primed organization are obtained from the conjugate maps M_A and M_B associated with the unprimed organization according to $M'_A = M_A M$ and $M'_B = M_B M$.

The discussion to this point formally justifies an isomorphism between correspondence maps related by (2.21) where M is an invertible linear map and where the isomorphism is specifically on the number of positive and negative eigenvalues. In the remainder of this subsection, we draw attention to some special linear maps and their effect on the organization $\mathcal{O}'$ according to (2.21) through (2.23). Figure 2-3 highlights three particular classes of linear maps using a Venn diagram and contrasts their effect on the organization. The first class, indicated in the figure as quadratic form preserving transformations, is composed of those linear transformations M for which

$$\langle M^T C M \mathbf{v}, \mathbf{v}\rangle_{\mathcal{V}} \quad = \quad \langle C\mathbf{v}, \mathbf{v}\rangle_{\mathcal{V}}, \qquad \mathbf{v} \in \mathcal{V}. \tag{2.25}$$

The collection of linear maps which satisfy (2.25) is the indefinite or split orthogonal group $O(N/2, N/2)$ [11, Theorem 3.1]. The second class of transformations, indicated in the figure as partition decomposition preserving transformations, is composed of all linear maps that the partition decomposition is invariant to, i.e. for which $M\mathcal{D}_p = \mathcal{D}_p$. These transformations correspond to block diagonal matrices whose block sizes equal the dimensions of the partition decomposition subspaces $\mathcal{V}_k$ for $k = 1, \ldots, K$. To describe the third group of transformations, let $\mathcal{S}$ denote the behavior of a linear map A as demonstrated by (2.4). An interesting subset of transformations, indicated in the figure as interconnection invariant transformations, are those linear maps M for which the vector subspace $M\mathcal{S}$ is realized by the same linear map A. Interconnection invariant transformations are described in more detail in the context of interconnective system representation in Section 3.5.2 where a procedure for numerically generating them is also presented.

2.4.3 | Conservative vector spaces

Conservation principles are closely related to orthogonality principles, so a natural question pertains to the distinction between pairwise and subspace orthogonality. To address this,

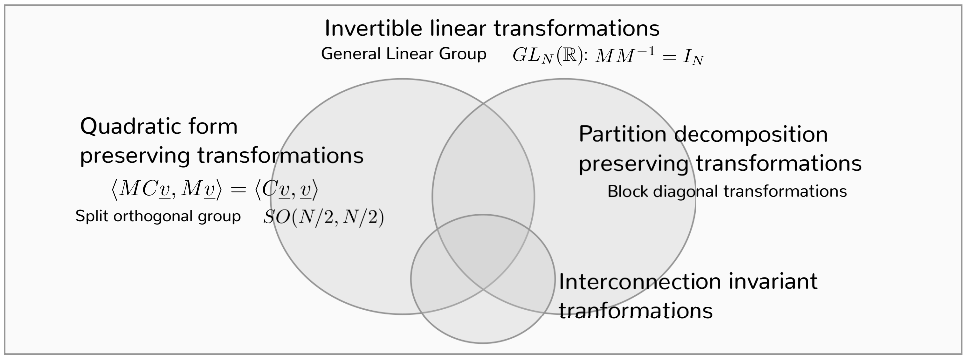

Invertible linear transformations
General Linear Group $GL_N(\mathbb{R})$: $MM^{-1} = I_N$
Quadratic form preserving transformations
$\langle MC\underline{v}, M\underline{v}\rangle = \langle C\underline{v}, \underline{v}\rangle$
Split orthogonal group $SO(N/2, N/2)$
Partition decomposition preserving transformations
Block diagonal transformations
Interconnection invariant tranformations

Figure 2-3: A Venn diagram illustrating several categories of linear transformations. Those in the regions labeled "quadratic form preserving transformations" and "partition decomposition preserving transformations" are used in the discussion on the isomorphism between conservation principles.

we discuss in this subsection two types of organizations for which a subspace $\mathcal{S} = S \subseteq \mathcal{V}$ may be conservative, and we do so by focusing on properties the organizations exhibit when viewed through the comparison space. In particular, the conservation principle expressed in the comparison space in (2.15) may be due to pairwise orthogonality between the particular elements $M_A\mathbf{v}^{(A)}$ and $M_B\mathbf{v}^{(B)}$ for each element $\mathbf{v} = \mathbf{v}^{(A)} + \mathbf{v}^{(B)} \in S$, or may be due to a stronger condition that requires the set of all possible elements $M_A\mathbf{v}^{(A)}$ to form a subspace that is orthogonal to the set of all possible elements $M_B\mathbf{v}^{(B)}$.

To formalize this distinction, we first relate the inner-product space $\mathcal{V}$ to the conjugate subspaces $\mathcal{V}_A$ and $\mathcal{V}_B$ using two oblique projection operators $P_A \colon \mathcal{V} \to \mathcal{V}$ and $P_B \colon \mathcal{V} \to \mathcal{V}$ satisfying range$(P_A) = \mathcal{V}_A$ and range$(P_B) = \mathcal{V}_B$. In addition, we require the projectors P_A and P_B to uniquely decompose elements of $\mathcal{V}$, i.e. $P_A + P_B = I$. Using these operators, we define two vector spaces S_A and S_B as vector subspaces of $\mathcal{V}$ given by

$$S_A \quad = \quad \{\mathbf{v}^{(A)} \colon \mathbf{v}^{(A)} = P_A\mathbf{v}, \mathbf{v} \in S\} \tag{2.26}$$

and

$$S_B \quad = \quad \{\mathbf{v}^{(B)} \colon \mathbf{v}^{(B)} = P_B\mathbf{v}, \mathbf{v} \in S\}. \tag{2.27}$$

An organized variable space is referred to as being *strongly conservative* provided that the vector space $M_A(S_A)$ is orthogonal to the vector space $M_B(S_B)$ under the comparison spaces' inner product, i.e. satisfies $M_A(S_A) \perp_{\mathcal{U}} M_B(S_B)$ which is shorthand for the formal condition

$$\langle M_A \mathbf{v}^{(A)}, M_B \mathbf{v}^{(B)} \rangle_{\mathcal{U}} \quad = \quad 0, \qquad \mathbf{v}^{(A)} \in S_A, \mathbf{v}^{(B)} \in \mathcal{S}_B. \tag{2.28}$$

In the next chapter, we shall make use of this particular characterization of strong conservation. Several equivalent conditions are provided in [11, Theorem 3.2]. A key property of this definition is that any linear combination of vectors in S_A will always be orthogonal to any linear combination of vectors in S_B. This property will allow for conservation principles in certain signal processing systems to be stated independently of the systems evolution.

2.5 | Mathematical optimization

A primary contribution of this thesis involves designing distributed algorithms for solving convex and non-convex optimization problems by assembling nonlinear signal processing systems that possess innate variational properties at their equilibrium states. With this in mind, we briefly review some fundamentals of optimization theory in this section and present the duality principle behind many popular classes of problems and associated algorithms. These preliminaries later assist in relating the contributions in this thesis to distinct classes of algorithms in the optimization literature. For a rigorous and complete development of this subject, we refer readers to classic texts such as [38–40].

In reviewing the fundamentals, we begin by considering a primal constrained optimization problem of the form

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g_k(\mathbf{x}) = 0, \quad k = 1, \ldots, K \\ & h_j(\mathbf{x}) \leq 0, \quad j = 1, \ldots, J \end{aligned} \tag{2.29}$$

where $f \colon \mathbb{R}^N \to \mathbb{R}$ is the objective or cost function, $g_k \colon \mathbb{R}^N \to \mathbb{R}$, for $k = 1, \ldots, K$, are the equality constraints, and $h_j \colon \mathbb{R}^N \to \mathbb{R}$, for $j = 1, \ldots, J$, are the inequality constraints. The collection of vectors $\mathbf{x}$ satisfying all of the constraints is referred to as the feasible set, and algorithms for solving (2.29) typically attempt to determine a feasible element

$\mathbf{x}^\star$ for which $f(\mathbf{x}^\star)$ is locally or globally minimum. In the sequel, we exclusively consider continuous feasible sets, e.g. closed subsets of $\mathbb{R}^N$. Tools to minimize over discrete sets are often combinatoric in nature or consist of solving sequences of continuous problems.

Convexity principles are ubiquitous in casting practical problems into the form (2.29). To elaborate on this, a function $T\colon \mathbb{R}^N \to \mathbb{R}$ is referred to as being *convex* if

$$T(\rho\mathbf{v} + (1-\rho)\mathbf{u}) \leq \rho T(\mathbf{v}) + (1-\rho)T(\mathbf{u}), \qquad \mathbf{u}, \mathbf{v} \in \mathbb{R}^N, \rho \in [0,1]. \tag{2.30}$$

If (2.30) holds with a strict inequality for $\rho \in (0,1)$ and $\mathbf{u} \neq \mathbf{v}$, then T is referred to as being *strictly convex.* Convex optimization problems are special cases of (2.29) where the cost and constraint functions are all convex. Convex problems conveniently equate local and global solutions whereas strictly convex problems also ensure the uniqueness of a solution.

2.5.1 | Duality principles and optimality conditions

In a broad sense, duality principles in optimization theory typically emphasize structural relationships between primal and dual functions or problems using injective transformations including symmetries, involutions, and inequalities. Examples of this include Kantorovich-Rubinstein duality for optimal transportation inequalities and linear programming [41, Theorem 5.10], Strassen duality for stochastic variables, and Fenchel duality in convex analysis [40, Theorem 31.1]. In Chapter 5, we draw upon the duality principle relating orthogonality and variational properties of conservative vector spaces. In this subsection, we focus on Lagrangian duality, which is, loosely speaking, well-known to be connected to Fenchel duality by representing convex sets using supporting hyperplanes.

In reviewing Lagrangian duality, we define the Lagrangian function $\mathcal{L}\colon \mathbb{R}^N \times \mathbb{R}^K \times \mathbb{R}^J \to \mathbb{R}$ associated with a primal optimization problem of the form (2.29) as

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\pi}, \boldsymbol{\lambda}) \quad = \quad f(\mathbf{x}) + \sum_{k=1}^{K} \boldsymbol{\pi}_k g_k(\mathbf{x}) + \sum_{j=1}^{J} \boldsymbol{\lambda}_j h_j(\mathbf{x}) \tag{2.31}$$

where $\boldsymbol{\pi}$ and $\boldsymbol{\lambda}$ are Lagrange multipliers and are also referred to as *dual variables.* From this, the Lagrange dual function $g \colon \mathbb{R}^K \times \mathbb{R}^J \to \mathbb{R}$ is defined according to

$$g(\boldsymbol{\pi}, \boldsymbol{\lambda}) \quad = \quad \inf_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \boldsymbol{\pi}, \boldsymbol{\lambda}). \tag{2.32}$$

By construction, evaluating $g(\boldsymbol{\pi}, \boldsymbol{\lambda})$ at any point $(\boldsymbol{\pi}, \boldsymbol{\lambda})$ with $\boldsymbol{\lambda} \geq \mathbf{0}$ (coordinate-wise) provides a lower bound to the optimal cost of (2.29), and a natural step is to identify the tightest possible lower bound. Thus, for the primal problem (2.29), the Lagrangian dual optimization problem is given by

$$\begin{aligned} \underset{\boldsymbol{\pi}, \boldsymbol{\lambda}}{\text{maximize}} \quad & g(\boldsymbol{\pi}, \boldsymbol{\lambda}) \\ \text{s.t.} \quad & \boldsymbol{\lambda}_j \geq 0 \qquad j = 1, \ldots, J. \end{aligned} \tag{2.33}$$

Observe that, independent of the convexity of the primal problem, the dual problem is concave. The quantity $f(\mathbf{x}^\star) - g(\boldsymbol{\pi}^\star, \boldsymbol{\lambda}^\star)$ for any $(\boldsymbol{\pi}^\star, \boldsymbol{\lambda}^\star)$ that solves (2.33) is referred to as the duality gap, and problems where the duality gap is zero are said to exhibit strong duality. In some cases, a solution $(\boldsymbol{\pi}^\star, \boldsymbol{\lambda}^\star)$ can be used to generate $\mathbf{x}^\star$ and vice versa. Indeed, solving both problems is often desirable as dual variables have various convenient interpretations, e.g. they are useful in understanding the sensitivity of primal solutions.

It is common to characterize optimal solutions to (2.29) in terms of conditions that are satisfied when such a solution is obtained. Let the Lagrangian in (2.31) be differentiable with respect to $\mathbf{x}$. Any optimal solution $\mathbf{x}^\star$ satisfying some regularity[3] is then associated with a dual solution $(\boldsymbol{\pi}^\star, \boldsymbol{\lambda}^\star)$ and collectively these solutions satisfy the conditions:

$$\begin{aligned} \text{Stationarity:} \quad & \nabla_{\mathbf{x}} f(\mathbf{x}^\star) + \sum_{k=1}^{K} \boldsymbol{\pi}_k^\star \nabla_{\mathbf{x}} g_k(\mathbf{x}^\star) + \sum_{j=1}^{J} \boldsymbol{\lambda}_j^\star \nabla_{\mathbf{x}} h_j(\mathbf{x}^\star) = 0 && \text{(2.34a)} \\ \text{Primal Feasibility:} \quad & g_k(\mathbf{x}^\star) = 0, \qquad k = 1, \ldots, K && \text{(2.34b)} \\ & h_j(\mathbf{x}^\star) \leq 0, \qquad j = 1, \ldots, J && \text{(2.34c)} \\ \text{Dual Feasibility:} \quad & \boldsymbol{\lambda}_j^\star \geq 0, \qquad j = 1, \ldots, J && \text{(2.34d)} \\ \text{Complementarity:} \quad & \boldsymbol{\lambda}_j^\star h_j(\mathbf{x}^\star) = 0, \qquad j = 1, \ldots, J && \text{(2.34e)} \end{aligned}$$

[3] There exist many forms of regularity or constraint qualification conditions. In this thesis, we use the condition that g_k and h_j are affine or, for convex problems, the feasible set has a well-defined relative interior.

where $\nabla_{\mathbf{x}}$ denotes the gradient with respect to $\mathbf{x}$. These necessary conditions are the celebrated KKT optimality conditions and are strengthened to sufficiency when the cost function f and inequality constraints h_j are convex and the equality constraints g_k are affine.

For problems with non-differentiable Lagrangians, the KKT conditions generalize in a straightforward way by proper application of subderivative relations. Moreover, if $J = 0$, the KKT conditions reduce to the method of Lagrange multipliers in elementary calculus and reduce further to the unconstrained first-order condition $\nabla_{\mathbf{x}} f(\mathbf{x}^\star) = 0$ if $K = 0$ too. Geometrically, constrained optimization problems are essentially unconstrained when the constraints are not active at the minimizer. More generally, when solutions are on the boundary of the feasible set, the complementarity conditions ensure that only active inequality constraints contribute to the stationarity condition. Similarly, the dual feasibility conditions ensure that the primal inequality constraints h_j contribute appropriately to the dual objective function. Methods aimed at directly solving the KKT conditions have the potential to yield solutions to both the primal and dual problems simultaneously.

2.5.2 | Optimization algorithms

In designing algorithms to solve distributed or large-scale optimization problems, a common strategy is to partition an iteration onto several processors that simultaneously perform computation and regularly exchange data. Convergence analysis for distributed or non-distributed computing platforms is essentially the same when data transfers follow pre-specified schedules or use round-robin token passing. Performance for these data exchange protocols may suffer in the presence of synchronization problems, e.g., arising from network congestion or processor heterogeneity. In response to this, many traditional algorithms have been adapted to alleviate these problems by allowing asynchronous communication between processors while maintaining convergence guarantees [42]. In this subsection, we review two algorithm classes of this type.

We first consider minimizing an unconstrained, differentiable objective function $f\colon \mathbb{R}^N \to \mathbb{R}$ that is separable into the sum of M terms according to

$$f(\mathbf{x}) \quad = \quad f_1(\mathbf{x}) + \cdots + f_M(\mathbf{x}) \tag{2.35}$$

where each summand $f_m \colon \mathbb{R}^N \to \mathbb{R}$, for $m = 1, \ldots, M$, might operate on only a subvector of $\mathbf{x}$. For example, (2.35) may represent an empirical estimate of the expectation of f in probabilistic settings. The *stochastic gradient descent* algorithm attempts to determine a solution $\mathbf{x}^\star$ by producing a sequence of vectors $\{\mathbf{x}^n \colon n \in \mathbb{N}_0\}$ according to the update

$$\mathbf{x}^n = \mathbf{x}^{n-1} - \rho \nabla_{\mathbf{x}} f_m(\mathbf{x}^{n-1}), \qquad n \in \mathbb{N} \tag{2.36}$$

where $m \in \{1, \ldots, M\}$ is a random integer chosen independently for each n and $\rho > 0$ is the step size. The change in $\mathbf{x}^n$ after multiple iterations is interpretable as moving in the direction of a noisy gradient estimate and provides robustness to minimizing non-smooth objective functions with many local minima. When individual terms f_m are not differentiable, $\nabla_{\mathbf{x}} f_m(\mathbf{x}^{n-1})$ can be replaced by subdifferentials of f_m at $\mathbf{x}^{n-1}$.

The fact that the iteration in (2.36) selects the parameter m without reference to the previous iteration suggests opportunity in distributing and asynchronously implementing the stochastic gradient descent algorithm. By assigning each summand f_m to a different processor, each processor independently retrieves the current state of the partial solution $\mathbf{x}$ from globally accessible memory, computes the update (2.36) or an incremental form of it, and writes the result back into memory. This strategy is similar to the `HogWild!` approach discussed in [43] where convergence is guaranteed under reasonable assumptions such as Lipschitz continuity of the gradients $\nabla_{\mathbf{x}} f_m$. When the gradient of each summand f_m happens to be sparse, asynchronous implementations converge in approximately the same amount of computation or equivalent iterations as their synchronous or batched counterparts.

Another common class of asynchronous and distributable algorithms pertain to solving linearly constrained optimization problems of the form

$$\begin{aligned} \underset{\mathbf{x},\,\mathbf{z}}{\text{minimize}} \quad & f_1(\mathbf{x}) + f_2(\mathbf{z}) \\ \text{s.t.} \quad & A\mathbf{x} + B\mathbf{z} = \mathbf{c} \end{aligned} \tag{2.37}$$

where f_1 and f_2 are possibly non-smooth convex functions. The *alternating direction method of multipliers* (ADMM) algorithm is widely used to solve problems within this class. In writing the ADMM algorithm, we define an augmented Lagrangian as

$$\mathcal{L}_\rho(\mathbf{x}, \mathbf{z}, \boldsymbol{\pi}) \quad = \quad f_1(\mathbf{x}) + f_2(\mathbf{z}) + \boldsymbol{\pi}^T(A\mathbf{x} + B\mathbf{z} - \mathbf{c}) + \frac{\rho}{2}\|A\mathbf{x} + B\mathbf{z} - \mathbf{c}\|^2 \tag{2.38}$$

where $\rho > 0$ is a tuning parameter. The augmentation provides the dual function associated with (2.38) to be differentiable for a broader set of choices for f_1 and f_2. Note that (2.38) is the standard Lagrangian associated with (2.37), except with the term $\frac{\rho}{2}\|A\mathbf{x} + B\mathbf{z} - \mathbf{c}\|^2$ added to the cost function. The ADMM algorithm seeks a solution by generating sequences for $\mathbf{x}$, $\mathbf{z}$, and $\boldsymbol{\pi}$ given by:

$$\mathbf{x}^n \quad = \quad \arg\min_{\mathbf{x}} \mathcal{L}_\rho(\mathbf{x}, \mathbf{z}^{n-1}, \boldsymbol{\pi}^{n-1}) \tag{2.39a}$$

$$\mathbf{z}^n \quad = \quad \arg\min_{\mathbf{z}} \mathcal{L}_\rho(\mathbf{x}^n, \mathbf{z}, \boldsymbol{\pi}^{n-1}) \tag{2.39b}$$

$$\boldsymbol{\pi}^n \quad = \quad \boldsymbol{\pi}^{n-1} + \rho(A\mathbf{x}^n + B\mathbf{z}^n - \mathbf{c}). \tag{2.39c}$$

Distributed implementations of the iteration above have been incorporated into the MapReduce programming model [44]. In addition, block splitting methods allow the linear constraints to be arbitrarily partitioned by row and column and assigned to different processors [22]. In terms of convergence, guarantees for asynchronous implementations of ADMM have been shown almost surely [45]. This differs from the mean-square convergence under which asynchronous implementations of the algorithms developed in this thesis are guaranteed to work. On a technical level, the stochastic and asynchronous convergence analysis in this thesis has similarities to the analysis of proximal algorithms in [46] and monotone operators in [47] wherein the approach is to focus on fixed-point properties rather than gradient based optimality conditions since the iterated operators are generally not directly related to the gradient of the cost function. Connections between gradient-based algorithms, ADMM and other related proximal algorithms, and the class of algorithms developed in this thesis are discussed in detail in Section 5.6.

Chapter 3

Interconnective framework

The general idea behind an interconnective description of a signal processing system is to model the system through the coupling of many independent constitutive modules and interconnecting networks. The interconnecting networks are used to characterize the linear and memoryless relationships internal to the system while the constitutive modules characterize the system's nonlinear features, memory, and overall system inputs and outputs. Separability of the constitutive modules is helpful in understanding the connectivity and communication required for distributed implementations of the system whereas separability of the interconnecting networks informs the choice between centralized and decentralized implementations. These separability properties, utilized in Chapter 6 for the purposes just mentioned, are invariant to the complexity of each module as well as to the domains in which the constraints associated with the modules are enforced.

This chapter begins with a generalization of the interconnective system representations presented in Section 2.3 by introducing new terminology with which to describe a signal processing system, and the remainder of the chapter is spent formulating the consequences of those terms. In particular, an interconnective description is developed with multiple levels of specificity by allowing the system to be described using different amounts of structure. Starting in the coordinate-free setting, a fixed-coordinate model is presented that naturally inspires an equivalence relation between systems whose behaviors are equal to within linear transformations of one another. Systematic procedures for translating behavioral models into computable signal processing structures are proposed so that algorithms can be produced by inserting delay modules to break delay-free loops, thereby instilling coordinated or uncoordinated dynamics into the structure as well as ensuring a well-defined precedence tree exists for scheduling purposes. Algorithms produced this way encapsulate a broad range of organizations and input-output configurations of a system's variables, and the focus is specifically on two types of algorithms: synchronous processing loops and asynchronous processing systems. The chapter then concludes with several examples.

3.1 | Interconnective descriptions of signal processing systems

Selection between different models or descriptions of a class of signal processing systems are often based on intended processing goals. In some contexts, for example, a particular description may be better suited to handle specific tasks when compared to its alternatives due to convenient properties the chosen description exhibits toward the present context. In this section and the next, we develop an interconnective description to represent and manipulate large-scale signal processing systems that is well-suited to the eventual processing goals in this thesis. Toward this end, we build upon the existing interconnective descriptions reviewed in Section 2.3. Key factors motivating the extension of these descriptions include the current focus on large-scale and decentralized systems involving multiple independent interconnecting networks and constitutive modules as well as a link to be discussed in the following section between equivalent systems operating on the same underlying graph for a particular notion of equivalence. In this section, we develop the foundations of an abstract interconnective description where connectivity and separability properties important to designing and implementing distributed signal processing systems are explicitly underscored, and then we define a notation for illustrating these properties using graphical structures.

3.1.1 | Interconnective decompositions

To characterize the behavior of a system through the coupling of a memoryless, linear interconnecting network to several independent constitutive modules, we shall make use of a particular decomposition of the vector space containing the system's behavior as well as a particular decomposition of the behavior itself. Utilizing the terminology developed by

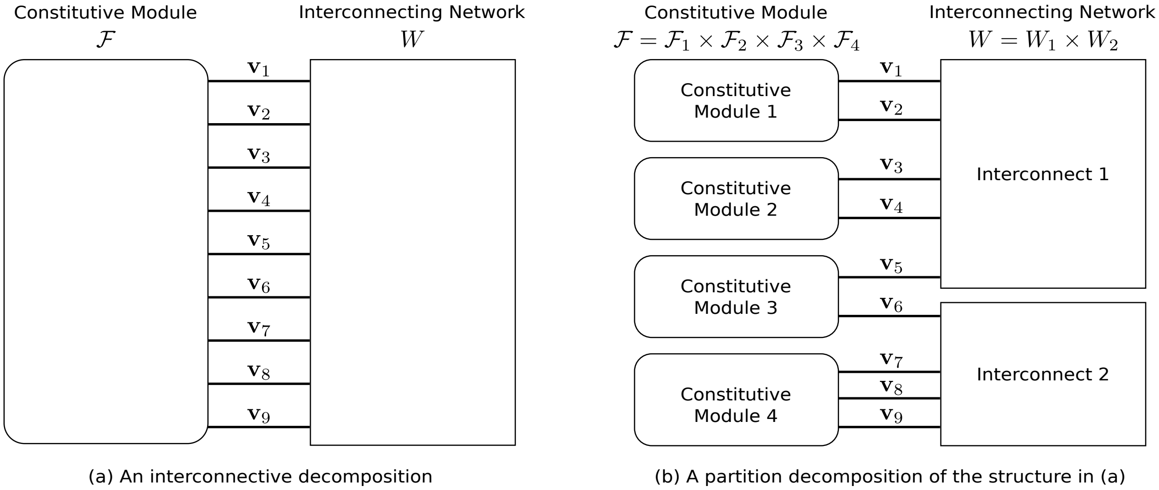


Figure 3-1: An illustration of the interconnective description of a signal processing system formed by coupling a constitutive module $\mathcal{F}$ to an interconnecting network W. (a) An interconnective decomposition. (b) A partition decomposition further decomposing the structure in (a).

Willems in [15], we shall refer to the vector space as the *terminal linear space*, or *terminal space* for short. Formally, the terminal space is defined as the finite-dimensional vector space $\mathcal{V}$ whose dimensionality equals the number of terminal variables used in the description of a system's behavior. From this, an *interconnective decomposition* $\mathcal{D}_i$ of a system's behavior $\mathcal{B} \subseteq \mathcal{V}$ is defined as a description of $\mathcal{B}$ using the components

$$\mathcal{D}_i \triangleq \{W, \mathcal{F}\} \tag{3.1}$$

where W and $\mathcal{F}$ are each subsets of $\mathcal{V}$, so that the behavior equals the intersection of W and $\mathcal{F}$, i.e. the behavior $\mathcal{B}$ decouples according to

$$\mathcal{B} = W \cap \mathcal{F}, \tag{3.2}$$

and where W in particular is a linear subspace of $\mathcal{V}$. Figure 3-1(a) depicts a graphical representation of an interconnective decomposition for a signal processing system whose behavior is expressed using nine terminal variables $\mathbf{v} = (\mathbf{v}_1, \ldots, \mathbf{v}_9)$.

Comparing (2.9) and (3.2), forming an interconnective decomposition is interpretable as generating separate linear and nonlinear systems whose behaviors form the overall system's behavior when connected together. In the sequel, the interconnective decomposition $\mathcal{D}_i$ will routinely be used to refer to the behavior of a system rather than referencing the intersection of W and $\mathcal{F}$ directly, and we do so with the understanding that we specifically mean their intersection as demonstrated by (3.2). Also, the vector subspace W or any linear map that realizes it according to (2.4) will be referred to as the *interconnecting network*, or *interconnect* for short. Similarly, the set $\mathcal{F}$ or any generally nonlinear relation or function that realizes it according to (2.1) will be referred to as the *constitutive module* or *relation*. Constitutive modules will also provide an external interface for systems with overall inputs and outputs. Finally, signal processing systems whose behaviors are expressed using an interconnective decomposition will be referred to as *interconnective systems*.

3.1.2 | Partition decompositions

Interconnective decompositions, introduced in the previous subsection, are universal in the sense that they can express the behavior of any signal processing system. To see this, let $\mathcal{F}$ denote a system's behavior and let W be any subspace that subsumes $\mathcal{F}$. Interconnective decompositions constructed in this way are not very descriptive, so, in this subsection, we proceed by introducing some additional structure through a decomposition of the terminal space $\mathcal{V}$ that serves to separate the constitutive module $\mathcal{F}$ into generally many independent constitutive modules or subsystems. When this decomposition is appropriately selected, it helps to restrict the choice of the interconnecting network subspace W so that the system's behavior $W \cap \mathcal{F}$ is a proper subset of both W and $\mathcal{F}$.

For an interconnective system with terminal space $\mathcal{V}$ and interconnective decomposition $\mathcal{D}_i = \{W, \mathcal{F}\}$, a *partition decomposition* corresponds to separating the terminal space $\mathcal{V}$ into several lower dimensional vector spaces such that the constitutive relation $\mathcal{F}$ acts as an independent set constraint on each. More formally, a partition decomposition is defined as a collection of K vector spaces $\{\mathcal{V}_k\}$ that decompose the terminal space $\mathcal{V}$ according to

$$\mathcal{V} \;=\; \mathcal{V}_1 \times \cdots \times \mathcal{V}_K \tag{3.3}$$

and that separate the constitutive module $\mathcal{F}$ into K constitutive modules $\{\mathcal{F}_k\}$ satisfying

$\mathcal{F}_k \subseteq \mathcal{V}_k$, for $k = 1, \ldots, K$, such that $\mathcal{F}$ can be reconstructed according to

$$\mathcal{F} = \mathcal{F}_1 \times \cdots \times \mathcal{F}_K. \tag{3.4}$$

The vector space $\mathcal{V}_k$ will be referred to as the k-th *partition subspace* of $\mathcal{V}$ and collectively the partition subspaces form a partition decomposition $\mathcal{D}_p = \{\mathcal{V}_1, \ldots, \mathcal{V}_K\}$. With access to a bona fide partition decomposition, there is no ambiguity in determining what the individual constitutive modules are, thus the formal definition of an interconnective description of a signal processing system does not need to explicitly include them.

The total number of partition subspaces provides some level of insight into how extensively a signal processing system can be distributed onto different processors that then implement the system by concurrently or sequentially executing their assigned processing instructions. On the other hand, separability of the interconnecting network dictates whether or not the group of processors can be decentralized. To illustrate this, consider the case where the interconnection subspace W decomposes using direct products as

$$W \quad = \quad W_1 \times \cdots \times W_L \tag{3.5}$$

where the case $L = 1$ subsumes the outcome that a proper direct product decomposition of W does not exist. Figure 3-1(b) illustrates a graphical representation of a partition decomposition for the interconnective system in (a) where the constitutive relation has been split into four independent modules and where the interconnecting network has additionally been separated into two independent interconnects. As indicated by the figure, the separation of the interconnects implies that any information exchange between the first two and the fourth constitutive modules will only be communicated through the third constitutive module, thus enabling the possibility of a decentralized implementation of the system.

As was the case with interconnective decompositions, partition decompositions are generally not unique. In response to this, we will typically be interested in partition decompositions whose partition subspaces satisfy a particular separability property, namely each subspace being two dimensional. This property is only possible, of course, for terminal spaces of even dimensionality, which is also a necessary condition for a system to be conservative. Partition decompositions whose partition subspaces satisfy this property will be referred to as *maximal partition decompositions.*

We conclude this subsection by commenting on the relationship between the definition of a partition decomposition made in this subsection and the definition made in [11] that was used to discuss conservation principles in signal processing systems in Section 2.4. To modify the definition above to be consistent with the decomposition in (2.12), the terminal space $\mathcal{V}$ would need to be decomposed using a direct sum of K subspaces $\{\mathcal{V}_k\}$ according to $\mathcal{V} = \mathcal{V}_1 \oplus \cdots \oplus \mathcal{V}_K$ rather than the direct product decomposition in (3.3). Then, the constitutive modules $\mathcal{F}_k$ would be generated according to

$$\mathcal{F}_k \quad = \quad \mathbf{proj}_{\mathcal{V}_k}\mathcal{F} + \bigoplus_{j \neq k} \mathcal{V}_j, \quad k = 1, \dots, K, \tag{3.6}$$

where $\mathbf{proj}_{\mathcal{V}_k}\mathcal{F}$ is the orthogonal projection of $\mathcal{F}$ onto $\mathcal{V}_k$, and the original constitutive relation $\mathcal{F}$ would be reconstructed using intersections according to $\mathcal{F} = \mathcal{F}_1 \cap \cdots \cap \mathcal{F}_K$ instead of using direct products as in (3.4). Notice that the notation for the partition subspaces $\mathcal{V}_k$ and subsystem modules $\mathcal{F}_k$ is now overloaded since they represent sets of vectors of different lengths depending on whether they correspond to decompositions using direct products or direct sums and intersections. Despite this ambiguity, the quantity $\dim\mathcal{V}_k$, which is essential to capturing the separability of an interconnective system, is invariant to the decomposition used, so we proceed with the convention that an interconnective description uses the direct product decompositions in (3.3) and (3.4) unless explicitly stated otherwise.

3.1.3 | Coordinate-free interconnective descriptions

Having established both interconnective and partition decompositions to describe behavioral models of large-scale signal processing systems, we are now equipped with an adequate amount of structure to define a meaningful representation of a system. This description specifically captures dependencies implied by a system's topology in an abstract way, i.e. without reference to a particular basis or coordinate system. This formulation will be referred to as a coordinate-free description and written using the notation $\mathcal{R}_{cf}$.

Definition 3.1.1 (Coordinate-free interconnective descriptions)**.** *A coordinate-free interconnective description of a signal processing system is defined as a triple*

$$\mathcal{R}_{cf} \;\triangleq\; (\mathcal{V},\ \mathcal{D}_p,\ \mathcal{D}_i) \tag{3.7}$$

with elements

$\mathcal{V}$: *a terminal linear space defined over an appropriate field,*

$\mathcal{D}_p$: *an associated partition decomposition, and*

$\mathcal{D}_i$: *an associated interconnective decomposition,*

where the elements composing the partition and interconnective decompositions uniquely decompose the constitutive relation $\mathcal{F}$ according to

$$\mathcal{F} \;=\; (\mathcal{F}_1 \cap \mathcal{V}_1) \times \cdots \times (\mathcal{F}_K \cap \mathcal{V}_K) \tag{3.8}$$

where K is the total number of partition subspaces.

3.1.4 | Graphical structures for interconnective systems

As is common in working with general signal processing systems, it is often useful to be able to write a succinct and unambiguous description of a system using a graphical language or structure. In this subsection, we define a graphical convention that focuses on issues pertinent to the design and implementation of large-scale systems using elements from the system's interconnective description. Also, we establish a notational convention for addressing certain subsets of terminal variables that will later be used to manipulate the individual constitutive modules and interconnecting networks composing the overall system.

Formally, an *interconnective graph* or *structure* is defined as a bipartite, undirected graph where the constitutive modules $\mathcal{F}_k$ form the first type of nodes and are depicted using blocks with rounded corners, and the interconnects W_l form the second type of nodes and are depicted using blocks with square corners. Edges of a graph are used to indicate that two adjacent nodes of opposite type constrain an overlapping subset of the terminal variables and that the nodes will require a direct communication link when implemented on different processors. Graphically, edges are depicted using bidirectional arrows of the form $\longleftrightarrow$ where the bidirectional notation underscores the fact that the communication links may need to be reciprocal in order to enforce the collective constraints the nodes place over their shared terminal variables with respect to the overall system's behavior as well as the fact that the terminal variables have yet to be designated as being inputs or outputs of the nodes. A similar convention appears in the bond graph literature where arrow bidirectionality is used to emphasize the reciprocal exchange of information or energy between physical subsystems [48]. In many places throughout this thesis, an interconnective system will be defined using a graphical structure rather than an analytic description with the understanding that the two are equivalent to one another, i.e. that an analytic description is completely determined by an interconnective graph as described above and vice versa.

With direct product decompositions (3.3) through (3.5) in place, two conventions for partitioning a length N terminal vector $\mathbf{v} = (\mathbf{v}_1, \ldots, \mathbf{v}_N)$ will play an essential role in manipulating interconnective systems and their graphs. In the first convention, a terminal vector $\mathbf{v}$ is split into K subvectors, denoted by $\mathbf{v}_k^{(CR)}$ for $k = 1, \ldots, K$, such that the k-th constitutive module $\mathcal{F}_k$ is the only constitutive module that constrains the subvector $\mathbf{v}_k^{(CR)}$ for each k. Similarly, in the second convention, the terminal vector $\mathbf{v}$ is split into L subvectors, denoted by $\mathbf{v}_l^{(LI)}$ for $l = 1, \ldots, L$, such that the l-th interconnect W_l is the only interconnect that constrains $\mathbf{v}_l^{(LI)}$ for each l. Written formally, these partitioning conventions are summarized by

$$\begin{aligned} \mathbf{v} &= (\mathbf{v}_1, \ \ldots, \ \mathbf{v}_N) && (3.9) \\ &= (\mathbf{v}_1^{(CR)}, \ \ldots, \ \mathbf{v}_K^{(CR)}) && (3.10) \\ &= (\mathbf{v}_1^{(LI)}, \ \ldots, \ \mathbf{v}_L^{(LI)}). && (3.11) \end{aligned}$$

The length of each subvector $\mathbf{v}_k^{(CR)}$ and $\mathbf{v}_l^{(LI)}$ is respectively denoted by $N_k^{(CR)}$ and $N_l^{(LI)}$ and collectively these lengths satisfy

$$\begin{aligned} N &= N_1^{(CR)} + \cdots + N_K^{(CR)} && (3.12) \\ &= N_1^{(LI)} + \cdots + N_L^{(LI)} && (3.13) \end{aligned}$$

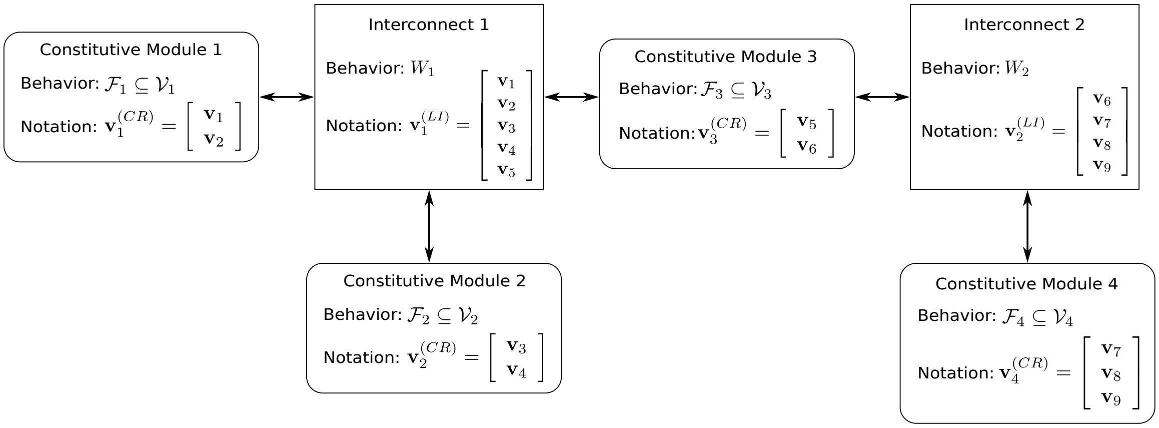

Constitutive Module 1
Behavior: $\mathcal{F}_1 \subseteq \mathcal{V}_1$
Notation: $\mathbf{v}_1^{(CR)} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix}$
Interconnect 1
Behavior: W_1
Notation: $\mathbf{v}_1^{(LI)} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \\ \mathbf{v}_4 \\ \mathbf{v}_5 \end{bmatrix}$
Constitutive Module 3
Behavior: $\mathcal{F}_3 \subseteq \mathcal{V}_3$
Notation: $\mathbf{v}_3^{(CR)} = \begin{bmatrix} \mathbf{v}_5 \\ \mathbf{v}_6 \end{bmatrix}$
Interconnect 2
Behavior: W_2
Notation: $\mathbf{v}_2^{(LI)} = \begin{bmatrix} \mathbf{v}_6 \\ \mathbf{v}_7 \\ \mathbf{v}_8 \\ \mathbf{v}_9 \end{bmatrix}$
Constitutive Module 2
Behavior: $\mathcal{F}_2 \subseteq \mathcal{V}_2$
Notation: $\mathbf{v}_2^{(CR)} = \begin{bmatrix} \mathbf{v}_3 \\ \mathbf{v}_4 \end{bmatrix}$
Constitutive Module 4
Behavior: $\mathcal{F}_4 \subseteq \mathcal{V}_4$
Notation: $\mathbf{v}_4^{(CR)} = \begin{bmatrix} \mathbf{v}_7 \\ \mathbf{v}_8 \\ \mathbf{v}_9 \end{bmatrix}$

Figure 3-2: The interconnective structure associated with a decentralized behavioral model of the signal processing system in Figure 3-1 illustrating the notational conventions used to describe systems graphically.

where $N_k^{(CR)} = \dim\mathcal{V}_k$ for $k = 1, \ldots, K$. To this point, the partitioning scheme has yet to designate the individual terminal variables composing the terminal vector $\mathbf{v}$ as being inputs or outputs from either the interconnects W_l or constitutive modules $\mathcal{F}_k$.

Before concluding this subsection, we present an example to demonstrate both the graphical conventions outlined above as well as the partitioning schemes summarized by (3.9) through (3.11). To do this, we continue with the the signal processing system whose interconnective description is shown in Figure 3-1. The coordinate-free interconnective description of this system is written according to $\mathcal{R}_{cf} = (\mathcal{V}, \mathcal{D}_p, \mathcal{D}_i)$ where the terminal space $\mathcal{V}$ is 9-dimensional and the elements composing the decompositions take the form

$$\mathcal{D}_p = \{\mathcal{V}_1,\ \mathcal{V}_2,\ \mathcal{V}_3,\ \mathcal{V}_4\} \tag{3.14}$$

$$\mathcal{D}_i = \{W_1 \times W_2,\ \mathcal{F}_1 \times \mathcal{F}_2 \times \mathcal{F}_3 \times \mathcal{F}_4\} \tag{3.15}$$

where $\dim\mathcal{V}_k$ is equal to 2 for $k = 1, 2, 3$, and 3 for $k = 4$. Figure 3-2 describes the system using an interconnective graph. The notation established for the various arrangements of the terminal vector $\mathbf{v}$ is also depicted. In particular, the first three constitutive modules constrain the first three pairs of terminal variables while the fourth constitutive module constrains the remaining three. The decentralized nature of the structure follows immediately from the separation of the interconnecting network into two independent interconnects.

3.2 | Equivalent interconnective descriptions

The core operating principle underlying many well-established signal processing algorithms is to transform an input signal to a domain for efficient processing, process the signal in that domain, and then transform the processed signal back to the original domain. Homomorphic filtering methods, originally used in speech contexts and more recently in deep learning networks [49, 50], illustrate this principle where the transform of the input signal is non-linear. Fast convolution methods are also consistent with this principle where fast implementations of the discrete Fourier transform are used to reduce the complexity associated with performing convolution. From the interconnective viewpoint, algorithms in this spirit are interpretable as performing the transforms via pre- and post-processing by the interconnects coupling adjacent constitutive modules where the modules specifically process signals in different coordinate systems. In this section, we build upon the interconnective description of a signal processing system to allow for invertible linear transformations of the system behavior and then identify conditions under which these transformations preserve the separability of the systems underlying graph.

A recurring and important theme emphasized throughout this thesis is that the organization of a system's behavior for which a useful property manifests itself is not necessarily the only organization that can take advantage of that property in practice. To assist with using this observation, we conclude this section by defining an equivalence relation between interconnective systems whose behaviors are intrinsically related to one another and then use the correspondences between those system's organizations to parameterize the associated equivalence classes. By doing this, properties derived from one organization can potentially be exploited by any system in the same equivalence class by understanding the effect reorganizing the system's behavior has on those properties.

3.2.1 | Coordinate maps

A *coordinate map* is defined as any invertible linear map M taking the terminal space $\mathcal{V}$ into itself. The basic utility of a coordinate map is to employ the transformation M so that a system's behavior $\mathcal{D}_i$ is viewed in $\mathcal{V}$ as $M\mathcal{D}_i$. We proceed with the following convention:

when M corresponds to a general linear operator then $\mathcal{V}$ remains an abstract vector space, but when M corresponds to a coordinate matrix then $\mathcal{V} = \mathbb{R}^N$ is assigned the standard basis $\{\mathbf{e}^{(1)}, \ldots, \mathbf{e}^{(N)}\}$ through which to view $M\mathcal{D}_i$, where $N = \dim\mathcal{V}$ and $\mathbf{e}^{(n)}$ is the length N vector that is unity in its n-th entry and zero elsewhere. The role of the coordinate map is important in many contexts and is discussed for the purposes of generating algorithms in Section 3.3 and stability analysis for uncoordinated data processing systems in Section 4.3.

Specifying a coordinate map can be thought of as selecting N linearly independent dual vectors[1] $\mathbf{m}_{(n)} \in \mathcal{V}^\star$, where $\mathcal{V}^\star$ denotes the dual vector space to $\mathcal{V}$, and arranging them into the linear transformation M according to

$$M = \begin{bmatrix} \mathbf{m}_{(1)} \\ \vdots \\ \mathbf{m}_{(N)} \end{bmatrix}. \tag{3.16}$$

The perspective provided by thinking of a coordinate map as a mapping from the dual vector space $\mathcal{V}^\star$ to the terminal space $\mathcal{V}$ is useful in interpreting the action of each dual vector $\mathbf{m}_{(n)}$ in the expression $M\mathcal{D}_i$ as specifically producing a new terminal variable as a linear combination of the terminal variables used to describe $\mathcal{D}_i$.

In dealing with interconnective systems, an important subset of coordinate maps are those for which a system's partition decomposition remains the same, i.e. for which $\mathcal{D}_p = M\mathcal{D}_p$, or, equivalently, for which $\mathcal{F}$ and $M\mathcal{F}$ decompose into the same partition subspaces. We shall refer to coordinate maps satisfying this property as being $\mathcal{D}_p$-invariant coordinate maps. These special maps later assist with designing algorithms that naturally distribute onto various processor networks and can easily be generated by selecting K bases for the dual partition subspaces $\mathcal{V}_k^\star$, for $k = 1, \ldots, K$. More concretely, the linear map M in (3.16) is $\mathcal{D}_p$-invariant if the N linearly independent dual vectors $\mathbf{m}_{(n)} \in \mathcal{V}^\star$, for $n = 1, \ldots, N$, also

[1] In this thesis, the parenthetical superscript notation $\mathbf{v}^{(n)}$ is used to indicate a vector in a vector space and the parenthetical subscript notation $\mathbf{v}_{(n)}$ to indicate a covector (linear functional) in a dual vector space.

satisfy the conditions

$$\mathcal{V}_1^* = \operatorname{span}\left\{\mathbf{m}_{(1)}, \ldots, \mathbf{m}_{(\dim \mathcal{V}_1)}\right\} \tag{3.17a}$$

$$\vdots$$

$$\mathcal{V}_K^* = \operatorname{span}\left\{\mathbf{m}_{(N-\dim \mathcal{V}_K+1)}, \ldots, \mathbf{m}_{(N)}\right\}. \tag{3.17b}$$

It is a straightforward exercise to show that the partition decomposition $\mathcal{D}_p$ is invariant to M when constructed using dual vectors satisfying (3.17).

As a matter of choice, we restrict ourselves moving forward to using $\mathcal{D}_p$-invariant coordinate maps whenever the interconnective description of a system uses direct product decompositions. By doing so, separability of the constitutive modules is preserved when applying the coordinate map M to $\mathcal{D}_i$. Selecting a coordinate map consistent with this requirement reduces to specifying a block-diagonal coordinate map M of the form

$$M = \begin{bmatrix} M_1 & 0 & \cdots & 0 \\ 0 & M_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & M_K \end{bmatrix} \tag{3.18}$$

where M is invertible if and only if each block M_k, for $k = 1, \ldots, K$, is an invertible $N_k^{(CR)} \times N_k^{(CR)}$ linear map taking $\mathcal{V}_k$ into itself. Note that while separability of the constitutive modules is preserved, separability of the interconnecting network may change when using $\mathcal{D}_p$-invariant coordinate maps, meaning only a subset of these maps allow algorithms to be derived from the transformed system that operate on the same decentralized graph.

If the behavior of a signal processing system is initially described using the coordinate map M, then a related system can be defined by describing the same system's behavior using any second coordinate map M' by appropriately changing coordinate maps. This process, referred to as a *coordinate transform*, corresponds to reorganizing the first system's behavior by applying the coordinate transform $M'M^{-1}$ to it. Since the inverse of a block-diagonal linear map is block-diagonal, the coordinate transform $M'M^{-1}$ used to move to the new behavioral model preserves the partition decomposition and sparsity pattern in (3.18) if M' and M are each block diagonal with conforming block sizes.

3.2.2 | Fixed-coordinate interconnective descriptions

In the same sense that a linear transformation can be represented as a matrix once bases have been chosen for the transformation's domain and codomain, the interconnective description of a signal processing system becomes more specific once a coordinate matrix has been selected. This level of description will be referred to as a fixed-coordinate interconnective description and will be written using the notation $\mathcal{R}_{fc}$.

Definition 3.2.1 (Fixed-coordinate interconnective descriptions). *A fixed-coordinate description of a signal processing system is defined as a quadruple*

$$\mathcal{R}_{fc} \;\triangleq\; (\mathcal{V},\ M\mathcal{D}_p,\ M\mathcal{D}_i,\ M) \tag{3.19}$$

with elements

$(\mathcal{V},\ \mathcal{D}_p,\ \mathcal{D}_i)$: *a coordinate-free interconnective representation, and*

M: *an associated coordinate map or matrix.*

In the remainder of the thesis, we work exclusively with fixed-coordinate descriptions of signal processing systems using both coordinate maps and matrices, thus $\mathcal{R}_{fc}$ will simply be referred to as *the* interconnective description. In principle, this description remains abstract and coordinate-free when the coordinate map M represents a general invertible linear map. However, once M becomes a coordinate matrix and $\mathcal{V}$ is set to $\mathbb{R}^N$, the description becomes numeric. As was mentioned previously, the coordinate map used to initially describe a signal processing system is not the only coordinate map that can provide a complete description of the system. The requirements of coordinate maps and matrices allow for a potentially large number of candidates for use in a system's interconnective description. This flexibility inspires an equivalence relation between interconnective systems whose behaviors are related through changing coordinate maps, the formal statement and justification of which is the subject of the next subsection.

3.2.3 | Interconnective equivalence classes

Consistent with the earlier discussion surrounding Figure 2-1, the correspondence between functions whose behaviors are related through invertible linear transformations is often nonlinear from the perspective of functional operations and manipulations. Generalizing this point to interconnective descriptions, similar correspondences become increasingly important in the coming chapters for relating systems in various contexts. The equivalence classes established in this subsection provide a straightforward way to identify these types of connections as well as distinguish between signal processing systems that are distinct from one another on a more fundamental level. Toward this goal, we state a binary relation that leads to an interconnective notion of equivalence between systems in the following definition.

Definition 3.2.2 (Interconnective binary relation)**.** *Let s_k denote a signal processing system whose fixed-coordinate interconnective description $\mathcal{R}_{fc}^{(k)}$ is given by*

$$\mathcal{R}_{fc}^{(k)} \quad = \quad (\mathcal{V},\ \mathcal{D}_p^{(k)},\ \mathcal{D}_i^{(k)},\ M_k). \tag{3.20}$$

Let s_{k_1} and s_{k_2} denote two signal processing systems whose interconnective descriptions are given by $\mathcal{R}_{fc}^{(k_1)}$ and $\mathcal{R}_{fc}^{(k_2)}$ according to (3.20)*, respectively. We define s_{k_1} and s_{k_2} to be equivalent interconnective systems if the description of s_{k_2} can be obtained by an appropriate coordinate transform applied to s_{k_1}, i.e. if and only if*

$$\mathcal{D}_p^{(k_2)} \quad = \quad M_{k_2} M_{k_1}^{-1} \mathcal{D}_p^{(k_1)} \tag{3.21}$$

and

$$\mathcal{D}_i^{(k_2)} \quad = \quad M_{k_2} M_{k_1}^{-1} \mathcal{D}_i^{(k_1)}. \tag{3.22}$$

When (3.21) *and* (3.22) *hold, this equivalence is written succinctly as $s_{k_1} \sim s_{k_2}$.*

For an invertible linear map M, the notation $M\mathcal{D}_p$ and $M\mathcal{D}_i$ in the previous definition specifically refers to the linear transformation of each element in the decompositions as demonstrated by (2.6). In principle, justifying that a binary relation is an equivalence relation requires the relation to satisfy three properties: (i) reflexivity, (ii) symmetry, and (iii) transitivity. For the specific definition of an interconnective relation $\sim$ established above, each of these properties is verified next by using the appropriate coordinate change.

Let s_k denote an interconnective system with fixed-coordinate representation $\mathcal{R}_{fc}^{(k)}$ in (3.20) for $k = 1, 2, 3$. The reflexivity property required of $\sim$ is satisfied if any interconnective system is equivalent to itself. The relationship $s_1 \sim s_1$ immediately follows from (3.21) and (3.22) by noting that $M_{k_2} = M_{k_1}$, thereby simplifying the linear map $M_{k_2} M_{k_1}^{-1}$ that is applied to the partition and interconnective decompositions to the identity map since $M_{k_1} M_{k_1}^{-1} = I$. The symmetry property required of $\sim$ is satisfied if the assumption $s_1 \sim s_2$ implies that $s_2 \sim s_1$. This follows again from (3.21) and (3.22) and the invertibility required of a coordinate map. Specifically, invertibility of the linear map M_2 can be used to manipulate (3.21) by multiplying the partition decomposition $\mathcal{D}_p^{(2)}$ by $M_1 M_2^{-1}$ to obtain

$$\mathcal{D}_p^{(1)} = M_1 M_2^{-1} \mathcal{D}_p^{(2)}, \tag{3.23}$$

and similarly to multiply the relationship in (3.22) to obtain

$$\mathcal{D}_i^{(1)} = M_1 M_2^{-1} \mathcal{D}_i^{(2)} \tag{3.24}$$

as required. Finally, the transitivity property required of $\sim$ is satisfied if the assumptions $s_1 \sim s_2$ and $s_2 \sim s_3$ imply that $s_1 \sim s_3$. By assumption, the relationships $\mathcal{D}^{(2)} = M_2 M_1^{-1} \mathcal{D}^{(1)}$ and $\mathcal{D}^{(3)} = M_3 M_2^{-1} \mathcal{D}^{(2)}$ hold where $\mathcal{D}$ is a placeholder for both the partition decomposition $\mathcal{D}_p$ and the interconnective decomposition $\mathcal{D}_i$. The desired result then follows from the associativity of composing linear maps. In particular, for the partition decompositions we obtain

$$\begin{aligned} \mathcal{D}_p^{(3)} &= M_3 M_2^{-1} \mathcal{D}_p^{(2)} && (3.25) \\ &= M_3 M_2^{-1} M_2 M_1^{-1} \mathcal{D}_p^{(1)} && (3.26) \\ &= M_3 M_1^{-1} \mathcal{D}_p^{(1)} && (3.27) \end{aligned}$$

and similarly for the interconnective decompositions we obtain

$$\begin{aligned}\mathcal{D}_i^{(3)} &= M_3 M_2^{-1} \mathcal{D}_i^{(2)} && (3.28)\\ &= M_3 M_2^{-1} M_2 M_1^{-1} \mathcal{D}_i^{(1)} && (3.29)\\ &= M_3 M_1^{-1} \mathcal{D}_i^{(1)}. && (3.30)\end{aligned}$$

These relationships together imply that $s_1 \sim s_3$, therefore the argument above justifies the assertion that the interconnective relation $\sim$ is in fact an equivalence relation between interconnective systems. The following theorem summarizes this result.

Theorem 3.2.1 (Interconnective equivalence relation)**.** *The interconnective relation $\sim$ as described in Definition 3.2.2 forms an equivalence relation between fixed-coordinate interconnective descriptions of a system.*

Equivalence relations in general mathematical settings provide a sound mechanism by which like objects may be discerned using structure derived from the objects themselves. Having established an interconnective equivalence relation, we are equipped with several additional tools beyond unambiguously determining whether or not the behavioral models of two systems are equivalent in the sense of (3.21) and (3.22). For example, we shall make substantial use of the equivalence classes associated with $\sim$ in the remainder of the thesis. Specifically, if s denotes an interconnective system with fixed-coordinate representation $\mathcal{R}_{fc}^{(s)} = (\mathcal{V},\ \mathcal{D}_p^{(s)},\ \mathcal{D}_i^{(s)},\ M_s)$, then the class of interconnective systems equivalent to s with respect to $\sim$, denoted using the standard notation $[s]$, is defined by

$$[s] \ \triangleq\ \left\{\left(\mathcal{V},\ M M_s^{-1} \mathcal{D}_p^{(s)},\ M M_s^{-1} \mathcal{D}_i^{(s)},\ M\right) : M M^{-1} = I\right\}. \qquad (3.31)$$

Section 3.4 in particular focuses on a class of interconnective systems for which the terminal vectors comprising the behavior of any system in the class satisfy a quadratic conservation principle. In this context, we will link to another standard tool provided by having established an equivalence relation, specifically the concept of class representatives.

3.3 | Implementing interconnective systems

Behavioral models such as the interconnective description of a signal processing system provide a higher level of abstraction than traditionally provided by algorithmic or functional models. In the coming chapters, we derive results primarily using behaviors while freely referencing implementations and without specifying the details associated with generating them. The methods developed in this section, preliminaries to which appear in [14, 17], justify this practice. Namely, the processing instructions associated with an implementation are obtainable from a system's interconnective description using the methods described next.

Depending on context, an implementation will refer to one of two kinds of algorithms. The first corresponds to an organization of a system's interconnective description into a synchronous processing loop and the second corresponds to an organization into an asynchronous processing system. For either type, the viewpoint we adopt in this thesis is that the key issues in generating the processing instructions are essentially the same. These issues are analogs to the issues found in designing compilers to convert declarative programming models into imperative ones [51]. Selecting the kind of algorithm in practice may involve understanding how synchronization affects the processing task at hand and the capabilities and overhead associated with synchronizing the available processing resources. For example, either can be used in the context of solving fixed-point problems formulated as solving for elements in the behavior of an associated signal processing system.

We use the following general strategy to generate algorithms from an interconnective description of a signal processing system:

(i) determine a coordinate matrix and input-output configuration of the terminal variables so that each constitutive module and the interconnecting network have well-defined functional realizations

(ii) obtain these functional realizations,

(iii) insert state or memory into the system as needed,

(iv) determine a protocol for exchanging state.

Chapter 6 builds upon this strategy while using the methods developed in this section to efficiently organize interconnective systems onto decentralized and multicore computing platforms by assigning constitutive modules to independent processors and distributing the interconnecting network accordingly.

3.3.1 | Realizing interconnective systems

The general strategy for generating algorithms from a system's interconnective description hinges on the ability of functional realizations of the processing nodes in the system's interconnective graph to be obtained. In this subsection, the goal is specifically to obtain these functions, i.e. to obtain a set of K functions f_k whose behaviors $\mathcal{B}_{f_k}$ equal the constitutive modules behaviors $\mathcal{F}_k$ and a set of L matrices A_l whose behaviors $\mathcal{B}_{A_l}$ equal the interconnect subspaces W_l. In the two subsections that follow, these functions and matrices are used to design a variety of synchronous and asynchronous algorithms.

A key preliminary step to generating these functions and matrices involves configuring the terminal variables into inputs and outputs of the processing nodes. Without loss of generality, we define this configuration with reference to the interconnecting network to which the constitutive modules are connected. By symmetry, this is equivalent to selecting the opposite configuration with reference to the constitutive modules.

Consistent with [17, 18], an *input-output configuration* is formally defined as a collection of L permutation matrices P_l that encode the configuration by partitioning each interconnection vector $\mathbf{v}_l^{(LI)}$ into two subvectors, denoted $\mathbf{v}_l^{(i)}$ and $\mathbf{v}_l^{(o)}$, according to

$$\begin{bmatrix} \mathbf{v}_l^{(i)} \\ \mathbf{v}_l^{(o)} \end{bmatrix} = P_l \mathbf{v}_l^{(LI)}, \qquad l = 1, \dots L \tag{3.32}$$

where P_l denotes the $N_l^{(LI)} \times N_l^{(LI)}$ permutation matrix associated with the l-th interconnect W_l, and $\mathbf{v}_l^{(i)}$ and $\mathbf{v}_l^{(o)}$ respectively denote the terminal variables that are assigned to be inputs to and outputs from W_l. The length of each subvector $\mathbf{v}_l^{(i)}$ and $\mathbf{v}_l^{(o)}$ is respectively denoted by $N_l^{(i)}$ and $N_l^{(o)}$ and collectively these lengths satisfy

$$N_l^{(LI)} = N_l^{(i)} + N_l^{(o)}, \quad l = 1, \dots L. \tag{3.33}$$

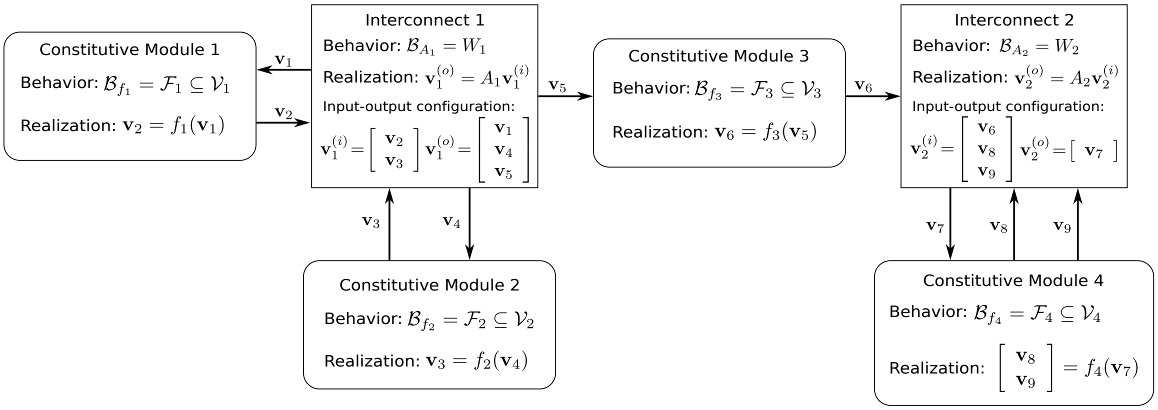

Constitutive Module 1
Behavior: $\mathcal{B}_{f_1} = \mathcal{F}_1 \subseteq \mathcal{V}_1$
Realization: $\mathbf{v}_2 = f_1(\mathbf{v}_1)$
$\mathbf{v}_1$
$\mathbf{v}_2$
Interconnect 1
Behavior: $\mathcal{B}_{A_1} = W_1$
Realization: $\mathbf{v}_1^{(o)} = A_1\mathbf{v}_1^{(i)}$
Input-output configuration:
$\mathbf{v}_1^{(i)} = \begin{bmatrix} \mathbf{v}_2 \\ \mathbf{v}_3 \end{bmatrix}$ $\mathbf{v}_1^{(o)} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_4 \\ \mathbf{v}_5 \end{bmatrix}$
$\mathbf{v}_5$
Constitutive Module 3
Behavior: $\mathcal{B}_{f_3} = \mathcal{F}_3 \subseteq \mathcal{V}_3$
Realization: $\mathbf{v}_6 = f_3(\mathbf{v}_5)$
$\mathbf{v}_6$
Interconnect 2
Behavior: $\mathcal{B}_{A_2} = W_2$
Realization: $\mathbf{v}_2^{(o)} = A_2\mathbf{v}_2^{(i)}$
Input-output configuration:
$\mathbf{v}_2^{(i)} = \begin{bmatrix} \mathbf{v}_6 \\ \mathbf{v}_8 \\ \mathbf{v}_9 \end{bmatrix}$ $\mathbf{v}_2^{(o)} = \begin{bmatrix} \mathbf{v}_7 \end{bmatrix}$
$\mathbf{v}_3$
$\mathbf{v}_4$
$\mathbf{v}_7$
$\mathbf{v}_8$
$\mathbf{v}_9$
Constitutive Module 2
Behavior: $\mathcal{B}_{f_2} = \mathcal{F}_2 \subseteq \mathcal{V}_2$
Realization: $\mathbf{v}_3 = f_2(\mathbf{v}_4)$
Constitutive Module 4
Behavior: $\mathcal{B}_{f_4} = \mathcal{F}_4 \subseteq \mathcal{V}_4$
Realization: $\begin{bmatrix} \mathbf{v}_8 \\ \mathbf{v}_9 \end{bmatrix} = f_4(\mathbf{v}_7)$

Figure 3-3: The interconnective graph associated with a decentralized behavioral model of the signal processing system in Figure 3-2 illustrating the notational conventions used to describe realized systems graphically. The input-output configuration in the figure is consistent with (3.34) and (3.35).

Continuing the graphical definition of an interconnective description of a signal processing system, Figure 3-3 illustrates the system in Figure 3-2 with functional realizations of the constitutive modules and interconnects consistent with the input-output configuration

$$\left\{ \begin{array}{ll} \text{Inputs:} & \mathbf{v}_2, \mathbf{v}_3 \\ \text{Outputs:} & \mathbf{v}_1, \mathbf{v}_4, \mathbf{v}_5 \end{array} \right. \quad \Longleftrightarrow \quad P_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.34}$$

and

$$\left\{ \begin{array}{ll} \text{Inputs:} & \mathbf{v}_6, \mathbf{v}_8, \mathbf{v}_9 \\ \text{Outputs:} & \mathbf{v}_7 \end{array} \right. \quad \Longleftrightarrow \quad P_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \tag{3.35}$$

More generally, to describe an input-output configuration graphically, the bidirectional arrows are replaced with directed arrows where the arrow heads indicate the direction of functional dependencies amongst the terminal variables. The example structure in Figure 3-3 additionally serves to illustrate that realized interconnective graphs can be used to recover behavioral interconnective graphs in a straightforward way.

Let s denote a signal processing system whose interconnective description uses the coordinate matrix M and input-output configuration P. A functional realization of s formally refers to having access to a collection of generally nonlinear functions f_k and matrices A_l, with respective behaviors $\mathcal{F}_k$ and W_l, where the functions and matrices are mutually consistent with the input-output configuration P. The general strategy we follow to generate such a realization is summarized by:

(i) If a realization of each constitutive module $\mathcal{F}_k$ exists and is consistent with P, determine the realizations f_k. If any of the realizations do not exist, change coordinate matrices and start over.

(ii) If a realization of each interconnect subspace W_l exists and is consistent with P, determine the realizations A_l. If any of the realizations do not exist, change coordinate matrices and start over.

All possible changes to the input-output configuration P can be accounted for by appropriately modifying the coordinate matrix M, see Section 3.5.1 for a complete description of this fact. Indeed, changing coordinate matrices corresponds to selecting different systems from the equivalence class $[s]$, as was discussed in Section 3.2.3. This strategy may only make sense in the context of certain processing tasks. For example, this strategy is used in Chapter 4 where fixed-point problems are related to interconnective systems so that identifying a fixed-point corresponding to the problem described by s effectively identifies a fixed-point of all problems described by the group of systems $[s]$. We proceed assuming an appropriate coordinate matrix has been selected so that the realizations f_k of the constitutive modules are available. The process of realizing the constitutive modules, however, may result in interconnects whose descriptions contain implicit constraints. For example, if an interconnect is initially described using a linear, memoryless signal-flow graph, then the interconnect after changing coordinate maps may possess delay-free loops. In light of this observation, we focus on realizing behavioral models of interconnects as matrices in the remainder of this subsection. An example illustrating the use of the procedure developed next in the context of eliminating linear delay-free loops in allpass-warped filtering systems

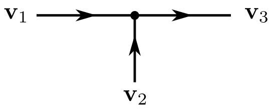
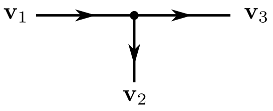
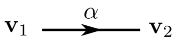
Element type	Summation node	Distribution node	Constant-coefficient multiplier
Signal-flow notation			
Constraint-form encoding	$\mathbf{v}_1 + \mathbf{v}_2 - \mathbf{v}_3 = 0$	$\mathbf{v}_1 - \mathbf{v}_2 = 0$ $\mathbf{v}_1 - \mathbf{v}_3 = 0$	$\alpha\mathbf{v}_1 - \mathbf{v}_2 = 0$

Figure 3-4: Linear signal-flow elements and their corresponding constraint form equations for (a) a summation node, (b) a distribution node, and (c) a constant-coefficient multiplier.

is presented in Section 3.6.1.

To produce functional realizations of the interconnect subspaces W_l as matrices A_l, we apply straightforward manipulations to certain characterizations of the finite-dimensional vector spaces W_l so that the matrices A_l naturally emerge. To this end, let B_l denote a matrix whose null space equals the subspace W_l. This condition, $\text{null}(B_l) = W_l$, is written more explicitly as

$$B_l \mathbf{v}_l^{(LI)} \quad = \quad \mathbf{0}, \qquad \mathbf{v}_l^{(LI)} \in W_l. \tag{3.36}$$

Procedurally, B_l can be populated row by row as follows: for each linear and memoryless constraint involving the terminal variables in $\mathbf{v}_l^{(LI)}$, append one or more rows to B_l where the coefficients in the row(s) are the coefficients of the constraints written in *constraint-form*, i.e. as homogeneous linear equations. Figure 3-4 depicts the constraint-form encoding of the signal-flow elements used in linear, memoryless signal-flow graphs. The decomposition of the interconnecting network W via direct products in (3.5) ensures that the only coupling between terminal variables in $\mathbf{v}_i^{(LI)}$ and $\mathbf{v}_j^{(LI)}$ for $i \neq j$ is through a constitutive module. Therefore, once the procedure above is repeated for all relevant constraints, the resulting matrix B_l satisfies (3.36) as desired.

Next, a second matrix C_l is generated from B_l such that it satisfies two key properties: (i) the rows of C_l are ordered such that a row partitioning yields an upper block $C_l^{(i)}$ corresponding to the $N_l^{(i)}$ input terminal variables $\mathbf{v}_l^{(i)}$ and a lower block $C_l^{(o)}$ corresponding to the $N_l^{(o)}$ output terminal variables $\mathbf{v}_l^{(o)}$, and (ii) the range of C_l equals the subspace $P_l W_l$.

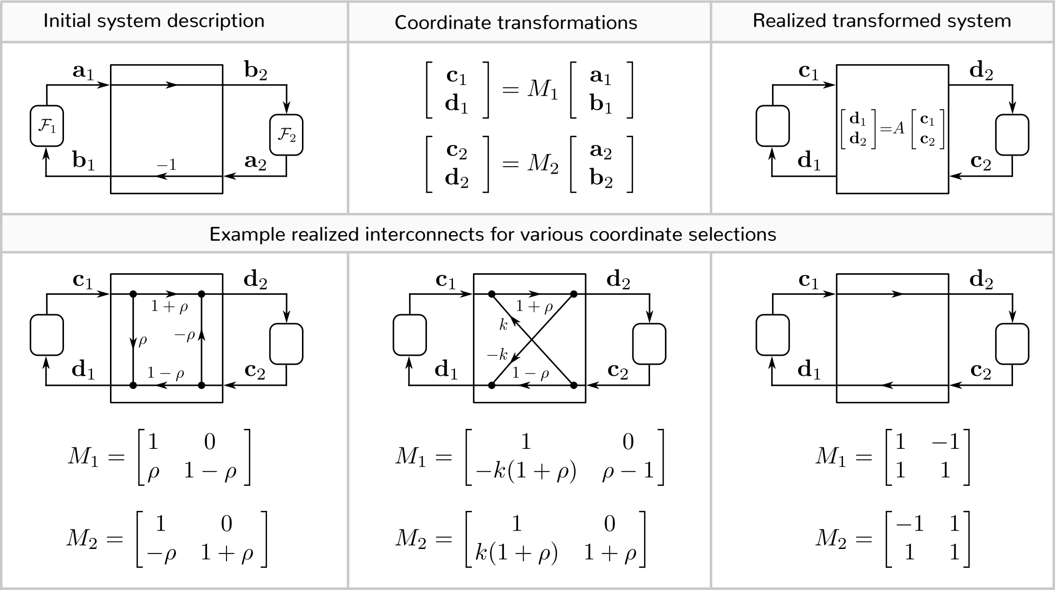

Initial system description
Coordinate transformations
Realized transformed system
$\mathbf{a}_1$
$\mathbf{b}_2$
$\mathcal{F}_1$
$\mathcal{F}_2$
$\mathbf{b}_1$
-1
$\mathbf{a}_2$
$\begin{bmatrix} \mathbf{c}_1 \\ \mathbf{d}_1 \end{bmatrix} = M_1 \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{b}_1 \end{bmatrix}$
$\begin{bmatrix} \mathbf{c}_2 \\ \mathbf{d}_2 \end{bmatrix} = M_2 \begin{bmatrix} \mathbf{a}_2 \\ \mathbf{b}_2 \end{bmatrix}$
$\mathbf{c}_1$
$\mathbf{d}_2$
$\begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{bmatrix} = A \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix}$
$\mathbf{d}_1$
$\mathbf{c}_2$
Example realized interconnects for various coordinate selections
$\mathbf{c}_1$
$\mathbf{d}_2$
$1+\rho$
ρ
$-\rho$
$1-\rho$
$\mathbf{d}_1$
$\mathbf{c}_2$
$\mathbf{c}_1$
$\mathbf{d}_2$
$1+\rho$
k
$-k$
$1-\rho$
$\mathbf{d}_1$
$\mathbf{c}_2$
$\mathbf{c}_1$
$\mathbf{d}_2$
$\mathbf{d}_1$
$\mathbf{c}_2$
$M_1 = \begin{bmatrix} 1 & 0 \\ \rho & 1-\rho \end{bmatrix}$
$M_2 = \begin{bmatrix} 1 & 0 \\ -\rho & 1+\rho \end{bmatrix}$
$M_1 = \begin{bmatrix} 1 & 0 \\ -k(1+\rho) & \rho-1 \end{bmatrix}$
$M_2 = \begin{bmatrix} 1 & 0 \\ k(1+\rho) & 1+\rho \end{bmatrix}$
$M_1 = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$
$M_2 = \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix}$

Figure 3-5: An illustration of three realized interconnects for different coordinate matrix selections and the same input-output configuration. The constitutive modules remain unspecified throughout.

Procedurally, C_l can be assembled by concatenating the singular vectors of B_l associated with zero-valued singular values and then permuting the rows to be consistent with the input-output configuration P_l for W_l. This procedure is summarized according to

$$P_l \text{null}\left(B_l\right) = \text{range}\left(\begin{bmatrix} C_l^{(i)} \\ C_l^{(o)} \end{bmatrix}\right). \tag{3.37}$$

Comparing (3.37) to (2.5), we immediately obtain that the desired matrix A_l, when it exists, is given by

$$A_l = C_l^{(o)} \left(C_l^{(i)}\right)^{-1}. \tag{3.38}$$

The matrix A_l will exist if and only if the submatrix $C_l^{(i)}$ is invertible. Although a straightforward observation, the consequence of this fact is important to selecting well-defined input-output configurations and realizing interconnective graphs. In particular, the interconnect W_l constrains a total of $N_l^{(LI)}$ terminal variables, and from (3.33) we know that $N^{(LI)} = N_l^{(i)} + N_l^{(o)}$. The consequence of (3.38) is that $N_l^{(i)}$ must equal the dimension of W_l. To see this, recall that multiplying a first matrix on the right by a second invertible matrix does not change the dimension of the subspace generated by the rows of the first matrix [34], so the dimension of the l-th interconnect equals the rank of C_l which must be equal to the number of input variables to W_l for $C_l^{(i)}$ to be invertible.

To conclude this subsection, we present an example of the procedure above for generating functional realizations of an interconnecting network after performing coordinate transforms to a vector space description of the interconnect for a simple signal processing system. This example also illustrate the breadth of interconnective structures that can be achieved in the same interconnective equivalence class. Consider the interconnective graph on the top left of Figure 3-5. The interconnective description of this system is given by $(\mathcal{V}, \mathcal{D}_p, \mathcal{D}_i, M)$ where the terminal space $\mathcal{V}$ is four dimensional, the coordinate matrix M is assumed to be the identity matrix without loss of generality, and the partition and interconnective decompositions are of the form

$$\mathcal{D}_p = \{\mathcal{V}_1, \mathcal{V}_2\} \tag{3.39}$$

$$\mathcal{D}_i = \{W, \mathcal{F}_1 \times \mathcal{F}_2\} \tag{3.40}$$

where $\dim\mathcal{V}_k = 2$ for $k = 1, 2$. The constitutive modules $\mathcal{F}_1$ and $\mathcal{F}_2$ respectively constraining the terminal subvectors $\mathbf{v}_1^{(CR)} = (\mathbf{a}_1, \mathbf{b}_1)$ and $\mathbf{v}_2^{(CR)} = (\mathbf{a}_2, \mathbf{b}_2)$ remain unspecified throughout the figure. The interconnecting network W is described by

$$W = \left\{ \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{b}_1 \\ \mathbf{a}_2 \\ \mathbf{b}_2 \end{bmatrix} \in \mathcal{V} \colon \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{b}_1 \\ \mathbf{a}_2 \\ \mathbf{b}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\}, \tag{3.41}$$

$$= \operatorname{range}\left(\begin{bmatrix} 1 & 0 \\ 0 & -1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \right) \tag{3.42}$$

where the description in (3.41) illustrates the use of a constraint-form encoding. The input-output configuration depicted in the figure corresponds selecting $\mathbf{v}_1^{(i)} = (\mathbf{a}_1, \mathbf{a}_2)$ and $\mathbf{v}_1^{(o)} = (\mathbf{b}_1, \mathbf{b}_2)$. The graph of the transformed system in the top right of Figure 3-5 depicts the general form of the realized interconnect as a matrix A consistent with an input-output configuration with inputs $(\mathbf{c}_1, \mathbf{c}_2)$ and outputs $(\mathbf{d}_1, \mathbf{d}_2)$. The coordinate transforms for a $\mathcal{D}_p$-invariant coordinate change are provided in the middle. The structures along the bottom correspond to common signal-flow interconnecting networks found in signal processing applications and are obtained by applying a coordinate transform to (3.42) using the coordinate submatrices M_1 and M_2 listed underneath.

3.3.2 | Synchronous processing loops

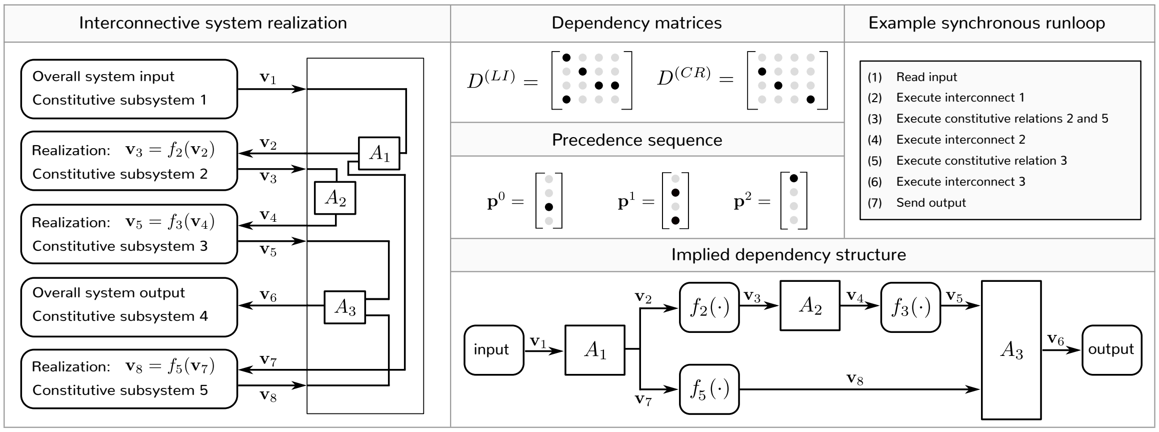
Conventional methods of organizing signal processing systems into ordered sequences of processing instructions have primarily focused on scheduling input-output computations to form overall iterations. For a broad class of signal-flow structures, schedules can be identified by using callback functions in data pull or data push arrangements or through similar graph-based techniques, as have been discussed in [21, 52, 53]. In these approaches, issues of computability directly relate to the existence of delay-free loops or cycles within various directed graphs derived from the model or system at hand. Consistent with the discussion in Section 2.2, methods including those in [26–31] handle some special cases where these techniques either fail or do not apply by breaking certain delay-free loops via inserting memory modules or applying the implicit function theorem as part of the iteration. The techniques presented in this subsection specifically extend the implementation procedures developed in [17] to incorporate the additional flexibility provided by having a coordinate matrix as part of the interconnective description of a system.

The procedure discussed in the previous subsection to produce functional realizations of an interconnective graph handled issues related to computability by appropriately selecting the coordinate matrix M and input-output configuration P. Jointly selecting M and P so that the functions f_k are well-defined is equivalent to eliminating implicit constraints from the constitutive modules. The method developed to realize an interconnect subspace as a matrix then dealt with implicit linear constraints through the constraint form encoding of the subspace. A sufficient condition for scheduling synchronous processing loops, established in [21, 52, 53] and translated into the present context, is the absence of delay-free paths starting from any terminal variable and ending with itself, where the edges of the graph are only traversed in directions that are consistent with the input-output configuration. Therefore, from the definition of a functional realization, any system that would be scheduled using [21, 52, 53] can also be scheduled using the procedure presented next.

To generate globally synchronous algorithms from a functional realization of an interconnective graph, we focus on two key steps. The first step involves inserting state or memory into the system and the second involves determining a protocol for state exchange, thereby introducing a semblance of evolution into the system. For the purpose of designing globally coordinated algorithms in the form of synchronous processing loops, i.e. ordered sets of processing instructions to be repeatedly executed, we take the constitutive modules to already contain any memory required to realize their behaviors. Referring to the interconnective graph in Figure 3-3, designing a synchronous iteration corresponds to identifying an execution order for the functions f_k that in turn implies how the terminal variables are to be processed through subsets of the interconnecting networks A_l.

A processing loop is designed in reverse order of its execution by properly satisfying the functional dependencies of the constitutive functions f_k as determined by the input-output configuration, system memory, and graph topology. To do this, let A denote the collective realization of W and define $D^{(LI)}$ and $D^{(CR)}$ as binary dependency matrices populated as follows. The entry $D^{(LI)}_{i,j} = 1$ if the i-th input to A interacts with the j-th output of A, i.e. $D^{(LI)}_{i,j}$ takes value one if $A_{i,j}$ is non-zero and is zero otherwise. Similarly, the entry $D^{(CR)}_{i,j} = 1$ if, within the same clock cycle, the i-th output of A is used to produce the j-th input to A as determined by the constitutive functions and is zero otherwise. A schedule will be identified if the matrix $D^{(LI)}D^{(CR)}$ is nilpotent, which can be certified using the fact a matrix D is nilpotent if $\text{trace}\left(D^k\right) = 0$ for any k between 1 and the dimension of D. The smallest value of k for which $D^k = 0$ is called the degree of the matrix.

Let $\mathbf{p}^n$ denote a sequence of length $N_1^{(o)} + \cdots + N_L^{(o)}$ precedence vectors where the i-th entry of each precedent vector corresponds to the i-th interconnect output variable, i.e. the i-th entry of $\mathbf{p}$ corresponds to the i-th output of A. A schedule is identified by running the


Interconnective system realization

Dependency matrices

Example synchronous runloop

(1) Read input
(2) Execute interconnect 1
(3) Execute constitutive relations 2 and 5
(4) Execute interconnect 2
(5) Execute constitutive relation 3
(6) Execute interconnect 3
(7) Send output

Precedence sequence

Implied dependency structure

Figure 3-6: A demonstration of the procedure for generating synchronous processing loops from interconnective graphs with functional realizations of the graph's constitutive modules and interconnects.

forward iteration

$$\mathbf{p}^n \quad = \quad \left(D^{(LI)}D^{(CR)}\right)^T \mathbf{p}^{n-1}, \qquad n = 1, \ldots, m \tag{3.43}$$

where m is the degree of $D^{(LI)}D^{(CR)}$ and the iteration is initialized by setting $\mathbf{p}_j^0 = 1$ if the j-th output of A is an overall system output. The execution order of the functions f_k is determined from the precedence tree described by the precedence sequence. Specifically, the processing loop is assembled by ordering the functions f_k in reverse of the order that first non-zero value appears in the associated row of $\mathbf{p}^n$. The processing loop is then completed by inserting the relevant rows of A between the ordered function calls. This procedure is demonstrated next.

Figure 3-6 demonstrates the scheduling procedure described above applied to the example interconnective graph depicted on the left. The constitutive modules and interconnects are assumed to have the functional realizations depicted in the figure. The sparsity pattern of the binary dependency matrices $D^{(LI)}$ and $D^{(CR)}$ indicates the functional dependencies between the terminal variables on a given clock cycle. The sparsity pattern of the precedence sequence $\mathbf{p}^n$, for $n = 0, 1, 2$, is also provided where the coordinates of $\mathbf{p}^n$ correspond to the terminal variables ($\mathbf{v}_2$, $\mathbf{v}_4$, $\mathbf{v}_6$, $\mathbf{v}_7$), which are the terminal variables that are outputs of the interconnecting network. The initial precedent vector is $\mathbf{p}^0 = (0, 0, 1, 0)$ indicating that the terminal variable $\mathbf{v}_6$ is the only overall system output. From the sparsity pattern of the displayed precedence sequence, the synchronous processing loop in the top right is generated by ordering the constitutive functions and inserting the relevant interconnects in between. An equivalent visualization of the precedent sequence is provided on the bottom and highlights the dependency structure amongst the system's terminal variables. As is evident in either the precedence sequence of equivalent structure, certain of the constitutive modules can be implemented in parallel, thereby allowing distributed processing loops to be designed based upon the joint separability of the constitutive modules and interconnects.

We briefly comment on how the processing loop design procedure can be augmented to account for multirate relations appearing in the description of an interconnective system. To handle this, each terminal variable $\mathbf{v}_k$ is associated with a clock rate $\mathbf{r}_k$. Using the previously defined dependency matrices $D^{(LI)}$ and $D^{(CR)}$, a system of nonlinear equations is assembled as follows: for each pair of terminal variables $\mathbf{v}_i$ and $\mathbf{v}_j$ that interact during the same clock cycle according to $D^{(LI)}$ or $D^{(CR)}$, collect a constraint of the form

$$\mathbf{r}_i \mathbf{r}_j^{-1} = \mu_{i,j} \tag{3.44}$$

where $\mu_{i,j}$ is the rate relation between $\mathbf{v}_i$ and $\mathbf{v}_j$. In particular, (3.44) is interpreted to mean that $\mathbf{v}_i$ operates at $\mu_{i,j}$ times the rate of $\mathbf{v}_j$. Substituting the transformed variables $\hat{\mathbf{r}}_k = \log \mathbf{r}_k$, we reduce (3.44) to a linear system of equations

$$\hat{\mathbf{r}}_i - \hat{\mathbf{r}}_j = \log \mu_{i,j}. \tag{3.45}$$

Any non-trivial solution to (3.45) uniquely specifies a rate vector $\mathbf{r}$. Sufficient conditions for designing a valid processing loop' according to the rates $\mathbf{r}$ are twofold. First, the right-hand side of (3.45) must lie in the range of the linear system. Second, the rates obtained must all be rational numbers, i.e. the vector $\mathbf{r}$ needs to be converted to a vector of integers to assign valid rates to the processing nodes. If the rate conversion modules in an interconnective system description are all upsampling or downsampling by integer amounts then the second condition will always be satisfied. The rates obtained in this way determine those iterations of the processing loop where particular functions are to be executed.

3.3.3 | Asynchronous processing systems

The processing goal behind a variety of signal and data processing problems is to determine a solution to a generally nonlinear system of equations where any solution is considered as good as any other. Problems of this type, sometimes referred to as constraint satisfaction problems, are predominately solved using iterative techniques after first reformulating the problem into fixed-point form. For example, many algorithms for solving convex optimization problems can be viewed as numerical fixed-point solvers where the fixed-point problem corresponds to recasting the KKT conditions in Section 2.5.2 into fixed-point form. Drawing upon the interconnective framework developed in this chapter, solving a broad class of fixed-point problems can be reduced to identifying a terminal vector $\mathbf{v}^\star$ in the behavior of an appropriately defined interconnective system s. Moreover, solving for this terminal vector simultaneously identifies a terminal vector in the behavior of every system in $[s]$, thus the fixed-point problems corresponding to the interconnective systems in $[s]$ are solved too.

A potential strategy for solving the fixed-point problem associated with an interconnective system s is to build a physical system to model the constraints of any system in $[s]$, and to use the steady-state behavior of the physical system provides a solution to within the appropriate coordinate transform. This strategy is consistent with the approach to solving network programming problems in [7] where analog circuits played the role of the physical system. Motivated in part by this, we formulate an interconnective description of fixed-point problems using operator methods in this subsection, and use the asynchronous implementation of the system to describe the corresponding asynchronous algorithm. This setup is in anticipation of solving a class of fixed-point problems described in detail in the following chapter. This setup can also be viewed as a discrete-time approximation of the strategy above where careful synchronization between the constitutive modules and interconnects is not required when they are implemented on different processors.

Returning to the procedure outlined at the beginning of this section for generating algorithms from interconnective structures, an *asynchronous processing system* is defined as any functional realization of the structure with memory inserted so that no delay-free loops remain. Every asynchronous processing system is associated with a *system operator* defined as the generally nonlinear mapping of the terminal variables at the outputs of the delay modules to the inputs of the same delay modules in a manner consistent with the constraints of the system. For example, a valid memory configuration for the interconnective structure in Figure 3-3 corresponds to placing delay modules along the edges directed toward constitutive modules so that the system operator maps the terminal subvector $(\mathbf{v}_1^{(o)}, \ldots, \mathbf{v}_L^{(o)})$ from the output of the delays through the modules and interconnects to the input of the same delays. It is straightforward to verify that system operators can be formed by designing a synchronous processing loop where the delay modules are used to define overall system inputs and outputs. In Chapter 4, system operator properties are used to state sufficient conditions for which an asynchronous processing system solves the associated fixed-point problem by simply letting the system settle. These asynchronous algorithms, of course, do not explicitly require the system operator associated with a problem to be formed.

To run an asynchronous processing system, an asynchronous processing protocol is defined for exchanging state by defining each delay module as an asynchronous delay element, i.e. a randomly triggered sample-and-hold element. Specifically, for the purpose of analysis, the behavior of an asynchronous delay element is modeled using a discrete-time sample-and-hold state module triggered by an independent Bernoulli process. For example, a scalar asynchronous delay element with output $\mathbf{y}^n$ and input $\mathbf{x}^n$ operates according to

$$\mathbf{y}^n = \begin{cases} \mathbf{x}^n, & \text{with probability } p \\ \mathbf{y}^{n-1}, & \text{with probability } 1-p \end{cases} \tag{3.46}$$

for each n independently. The associated protocol with this state exchange behavior for an asynchronous processing system is stated formally in the following definition.

Definition 3.3.1 (Asynchronous processing protocol)**.** *The protocol for an asynchronous processing system with associated system operator $T\colon \mathbb{R}^N \to \mathbb{R}^N$, and starting from initial state $\mathbf{v}^0 \in \mathbb{R}^N$, corresponds to the state sequence $\{\mathbf{v}^n \in \mathbb{R}^N : n \in \mathbb{N}_0\}$ generated according to*

$$\mathbf{v}^n = D^{(p)} T\left(\mathbf{v}^{n-1}\right) + \left(I_N - D^{(p)}\right)\mathbf{v}^{n-1}, \quad n \in \mathbb{N} \tag{3.47}$$

where $D^{(p)}$ is an $N \times N$, stochastic, binary, diagonal matrix whose diagonal elements are i.i.d. Bernoulli and independent of n, taking values $D^{(p)}_{i,i} = 1$ with probability p and $D^{(p)}_{i,i} = 0$ with probability $1 - p$.

The interpretation of the role played by the stochastic matrix $D^{(p)}$ in (3.47) in the context of a decentralized implementation of an interconnective system is that the entries of the terminal vector $\mathbf{v}$ that get updated correspond to those processors in the network which communicate at that time instant and those entries which do not get updated correspond to those processors which do not communicate. Furthermore, the definition above naturally encompasses synchronous implementations of the processing system by setting $p = 1$. Identifying sufficient conditions on the system operator T so that running an asynchronous processing system according to the protocol above produces a state sequence that tends to an invariant state or fixed-point of T is the central focus of Section 4.3.

3.4 | Strongly-conservative interconnecting networks

A widespread practice in deploying signal processing algorithms is to map a mathematical description of an algorithm into a predefined structure or form whose properties are well-understood and that can be readily executed by off-the-shelf processors. For example, in digital filtering applications where forward and inverse systems are required, it may be convenient to use lattice structures so that coefficient quantization effects in the forward system are easily accounted for by the inverse system [35]. Similarly, in implementing multirate filtering systems, polyphase structures can be used to eliminate the computational burden associated with subcomputations that do not affect the overall system output [54]. In this section, we discuss the role played by the interconnecting network with regard to conservation principles and call attention to two organizations of an interconnecting network that exhibit particular strong conservation principles and then derive the coordinate transforms used to translate between these organizations.

The two forms of an interconnective description defined in this section, respectively referred to as *canonical-form* and *scattering-form*, can be easily identified by their interconnecting networks and conservation principles. In particular, terminal vectors consistent with the interconnecting network of a system organized into canonical-form satisfy quadratic conservation principles similar to the expression

$$\mathbf{v}_1\mathbf{v}_2 + \mathbf{v}_3\mathbf{v}_4 + \mathbf{v}_5\mathbf{v}_6 \quad = \quad 0 \tag{3.48}$$

while terminal vectors consistent with the interconnecting network of a system organized into scattering-form satisfy quadratic conservation principles similar to the expression

$$\mathbf{v}_1^2 - \mathbf{v}_2^2 + \mathbf{v}_3^2 - \mathbf{v}_4^2 + \mathbf{v}_5^2 - \mathbf{v}_6^2 \quad = \quad 0. \tag{3.49}$$

The coordinate transforms relating these two forms will be referred to as *scattering coordinate transforms*. The chosen nomenclature reflects the fact that the quadratic conservation principle in (3.48) is reminiscent of the conservation principle satisfied by current and voltage terminal variables in electrical networks while the quadratic conservation principle in (3.49) corresponds to the conservation principle satisfied by the scattering parameter representation of an electrical network [55]. These two descriptions or forms set the stage for the following two chapters where, in Chapter 5, connections between canonical-form systems and optimality conditions for solving certain optimization problems are established, and this connection can be taken advantage of by the scattering-form description of the system, which is shown in Chapter 4 to be robust to asynchronous processing issues for a wide variety of data processing problems. Before defining these forms, we first discuss the interplay between conservation principles, coordinate transforms, and interconnective equivalence classes.

3.4.1 | Conservation principles for equivalent interconnecting networks

Let s_k denote a signal processing system for each value of k whose fixed-coordinate interconnective description $\mathcal{R}_{fc}^{(k)}$ is given by

$$\mathcal{R}_{fc}^{(k)} \quad = \quad (\mathcal{V},\, \mathcal{D}_p^{(k)},\, \mathcal{D}_i^{(k)},\, M_k) \tag{3.50}$$

where the terminal space $\mathcal{V}$ is even-dimensional and defined over a real field. Using the conservation framework originally developed in [11] and summarized in Section 2.4, let $\mathcal{O}^{(k)}$

denote an organization of $(\mathcal{V}, \langle \cdot, \cdot \rangle_{\mathcal{V}})$ where $\langle \cdot, \cdot \rangle_{\mathcal{V}}$ denotes the standard inner product on $\mathcal{V}$ and the organization $\mathcal{O}^{(k)}$ is of the form

$$\mathcal{O}^{(k)} \quad = \quad (C_k, \mathcal{D}_p^{(k)}, \mathcal{D}_c^{(k)}) \tag{3.51}$$

where the connection between the partition decompositions $\mathcal{D}_p^{(k)}$ in (3.50) and (3.51) was established in the discussion surrounding (3.6). Drawing upon the definition of a conservation principle in Section 2.4.2, the behavioral model of the interconnecting network described as the vector space $W^{(k)}$ is said to be conservative with regard to the organization $\mathcal{O}^{(k)}$ if

$$\langle C_k \mathbf{v}, \mathbf{v} \rangle_{\mathcal{V}} \quad = \quad 0, \qquad \mathbf{v} \in W^{(k)}. \tag{3.52}$$

In the remainder of this subsection, we assume (3.52) holds for $k = 1$.

The central result in this subsection is that every interconnective system $s_k \in [s_1]$ is conservative with regard to the organization $\mathcal{O}^{(k)}$ in (3.51) where the elements composing the organization $\mathcal{O}^{(k)}$ are related to the elements of the organization $\mathcal{O}^{(1)}$ according to

$$\begin{aligned} C_k \quad &= \quad (M_1 M_k^{-1})^T C_1 M_1 M_k^{-1} && (3.53) \\ \mathcal{D}_p^{(k)} \quad &= \quad M_k M_1^{-1} \mathcal{D}_p^{(1)} && (3.54) \\ \mathcal{D}_c^{(k)} \quad &= \quad M_k M_1^{-1} \mathcal{D}_c^{(1)}. && (3.55) \end{aligned}$$

The assertion in (3.53) is justified by application of the identity in (2.24) as follows:

$$\begin{aligned} q(\mathbf{v}) &= \langle C_1 \mathbf{v},\ \mathbf{v} \rangle, && \mathbf{v} \in M_1 M_k^{-1} W_k && (3.56) \\ &= \langle C_1 M_1 M_k^{-1} \mathbf{v},\ M_1 M_k^{-1} \mathbf{v} \rangle, && \mathbf{v} \in W_k && (3.57) \\ &= \langle \underbrace{(M_1 M_k^{-1})^T C_1 M_1 M_k^{-1}}_{C_k} \mathbf{v},\ \mathbf{v} \rangle, && \mathbf{v} \in W_k. && (3.58) \end{aligned}$$

Furthermore, the transformation of the partition decomposition in (3.54) agrees with the transformation required by the assumption $s_1 \sim s_k$ in (3.21). It is straightforward to verify that if the coordinate matrices M_1 and M_k in (3.54) are block diagonal with matching blocks sizes, then any partition decomposition described using direct products as demonstrated by (3.3) will be invariant to the coordinate transform $M_k M_1^{-1}$. To summarize, every interconnective system that can be obtained by applying a coordinate transform to an initial conservative system is itself a conservative system, and the interconnecting network of the initial system being conservative is sufficient for the transformed system to be conservative under any such coordinate transform.

3.4.2 | Canonical-form interconnective systems

In this subsection, we define a *canonical-form* interconnective description to model the subset of strongly conservative signal processing systems whose terminal vectors satisfy conservation principles akin to the quadratic form in (3.48). To do this, we first define the components of a canonical-form interconnective description through skew-symmetry properties of its interconnecting network, and then we associate the description with a canonical-form organization. For brevity in future use of this form, we then express the class of canonical-form systems using an interconnective graph. Properties of canonical-form systems are used in Chapter 5 to derive variational interpretations of conservative signal processing systems, and this analysis allows for the straightforward modular design of optimization algorithms realized as conservative signal processing systems.

Let $s^{(c)}$ denote a signal processing system whose interconnective description $\mathcal{R}_{fc}^{(c)}$ is defined as being in canonical form and is given by

$$\mathcal{R}_{fc}^{(c)} \triangleq (\mathcal{V}^{(c)},\ \mathcal{D}_p^{(c)},\ \mathcal{D}_i^{(c)},\ M_c) \tag{3.59}$$

where the elements composing (3.59) take the general form

$$\mathcal{V}^{(c)} = \mathbb{R}^{2(R^{(c)}+L^{(c)})} \tag{3.60}$$

$$\mathcal{D}_p^{(c)} = \mathrm{span}(\mathbf{e}^{(1)},\ldots,\mathbf{e}^{(2(R^{(c)}+L^{(c)}))}) \tag{3.61}$$

$$\mathcal{D}_i^{(c)} = \{W^{(c)},\ \mathcal{F}^{(c)}\} \tag{3.62}$$

and where the coordinate map M_c is a $2(R^{(c)}+L^{(c)})\times 2(R^{(c)}+L^{(c)})$-dimensional linear map or matrix. Referring to the interconnective decomposition $\mathcal{D}_i^{(c)}$ in (3.62), the constitutive module $\mathcal{F}^{(c)}$ is an arbitrary memoryless relation defined on the entire set of terminal variables, and the interconnecting network $W^{(c)}$ is specifically a subspace of $\mathcal{V}^{(c)}$ generated by a matrix $A\colon \mathbb{R}^{R^{(c)}} \to \mathbb{R}^{L^{(c)}}$ and takes the form

$$\begin{aligned} W^{(c)} &= \left\{ \mathbf{v} \in \mathcal{V}^{(c)} \colon \mathbf{v} = \begin{bmatrix} \mathbf{a}_1^{(c)} \\ \mathbf{b}_1^{(c)} \\ \mathbf{a}_2^{(c)} \\ \mathbf{b}_2^{(c)} \end{bmatrix}, \begin{bmatrix} A & 0 & -I_{L^{(c)}} & 0 \\ 0 & I_{R^{(c)}} & 0 & A^T \end{bmatrix} \begin{bmatrix} \mathbf{a}_1^{(c)} \\ \mathbf{b}_1^{(c)} \\ \mathbf{a}_2^{(c)} \\ \mathbf{b}_2^{(c)} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \right\} && (3.63) \\ &= \operatorname{range} \left(\begin{bmatrix} I_{R^{(c)}} & 0 \\ 0 & -A^T \\ A & 0 \\ 0 & I_{L^{(c)}} \end{bmatrix} \right). && (3.64) \end{aligned}$$

Note that the description of the vector space $W^{(c)}$ in (3.63) corresponds to a constraint-form encoding of the interconnecting network.

In dealing with equivalence classes in general settings, the concept of a class representative is often used to designate a particular member of each equivalence class that is special in some sense. Formally defining a representative requires an injective mapping to exist from an equivalence class to the representative. In the context of interconnective equivalence classes, the conditions required for an interconnective description to be in canonical form are not sufficient for $s^{(c)}$ to be a representative of $[s^{(c)}]$. Indeed, applying any interconnection invariant coordinate transform from Section 3.5.2 produces a system $s' \in [s^{(c)}]$ that also has a canonical form representation. Also, a representative must be defined on all equivalence classes, including those that do not contain conservative systems.

Moving forward, we equip the terminal space $\mathcal{V}^{(c)}$ in (3.60) with the standard inner product and associate the canonical-form interconnective description of a system with a canonical organization $\mathcal{O}^{(c)}$ defined by

$$\mathcal{O}^{(c)} \triangleq (C_c, \mathcal{D}_p^{(c)}, \mathcal{D}_c^{(c)}) \qquad (3.65)$$

where the connection between the partition decompositions $\mathcal{D}_p^{(k)}$ in (3.61) and (3.65) was established in the discussion surrounding (3.6). The correspondence map and conjugate decomposition in (3.65) take the form

$$C_c = \frac{1}{2}\begin{bmatrix} 0 & I_{R^{(c)}} & 0 & 0 \\ I_{R^{(c)}} & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{L^{(c)}} \\ 0 & 0 & I_{L^{(c)}} & 0 \end{bmatrix} \tag{3.66}$$

$$\mathcal{D}_c^{(c)} = \left\{\text{span}\left(\mathbf{e}^{(1)}, \ldots, \mathbf{e}^{(R^{(c)}+L^{(c)})}\right),\ \text{span}\left(\mathbf{e}^{(R^{(c)}+L^{(c)}+1)}, \ldots, \mathbf{e}^{(2(R^{(c)}+L^{(c)}))}\right)\right\} \tag{3.67}$$

In describing canonical organizations, the associated comparison space is $(\mathbb{R}^{R^{(c)}+L^{(c)}}, \langle\cdot,\cdot\rangle_{\mathbb{R}^{R^{(c)}+L^{(c)}}})$ where the inner product is the standard inner product and the conjugate maps are given by

$$M_A^{(c)} = \sqrt{2}\begin{bmatrix} I_{R^{(c)}} & 0 & 0 & 0 \\ 0 & 0 & I_{L^{(c)}} & 0 \end{bmatrix} \tag{3.68}$$

and

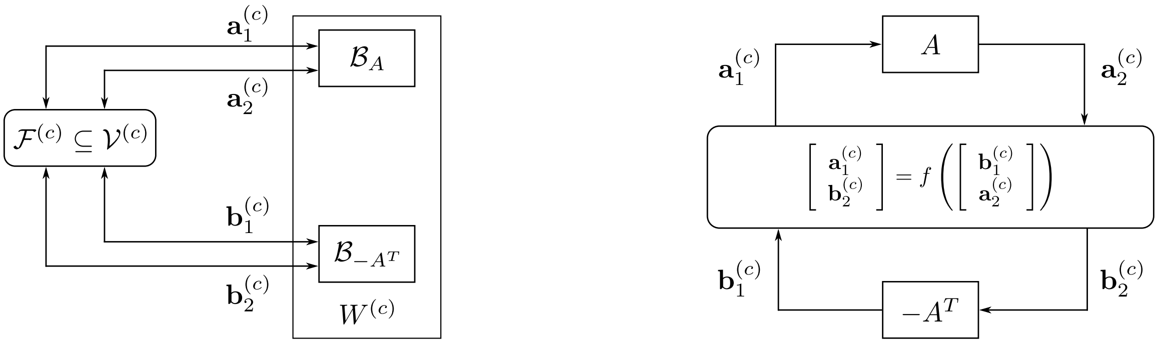
$$M_B^{(c)} = \sqrt{2}\begin{bmatrix} 0 & I_{R^{(c)}} & 0 & 0 \\ 0 & 0 & 0 & I_{L^{(c)}} \end{bmatrix}. \tag{3.69}$$

For these conjugate maps and the partitioning scheme of the terminal vector $\mathbf{v}$ in (3.63), the conservation principle associated with a canonical organization may be written in the comparison space as

$$\left\langle \begin{bmatrix} \mathbf{a}_1^{(c)} \\ \mathbf{a}_2^{(c)} \end{bmatrix}, \begin{bmatrix} \mathbf{b}_1^{(c)} \\ \mathbf{b}_2^{(c)} \end{bmatrix} \right\rangle_{\mathbb{R}^{R^{(c)}+L^{(c)}}} = 0, \qquad (\mathbf{a}_1^{(c)}, \mathbf{b}_1^{(c)}, \mathbf{a}_2^{(c)}, \mathbf{b}_2^{(c)}) \in W^{(c)}, \tag{3.70}$$

which is reminiscent of the form (3.48) where the terminal vector $\mathbf{v}$ has been partitioned into an equal number of $\mathbf{a}$ and $\mathbf{b}$ variables.

Interconnective structures describing signal processing systems in canonical form are illustrated in Figure 3-7 where the structure in (a) depicts a behavioral model represented



(a) General canonical-form structure

(b) Realized canonical-form structure

Figure 3-7: Interconnective descriptions of a signal processing system in canonical form. (a) The interconnective structure corresponding to a generic canonical-form description. (b) A realization of the interconnective structure in (a) corresponding to the input-output configuration in (3.71).

as a graph according to the partitioning of $\mathbf{v}$ in (3.63) and the structure in (b) corresponds to a functional realization of the structure in (a) according to the input-output configuration

$$\begin{cases} \text{Inputs:} & \mathbf{a}_1^{(c)},\ \mathbf{b}_2^{(c)} \\ \text{Outputs:} & \mathbf{b}_1^{(c)},\ \mathbf{a}_2^{(c)} \end{cases} \iff P = \begin{bmatrix} I_{R^{(c)}} & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{L^{(c)}} \\ 0 & I_{R^{(c)}} & 0 & 0 \\ 0 & 0 & I_{L^{(c)}} & 0 \end{bmatrix}. \tag{3.71}$$

The functional realization of the interconnecting network as a matrix that is consistent with the input-output configuration P is readily obtained using the procedure in Section 3.3 or by inspection of the graph and is given by

$$\begin{bmatrix} \mathbf{b}_1^{(c)} \\ \mathbf{a}_2^{(c)} \end{bmatrix} = \begin{bmatrix} 0 & -A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \mathbf{a}_1^{(c)} \\ \mathbf{b}_2^{(c)} \end{bmatrix}. \tag{3.72}$$

In the coming chapters, we will frequently be interested in assembling large-scale signal processing systems by connecting together a number of constitutive modules and interconnects to form a canonical-form system where the partition decomposition is specifically chosen to be maximal. To be consistent with the definition above, the partition decomposition in (3.61) is organized for these systems into a collection of $R^{(c)} + L^{(c)}$ two-dimensional subspaces (using direct sums for notational convenience) according to

$$\begin{aligned}\mathcal{D}_p^{(c)} \quad = \quad & \Big\{\text{span}\left(\mathbf{e}^{(1)}, \mathbf{e}^{(R^{(c)}+1)}\right), \ldots, \text{span}\left(\mathbf{e}^{(R^{(c)})}, \mathbf{e}^{(2R^{(c)})}\right), \\ & \text{span}\left(\mathbf{e}^{(2R^{(c)}+1)}, \mathbf{e}^{(2R^{(c)}+L^{(c)}+1)}\right), \ldots, \text{span}\left(\mathbf{e}^{(2R^{(c)}+L^{(c)})}, \mathbf{e}^{(2(R^{(c)}+L^{(c)}))}\right)\Big\}.\end{aligned} \tag{3.73}$$

The constitutive modules are then each decomposable into individual relations in $\mathbb{R}^2$. When a system in canonical-form utilizes (3.73), we shall refer to the system as being in *maximal* canonical form.

3.4.3 | Scattering-form interconnective systems

In this subsection, we define a *scattering-form* interconnective description to model the subset of strongly conservative signal processing systems whose terminal vectors satisfy conservation principles akin to the quadratic form in (3.49). To do this, we first define the components of a scattering-form interconnective description through orthogonality properties of its interconnecting network, and then we associate the description with a scattering-form organization. For brevity in future use of this form, we then express the class of scattering-form systems using an interconnective graph. Scattering-form systems aid the presentation in Chapter 4 where fixed-point and constraint satisfaction problems are linked to conservative signal processing systems, and implementations of scattering-form systems, referred to as *scattering algorithms*, are shown to possess strong stability and robustness properties when organized into asynchronous processing systems.

Let $s^{(s)}$ denote a signal processing system whose interconnective description $\mathcal{R}_{fc}^{(s)}$ is defined as being in scattering form and is given by

$$\mathcal{R}_{fc}^{(s)} \quad \triangleq \quad (\mathcal{V}^{(s)},\ \mathcal{D}_p^{(s)},\ \mathcal{D}_i^{(s)},\ M_s) \tag{3.74}$$

where the elements composing (3.74) take the general form

$$\mathcal{V}^{(s)} = \mathbb{R}^{2(R^{(s)}+L^{(s)})} \tag{3.75}$$

$$\mathcal{D}_p^{(s)} = \operatorname{span}(\mathbf{e}^{(1)}, \ldots, \mathbf{e}^{(2(R^{(s)}+L^{(s)}))}) \tag{3.76}$$

$$\mathcal{D}_i^{(s)} = \{W^{(s)}, \mathcal{F}^{(s)}\} \tag{3.77}$$

and where the coordinate map M_s is a $2(R^{(s)}+L^{(s)}) \times 2(R^{(s)}+L^{(s)})$-dimensional linear map or matrix. Referring to the interconnective decomposition $\mathcal{D}_i^{(s)}$ in (3.77), the constitutive module $\mathcal{F}^{(s)}$ is an arbitrary memoryless relation defined on the entire set of terminal variables, and the interconnecting network $W^{(s)}$ is specifically a subspace of $\mathcal{V}^{(s)}$ generated by an orthogonal matrix $Q\colon \mathbb{R}^{R^{(s)}+L^{(s)}} \to \mathbb{R}^{R^{(s)}+L^{(s)}}$ and takes the form

$$W^{(s)} = \left\{ \mathbf{v} \in \mathcal{V}^{(s)} : \mathbf{v} = \begin{bmatrix} \mathbf{c}_1^{(s)} \\ \mathbf{d}_1^{(s)} \\ \mathbf{c}_2^{(s)} \\ \mathbf{d}_2^{(s)} \end{bmatrix}, \begin{bmatrix} \mathbf{d}_1^{(s)} \\ \mathbf{d}_2^{(s)} \end{bmatrix} - Q \begin{bmatrix} \mathbf{c}_1^{(s)} \\ \mathbf{c}_2^{(s)} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \right\}. \tag{3.78}$$

Note that the description of the scattering-form interconnecting network in (3.78) uses a constraint-form encoding. Similar to the previous discussion of equivalence class representatives for canonical-form systems, the conditions required for an interconnective description be in scattering form are not sufficient for a representative of an equivalence class $[s^{(s)}]$ to be defined.

Equipping the terminal space $\mathcal{V}^{(s)}$ in (3.75) with the standard inner product, we associate the scattering-form interconnective description of a system with a scattering organization $\mathcal{O}^{(s)}$ defined according to

$$\mathcal{O}^{(s)} \triangleq (C_s, \mathcal{D}_p^{(s)}, \mathcal{D}_c^{(s)}) \tag{3.79}$$

where the connection between the partition decompositions $\mathcal{D}_p^{(s)}$ in (3.76) and (3.79) was established in the discussion surrounding (3.6). The correspondence map and conjugate decomposition in (3.79) take the form

$$C_s = \begin{bmatrix} I_{R^{(s)}} & 0 & 0 & 0 \\ 0 & -I_{R^{(s)}} & 0 & 0 \\ 0 & 0 & I_{L^{(s)}} & 0 \\ 0 & 0 & 0 & -I_{L^{(s)}} \end{bmatrix} \tag{3.80}$$

$$\mathcal{D}_c^{(s)} = \left\{ \text{span}\left(\mathbf{e}^{(1)}, \ldots, \mathbf{e}^{(R^{(s)}+L^{(s)})}\right),\ \text{span}\left(\mathbf{e}^{(R^{(s)}+L^{(s)}+1)}, \ldots, \mathbf{e}^{(2(R^{(s)}+L^{(s)}))}\right)\right\} \tag{3.81}$$

In describing scattering organizations, the associated comparison space is $(\mathbb{R}^{R^{(s)}+L^{(s)}}, \langle\cdot,\cdot\rangle_{\mathbb{R}^{R^{(s)}+L^{(s)}}})$ where the inner product is the standard inner product and the conjugate maps are given by

$$M_A^{(s)} = \begin{bmatrix} I_{R^{(s)}} & I_{R^{(s)}} & 0 & 0 \\ 0 & 0 & I_{L^{(s)}} & I_{L^{(s)}} \end{bmatrix} \tag{3.82}$$

and

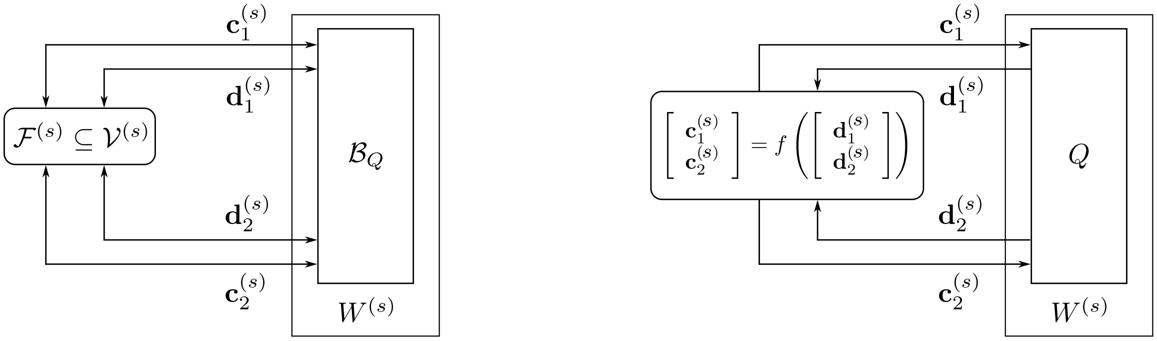
$$M_B^{(s)} = \begin{bmatrix} I_{R^{(s)}} & -I_{R^{(s)}} & 0 & 0 \\ 0 & 0 & I_{L^{(s)}} & -I_{L^{(s)}} \end{bmatrix}. \tag{3.83}$$

For these conjugate maps and the partitioning scheme of the terminal vector $\mathbf{v}$ in (3.78), the conservation principle associated with a scattering organization may be written in the comparison space as

$$\left\langle \begin{bmatrix} \mathbf{c}_1^{(s)} + \mathbf{d}_1^{(s)} \\ \mathbf{c}_2^{(s)} + \mathbf{d}_2^{(s)} \end{bmatrix}, \begin{bmatrix} \mathbf{c}_1^{(s)} - \mathbf{d}_1^{(s)} \\ \mathbf{c}_2^{(s)} - \mathbf{d}_2^{(s)} \end{bmatrix} \right\rangle_{\mathbb{R}^{R^{(s)}+L^{(s)}}} = \left\| \begin{bmatrix} \mathbf{c}_1^{(s)} \\ \mathbf{c}_2^{(s)} \end{bmatrix} \right\|^2 - \left\| \begin{bmatrix} \mathbf{d}_1^{(s)} \\ \mathbf{d}_2^{(s)} \end{bmatrix} \right\|^2 \tag{3.84}$$

and evaluates to zero for all $(\mathbf{c}_1^{(s)}, \mathbf{d}_1^{(s)}, \mathbf{c}_2^{(s)}, \mathbf{d}_2^{(s)}) \in W^{(s)}$. This conservation principle is reminiscent of the form (3.49) where the terminal vector $\mathbf{v}$ has been partitioned into an equal number of $\mathbf{c}$ and $\mathbf{d}$ variables.

Interconnective structures describing signal processing systems in scattering form are illustrated in Figure 3-8 where the structure in (a) depicts a behavioral model represented as a graph according to the partitioning of $\mathbf{v}$ in (3.78) and the structure in (b) corresponds


$\mathbf{c}_1^{(s)}$
$\mathbf{d}_1^{(s)}$
$\mathcal{F}^{(s)} \subseteq \mathcal{V}^{(s)}$
$\mathcal{B}_Q$
$\mathbf{d}_2^{(s)}$
$\mathbf{c}_2^{(s)}$
$W^{(s)}$
$\mathbf{c}_1^{(s)}$
$\mathbf{d}_1^{(s)}$
$\begin{bmatrix} \mathbf{c}_1^{(s)} \\ \mathbf{c}_2^{(s)} \end{bmatrix} = f\left(\begin{bmatrix} \mathbf{d}_1^{(s)} \\ \mathbf{d}_2^{(s)} \end{bmatrix}\right)$
Q
$\mathbf{d}_2^{(s)}$
$\mathbf{c}_2^{(s)}$
$W^{(s)}$

(a) General scattering-form structure

(b) Realized scattering-form structure

Figure 3-8: Interconnective descriptions of a signal processing system in scattering form. (a) The interconnective structure corresponding to a generic scattering-form description. (b) A realization of the interconnective structure in (a) corresponding to the input-output configuration in (3.85).

to a functional realization of the structure in (a) according to the input-output configuration

$$\begin{cases} \text{Inputs:} & \mathbf{c}_1^{(s)},\ \mathbf{c}_2^{(s)} \\ \text{Outputs:} & \mathbf{d}_1^{(s)},\ \mathbf{d}_2^{(s)} \end{cases} \implies P = \begin{bmatrix} I_{R^{(s)}} & 0 & 0 & 0 \\ 0 & 0 & I_{L^{(s)}} & 0 \\ 0 & I_{R^{(s)}} & 0 & 0 \\ 0 & 0 & 0 & I_{L^{(s)}} \end{bmatrix}. \tag{3.85}$$

The functional realization of the interconnecting network as a matrix that is consistent with the input-output configuration P is readily obtained using the procedure in Section 3.3 or by inspection of the graph and is given by

$$\begin{bmatrix} \mathbf{d}_1^{(s)} \\ \mathbf{d}_2^{(s)} \end{bmatrix} = Q \begin{bmatrix} \mathbf{c}_1^{(s)} \\ \mathbf{c}_2^{(s)} \end{bmatrix}. \tag{3.86}$$

Similar to the case of a canonical-form description, we will frequently be interested in assembling large-scale signal processing systems by connecting together many independent constitutive modules and interconnects to form a scattering-form system where the partition decomposition is specifically designed to be maximal. To be consistent with the definition above, the partition decomposition in (3.76) is organized for these systems into a collection of $R^{(s)} + L^{(s)}$ two-dimensional subspaces (using direct sums for notational convenience) according to

$$\begin{aligned}\mathcal{D}_p^{(s)} \;=\; &\Big\{\text{span}\left(\mathbf{e}^{(1)}, \mathbf{e}^{(R^{(s)}+1)}\right), \ldots, \text{span}\left(\mathbf{e}^{(R^{(s)})}, \mathbf{e}^{(2R^{(s)})}\right), \\ &\text{span}\left(\mathbf{e}^{(2R^{(s)}+1)}, \mathbf{e}^{(2R^{(s)}+L^{(s)}+1)}\right), \ldots, \text{span}\left(\mathbf{e}^{(2R^{(s)}+L^{(s)})}, \mathbf{e}^{(2(R^{(s)}+L^{(s)}))}\right)\Big\}.\end{aligned} \tag{3.87}$$

The constitutive modules are then each decomposable into individual relations in $\mathbb{R}^2$. When a system in scattering form utilizes (3.87), we shall refer to the system as being in *maximal* scattering form.

3.4.4 | Scattering coordinate transforms

In this subsection, we present the *scattering coordinate transform* used to move between canonical-form and scattering-form descriptions of a system. To do this, we focus on manipulating the interconnective description of a system that has a maximal partition decomposition and is initially represented in canonical form. By symmetry, and the fact that coordinate transforms are required to be invertible, this presentation also defines the scattering coordinate transform used to manipulate a system initially described in scattering form to a canonical form description, assuming a canonical-form description exists. One benefit to focusing on canonical and scattering descriptions with maximal partition decompositions is that the same coordinate transform will also work between systems with non-maximal partition decompositions.

Without loss of generality, let the coordinate matrix M_c associated with a canonical-form structure be the identity matrix and let $R = R^{(c)} = R^{(s)}$ and $L = L^{(c)} = L^{(s)}$. Then, by inspection of the canonical and scattering organizations, and (3.68) through (3.70) and (3.82) through (3.84) in particular, it is straightforward to verify that a valid scattering coordinate transform is summarized by

$$\begin{bmatrix} \mathbf{c}_1^{(s)} \\ \mathbf{d}_1^{(s)} \\ \mathbf{c}_2^{(s)} \\ \mathbf{d}_2^{(s)} \end{bmatrix} = \underbrace{\begin{bmatrix} I_R & -I_R & 0 & 0 \\ I_R & I_R & 0 & 0 \\ 0 & 0 & -I_L & I_L \\ 0 & 0 & I_L & I_L \end{bmatrix}}_{M_s M_c^{-1}} \begin{bmatrix} \mathbf{a}_1^{(c)} \\ \mathbf{b}_1^{(c)} \\ \mathbf{a}_2^{(c)} \\ \mathbf{b}_2^{(c)} \end{bmatrix}. \tag{3.88}$$

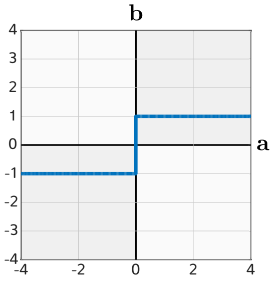
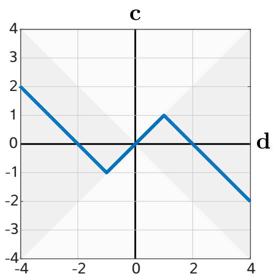
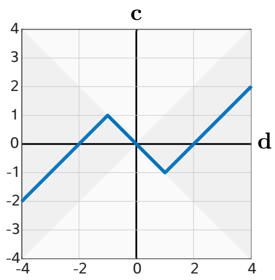
Coordinate transform	n/a	$\begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}$	$\begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}$
Behavior			
Functional realization	n/a	$\mathbf{c} = \begin{cases} \mathbf{d}, & \lvert\mathbf{d}\rvert \leq 1 \\ 2\operatorname{sgn}(\mathbf{d}) - \mathbf{d}, & \lvert\mathbf{d}\rvert > 1 \end{cases}$	$\mathbf{c} = \begin{cases} -\mathbf{d}, & \lvert\mathbf{d}\rvert \leq 1 \\ \mathbf{d} - 2\operatorname{sgn}(\mathbf{d}), & \lvert\mathbf{d}\rvert > 1 \end{cases}$

Figure 3-9: An illustration of the coordinate transform in (3.88) applied to a relation belonging to maximal canonical-form structure. The shaded regions indicate that passivity in the $(\mathbf{a}, \mathbf{b})$ coordinate system results in passivity in the $(\mathbf{c}, \mathbf{d})$ coordinate system [1].

This transformation matrix is easily verified to be unique to within matched pairwise multiplication of columns that preserve any partition decomposition by ±1 and matched column scalings where the matching refers to columns i and $i + R$ for $1 \leq i \leq R$ and columns i and $i + L$ for $2R + 1 \leq i \leq 2R + L$. Importantly, the sparsity pattern in (3.88) is sufficient to preserve the partition decompositions in (3.61) and (3.76) and preserve the maximal partition decompositions in (3.73) and (3.87). The action of this coordinate transformation, and by symmetry the inverse coordinate transformation, is discussed next with regard to the constitutive modules and interconnecting networks separately.

Consider an interconnective system organized into a maximal canonical-form description where the canonical constitutive module $\mathcal{F}^{(c)}$ in (3.62) decomposes according to

$$\mathcal{F}^{(c)} \quad = \quad \mathcal{F}_1^{(1)} \times \cdots \times \mathcal{F}_R^{(1)} \times \mathcal{F}_1^{(2)} \times \cdots \times \mathcal{F}_L^{(2)} \tag{3.89}$$

where $\mathcal{F}_r^{(1)}$ denotes the relation coupling the variables $(\mathbf{a}, \mathbf{b}) = (\mathbf{a}_{1,r}^{(c)}, \mathbf{b}_{1,r}^{(c)})$ corresponding to the r-th entries of $\mathbf{a}_1^{(c)}$ and $\mathbf{b}_2^{(c)}$ and $\mathcal{F}_l^{(2)}$ similarly denotes the relation coupling the variables $(\mathbf{a}, \mathbf{b}) = (\mathbf{a}_{2,l}^{(c)}, \mathbf{b}_{2,l}^{(c)})$ corresponding to the l-th entries of $\mathbf{a}_2^{(c)}$ and $\mathbf{b}_2^{(c)}$. The conservation principle in (3.70) is expressed using this notation as the sum of terms $\mathbf{ab} = 0$ where the sum is over the $R + L$ relations in (3.89). The variables $(\mathbf{c}, \mathbf{d})$ used in the scattering-form description of the system can be obtained using (3.88) or using $R + L$ smaller transformations as follows. For each relation $\mathcal{F}_r^{(1)}$ the transformed variables $(\mathbf{c},\mathbf{d}) = (\mathbf{c}_{1,r}^{(s)},\mathbf{d}_{1,r}^{(s)})$ corresponding to the r-th entries of $\mathbf{c}_1^{(s)}$ and $\mathbf{d}_1^{(s)}$ are generated from $(\mathbf{a},\mathbf{b}) = (\mathbf{a}_{1,r}^{(c)},\mathbf{b}_{1,r}^{(c)})$ according to

$$\begin{bmatrix}\mathbf{c}\\ \mathbf{d}\end{bmatrix} = \underbrace{\begin{bmatrix}1 & -1\\ 1 & 1\end{bmatrix}}_{\triangleq M^{(i)}}\begin{bmatrix}\mathbf{a}\\ \mathbf{b}\end{bmatrix}. \tag{3.90}$$

Similarly, for each relation $\mathcal{F}_l^{(2)}$ the transformed variables $(\mathbf{c},\mathbf{d}) = (\mathbf{c}_{2,l}^{(s)},\mathbf{d}_{2,l}^{(s)})$ corresponding to the l-th entries of $\mathbf{c}_2^{(s)}$ and $\mathbf{d}_2^{(s)}$ are generated from $(\mathbf{a},\mathbf{b}) = (\mathbf{a}_{2,l}^{(c)},\mathbf{b}_{2,l}^{(c)})$ according to

$$\begin{bmatrix}\mathbf{c}\\ \mathbf{d}\end{bmatrix} = \underbrace{\begin{bmatrix}-1 & 1\\ 1 & 1\end{bmatrix}}_{\triangleq M^{(o)}}\begin{bmatrix}\mathbf{a}\\ \mathbf{b}\end{bmatrix}. \tag{3.91}$$

A maximal scattering-form system has relations that constrain $(\mathbf{c},\mathbf{d})$ pairs satisfying the conservation principle $\mathbf{c}^2 - \mathbf{d}^2 = 0$. The matrix $M^{(i)}$ defined in (3.90) corresponds to a scaled rotation by $\frac{\pi}{4}$ and the matrix $M^{(o)}$ defined in (3.91) corresponds to a squeeze mapping, i.e. a transformation which preserves Euclidean area in the plane which is not a rotation or a sheer transformation. Figure 3-9 illustrates these transformations for the example relation in (2.7) which is repeated on the left of the figure for completeness. More generally, the figure illustrates that relations in the $(\mathbf{a},\mathbf{b})$-plane contained within the first and third quadrants map to conic sections in the $(\mathbf{c},\mathbf{d})$-plane as indicated by the shaded regions. This well-known result is referred to as incremental passivity in the controls community and will play an important role in Chapter 5 [1].

A closed-form expression for the matrix Q denoting a realization of the scattering-form interconnecting network that is consistent with the input-output configuration P in (3.85) and that is stated in terms of the matrix A from the canonical-form interconnecting network $W^{(c)}$ in (3.63) can be obtained by following the procedure developed in Section 3.3.1. Specifically, by following the steps associated with (3.37), the transformed interconnect $W^{(s)} = M_s M_c^{-1} W^{(c)}$ with rows permuted to conform to the input-output configuration can be written as the range of an appropriately defined matrix. For the setup in this subsection, this corresponds to

$$\begin{bmatrix} \mathbf{c}^{(1)} \\ \mathbf{c}^{(2)} \\ \mathbf{d}^{(1)} \\ \mathbf{d}^{(2)} \end{bmatrix} \in \operatorname{range}\left(\underbrace{\begin{bmatrix} I_R & 0 & 0 & 0 \\ 0 & 0 & I_L & 0 \\ 0 & I_R & 0 & 0 \\ 0 & 0 & 0 & I_L \end{bmatrix}}_{\text{input-output config. in (3.85)}} \underbrace{\begin{bmatrix} I_R & -I_R & 0 & 0 \\ I_R & I_R & 0 & 0 \\ 0 & 0 & -I_L & I_L \\ 0 & 0 & I_L & I_L \end{bmatrix}}_{\text{coordinate transform in (3.88)}} \underbrace{\begin{bmatrix} I_R & 0 \\ 0 & -A^T \\ A & 0 \\ 0 & I_L \end{bmatrix}}_{W^{(c)} \text{ in (3.63)}} \right) \tag{3.92}$$

$$\in \quad \operatorname{range}\left(\begin{bmatrix} I_R & A^T \\ -A & I_L \\ I_R & -A^T \\ A & I_L \end{bmatrix} \right). \tag{3.93}$$

By direct application of (3.38) we immediately obtain that the realization Q is given by

$$Q \quad = \quad \begin{bmatrix} I_R & -A^T \\ A & I_L \end{bmatrix} \begin{bmatrix} I_R & A^T \\ -A & I_L \end{bmatrix}^{-1} \tag{3.94}$$

$$= \quad \begin{bmatrix} \left(I_R - A^T A\right)\left(I_R + A^T A\right)^{-1} & -2A^T\left(I_L + AA^T\right)^{-1} \\ 2A\left(I_R + A^T A\right)^{-1} & \left(I_L - AA^T\right)\left(I_L + AA^T\right)^{-1} \end{bmatrix} \tag{3.95}$$

where the matrix inverse in (3.94) is guaranteed to be well-defined and (3.95) is obtained from (3.94) by application of the matrix inversion lemma. The inversion of (3.94) follows from the observation that the inverse is applied to an identity matrix added to a skew-symmetric matrix which cannot result in a zero-valued eigenvalue. Indeed, further inspection of (3.94) yields that Q is the Cayley transform[2] of the negative of the realization of the canonical form matrix in (3.72) and that the block diagonal components of Q in (3.95) are respectively the Cayley transform of $A^T A$ and AA^T. The implication of this observation is that Q will always have eigenvalues bounded away from -1 on the unit circle, and will become important in Chapter 4 in the context of stability analysis. Note that only special orthogonal matrices

[2] The Cayley transform is a single-parameter family of transforms. The parameter will not matter to us due to the invariance of (3.88) to matched column scalings.

Q will map through the inverse Cayley transform to produce a canonical-form interconnect, and so we will primarily be interested in orthogonal form structures where this is possible.

When A is itself an orthogonal matrix, the expression for Q in (3.95) simplifies to

$$Q = \begin{bmatrix} 0 & -A^T \\ A & 0 \end{bmatrix} \tag{3.96}$$

which is identical to the realization of the canonical form interconnect for the input-output configuration in (3.71). Said another way, the coordinate transform between scattering-form and canonical-form descriptions in (3.88) is an interconnection invariant coordinate transform when A is an orthogonal matrix. In the context of implementing interconnective systems, the consequence of (3.96) is that when A denotes an orthogonal transform for which a fast implementation exists, such as a fast Fourier transform algorithm, then the same fast implementation may be used to implement Q.

3.5 | Properties of equivalent interconnective systems

So far in this chapter, we have seen that every interconnective system s belongs to an equivalence class $[s]$ and that, by properly selecting the coordinate matrix and input-output configuration, some of the systems in $[s]$ may be organized into realized interconnective structures and then algorithms. In this section, we present several properties pertaining to groups of systems belonging to the same interconnective equivalence class.

3.5.1 | Redundancy in coordinate maps and input-output configurations

The design task in many signal processing contexts is specifically to invert a generally nonlinear and time-varying forward system, assuming of course that an inverse system exists. Previous approaches in the literature have had varying degrees of success with this task, often resorting to blending heuristics with graph based techniques [18, 56–58]. As was briefly discussed in [17], methods similar to the procedure in Section 3.3 for designing synchronous processing loops can be leveraged to automate the inversion of many signal processing systems. In this subsection, we state this more clearly in the context of the interconnective framework and use it to illustrate the redundancy that exists between selecting coordinate matrices and input-output configurations.

The behavior of an invertible forward operator is closely related to the behavior of the inverse operator, as demonstrated by (2.1) through (2.3). Similar to this, the interconnective description of a forward signal processing system is in the same interconnective equivalence class as the interconnective description of a signal processing system that inverts it, again assuming that an inverse system is well-defined. To see this, let s denote an interconnective system using coordinate matrix M and whose implementation corresponding to input-output configuration P is known to be invertible. Without loss of generality, assume there are an equal number of overall system inputs and outputs and that they are organized into the first constitutive module $\mathcal{F}_1$ such that the inputs appear above the outputs in $\mathbf{v}_1^{(CR)}$. Let s' denote the inverse system. The coordinate matrix M' and input-output configuration P' associated with s' can be related to M and P in at least two ways. In the first, the coordinate matrix $M' = M$ is preserved and the input-output configuration P' is generated from P by modifying the overall system inputs and outputs to be outputs and inputs, respectively. In the second, the input-output configuration $P' = P$ is preserved and the coordinate submatrix M_1' is generated from M_1 according to the permutation

$$M_1' = \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} M_1. \tag{3.97}$$

The symmetry between these definitions of the system s' highlight an important complementarity property between general coordinate matrices and input-output configurations. From either description of s', the procedure for obtaining a synchronous processing loop can then be used to generate an implementation of s' to invert s. Hence, this property suggests a straightforward way to automate the inversion of many signal processing systems. However, the fact that s and s' are equivalent descriptions of the same underlying behavioral model is not sufficient to conclude that a schedule for s' will be successfully identified without imposing additional structure on s.

3.5.2 | Interconnection invariant coordinate transforms

In Section 3.3.1 we presented a general approach as well as developed some numerical tools for obtaining a collection of functions f_k and matrices A_l with respective behaviors $\mathcal{F}_k$ and W_l. The key step in that process is the selection an appropriate coordinate matrix and input-output configuration so that the mapping objects are well-defined. In this subsection, we characterize the set of all coordinate matrices for which the realization of the interconnecting network remains the same and provide a straightforward numerical recipe to generate them. To elaborate, let s denote an interconnective system with coordinate matrix M. We shall say that the coordinate transform $M'M^{-1}$ used to represent s as an equivalent system $s' \in [s]$ with coordinate matrix M' is an *interconnection invariant coordinate transform* if the realization of the interconnecting network for both systems is the same matrix. The realization of the constitutive modules, when they exist, will generally be different. This observation lends itself to generating implementations of interconnective systems for which a decentralized architecture for the interconnects has been found but for which the constitutive modules are not directly realizable. In particular, performing coordinate transforms of this type can be used to generate functional realizations of the constitutive modules while retaining the robustness and separability of the interconnecting network, thereby allowing algorithms to be found for decentralized systems without modifying the topology of the decentralized interconnective graph.

Let A and P respectively denote a realization of the interconnecting network and the associated input-output configuration of an interconnective system s. The collection of all coordinate matrices associated with systems in the equivalence class $[s]$ with the same interconnecting network realization A and input-output configuration P are parameterized using the generic coordinate matrix M according to

$$\begin{bmatrix} \\ P \\ \\ \end{bmatrix} \begin{bmatrix} \\ M \\ \\ \end{bmatrix} \begin{bmatrix} I \\ A \end{bmatrix} [R] = \begin{bmatrix} I \\ A \end{bmatrix} \tag{3.98}$$

where R is any $\dim(W) \times \dim(W)$ invertible matrix. If we restrict ourselves to $\mathcal{D}_p$-invariant coordinate matrices then the sparsity pattern of M is further constrained by the direct product decomposition of $\mathcal{V}$ used to define the partition decomposition $\mathcal{D}_p$, i.e. M is composed of K diagonal submatrices M_k each of size $\dim\mathcal{V}_k \times \dim\mathcal{V}_k$. The use of the matrix R in (3.98) follows from the fact that the subspace generated by the span of the columns of a matrix is invariant to the right multiplication of the matrix by any invertible matrix [34]. In the context of realizing interconnective systems, a complementary approach that reverses the steps outlined in Section 3.3.1 is to specifically generate a realization of the interconnecting network first and then draw coordinate matrices from (3.98) until the constitutive modules can also be realized as functions.

We now present a numerical procedure to generate interconnection invariant coordinate matrices M satisfying (3.98). To assist with the notation in doing this, we define a matrix C whose range is equal to the interconnect subspace W using the realization A, i.e. a natural choice for C is given by

$$C = \begin{bmatrix} I \\ A \end{bmatrix}, \tag{3.99}$$

and define the function $d\colon \mathbb{R}^{N\times N} \times \mathbb{R}^{\dim(W)\times\dim(W)} \rightarrow \mathbb{R}$ according to

$$d(M,R) = \|PMCR - C\|_F^2 \tag{3.100}$$

where $\|A\|_F = \sqrt{\text{trace}(A^T A)}$ denotes the Frobenius matrix norm. By construction, any appropriately block diagonal matrix M' and invertible matrix R' for which $d(M',R') = 0$ defines an interconnection invariant coordinate transform from an initial coordinate matrix M associated with A to the coordinate matrix M' according to $M'M^{-1}$. The function d is biconvex, i.e. the function $d(M,\cdot)$ is convex in its second argument for fixed M and the function $d(\cdot,R)$ is convex in its first argument for fixed R. This property suggests an efficient method to solve for a block diagonal matrix M is to use an alternating projection approach. Specifically, let $M^{(0)}$ denote an initial coordinate matrix and $R^{(0)}$ an initial invertible matrix.

The procedure is to iteratively solve the least squares problems

$$\begin{aligned} M^{(k)} &= \arg\min_{M} \|PMCR^{(k)} - C\|_F^2 \text{ s.t. } M \text{ block diagonal} && \text{(3.101a)} \\ R^{(k+1)} &= \arg\min_{R} \|PM^{(k)}CR - C\|_F^2 && \text{(3.101b)} \end{aligned}$$

until $d\left(M^{(k)}, R^{(k+1)}\right) = 0$. Each step in (3.101) can be solved in closed form since they are simply linearly constrained and unconstrained least squares problems, respectively. The stopping criterion implies that R is invertible as required. As a final remark, note that the initial guess $M^{(0)}$ is critical to identifying coordinate transformations which are not stationary, e.g. the initial coordinate matrix M and $R = I$ is a solution to the iterative procedure that yields a fairly uninteresting coordinate transformation.

3.5.3 | Numerical sensitivity of interconnecting networks

Large-scale signal processing systems, notably those involved in solving machine learning and big data problems, make ubiquitous use of linear algebra subroutines. In the context of implementing decentralized interconnective systems, numerical errors inevitably arise during the execution of the interconnecting network when the individual interconnects are implemented using different digital processors which may further have varying bit-depths or number systems. In asynchronous processing systems, for example, each processor, independent of every and all other processors, is assigned and continually implements a different matrix, therefore propagating numerical errors throughout the graph in a relatively unpredictable manner. These types of errors are typically understood through an assortment of perturbation and sensitivity analysis tools. For example, finite-precision effects such as coefficient quantization can be quantified by invoking sensitivity theorems as discussed, e.g., in [35]. These sources of error contribute significantly to the total accumulated error in a system's output, especially as problem sizes continue to scale. To evaluate the numerical sensitivity of decentralized interconnecting networks in this thesis, we will primarily restrict our analysis to the sensitivity and robustness of the global interconnecting network rather than, for example, the individual interconnects or constitutive modules. In Chapter 4, in the context of synchronous and asynchronous stability analysis, we will consider a broader classification of the entire system's robustness to a variety of network disturbances using conic properties and well-known concepts of continuity.

Motivated by the extensive use of floating-point arithmetic in modern digital signal processing platforms, we measure the sensitivity of a matrix to many perturbation sources using its relative condition number, i.e. the ratio of the matrix's extremal singular values. Loosely speaking, if this measure is on the order 10^k, then approximately $d - k$ digits of the processed output are reliable where d is the maximum number of digits available on the chosen computational platform [2]. This loss of accuracy is inherent to the matrix itself and says nothing about further degradations that accumulate due to a particular constitutive modules stability nor to errors that arise due to round-off noise and coefficient quantization. When evaluating the numerical sensitivity of a realization A of the interconnect W, we will use a suitably modified version of the relative condition number where we take the ratio of the largest to smallest non-zero singular values of A, i.e. we use the modified condition number $\hat{\kappa}(A)$ given by

$$\hat{\kappa}(A) \quad = \quad \|A\|\|A^{\dagger}\|^{-1} \tag{3.102}$$

where $A^{\dagger}$ denotes the Moore-Penrose pseudo inverse of A and the term $\|A^{\dagger}\|$ evaluates to the smallest non-zero singular value of A and can be equivalently expressed according to

$$\|A^{\dagger}\| \quad = \min_{\substack{\mathbf{x}\in\mathrm{range}(A)\\ \|\mathbf{x}\|=1}} \|A^T\mathbf{x}\|. \tag{3.103}$$

The expression (3.102) also has a simple interpretation: it is the ratio between the relative change or error in the interconnecting network due to a small perturbation of its input with respect to the relative size of the perturbation itself where the modification is justified using the fact that the number of interconnect inputs $N^{(i)}$ and outputs $N^{(o)}$ are frequently not the same and having repeated outputs generates a non-trivial null space that is not a reflection of numerical instability. In what follows, we refer to to interconnecting network as being either *ill-conditioned* or *well-conditioned*, consistent with the traditional treatment of this topic. If an interconnecting network is ill-conditioned, i.e. if $\hat{\kappa}(A) \gg 0$, it means that the matrix A is extremely sensitive, in the relative sense, to perturbations in its input. On the contrary, if the interconnecting network is well-conditioned, i.e. if $\kappa(A) \approxeq 1$, then the matrix does not rapidly change in value as small perturbations to the input are made.

The numerical sensitivity of the interconnecting network associated with interconnective structures within the same equivalence class can generally range from ill-conditioned to well-conditioned. For example, the modified condition number of the functional realization of the interconnecting network associated with a canonical-form system can be arbitrarily large, while the conditioning of the interconnecting network associated with the equivalent scattering-form system is perfect. Of course, this fact assumes that the numerical computation described by (3.95) for generating the orthogonal matrix Q from the skew-symmetric matrix A is unaffected by the ill-conditioning of the matrix A.

3.6 | Examples of interconnective systems

In this chapter, we established an interconnective framework to handle behavioral models of large-scale signal processing systems and then used it to various ends. In this section, we present two examples to further illustrate the utility of the framework. First, the constraint-form encoding of a system's interconnecting network presented in Section 3.3.1 is used to eliminate delay-free loops in allpass-warped signal-flow graphs. Second, a nonuniform sampling system that characterizes bandlimited signals using level crossings that correspond to nonuniform sampling instants is described. Using the interconnective viewpoint, we show that this procedure is essentially mapping the behavior of the input signal through a particular coordinate transform followed by standard periodic sampling. This example serves a dual purpose in that it provides direction for generalizing the tools developed in this chapter to handle broader classes of signal processing systems than we consider in this thesis as well introduces the notion of an interconnective signal model. In closing, we remark that a promising strategy for future research involves characterizing known signal processing systems using the interconnective framework and then using the interconnective equivalence classes established in Section 3.2.3 to provide new insight or benefit, e.g. for purposes of analysis, implementation, or interpretation.

3.6.1 | Delay-free loop reduction in allpass-warped filter structures

Linear and nonlinear signal-flow graphs, commonly used to represent signal processing and filtering systems graphically, are often interpretable as mixtures of both behavioral and input-output descriptions of a system. Signal-flow representations of allpass-warped filtering structures, for example, may contain elements with well-defined input-output relationships as well as elements that readily form implicit algebraic constraints. In this subsection, we demonstrate the utility of the procedure that was developed in Section 3.3.1 for eliminating implicit constraints manifesting themselves in a signal-flow graph as delay-free loops or cycles. In doing so, the interconnective description of a system is used as an intermediary representation of these constraints and is also used to preserve the input-output configuration from the initial signal-flow description of the system while manipulating the behavior of the interconnecting network into a computable form.

Let $H(z)$ denote the system function or z-transform of an impulse response corresponding to a stable, all-pole second order filtering system of the form

$$H(z) = \frac{1}{1 - a_1 z^{-1} - a_2 z^{-2}}. \tag{3.104}$$

A signal-flow implementation of $H(z)$ may use any number of well-known structures. We proceed focusing on direct-form structures with an understanding that the same sequence of steps would apply to other forms as well, e.g. lattice or farrow structures. Next, the filtering system is composed by taking the flow-graph implementation of $H(z)$ and replacing each delay element z^{-1} with the first-order all-pass system $H_{ap}(z)$ given by

$$H_{ap}(z) = \frac{z^{-1} - \alpha}{1 - \alpha z^{-1}} \tag{3.105}$$

where the free parameter $\alpha \in (-1, 1)$ controls the warping effect on the phase response in the Fourier transform of the original filtering system and may be desired for the applications identified in [59]. The signal-flow structure of the warped filtering system is illustrated in Figure 3-10(a) and can be interpreted as indicating that the overall system output $\mathbf{y}^n$ is to be explicitly computed as a function of the overall system input $\mathbf{x}^n$ and the two previous

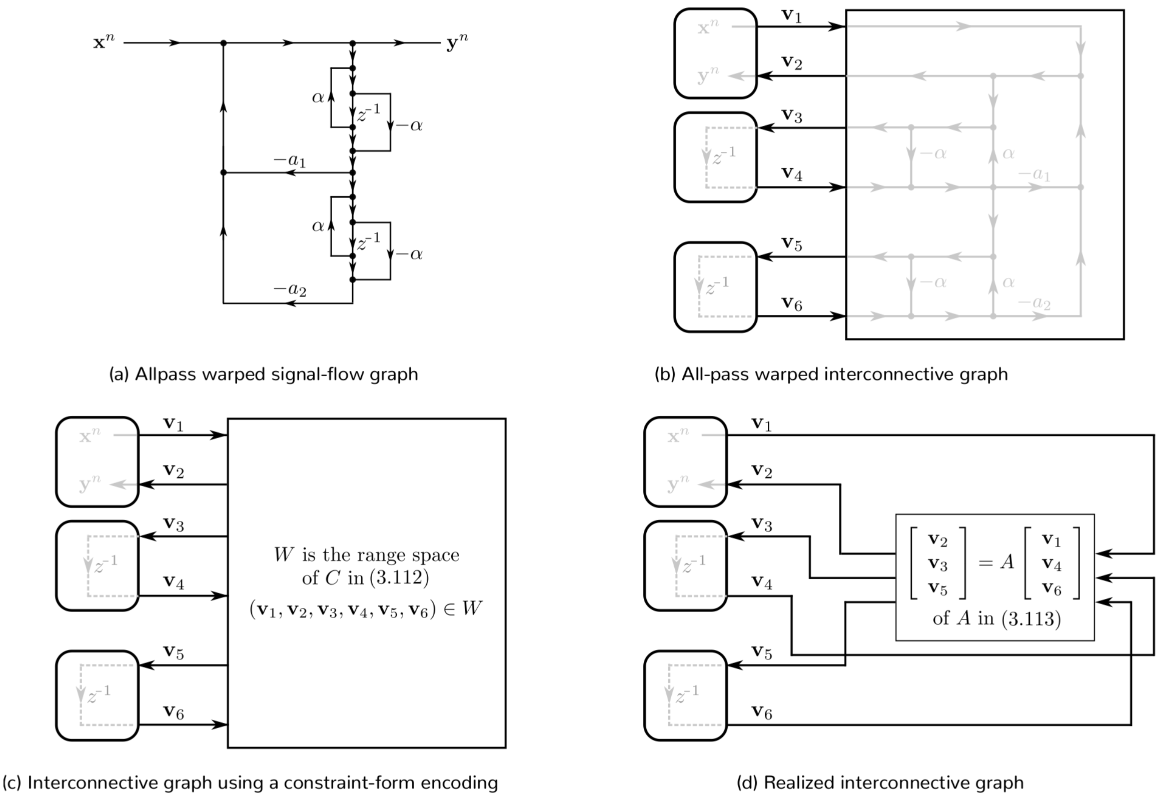


Figure 3-10: An illustration of the procedure for realizing interconnecting networks, shown for a second-order all-pole system in direct form where the delays have been replaced by first-order allpass elements. (a) The signal-flow graph containing delay-free loops. (b) The interconnective description of the system in (a). (c) The interconnective description using a constraint-form encoding of the interconnecting network. (d) A computable system with a matrix A whose behavior is W from (c).

outputs $\mathbf{y}^{n-1}$ and $\mathbf{y}^{n-2}$, and also that the computation should satisfy a set of implicit algebraic constraints despite the fact that the functional form of the overall computation is not immediately visible by inspection of the signal-flow graph.

An interconnective description of the filtering system after the warping or composition has been performed is depicted in Figure 3-10(b) and corresponds to an equivalent analytic description of the system given by

$$\mathcal{V} = \mathbb{C}^6 \tag{3.106}$$

$$\mathcal{D}_p = \{\mathbb{C}^2, \mathbb{C}^2, \mathbb{C}^2\} \tag{3.107}$$

$$\mathcal{D}_i = \{W, \mathcal{F}_1 \times \mathcal{F}_2 \times \mathcal{F}_3\} \tag{3.108}$$

$$M = \begin{bmatrix} I_2 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_2 \end{bmatrix} \tag{3.109}$$

where the first constitutive module $\mathcal{F}_1$ contains the overall system input and output, the second and third constitutive modules $\mathcal{F}_2$ and $\mathcal{F}_3$ correspond to synchronous delay modules, and the vector space description of the interconnecting network W is expressed using a constraint-form encoding according to

$$W = \left\{ \mathbf{v} \in \mathbb{C}^6 : \begin{bmatrix} 0 & 1 & -1 & \alpha & 0 & 0 \\ 0 & 0 & -\alpha & 1 & -1 & \alpha \\ -1 & 1 & -\alpha a_1 & a_1 & -\alpha a_2 & a_2 \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \\ \mathbf{v}_4 \\ \mathbf{v}_5 \\ \mathbf{v}_6 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right\}. \tag{3.110}$$

Referring to Figure 3-10, the sequence of steps depicted by the interconnective structures in (b) through (d) illustrate the effectiveness of the proposed realization strategy in synthesizing a computable interconnecting network from the composed signal-flow representation containing delay-free loops, and in this sense represent an automated way to obtain a solution to the class of problems discussed in [26]. Referring to structures (b) through (d) in the figure, the descriptions conform to the input-output configuration

$$\left\{\begin{array}{ll} \text{Inputs:} & \mathbf{v}_1,\ \mathbf{v}_4,\ \mathbf{v}_6 \\ \text{Outputs:} & \mathbf{v}_2,\ \mathbf{v}_3,\ \mathbf{v}_5 \end{array}\right. \iff P = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{3.111}$$

The matrix used in the description of the interconnecting network in (3.110) corresponds to a constraint-form encoding of the linear and memoryless constraints in the structure in Figure 3-10(b), and from this we obtain that the behavior of the interconnecting network can also be written as the range of a matrix C as demonstrated by (3.37). For the chosen input-output configuration, C is given by

$$C = \begin{bmatrix} \frac{\alpha^2-\alpha a_1+a_2}{\alpha^2} & \frac{\left(\alpha^2-1\right)\left(a_2-\alpha a_1\right)}{\alpha^2} & \frac{\left(1-\alpha^2\right)a_2}{\alpha} \\ -\alpha^{-1} & \alpha^{-1} & 0 \\ \alpha^{-2} & \frac{\alpha^2-1}{\alpha^2} & \alpha^{-1} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.112}$$

Observe that the top three rows of (3.112) correspond to a matrix that is the realization of the interconnecting network for an inverse filtering system with the opposite input-output configuration and is consistent with the earlier discussion that forward and inverse systems belong to the same equivalence class. Making use of (3.38), the interconnecting network can be realized by a matrix A given by

$$A = \frac{1}{1-\alpha a_1-\alpha^2 a_2} \begin{bmatrix} 1 & \left(\alpha^2-1\right)\left(a_1-\alpha a_2\right) & \left(\alpha^2-1\right)a_2 \\ 1 & \alpha-a_1+\alpha a_2 & \left(\alpha^2-1\right)a_2 \\ -\alpha & \left(1-\alpha^2\right) & \alpha\left(1-\alpha a_1+a_2\right) \end{bmatrix}. \tag{3.113}$$

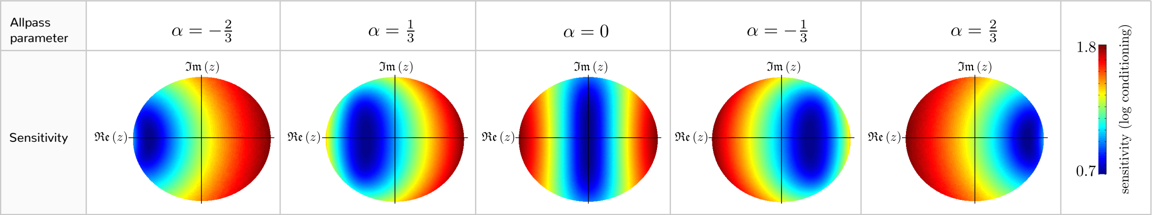


Figure 3-11: The numerical sensitivity of second-order filters, plotted for various pole locations when realized using an allpass-warped structure as in Figure 3-10 for allpass parameters $\alpha = 0, \pm\frac{1}{3}, \pm\frac{2}{3}$. For each allpass parameter, the realization is generated and analyzed for all complex conjugate pole pairs located within the unit circle of the z-plane.

As expected, the realization of the interconnecting network for the original direct-form filtering system described as an interconnective system corresponds to selecting the allpass warping parameter α in A to be zero.

For applications of allpass-warped filtering systems, a well-known design strategy is to select the allpass parameter α so that the effective pole locations of the composed filter lie in regions of the complex plane with desirable sensitivity properties. For signal-flow graphs such as the one in Figure 3-10(a), understanding the sensitivity of the interconnecting network in the modified sense of Section 3.5.3 is straightforward when no delay-free loops are present in the system and less so in the general case. Using the realized interconnecting network illustrated in Figure 3-10(d), Figure 3-11 depicts the sensitivity of the matrix A assuming the two poles are chosen to be in complex conjugate pairs within the unit circle for several values of the parameter α. The optimal regions of the z-plane for a fixed allpass parameter α using this sensitivity metric are consistent with the expected locations derived using traditional conformal mapping theory [59].

3.6.2 | Interconnective signal models for nonuniform sampling and time encodings of bandlimited signals

The application of the interconnective framework to manipulating behavioral models is focused in this thesis on models of systems rather than signals. More concretely, we focus specifically on terminal spaces $\mathcal{V}$ that correspond to finite-dimensional coordinate spaces and behaviors $\mathcal{B}$ that represent internal and external relationships characterizing a system in response to arbitrary inputs and initial conditions. In this final subsection, we allude to the opportunity in extending this framework to handle more general function and signal spaces and, by extension, their representation and acquisitions systems too. In particular, we examine the representation of bandlimited signals through uniformly and non-uniformly spaced samples from the interconnective viewpoint. In doing so, the discussion in this section highlights a potential future research direction by informally extending the body of work in this chapter to handle signal acquisition models.

Nonuniform sampling naturally occurs in many real-world signal acquisition systems and is typically attributed to irregular phenomena such as jitter, the asynchronous coordination of interleaved analog-to-digital converters, or the imprecision in spatially distributed sensors. Fortunately, the representation of a bandlimited signal as a collection of nonuniformly spaced samples remains unique as long as, roughly speaking, the sampling density is strictly greater than twice the bandwidth of the signal [60]. This fact suggests opportunity in intentionally sampling a signal in this way so that the samples contain enough redundancy for a particular feature to be robustly identified, either by further processing or visually by human inspection. In the following discussion, we consider a sampling system where the sampling rate is proportional to the amplitude of the first derivative of the input signal and may be desired for applications like non-intrusive load monitoring [61] where the processing goal is to detect envelopes of real-valued exponential signals.

The model of a signal acquisition system that samples a bandlimited signal $f\colon \mathbb{R} \to \mathbb{R}$ at a rate proportional to its first derivative is illustrated by the signal-chain in Figure 3-12(a). Rather than formally characterizing this system using an interconnective description, the focus will be on using the interconnective viewpoint to analyze the acquisition system since it is equivalent to analyzing the representation of f through its samples. Referring to the figure, the input signal f is added to a ramp, denoted by $\alpha\mathbf{t}$, in order to produce a monotone signal $g\colon \mathbb{R} \to \mathbb{R}$ that is subsequently differentiated after passing through an ideal uniform quantizer with granularity Δ. The result of this procedure is an impulse train whose timing instants encode a unique representation of f through the time series $\{(\mathbf{t}_n, n\Delta)\colon n \in \mathbb{Z}\}$.

An equivalent input-output model for this sampling system involves uniformly sampling a related signal $h\colon \mathbb{R} \to \mathbb{R}$ defined on a different independent variable $\mathbf{y}$. Specifically, the

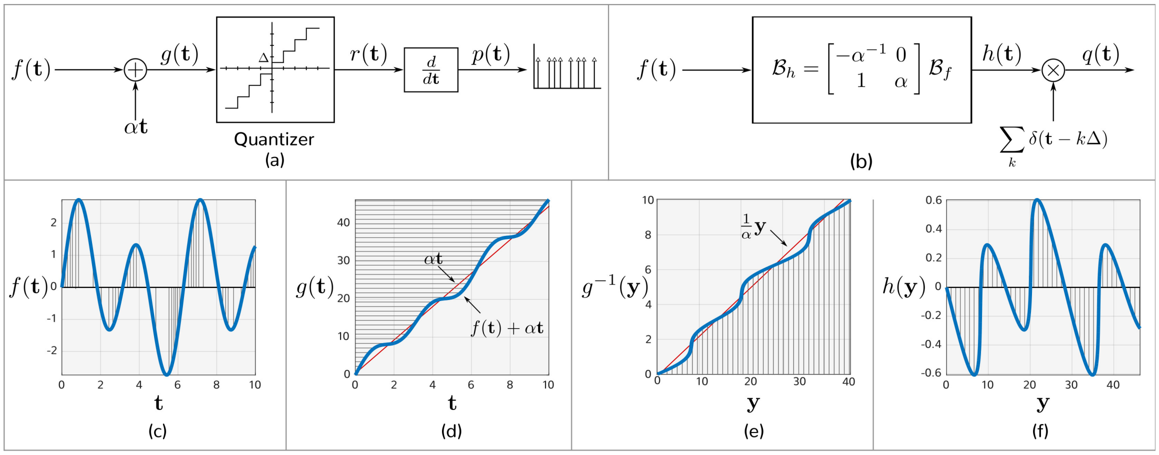


Figure 3-12: (a) A time-encoding of a bandlimited signal f at a rate proportional to its first derivative. (b) An equivalent sampling system using a coordinate transform. (c) A windowed segment of the input signal f. (d) The signal g obtained by adding f to a ramp. (e) The signal $g^{-1}(\mathbf{y})$ obtained by setting $\mathbf{y} = g(\mathbf{t})$. (f) The signal h. The gray lines in (c)-(f) correspond to the sampling instances associated with uniformly sampling the signal h.

signal h is generated according to

$$h(\mathbf{y}) \;=\; g^{-1}(\mathbf{y}) - \frac{1}{\alpha}\mathbf{y} \tag{3.114}$$

where $g(\mathbf{t}) = f(\mathbf{t}) + \alpha\mathbf{t}$ and the design parameter α is chosen sufficiently large so that g^{-1} is well-defined, i.e. $\alpha > \sup_{\mathbf{t}} \frac{d}{d\mathbf{t}} g(\mathbf{t})$. Uniformly spaced samples of the signal h are denoted in the figure by $q(n\Delta)$ and have a one-to-one correspondence with the previously encoded timing instants through the relationship

$$q(\mathbf{n}\Delta) \;=\; \mathbf{t}_n - \frac{1}{\alpha}\mathbf{n}\Delta \tag{3.115}$$

where $\mathbf{t}_n$ is the time of the $\mathbf{n}$-th sample of $p(\mathbf{t})$. This correspondence is demonstrated graphically by the vertical sampling bars in the processing stages mapping f to h in Figure 3-12(c) through (f).

From the interconnective viewpoint, the signals f and h can be related through coordinate transforms between their behaviors. Specifically, the action of the acquisition system transforming the input signal f to h equals an invertible linear transformation of the behavior of f according to

$$\begin{bmatrix} h(\mathbf{y}) \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} -\alpha^{-1} & 0 \\ 1 & \alpha \end{bmatrix} \begin{bmatrix} f(\mathbf{t}) \\ \mathbf{t} \end{bmatrix} \tag{3.116}$$

where α can be interpreted here as the parameter which ensures the behavior $\mathcal{B}_h$ has a well-defined functional realization. Allowing the behaviors of the signals f and h to respectively define the signal models s_f and s_h, (3.116) implies that $s_f \sim s_h$. From the symmetry property of the interconnective equivalence relation, we also have that $s_h \sim s_f$, so for any h produced using (3.116) we can recover f according to

$$\begin{bmatrix} f(\mathbf{t}) \\ \mathbf{t} \end{bmatrix} = \begin{bmatrix} -\alpha & 0 \\ 1 & \alpha^{-1} \end{bmatrix} \begin{bmatrix} h(\mathbf{y}) \\ \mathbf{y} \end{bmatrix}. \tag{3.117}$$

The mapping (3.117) can be thought of as the interconnective coordinate transform used by the reconstruction system. The relationship between the samples at the output of the signal chains in Figs. 3-12(a) and (b) are related by the transformations described in Section 3.5.1, and suggest methods for on-line computation of the processing system without needing to numerically invert the function g. The statement that $s_f \sim s_h$ is meant informally here since the signal spaces required to handle a complete characterization of the processing systems in Figure 3-12(a)-(b) would require terminal spaces $\mathcal{V}$ to be more general than the finite-dimensional vector spaces considered in this chapter.

Chapter 4

Stability and robustness properties of scattering structures

A variety of large-scale signal processing problems, including those blending data analytics and alternating projection principles [12], can be formulated as nonlinear systems solving fixed-point or constraint satisfaction problems involving decentralized and/or heterogeneous data sets. Consistent with this, a variety of computational tools have been developed to provide scalability and fault tolerance in many distributed computing models [19,20]. However, as the availability and geographic displacement of data and computing resources continues to grow, the benefits provided by reorganizing algorithms onto distributed systems diminish as global and local synchronization issues compound, e.g., including issues caused by the unreliability and intermediate latency of certain processors and interprocessor communication links [42, 62]. Common remedies work by introducing redundancy, thereby gaining functionality at the expense of resource usage and task replication [63].

Asynchronous computing models provide an alternative paradigm through which many of these issues can be handled, especially as the storage, retrieval, and movement of data becomes increasingly bottlenecked by networks prone to time-varying disruptions, congestion, and resource outages. Asynchronous communication protocols also allow for data exchanges to occur between processing nodes that are spread across dynamically changing networks where synchronized communication including some forms of token passing can become essentially impractical. In a broad sense, this thesis addresses pertinent aspects of the design, implementation, and analysis of both synchronous and asynchronous data processing algorithms using a distributed signal processing framework.

In this chapter, we specifically consider two kinds of data processing problems. The first corresponds to the class of constraint satisfaction and fixed-point problems that reduce to finding invariant states of nonlinear equations represented in scattering-form. With the intent of solving a broad range of these problems, we develop sufficient conditions for which simply implementing the associated interconnective system, either synchronously or asynchronously, effectively provides a solution. Moreover, we show that additional classes of problems can be solved by first-order filtering of the delay modules in these systems.

The second kind of problem corresponds to finding sparse solutions to convex systems of equations. The approach we take is to recast these problems as problems of graph discovery, i.e. learning the structure of an associated tree graph. To uncover this structure, several generic processing instructions are proposed that can readily be assembled into various distributed greedy algorithms. These algorithms are interpretable as producing sequences of embedded convex polytopes whose elements are increasingly sparse. To conclude this chapter, numerical experiments are presented that have been specifically chosen to complement the theoretical analysis associated with solving fixed-point problems by asynchronously implementing interconnective systems and to demonstrate the utility of the greedy processing instructions in designing frequency-selective filtering systems with sparse impulse responses.

4.1 | Constraint satisfaction and fixed-point problems

The general strategy behind many well-established signal and data processing algorithms is to formulate a behavioral description of the constraints associated with a problem, which are then used to incrementally refine an initial state into a space of solutions or set among which any solution is considered as good as any other. Gossip algorithms [64] and the methods used to perform spectral analysis and filtering for signal processing systems defined on graphs [65] are notable examples consistent with this. Mathematically, this strategy is consistent with iterating a collection of K relations $T_k\colon \mathbb{R}^K \to \mathbb{R}$, for $k = 1, \ldots, K$, according to a pseudo-iteration of the form

$$\begin{aligned}\mathbf{v}_1 \;&\leftarrow\; T_1((\mathbf{v}_1, \ldots, \mathbf{v}_K)) && \text{(4.1a)}\\ &\;\vdots \\ \mathbf{v}_K \;&\leftarrow\; T_K((\mathbf{v}_1, \ldots, \mathbf{v}_K)) && \text{(4.1b)}\end{aligned}$$

where the associated algorithm is formed by assigning an execution order to (4.1).

Referring to (4.1), standard successive approximation algorithms produce state sequences $\mathbf{v}^n$ by executing all K relations before incrementing n while Gauss-Seidel style algorithms sequentially execute the relations and increment n after each. In parallel computing environments where $\mathbf{v}$ is stored in globally accessible memory, asynchronous algorithms can execute the relations *ad infinitum* on different processors without concurrency rules. Common stopping criteria include identifying fixed-points $\mathbf{v}^\star = T(\mathbf{v}^\star)$ or after the size of $\mathbf{v}^{n+1} - \mathbf{v}^n$ falls below a threshold. To design time and frequency constrained filters, for example, the updates may represent orthogonal projections onto convex sets [66, 67]. Similarly, many parametric signal denoising algorithms apply sequences of oblique projectors until an adequately cleaned signal is identified in the intersection of the subspaces corresponding to the range of each operator [68]. Furthermore, many sparse signal recovery algorithms alternate linear and nonlinear operators until sufficiently sparse solutions are identified [69–71]. In this thesis, these and additional related problems are formulated using a constraint satisfaction viewpoint, and the interconnective framework is used to design algorithms for solving them.

Traditional constraint satisfaction problems (CSPs) are defined as triples $(\mathcal{X}, \mathcal{D}, \mathcal{C})$ where $\mathcal{X}$ denotes a set of variables, $\mathcal{D}$ denotes a set of corresponding domains over which the variables are defined, and $\mathcal{C}$ denotes a set of constraints between the variables. More formally, we define the elements of a CSP as

$$\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_K\} \tag{4.2}$$

$$\mathcal{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_K\} \tag{4.3}$$

$$\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_R\} \tag{4.4}$$

with each variable $\mathbf{x}_k$ satisfying $\mathbf{x}_k \in \mathcal{D}_k$, for $k = 1, \ldots, K$, and with each constraint $\mathcal{C}_r$, for $r = 1, \ldots, R$, being representable as a set constraint or relation imposed on a particular subset $\{\mathbf{x}_k\}$ of the variables in $\mathcal{X}$. Algorithms for solving CSPs, consistent with the strategy summarized by (4.1), seek to identify vectors $\mathbf{x}^\star$ satisfying the constraints $\mathcal{C}$ where no preference is given among the set of elements in $\mathcal{D}$ that jointly satisfy $\mathcal{C}$. In the remainder of this section, we define two types of CSPs and then spend the rest of the chapter developing methods to solve them. Before doing this, a brief survey of key features appearing in iterative algorithms for solving nonlinear systems of equations is provided.

4.1.1 | Fixed-point problems and iterative algorithms

A straightforward procedure for solving certain classes of nonlinear equations is to assemble a physical system that mimics the equations so that the steady state behavior of the system provides a solution. An analogous procedure was discussed in Section 3.3 where the interconnective description of the equations was assembled into an interconnective structure, and then memory was inserted to produce a well-defined iteration. The placement of memory in the structure is vital to whether repeatedly executing the corresponding iteration will produce a solution for certain initial conditions, since each placement can correspond to a different algorithm. In this subsection, we survey some of the recurring themes pertinent to understanding the stability and robustness of iterative algorithms. A more complete treatment of this review material can be found in standard texts such as [72, 73].

Let $T\colon \mathcal{V} \to \mathcal{V}$ denote a generally nonlinear operator and let $\mathcal{F}_T$ denote the associated fixed-point set, i.e.

$$\mathcal{F}_T \quad \triangleq \quad \{\mathbf{v}^\star \in \mathcal{V}\colon T(\mathbf{v}^\star) = \mathbf{v}^\star\}\,. \tag{4.5}$$

Questions pertaining to the existence of elements $\mathbf{v}^\star \in \mathcal{F}_T$ arise in many contexts, and procedures for generating them often must be iterative. A classic linear example is solving for roots of a quintic or higher order polynomial, which is equivalent to an eigenvalue problem using the polynomial's companion matrix, and which can only be solved iteratively. Similarly, the existence and uniqueness of solutions to certain ordinary differential equations correspond to examining fixed-point properties of related nonlinear operators. As is well-known, if an operator T is a contractive mapping, i.e. satisfies $\|T(\mathbf{x}) - T(\mathbf{y})\|_2 \leq \alpha \|\mathbf{x} - \mathbf{y}\|_2$ for some $\alpha \in [0, 1)$ and all $\mathbf{x}, \mathbf{y} \in \mathcal{V}$, then the Banach fixed-point theorem [33, Theorem 9.23] ensures that a unique fixed-point $\mathbf{v}^\star$ exists, and that the simple iteration

$$\mathbf{v}^n \quad = \quad T(\mathbf{v}^{n-1}), \qquad n \in \mathbb{N} \tag{4.6}$$

converges linearly[1] to the solution $\mathbf{v}^\star$ independent of $\mathbf{v}^0$. When an operator T is not guaranteed to be contractive, two main issues arise. Namely, the fixed-point set $\mathcal{F}_T$ may be empty, and even when it is not, the iteration (4.6) may not converge. Assuming a fixed-point exists, a common remedy is to modify the iteration in (4.6) by using a sequence of iterations described by the maps $g^{(n)} \colon \mathcal{V} \times \mathcal{V} \to \mathcal{V}$, for $n \in \mathbb{N}$, where the n-th iteration takes the form

$$\mathbf{v}^n \quad = \quad g^{(n)}\left(T(\mathbf{v}^{n-1}), \mathbf{v}^{n-1}\right), \qquad n \in \mathbb{N}. \tag{4.7}$$

When the update iteration can be described using a map $g \colon \mathcal{V} \times \mathcal{V} \to \mathcal{V}$ that is independent of n, the corresponding iteration is referred to as *pure* or *stationary* and takes the form

$$\mathbf{v}^n \quad = \quad g\left(T(\mathbf{v}^{n-1}), \mathbf{v}^{n-1}\right), \qquad n \in \mathbb{N}. \tag{4.8}$$

Key issues in designing iterations like (4.7) and (4.8) include requiring that the update map only has stationary points equal to $\mathcal{F}_T$ and that the iterative process converges to one such point. Concerns also include improving the rate of this convergence as well as the overall numerical stability and robustness of the iteration.

When fixed-point or constraint satisfaction problems admit both direct and iterative methods, iterative algorithms sometimes offer advantages that direct algorithms cannot. Figure 4-1 illustrates two potential advantages iterative algorithms provide in solving linear systems of equations $T\mathbf{v} = \mathbf{u}$ where T is a $K \times K$ matrix. First, iterative methods generally provide approximate solutions after small increments of work are completed and may be

[1] Linear convergence refers to the exponent in the distance to the fixed-point, describing precision gained per iteration for floating-point number systems. More formally, a sequence $\mathbf{x}^n$ converges linearly to $\mathbf{x}^\star$ if $\lim_{n\to\infty} \frac{\|\mathbf{x}^n - \mathbf{x}^\star\|}{\|\mathbf{x}^{n-1} - \mathbf{x}^\star\|} \in (0, 1)$.

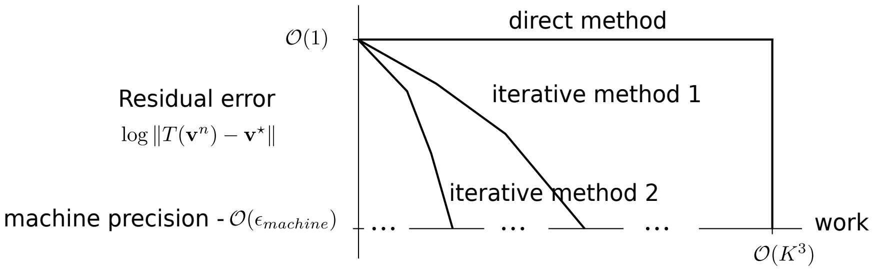


Figure 4-1: A schematic illustration adapted from [2, Figure 32.1] comparing the residual $\log\|T(\mathbf{v}^n) - \mathbf{v}^\star\|$ after fixed amounts of computation for three methods: one direct and two iterative. The conditioning of the $K \times K$ matrix T is assumed to be unity so that machine precision $\mathcal{O}(\epsilon_{machine})$ is achievable.

terminated early if limited precision is required in the solution. Second, iterative methods are able to solve for particular vectors $\mathbf{u}$ or particular problem instances more generally rather than solving for arbitrary vectors $\mathbf{u}$ and therefore can require less work than direct methods do for certain matrices T. This is equivalent to trading computational recyclability for efficiency as is often the case with linear Krylov methods [74]. As was mentioned earlier, for many nonlinear and transcendental problems, direct method do not always exist.

4.1.2 | Conservative constraint satisfaction problems

The first type of CSP we focus on in this chapter relates fixed-point problems to the scattering-form interconnective description introduced in Section 3.4.1.

Definition 4.1.1 (Conservative constraint satisfaction problem)**.** *A conservative constraint satisfaction problem (CCSP) is defined as being reducible to a CSP described by a triple* $(\mathcal{X},\ \mathcal{D},\ \mathcal{C})$ *having elements that take the following form:*

$$\mathcal{X} = \{(\mathbf{c}_1, \ldots, \mathbf{c}_K), (\mathbf{d}_1, \ldots, \mathbf{d}_K)\} \tag{4.9}$$

$$\mathcal{D} = \{\mathbb{R}^K,\ \mathbb{R}^K\} \tag{4.10}$$

$$\mathcal{C} = \{W,\ \mathcal{M}\} \tag{4.11}$$

where W *and* $\mathcal{M}$ *are each constraints imposed on the entire set of variables* $(\mathbf{c}, \mathbf{d})$*, and where*

W in particular is a linear subspace of $\mathbb{R}^K \times \mathbb{R}^K$ that satisfies the following property:

$$\|\mathbf{c}\|^2 - \|\mathbf{d}\|^2 = 0, \qquad \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix} \in W. \tag{4.12}$$

The interconnective equivalence classes defined in Section 3.2.3 provide a straightforward way to identify whether a particular CSP either is or can be transformed into a CCSP. This is done by first representing the CSP using an interconnective description and then changing coordinate maps. More broadly, algebraic reductions and manipulations beyond coordinate transforms may be used to reduce an initial CSP into a form that satisfies (4.9) through (4.12) where solutions to the CCSP can be used to generate solutions to the initial problem. For example, determining steady-state voltage and current distributions in linear or nonlinear electrical networks is reducible to a CCSP since electrical networks can be cast into canonical-form structures as defined in Section 3.4.2. In addition, we develop a class of conservative optimization problems in Chapter 5 that serve as a large and important class of problems that reduce to CCSPs. In that context, additional algebraic techniques are introduced to reduce the dimensionality of a CCSP.

There are many algebraic descriptions consistent with the form in (4.9) through (4.12) that can be used to describe a particular CCSP. In light of this, we focus our attention in this chapter to CCSPs for which the set constraints W and $\mathcal{M}$ can be realized using respective functional relationships specified as

$$W = \left\{ \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix} \in \mathbb{R}^K \times \mathbb{R}^K : \mathbf{d} = Q\mathbf{c} \right\}, \tag{4.13}$$

where $Q\colon \mathbb{R}^K \to \mathbb{R}^K$ denotes an orthogonal matrix and

$$\mathcal{M} = \left\{ \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix} \in \mathbb{R}^K \times \mathbb{R}^K : \mathbf{c} = m(\mathbf{d}) \right\}, \tag{4.14}$$

where $m\colon \mathbb{R}^K \to \mathbb{R}^K$ denotes a generally nonlinear, memoryless operator. With these descriptions in place, the techniques for realizing behaviors discussed in Section 3.3.1 can be used to obtain these functions for a given CCSP.

From the characterization of the constraint sets in (4.13) and (4.14), it follows that an equivalent statement of a CCSP can be posed by making use of the operators Q and m. In particular, consider the fixed-point problem of identifying any vector $(\mathbf{c}^\star, \mathbf{d}^\star) \in \mathbb{R}^K \times \mathbb{R}^K$ which satisfies the implicit and often transcendental system of equations

$$\mathbf{d}^\star = Q\mathbf{c}^\star \tag{4.15}$$

$$\mathbf{c}^\star = m(\mathbf{d}^\star). \tag{4.16}$$

Sufficient conditions under which asynchronously implementing the interconnective description of (4.15) and (4.16) as a scattering-form system successfully produces a solution $(\mathbf{c}^\star, \mathbf{d}^\star)$ is the primary focus of Sections 4.2 and 4.3. A key link between this approach and the iterative methods summarized by (4.1) is that the execution order and state exchange protocol are given by the asynchronous processing protocol in Definition 3.3.1.

4.1.3 | Sparse constraint satisfaction problems

Sparsity principles, in a broad sense, are often associated with desirable properties spanning a wide range of signal processing and computational applications including sampling theory, model-order reduction, and the design of power efficient systems and architectures. The second type of CSP we focus on in this chapter is closely related to many classical sparse recovery problems.

Definition 4.1.2 (Sparse constraint satisfaction problem)**.** *A sparse constraint satisfaction problem (SCSP) is defined as being reducible to a CSP described by a triple* $(\mathcal{X}, \mathcal{D}, \mathcal{C})$ *having elements that take the following form:*

$$\mathcal{X} = \{(\mathbf{v}_1, \ldots, \mathbf{v}_K)\} \tag{4.17}$$

$$\mathcal{D} = \{\mathbb{R}^K\} \tag{4.18}$$

$$\mathcal{C} = \{\mathcal{S}, \mathcal{T}\} \tag{4.19}$$

where $\mathcal{S}$ *and* $\mathcal{T}$ *are each constraints imposed on the entire set of variables* $\{\mathbf{v}\}$*, and where*

S is a generally closed convex set in $\mathbb{R}^K$ and $\mathcal{T}$ imposes a sparsity constraint of the form:

$$\|\mathbf{v}\|_0 \leq t \tag{4.20}$$

where $\|\cdot\|_0$ indicates the number of non-zero entries of its argument and t is an integer satisfying $0 \leq t \leq K$.

In the most general case, solving sparse CSPs for either maximally sparse or t-sparse solutions is a computationally difficult problem [75]. Often the smallest value of t for which $\mathcal{S}$ contains a t-sparse element is unknown, so the non-convex optimization problem statement

$$\begin{aligned} \mathbf{v}^\star \in \arg \underset{\mathbf{v}}{\text{minimize}} \quad & \|\mathbf{v}\|_0 \\ \text{s.t.} \quad & \mathbf{v} \in \mathcal{S} \end{aligned} \tag{4.21}$$

is used instead. Conventional algorithms for solving these problems can be categorized into two classes: convex optimization algorithms and greedy iterative methods. When $\mathcal{S}$ is a subspace whose matrix realization satisfies certain constraint qualification conditions, e.g. the restricted isometry property or coherence conditions, recovery guarantees for particular algorithms can often be made. For example, (4.21) and a convex relaxation of the cost function to the 1-norm have been shown to produce identical solutions [3]. The verification of such qualifications, however, is often itself computationally expensive or intractable. More importantly, several problems of interest to the signal processing community and elsewhere do not meet such constraint qualifications, e.g. the design of frequency-selective sparse impulse responses. Specialized algorithms that make use of heuristics and/or side information are often utilized instead and may lead to sufficiently sparse solutions for these problems.

Understanding the conditions under which various guarantees are possible as well as the grace with which such assurances degrade when the conditions start to be violated has been the focus of much attention in the literature, especially surrounding the performance of greedy methods. In fact, greedy methods often probably result in maximally sparse solutions for problems meeting certain constraint qualifiers [76]. Well-known algorithms tend to sequentially decrease the sparsity of a single solution in agreement with a suitable error metric until a particular stopping criterion is met. For example, (orthogonal) matching pursuit methods are of this type [77]. Another trend, encompassing methods such as iterative hard thresholding (IHT), subspace pursuit (SP), and compressive sampling matching pursuit (CoSaMP), is to iteratively threshold a single dense solution until a predetermined sparsity level while simultaneously attempting to decrease an error metric [69–71].

In Section 4.4, several generic processing instructions are developed that can be assembled into various greedy algorithms for solving SCSPs for fixed values of t or for solving (4.21) for the smallest possible t. To do this, the approach is to monotonically increase the sparsity of a bunch of elements via tree-based processing where feasibility with respect to $\mathcal{S}$ is maintained for every element at every stage. These greedy algorithms are easy to distribute and do not require constraint qualifiers, and therefore are of use when such qualifications either fail or cannot be verified. Furthermore, side information or heuristic knowledge on the sparsity pattern of optimal solutions is easy to incorporate into these algorithms with only minimal adjustment.

4.2 | Solving CCSPs using asynchronous scattering algorithms

In modeling systems in the physical sciences, conservation principles often underlie the predictability and stability of system behavior at macroscopic scales, thereby eliminating a need for careful analysis of the interactions between features at the subsystem levels. Conservation of energy, for example, emerges in Hamiltonian physics as a consequence of the time-invariance of physical principles governing isolated bodies as described by Noether's theorem and allows for the analysis of many physical processes without detailed handling of time-valued boundary conditions. As previously discussed in Section 2.4, conservation principles in signal processing systems are often far removed from having physical interpretations, due in part to the fact that the system variables are not required to have physical meanings and the system dynamics are not subject to the laws of physics. However, as we emphasize in this chapter and the next, the benefits provided by their existence are not readily diminished by the lack of a physical reference system.

The main purpose of this section is to define several categories of system operators, as well as modifications thereof that are consistent with (4.7) and (4.8), in anticipation

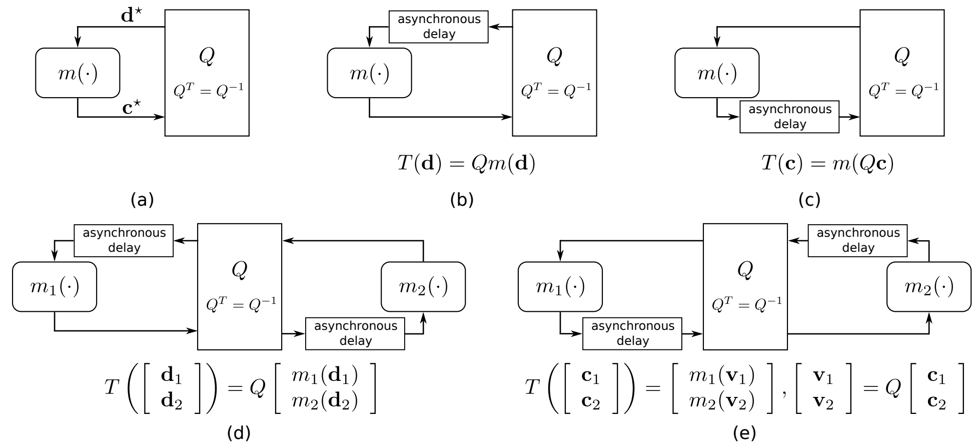

$\mathbf{d}^\star$
$\mathbf{c}^\star$
$m(\cdot)$
Q
$Q^T = Q^{-1}$
(a)
asynchronous delay
$T(\mathbf{d}) = Qm(\mathbf{d})$
(b)
$T(\mathbf{c}) = m(Q\mathbf{c})$
(c)
$m_1(\cdot)$
$m_2(\cdot)$
$T\left(\begin{bmatrix}\mathbf{d}_1\\ \mathbf{d}_2\end{bmatrix}\right) = Q\begin{bmatrix}m_1(\mathbf{d}_1)\\ m_2(\mathbf{d}_2)\end{bmatrix}$
(d)
$T\left(\begin{bmatrix}\mathbf{c}_1\\ \mathbf{c}_2\end{bmatrix}\right) = \begin{bmatrix}m_1(\mathbf{v}_1)\\ m_2(\mathbf{v}_2)\end{bmatrix}, \begin{bmatrix}\mathbf{v}_1\\ \mathbf{v}_2\end{bmatrix} = Q\begin{bmatrix}\mathbf{c}_1\\ \mathbf{c}_2\end{bmatrix}$
(e)

Figure 4-2: (a) The description of a CCSP as a scattering-form interconnective structure. (b)-(e) Example scattering algorithms formed by inserting memory to form asynchronous processing systems. The system operator associated with each algorithm is also provided.

of the subsequent stability and robustness analysis in Section 4.3. In discussing this, we shall refer to the description of a CCSP illustrated in Figure 4-2(a) as a scattering-form structure. The strategy in using this structure to solve CCSPs is specifically to implement the associated scattering algorithm corresponding to interconnective systems like those in Figure 4-2(b) through (e), where, in steady-state, the processing system can be replaced by the structure in (a), which graphically represents a solution to the fixed-point problem in (4.15) and (4.16). Referring to the definition of scattering-form interconnective systems in Section 3.4.3, the structures depicted in Figure 4-2(d) and (e) demonstrate additional separability of their partition decompositions using a single direct product decomposition of the terminal space and generalize inductively to any number. Concerning the deployment of asynchronous scattering algorithms in practice, the modifications discussed in this section are easily reflected into actual processing systems and require minimal adjustments to the system. The details associated with doing this are provided in Chapter 6.

4.2.1 | Categorization of system operators

In the sequel, the convergence of an asynchronous algorithm is closely related to stability and robustness properties of the system operator associated with the algorithms interconnective description. In this subsection, a broad class of system operators are categorized to assist with this analysis. We initially restrict our attention to those interconnective structures whose system operators satisfy the following definition.

Definition 4.2.1 (α-conic system operators)**.** *A system operator $T\colon \mathbb{R}^K \to \mathbb{R}^K$ is referred to as being α-conic about $\mathbf{v} \in \mathbb{R}^K$ provided a non-negative constant α exists such that*

$$\sup_{\mathbf{u}\neq\mathbf{0}} \frac{\|T(\mathbf{v}+\mathbf{u}) - T(\mathbf{v})\|}{\|\mathbf{u}\|} \leq \alpha. \tag{4.22}$$

If T satisfies (4.22) *for all $\mathbf{v} \in \mathbb{R}^K$ then T is referred to as being α-conic everywhere.*

We briefly call attention to the relationship between (4.22) and existing notions of continuity. In particular, the assumption that T is α-conic everywhere is sufficient to conclude that T is also Lipschitz continuous with Lipschitz constant α whereas the assumption that T is α-conic about $\mathbf{v}$ is weaker since bounded discontinuities away from $\mathbf{v}$ are admissible. To see this, observe that the inequality $\|T(\mathbf{v}+\mathbf{u})-T(\mathbf{v})\| \leq \alpha\|\mathbf{u}\|$ is implied by (4.22) since it is assumed that T is α-conic for all $\mathbf{v} \in \mathbb{R}^K$ and that α is the least upper bound for all $\mathbf{u} \neq \mathbf{0}$. Rewriting this inequality using the translated arguments $\mathbf{v}'$ and $\mathbf{u}'$ where $\mathbf{v}' = \mathbf{v}+\mathbf{u}$ and $\mathbf{u}' = \mathbf{v}$ yields

$$\|T(\mathbf{v}') - T(\mathbf{u}')\| \leq \alpha\|\mathbf{v}' - \mathbf{u}'\|, \qquad \mathbf{v}', \mathbf{u}' \in \mathbb{R}^K \tag{4.23}$$

hence (4.23) is a sufficient condition for T to be α-conic everywhere with the same constant.

Next, the class of α-conic system operators are divided into three groups that, roughly speaking, are used to organize the stability and robustness conditions presented in Sections 4.3.2 through 4.3.4. The nomenclature chosen to describe these categories is consistent with the the classes of functions defined by Zames in [1, 5] to discuss the stability of certain feedback control systems.

Definition 4.2.2 (α-dissipative, passive, and α-expansive system operators)**.** *Let $T\colon \mathbb{R}^K \to \mathbb{R}^K$ denote a system operator that is α-conic about $\mathbf{v}$. Then T is referred to as being:*

(i) α-dissipative about $\mathbf{v}$ provided $\alpha \in [0, 1)$;

(ii) passive about $\mathbf{v}$ provided $\alpha = 1$;

(iii) α-expansive about $\mathbf{v}$ provided $\alpha > 1$.

Furthermore, if T is α-conic everywhere, i.e. satisfies (4.22) *for all* $\mathbf{v} \in \mathbb{R}^K$, *then T is referred to as being:*

(i) α-dissipative everywhere provided $\alpha \in [0, 1)$;

(ii) passive everywhere provided $\alpha = 1$;

(iii) α-expansive everywhere provided $\alpha > 1$.

Let $T^{(i)} : \mathbb{R}^K \to \mathbb{R}^K$ denote a system operator that is α_i-conic everywhere for $i = 1, \ldots, m$. By induction it follows that the composite system operator $T^{(m)} \circ \cdots \circ T^{(1)}$ is also α-conic everywhere with parameter $\alpha = \prod_{i=1}^{m} \alpha_i$. This fact is important to determining the conic parameter of system operators derived from interconnective graphs where the interconnecting network and constitutive modules are both realized by α-conic operators. For example, the composition of any number passive everywhere system operators with a single dissipative operator results in an overall dissipative composite operator.

4.2.2 | Adaptive and homotopy system operators

A common technique used to generate a fixed point of a nonlinear operator, and that is consistent with the iteration-dependent methods summarized by (4.7), is to parameterize the nonlinear operator using a so-called homotopy map to which an associated numerical continuation scheme is then applied [78]. The basic idea behind this approach is to continuously deform a well-behaved operator into the original nonlinear operator, and to generate a sequence of intermediate fixed points that terminates when a fixed point of the original operator is obtained. The map used to describe these deformations is referred to as a *homotopy map*. Often the intermediary fixed points are only approximately determined. In this thesis, we view this continuation scheme through the lens of traditional adaptive system techniques [79] where we additionally restrict ourselves to the simple analytic form of a homotopy map appearing in the following definition, although alternative homotopy maps may generally be used by the same continuation scheme in practice.

Definition 4.2.3 (Adaptive system operator T_a)**.** *Let $T^{(0)} : \mathbb{R}^K \to \mathbb{R}^K$ and $T^{(1)} : \mathbb{R}^K \to \mathbb{R}^K$ denote two system operators. The homotopy map or adaptive system operator $T_a : [0, 1] \times$*

$\mathbb{R}^K \to \mathbb{R}^K$ *associated with* $T^{(1)}$ *is defined as*

$$T_a\,(\rho, \mathbf{v}) \;\;\triangleq\;\; \rho T^{(1)}(\mathbf{v}) + (1-\rho)T^{(0)}(\mathbf{v}). \tag{4.24}$$

The family of fixed-point problems associated with T_a is described by $T_a(\rho, \mathbf{v}) = \mathbf{v}$ for $\rho \in [0, 1]$, and the standard continuation strategy is to track the fixed points $\mathbf{v} = T_a(\rho, \mathbf{v})$ starting from $(\rho, \mathbf{v}) = (0, \mathbf{v}^*)$ as ρ progresses from 0 to 1, where $\mathbf{v}^*$ is an easily obtained fixed point of $T^{(0)}$. Typical implementations discretize the interval $[0, 1]$ and sequentially increment ρ by a sufficiently small amount where, under some mild regularity conditions, e.g., Lipschitz continuity, the procedure results in a pair $(\rho, \mathbf{v}) = (1, \mathbf{v}^\star)$ where $T^{(1)}(\mathbf{v}^\star) = \mathbf{v}^\star$. Thus any fixed point of the adaptive system operator $T_a(1, \cdot)$ is also a fixed point of $T^{(1)}$, as desired. Sections 5.4 and 6.1.1 use this method to decentralize an interconnecting network.

4.2.3 | Filtered system operators

Selecting the simple system operator $T^{(0)}$ to be the identity operator on $\mathbb{R}^K$ results in an adaptive system operator T_a and an associated numerical continuation scheme where a fixed point of $T^{(1)}$ is immediately obtained after generating any fixed point of $T_a(\rho, \cdot)$ for any non-zero value of the parameter ρ. So, in light of this observation and the fact that it remains true if ρ is not constrained to the unit interval, we next develop a stationary iteration consistent with (4.8) which subsumes these observations.

Definition 4.2.4 (Filtered system operator T_f)**.** *Let* $T\colon \mathbb{R}^K \to \mathbb{R}^K$ *denote a system operator. The filtered system operator* $T_f\colon \mathbb{R} \times \mathbb{R}^K \to \mathbb{R}^K$ *is defined as the operator whose action is to take an affine combination of* $\mathbf{v}$ *and* $T(\mathbf{v})$ *of the form*

$$T_f\,(\rho, \mathbf{v}) \triangleq \rho T(\mathbf{v}) + (1-\rho)\mathbf{v} \tag{4.25}$$

where ρ *is referred to as the filtering parameter.*

We briefly justify the motivating observation above by fixing the filtering operator $T_f(\rho, \cdot)\colon \mathbb{R}^K \to \mathbb{R}^K$ for a given non-zero value of ρ. Let $\mathbf{v}^\star$ denote a fixed point of $T_f(\rho, \cdot)$, i.e. $\mathbf{v}^\star \in \mathcal{F}_{T_f(\rho,\cdot)}$. After some straightforward manipulations to the fixed-point equation

$T_f(\rho, \mathbf{v}^\star) = \mathbf{v}^\star$ we obtain the fixed-point equation $T(\mathbf{v}^\star) = \mathbf{v}^\star$. Thus, it follows that any fixed point of $T_f(\rho, \cdot)$ is also a fixed point of T. The analysis holds for any non-zero value of the parameter ρ and the converse of this fact is also true.

Numerical iterations resembling the direct and synchronous implementation of a filtered system operator appear in many signal processing contexts where, instead of taking a weighted average or convex combination (if $\rho \in [0, 1]$) of a system's state, the operator instead acts on design parameters defining a real-time system. In these cases, the filtering parameter ρ is reminiscent of the so-called forgetting factor used to stabilize numerous weight adjustment protocols using in adaptive filtering systems. For example, least mean squares filters form a special case of the stochastic gradient descent algorithm discussed in Section 2.5.2 where the step size parameter plays the role of the filtering parameter in (4.25).

In the following lemma, we characterize the distance of a filtered system operator applied to an arbitrary state to any of its fixed points, and afterwards we make a connection between this characterization and well-known results in Euclidean geometry.

Lemma 4.2.1 (Distance of solutions to filtered system operators). *Let T denote a system operator with a non-empty fixed-point set $\mathcal{F}_T$. The distance of the continuum of states that are achievable by applying T_f to an arbitrary element $\mathbf{v}$ to any fixed point $\mathbf{v}^\star \in \mathcal{F}_T$ is given by*

$$\|T_f(\rho, \mathbf{v}) - \mathbf{v}^\star\|^2 = \rho\|T(\mathbf{v}) - \mathbf{v}^\star\|^2 + (1-\rho)\|\mathbf{v} - \mathbf{v}^\star\|^2 - \rho(1-\rho)\|T(\mathbf{v}) - \mathbf{v}\|^2. \quad (4.26)$$

Proof. The assertion above is justified by performing the following manipulations:

$$\begin{aligned}
\|T_f(\rho, \mathbf{v}) - \mathbf{v}^\star\|^2 &= \rho^2\|T(\mathbf{v}) - \mathbf{v}^\star\|^2 + (1-\rho)^2\|\mathbf{v} - \mathbf{v}^\star\|^2 && (4.27)\\
&\quad +2\rho(1-\rho)\langle T(\mathbf{v}) - \mathbf{v}^\star, \mathbf{v} - \mathbf{v}^\star\rangle \\
&= (\rho - \rho(1-\rho))\|T(\mathbf{v}) - \mathbf{v}^\star\|^2 + [(1-\rho) - \rho(1-\rho)]\|\mathbf{v} - \mathbf{v}^\star\|^2 && (4.28)\\
&\quad +2\rho(1-\rho)\langle T(\mathbf{v}) - \mathbf{v}^\star, \mathbf{v} - \mathbf{v}^\star\rangle \\
&= \rho\|T(\mathbf{v}) - \mathbf{v}^\star\|^2 + (1-\rho)\|\mathbf{v} - \mathbf{v}^\star\|^2 && (4.29)\\
&\quad -\rho(1-\rho)[\|T(\mathbf{v}) - \mathbf{v}^\star\|^2 + \|\mathbf{v} - \mathbf{v}^\star\|^2 - 2\langle T(\mathbf{v}) - \mathbf{v}^\star, \mathbf{v} - \mathbf{v}^\star\rangle] \\
&= \rho\|T(\mathbf{v}) - \mathbf{v}^\star\|^2 + (1-\rho)\|\mathbf{v} - \mathbf{v}^\star\|^2 - \rho(1-\rho)\|T(\mathbf{v}) - \mathbf{v}\|^2 && (4.30)
\end{aligned}$$

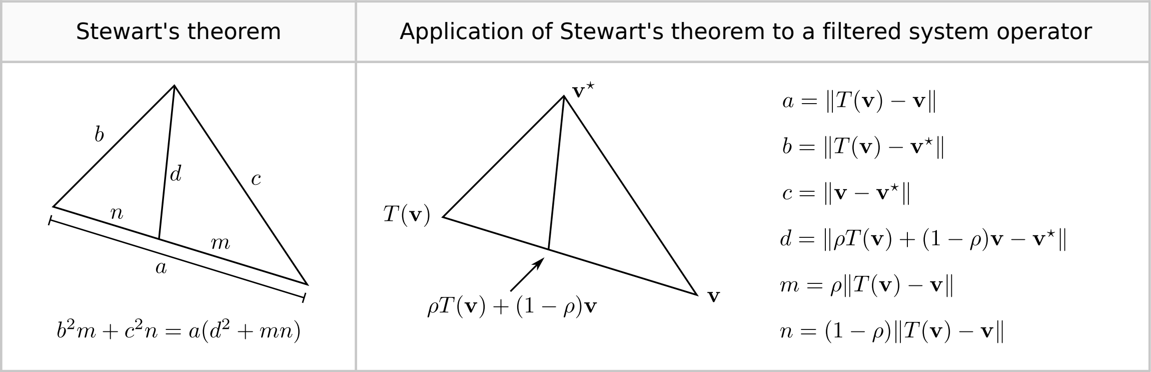

Stewart's theorem
Application of Stewart's theorem to a filtered system operator
b
d
c
n
m
a
$b^2m + c^2n = a(d^2 + mn)$
$\mathbf{v}^\star$
$T(\mathbf{v})$
$\mathbf{v}$
$\rho T(\mathbf{v}) + (1-\rho)\mathbf{v}$
$a = \|T(\mathbf{v}) - \mathbf{v}\|$
$b = \|T(\mathbf{v}) - \mathbf{v}^\star\|$
$c = \|\mathbf{v} - \mathbf{v}^\star\|$
$d = \|\rho T(\mathbf{v}) + (1-\rho)\mathbf{v} - \mathbf{v}^\star\|$
$m = \rho\|T(\mathbf{v}) - \mathbf{v}\|$
$n = (1-\rho)\|T(\mathbf{v}) - \mathbf{v}\|$

Figure 4-3: An illustration of Stewart's theorem (left) and the application of Stewarts theorem to the filtered system operator T_f (right) where the vertices of the triangle correspond to an arbitrary state $\mathbf{v}$, the application of T to $\mathbf{v}$ and a fixed-point $\mathbf{v}^\star$.

where the first equality is due to the fact $\|\mathbf{v} + \mathbf{u}\|^2 = \|\mathbf{v}\|^2 + \|\mathbf{u}\|^2 + 2\langle \mathbf{v}, \mathbf{u}\rangle$, the second equality is due to the fact $\rho^2 = \rho - \rho(1-\rho)$ and $(1-\rho)^2 = 1 - 2\rho + \rho^2 = (1-\rho) - \rho(1-\rho)$, and the fourth equality is due to the fact $\|\mathbf{v} - \mathbf{u}\|^2 = \|\mathbf{v}\|^2 + \|\mathbf{u}\|^2 - 2\langle \mathbf{v}, \mathbf{u}\rangle$. We reiterate that the identity in (4.26) holds for any real value of ρ. ■

A geometric interpretation of the distance in (4.26) with the filtering parameter restricted to the interval $\rho \in [0, 1]$ is established through a connection with Stewart's theorem from Euclidean geometry [80, Theorem 3]. In particular, Stewart's theorem, illustrated in the left pane of Figure 4-3, states that a given triangle with sides of lengths a, b, and c, and a cevian to the size of length a with length d, where the cevian divides a into two pieces of lengths n and m where m is adjacent to c and n is adjacent to d satisfy the relationship $b^2m + c^2n = a(d^2 + mn)$ which we rewrite prophetically as

$$d^2 = \frac{m}{a}b^2 + \frac{n}{a}c^2 - mn. \tag{4.31}$$

By proper assignment of the vertices of the triangle (listed in Figure 4-3 for completeness), the identity (4.30) follows immediately from (4.31). As a further connection, the special case of (4.30) for $\rho = \frac{1}{2}$ is then interpretable by a similar application of Apollonius' theorem [80, Exercise 3.5]. The fact that the continuum of states achievable by applying T_f forms the collinear subspace to the side of the triangle opposite the fixed-point vertex underlies the stability analysis of filtered system operators considered in Sections 4.3.3 and 4.3.4.

4.3 | Stability and robustness of interconnective structures

As asynchronous algorithms organized into distributed signal processing systems become increasingly used to solve large-scale constraint satisfaction and fixed-point problems, it will also become increasingly important to identify sufficient conditions that are easy to certify in practice to guarantee that these algorithms work. In this section, we study the convergence of asynchronous algorithms through properties of the system operator derived from their interconnective description. In particular, convergence of the algorithm corresponds to the stability and robustness of the interconnective system, and the conditions are therefore straightforward to identify in many signal processing systems encountered in practice.

In the context of designing asynchronous algorithms as interconnective systems, conditions that allow local and independent certificates of the individual subsystems to provide global stability and robustness guarantees are especially desirable. Algorithms whose interconnective descriptions are in scattering form, for example, allow the certificates derived in this section to be used in practice by certifying only the individual constitutive modules, and do not need the interconnecting network to be certified or aggregated when decentralized. Some of these conditions have appeared previously in [14] and preliminaries to those appeared in [81]. In addition, some of these conditions generalize known deterministic convergence results corresponding to synchronous and unfiltered implementations, respectively generated by setting $\rho = 1$ and $p = 1$ in (4.25) and (3.47).

4.3.1 | Stability and robustness definitions

Stability conditions in signal processing and control theory often refer to various criteria that ensure the output of a system remains bounded in response to any bounded input. In the context of solving fixed-point problems using globally clocked, synchronous implementations of an interconnective system, the system will be called stable if it's terminal variables tend to an equilibrium state, i.e. a state that is invariant to the dynamics of the iteration. Equivalently, stability can be formulated in terms of the associated system operator by connecting the iteration $\mathbf{v}^{n+1} = T(\mathbf{v}^n)$ to classical notions of convergence of the state sequence to a fixed-point. More concretely, let let $T\colon \mathbb{R}^K \to \mathbb{R}^K$ denote the system operator associated with an interconnective system and let $\mathbf{v}^n$, for $n \in \mathbb{N}$, denote the associated state sequence. Then the system operator is said to be r-stable if the state sequence satisfies the condition $\lim_{n\to\infty} \|\mathbf{v}^n - \mathbf{v}^\star\|^r = 0$ where $\mathbf{v}^\star$ is a fixed-point of T and the norm is the standard 2-norm.

Robust signal processing and feedback control systems typically refer to systems that perform properly when uncertainty is accounted for in some way. For example, uncertainty can be incorporated into the system operator itself [82] or through disturbances to the iterations dynamics [42]. In viewing asynchronous algorithms as interconnective systems, robustness properties will essentially refer to stability properties when the subsystem modules communicate using asynchronous protocols. This is consistent with uncertainty being incorporated through random deviations from the nominal synchronous operating mode. Indeed, an interconnective system will be referred to as being r-robust if it is r-stable when the delay modules in the interconnective structure behave as coordinatewise discrete-time sample-and-hold systems triggered by independent Bernoulli processes. To avoid being pedantic in stating synchronous stability properties separate from asynchronous robustness properties, we proceed to use the term *stability* to refer to both. In particular, stability will refer to the stochastic convergence of the state sequence produced using the asynchronous iteration in (3.47), where the convergence is specifically of the sequence of random vectors. The synchronous result is a the straightforward special case of the robustness property where the delay probability parameter p is set to $p = 1$.

Definition 4.3.1 (Stability in r-th mean)**.** *Let $T\colon \mathbb{R}^K \to \mathbb{R}^K$ denote the system operator associated with an interconnective structure. Then T is defined as being stable in r-th mean if the sequence $\{\mathbf{v}^n\colon n \in \mathbb{N}_0\}$ produced by the asynchronous protocol* (3.47) *satisfies*

$$\lim_{n\to\infty} \mathbb{E}\left[\|\mathbf{v}^n - \mathbf{v}^\star\|^r\right] = 0 \tag{4.32}$$

for some state $\mathbf{v}^\star$ that is a fixed point of T, i.e. $\mathbf{v}^\star \in \mathcal{F}_T$. When (4.32) *holds it is written succinctly as $\mathbf{v}^n \xrightarrow{r} \mathbf{v}^\star$.*

A straightforward consequence that results from application of Jensen's inequality to (4.32) is that any system operator that is stable in r-th mean is also stable in t-th mean for all t satisfying $1 \leq t \leq r$ [83]. Indeed, stability in the sense of (4.32) provides additional insight into the dynamics of stable asynchronous algorithms. In particular, convergence of the state sequence $\{\mathbf{v}^n \colon n \in \mathbb{N}_0\}$ in mean square ($r = 2$) implies both that a subsequence of $\mathbf{v}^n$ converges almost surely and that the state sequence itself converges in probability and distribution. We note separately that convergence in r-th mean may or may not coincide with convergence of the complete sequence almost surely. When a filtered system operator T_f is stable in r-th mean, the original system operator T that T_f was derived from is also stable in r-th mean if T_f is stable for a range of filter parameters that includes $\rho = 1$.

With formal definitions of stability and robustness in place, we remark that an asynchronous algorithm can generally reduce its overall communication bandwidth by only performing computation of the subsystems corresponding to the subset delays that trigger on each iteration n. This property is desirable in distributed computing environments and is readily exploited in Chapter 6 in the context of heterogeneous networks and uncoordinated processing nodes. As a straightforward example, if T is an arbitrary nonlinear function operating on each entry of $\mathbf{v}$ independently, then an efficient asynchronous implementation of T only computes the subset of coordinates indicated by the ones in the matrix $D^{(p)}$ for each n rather than computing all K and discarding the complement of that subset, as is suggested by the form of the iteration in (3.47).

In anticipation of the analysis of filtered asynchronous algorithms implemented using the stochastic update mechanism in (3.47), we next state a useful identity that will frequently appear in our analysis. Specifically, for any value $r \geq 1$, it follows that

$$\mathbb{E}[\|\mathbf{v}^n - \mathbf{v}^\star\|^r] = p\mathbb{E}[\|\mathbf{v}^n - \mathbf{v}^\star\|^r \mid D^{(p)} = I_K] + (1-p)\mathbb{E}[\|\mathbf{v}^n - \mathbf{v}^\star\|^r \mid D^{(p)} = 0] \quad (4.33)$$

$$= p\mathbb{E}[\|T_f(\rho, \mathbf{v}^{n-1}) - \mathbf{v}^\star\|^r] + (1-p)\mathbb{E}[\|\mathbf{v}^{n-1} - \mathbf{v}^\star\|^r] \quad (4.34)$$

where the first equality is due to the law of total expectation. The first term in (4.34) can be appropriately modified when the filtered operator T_f is instead replaced by a different system operator like T_h. Finally, we define $B(\mathbf{c}, r)$ as the non-empty, closed Euclidean ball or basin of radius $r > 0$ and center $\mathbf{c} \in \mathbb{R}^K$ given by

$$B(\mathbf{c}, r) \triangleq \{\mathbf{v} \in \mathbb{R}^K \colon \|\mathbf{c} - \mathbf{v}\| \leq r\}. \quad (4.35)$$

4.3.2 | Dissipative system operators

The first category of system operators we address include those that are α-dissipative in some manner. For each stability and robustness condition presented, the approach we take is to ensure that a unique fixed-point exists or that a fixed point is unique when one has been assumed to exist, and that the state sequence produced by the asynchronous algorithm does, in fact, tend to the fixed-point under the given assumptions. Note that the existence and uniqueness of a fixed-point is a property belonging to the system operator itself, and is independent of any particular iteration or analysis used to establish it.

To begin, consider the class of system operators that are α-dissipative about a state $\mathbf{v}$. Note that this restriction does not preclude a system operator from being α-conic about additional states for values of $\alpha \geq 1$ nor does it require continuity of a system operator around any states other than $\mathbf{v}$. This flexibility allows for the stability and robustness of an interconnective structure to be understood when the associated system operator possesses discontinuities and other similar behavior bounded away from $\mathbf{v}$. The following two theorems summarize stability and robustness properties for these system operators when $\mathbf{v}$ is and is not a fixed-point, respectively.

Theorem 4.3.1 (Stability about a fixed-point; $\alpha < 1$)**.** *Let $T\colon \mathbb{R}^K \to \mathbb{R}^K$ denote a system operator that is α-dissipative about $\mathbf{v}^\star$ where $\mathbf{v}^\star$ is a fixed point of T. Then the filtered system operator T_f associated with T is stable in mean square, i.e. $\mathbf{v}^n \xrightarrow{2} \mathbf{v}^\star$, provided $\rho \in (0, \frac{2}{1+\alpha^2})$. Furthermore, $\mathbf{v}^\star$ is the unique fixed point of T.*

Proof: The average squared distance of the state sequence to any fixed point $\mathbf{v}^\star$ after a single iteration is upper bounded according to

$$\begin{aligned}
\mathbb{E}\left[\|\mathbf{v}^n - \mathbf{v}^\star\|^2\right] &= p\mathbb{E}\left[\|T_f(\rho, \mathbf{v}^{n-1}) - \mathbf{v}^\star\|^2\right] + (1-p)\mathbb{E}\left[\|\mathbf{v}^{n-1} - \mathbf{v}^\star\|^2\right] && (4.36)\\
&\leq p\rho^2\mathbb{E}\left[\|T(\mathbf{v}^{n-1}) - \mathbf{v}^\star\|^2\right] + ((1-p) + p|1-\rho|^2)\mathbb{E}\left[\|\mathbf{v}^{n-1} - \mathbf{v}^\star\|^2\right] && (4.37)\\
&\leq \underbrace{\left(p\rho^2\alpha^2 + 1 - p + p|1-\rho|^2\right)}_{\phi(\rho)} \mathbb{E}\left[\|\mathbf{v}^{n-1} - \mathbf{v}^\star\|^2\right] && (4.38)
\end{aligned}$$

where the equality is due to (4.26). Aggregating the assumptions on α and p, respectively from the assumed dissipativity and asynchronous processing protocol, and requiring the function $\phi(\rho)$ in the single iteration inequality to satisfy $\phi(\rho) \in [0, 1)$, the range of acceptable values for the filtering parameter ρ follow as

$$\begin{cases} 0 \leq p\rho^2\alpha^2 + 1 - p + p|1-\rho|^2 < 1 \\ 0 \leq \alpha < 1 \\ 0 < p \leq 1 \end{cases} \implies \rho \in \left(0, \frac{2}{1+\alpha^2}\right). \tag{4.39}$$

Proceeding with this restriction on ρ, iterating the inequality in (4.38) n times and then taking a limit gives the desires stability result:

$$\begin{aligned} \lim_{n\to\infty} \mathbb{E}\left[\|\mathbf{v}^n - \mathbf{v}^\star\|^2\right] &\leq \left\|\mathbf{v}^0 - \mathbf{v}^\star\right\|^2 \lim_{n\to\infty} \phi(\rho)^n & (4.40) \\ &= 0. & (4.41) \end{aligned}$$

It remains to be seen that $\mathbf{v}^\star$ is the unique fixed point of T; we prove this next by producing a contradiction. Let $\mathbf{v}^\star$ and $\mathbf{u}^\star$ denote two distinct fixed points of T. Then, using the definition of a fixed point and (4.22) it follows that

$$\begin{aligned} \|\mathbf{u}^\star - \mathbf{v}^\star\| &= \|T(\mathbf{v}^\star + (\mathbf{u}^\star - \mathbf{v}^\star)) - T(\mathbf{v}^\star)\| & (4.42) \\ &\leq \alpha\|\mathbf{u}^\star - \mathbf{v}^\star\| & (4.43) \end{aligned}$$

which provides a contradiction since $\alpha < 1$, therefore $\mathbf{v}^\star = \mathbf{u}^\star$ must be true. ■

Verifying the conditions stated in Theorem 4.3.1 can be troublesome in practice since they involve both knowledge that a fixed-point exists as well as knowledge of the system operator's behavior around it. It is also possible to characterize the boundedness of the state sequence produced by a system operator T that is α-dissipative about an arbitrary point $\mathbf{c} \in \mathbb{R}^K$ that is not necessarily a fixed-point of T. Motivated by this, the following theorem asserts that the state sequence produced by such an interconnective system will on average be contained within a closed Euclidean ball centered at $\mathbf{c}$ after finite transient effects die off.

Theorem 4.3.2 (Finite-time entrapment; $\alpha < 1$)**.** *Let* $T \colon \mathbb{R}^K \to \mathbb{R}^K$ *denote a system* *operator that is α-dissipative about an arbitrary point $\mathbf{c}$ that is not necessarily a fixed-point of T. Then, for every $\epsilon > 0$ there exists a finite integer N such that the asynchronous implementation of the filtered system operator T_f associated with T produces a state sequence $\{\mathbf{v}^n \colon n \in \mathbb{N}_0\}$ satisfying*

$$\mathbb{E}\left[\|\mathbf{v}^n - \mathbf{c}\|\right] \leq \frac{\rho \left\|T(\mathbf{c}) - \mathbf{c}\right\|}{1 - \rho\alpha - |1-\rho|} + \epsilon, \quad n \geq N, \tag{4.44}$$

provided $\rho \in (0, \frac{2}{1+\alpha})$.

Proof: The average distance of the state sequence to the state $\mathbf{c}$ after a single iteration is upper bounded by

$$\begin{aligned}
\mathbb{E}\left[\|\mathbf{v}^n - \mathbf{c}\|\right] &= p\mathbb{E}\left[\left\|T_f(\rho, \mathbf{v}^{n-1}) - \mathbf{c}\right\|\right] + (1-p)\mathbb{E}\left[\left\|\mathbf{v}^{n-1} - \mathbf{c}\right\|\right] && (4.45)\\
&\leq p\rho\mathbb{E}\left[\left\|T(\mathbf{v}^{n-1}) - T(\mathbf{c})\right\|\right] + p|1-\rho|\mathbb{E}\left[\left\|\mathbf{v}^{n-1} - \mathbf{c}\right\|\right] && (4.46)\\
&\quad + (1-p)\mathbb{E}\left[\left\|\mathbf{v}^{n-1} - \mathbf{c}\right\|\right] + p\rho\left\|T(\mathbf{c}) - \mathbf{c}\right\| \\
&\leq \left(p\rho\alpha + p|1-\rho| + (1-p)\right)\mathbb{E}\left[\left\|\mathbf{v}^{n-1} - \mathbf{c}\right\|\right] + p\rho\left\|T(\mathbf{c}) - \mathbf{c}\right\| && (4.47)
\end{aligned}$$

where the first inequality is due to repeated application of the triangle inequality and the second inequality makes use of the fact that T is α-dissipative at $\mathbf{c}$. Iterating this inequality n times yields an upper bound of the form

$$\mathbb{E}\left[\|\mathbf{v}^n - \mathbf{c}\|\right] \leq \phi^n(\rho)\left\|\mathbf{v}^0 - \mathbf{c}\right\| + p\rho\left\|T(\mathbf{c}) - \mathbf{c}\right\|\left(\sum_{k=0}^{n-1}\phi^k(\rho)\right) \tag{4.48}$$

where $\phi(\rho) = p\rho\alpha + p|1-\rho| + (1-p)$ is convex, piece-wise linear, passes through $(\rho, \phi(\rho)) = (0, 1)$, and has a global minimum at $\rho = 1$. It is straightforward to verify that $\phi(\rho) \in [0, 1)$ provided that $\rho \in (0, \frac{2}{1+\alpha})$. We proceed by restricting the filtering parameter to this interval in order to loosen the inequality (4.48) to

$$\mathbb{E}\left[\|\mathbf{v}^n - \mathbf{c}\|\right] \leq \phi^n(\rho)\left\|\mathbf{v}^0 - \mathbf{c}\right\| + \frac{\rho\left\|T(\mathbf{c}) - \mathbf{c}\right\|}{1 - \rho\alpha - |1-\rho|}. \tag{4.49}$$

Therefore, there will always exists a finite value of n sufficiently large to ensure that

$\phi^n(\rho)\left\|\mathbf{v}^0 - \mathbf{c}\right\| < \epsilon$ which concludes the proof. ■

The entrapment of the state sequence discussed in the previous theorem neither assumes nor concludes anything about fixed points since in general fixed points of these system operators may not exist. However, if $\mathbf{c}$ happens to be a fixed point of T then it is unique, and Theorem 4.3.2 implies that the state sequence on average gets arbitrarily close to $\mathbf{c}$ in finite-time. This result, of course, is consistent with Theorem 4.3.1 in the sense that the radii achieved reduces to ϵ which may be made arbitrarily small by taking n large enough. This type of agreement is consistent with many classical finite iteration convergence results. Aside from studying dissipative system operators, Theorems 4.3.1 and 4.3.2 can be used to assemble stability conditions for more general system operators since they inherently require only local assumptions to be met. We shall return to them in this context in Section 4.3.4.

Next, we move on to synchronous and asynchronous algorithms produced by system operators that are α-dissipative everywhere. As was mentioned earlier, the Banach fixed-point theorem states that synchronously ($p = 1$) implementing such a system operator without filtering ($\rho = 1$) produces a unique fixed point at a linear rate. The following theorem extends this result into the asynchronous setting and suggests the use of the filtering parameter to achieve better rates.

Theorem 4.3.3 (Stability in $\mathbb{R}^K$; $\alpha < 1$). *Let $T\colon \mathbb{R}^K \to \mathbb{R}^K$ denote a system operator that is α-dissipative everywhere. Then T has a unique fixed point $\mathbf{v}^\star$ and the filtered system operator T_f associated with T is stable in mean square, i.e. $\mathbf{v}^n \xrightarrow{2} \mathbf{v}^\star$, provided $\rho \in (0, \frac{2}{1+\alpha^2})$.*

Proof: To begin, we prove the existence and then the uniqueness of the fixed point $\mathbf{v}^\star$ for completeness. Recall that such facts are properties of T and not the iteration. To do this, we combine two inequalities for the simple iteration $\mathbf{v}^n = T(\mathbf{v}^{n-1})$. The first inequality follows from substituting $\mathbf{v}' = \mathbf{v}^n$ and $\mathbf{u}' = \mathbf{v}^{n-1}$ into (4.23) and then iterating the inequality $n-1$ times:

$$\begin{aligned} \|\mathbf{v}^n - \mathbf{v}^{n-1}\| &\le \alpha\|\mathbf{v}^{n-1} - \mathbf{v}^{n-2}\| && (4.50)\\ &\le \alpha^{n-1}\|\mathbf{v}^1 - \mathbf{v}^0\|. && (4.51) \end{aligned}$$

Let n and m denote two positive integers satisfying $m < n$. The second inequality follows from repeated application of the triangle inequality to (4.23) to obtain

$$\|\mathbf{v}^n - \mathbf{v}^m\| \;\leq\; \sum_{k=m}^{n-1} \|\mathbf{v}^{k+1} - \mathbf{v}^k\|. \tag{4.52}$$

Together the inequalities (4.51) and (4.52) imply

$$\begin{aligned} \|\mathbf{v}^n - \mathbf{v}^m\| &\leq \left(\sum_{k=m}^{n-1} \alpha^k\right) \left\|\mathbf{v}^1 - \mathbf{v}^0\right\| && (4.53)\\ &= \frac{\alpha^m \left(1 - \alpha^{n-m}\right)}{1-\alpha} \left\|\mathbf{v}^1 - \mathbf{v}^0\right\| && (4.54) \end{aligned}$$

from which it is straightforward to show that the sequence $\{\mathbf{v}^n \colon n \in \mathbb{N}_0\}$ is Cauchy. Subsequently, (4.54) establishes the existence of a limit point $\mathbf{v}^\star$ and that $\mathbf{v}^n$ converges to it. Moreover, the continuity of T implied by the α-conic everywhere assumption ensures that

$$T(\mathbf{v}^\star) = T\left(\lim_{n\to\infty} \mathbf{v}^n\right) = \lim_{n\to\infty} T(\mathbf{v}^n) = \lim_{n\to\infty} v^{n+1} = \mathbf{v}^\star \tag{4.55}$$

thus $\mathbf{v}^\star$ is indeed a fixed point of T. The remainder of this theorem follows from Theorem 4.3.1 since: (i) T has a fixed point $\mathbf{v}^\star$ and (ii) T being α-dissipative everywhere clearly implies that T is α-dissipative about $\mathbf{v}^\star$. ■

The next category of system operators we consider form the middle ground between the system operators analyzed in Theorems 4.3.1 and 4.3.2 (α-dissipative about a point) and the system operators analyzed in Theorem 4.3.3 (α-dissipative everywhere). Specifically, we consider the class of system operators that are α-dissipative over a basin but not necessarily elsewhere. For this case, stability and robustness properties follow if the initial state is chosen inside the basin and if the center of the basin does not move too far under the action of the system operator. These conditions are restated more formally in the following theorem where the basin is specifically characterized in terms of a closed Euclidean ball. More generally, the argument remains true for an open ball since the uniform continuity of the system operator inherited from (4.22) is sufficient to extend it to the closure of the ball while retaining the same conic parameter α on the perimeter.

Theorem 4.3.4 (Stability in a basin; $\alpha < 1$)**.** *Let $T\colon \mathbb{R}^K \to \mathbb{R}^K$ denote a system operator that is α-dissipative about all $\mathbf{v} \in B(\mathbf{c}, r)$. Then T has a unique fixed point $\mathbf{v}^\star$ in $B(\mathbf{c}, r)$ and the filtered system operator T_f associated with T is stable in mean, i.e. $\mathbf{v}^n \xrightarrow{1} \mathbf{v}^\star$, provided the initial element satisfies $\mathbf{v}^0 \in B(\mathbf{c}, r)$, the filtering parameter satisfies $\rho \in (0, 1]$, and*

$$\frac{\|\mathbf{c} - T(\mathbf{c})\|}{1-\alpha} \quad \leq \quad r. \tag{4.56}$$

Proof: The existence and uniqueness of $\mathbf{v}^\star$ is inherited from Theorem 4.3.3 since $(B(\mathbf{c}, r), \|\cdot\|)$ is a complete metric space isomorphic to $(\mathbb{R}^K, \|\cdot\|)$. We next prove that the average state sequence initialized within the basin converges to it. To do this, consider first the application of the filtered system operator T_f to a state $\mathbf{v} \in B(\mathbf{c}, r)$:

$$\begin{aligned} \|T_f(\rho, \mathbf{v}) - \mathbf{c}\| &= \|\rho T(\mathbf{v}) + (1-\rho)\mathbf{v} - \rho T(\mathbf{c}) + \rho T(\mathbf{c}) - \rho\mathbf{c} - (1-\rho)\mathbf{c}\| && (4.57)\\ &\leq \rho\left(\|T(\mathbf{v}) - T(\mathbf{c})\| + \|T(\mathbf{c}) - \mathbf{c}\|\right) + |1-\rho|\,\|\mathbf{v} - \mathbf{c}\| && (4.58)\\ &\leq \left(\rho\alpha + |1-\rho|\right) r + \rho(1-\alpha) r. && (4.59) \end{aligned}$$

Therefore, by restricting ρ to the interval $(0, 1]$ we conclude that $T_f(\rho, \mathbf{v})$ is also contained in $B(\mathbf{c}, r)$. Then, using the fixed point $\mathbf{v}^\star$ and assuming $\mathbf{v}^0 \in B(\mathbf{c}, r)$ we obtain the inequality:

$$\begin{aligned} \mathbb{E}\left[\|\mathbf{v}^n - \mathbf{v}^\star\|\right] &= p\mathbb{E}\left[\left\|T_f(\rho, \mathbf{v}^{n-1}) - \mathbf{v}^\star\right\|\right] + (1-p)\mathbb{E}\left[\left\|\mathbf{v}^{n-1} - \mathbf{v}^\star\right\|\right] && (4.60)\\ &\leq \underbrace{(p\rho\alpha + p|1-\rho| + 1 - p)}_{\phi(\rho)} \mathbb{E}\left[\left\|\mathbf{v}^{n-1} - \mathbf{v}^\star\right\|\right] && (4.61) \end{aligned}$$

where $\phi(\rho)$ is in the interval $(0, 1)$ as long a $\rho \in (0, \frac{2}{1+\alpha})$. Taking the intersection of the intervals identified for ρ provides the range $\rho \in (0, 1]$. Proceeding with this, it follows that

$$\begin{aligned} \lim_{n\to\infty} \mathbb{E}\left[\|\mathbf{v}^n - \mathbf{v}^\star\|\right] &\leq \left\|\mathbf{v}^0 - \mathbf{v}^\star\right\| \lim_{n\to\infty} \phi^n(\rho) && (4.62)\\ &= 0 && (4.63) \end{aligned}$$

which concludes the proof that $\mathbf{v}^n \xrightarrow{1} \mathbf{v}^\star$. ■

The argument used in the proof of the previous theorem is reminiscent of the argument used in the proof of Theorem 4.3.3 if we interpret $\mathbb{R}^K$ as a Euclidean ball with an arbitrarily large radius. From this perspective, the upper bound in (4.56) can be selected to be arbitrarily large meaning the location of the initial state becomes irrelevant. Certifying the basin condition in Theorem 4.3.4 requires only a single application of the system operator. If $\mathbf{c} = \mathbf{0}$, then this condition reduces to an upper bound on the operator norm of the system operator. For realizations of interconnecting networks, this condition is simply a bound on the singular values of the realization.

Beyond situations similar to the proof of Theorem 4.3.4, the application of Theorem 4.3.3 is useful in some additional contexts. One such context, formalized in the next theorem, pertains to general system operators that are not necessarily α-conic themselves but become so after some finite number of self-compositions. We consider this setup using a direct and synchronous implementation of the original system operator since the stability of asynchronous and filtered implementations of the composite operator follow immediately from Theorem 4.3.3.

Theorem 4.3.5 (Stability for composition; $\alpha < 1$). *Let $T\colon \mathbb{R}^K \to \mathbb{R}^K$ denote a system operator. If there exists a finite integer $m \in \mathbb{N}$ such that $T^m = T \circ T^{m-1}$ is α-dissipative everywhere, then T has a unique fixed point $\mathbf{v}^\star$ and the state sequence generated by $\mathbf{v}^n = T(\mathbf{v}^{n-1})$, for $n \in \mathbb{N}$, converges to it.*

Proof: Since T^m is assumed to be α-dissipative everywhere, we conclude that T^m has a unique fixed point $\mathbf{u}^\star$ by direct application of Theorem 4.3.3. We next prove that $\mathbf{u}^\star$ is also a fixed point of T and that $\mathbf{u}^\star = \mathbf{v}^\star$. Indeed, since T^m satisfies the functional translation property it follows that

$$T^m \circ T(\mathbf{u}^\star) = T \circ T^m(\mathbf{u}^\star) = T(\mathbf{u}^\star). \tag{4.64}$$

Hence, by the uniqueness of the fixed point of T^m we have that $\mathbf{u}^\star = T(\mathbf{u}^\star)$ and therefore by definition $\mathbf{v}^\star = \mathbf{u}^\star$.

We conclude this proof by showing that the state sequence tends to $\mathbf{v}^\star$ and that $\mathbf{v}^\star$ is unique. For each integer i satisfying $0 \leq i \leq m-1$, the α-dissipative everywhere system operator T^m generates a subsequence of the sequence $\{\mathbf{v}^n \colon n \in \mathbb{N}_0\}$ produced by $\mathbf{v}^n = T(\mathbf{v}^{n-1})$ consisting of the states $\{\mathbf{v}^{mp+i} \colon p \in \mathbb{N}_0\}$, which is further illustrated by the relation

$$\mathbf{v}^{mp+i} = \underbrace{T^m \circ T^m \circ \cdots \circ T^m}_{p \text{ compositions}} \circ T^i(\mathbf{v}^0), \quad p \geq 0, \tag{4.65}$$

where T^0 is the identity operator and $\mathbf{v}^0$ is the initial state. Said another way, the state sequence $\{\mathbf{v}^n \colon n \in \mathbb{N}_0\}$ consists of the interleaved elements generated from m synchronous implementations of T^m with respective initial states $T^i(\mathbf{v}^0)$, $i = 0, 1, \ldots, m-1$. Therefore, by Theorem 4.3.3 with $\rho = p = 1$ it follows that

$$\lim_{p \to \infty} \left\| \mathbf{v}^{mp+i} - \mathbf{v}^\star \right\|^2 = 0, \qquad i = 0, 1, \ldots, m-1. \tag{4.66}$$

Observe that this argument is precisely proving that each channel in the polyphase decomposition of the sequence $\mathbf{v}^n$ converges to the same element. Thus, utilizing the fact that this convergence is independent of i, it is straightforward to conclude that the fixed point is unique. ■

The provision that composite system operators be α-conic everywhere does not imply that the original system operators are, nor does it imply that they are continuous in any sense. As a straightforward example of the former, consider the scalar operator $T \colon \mathbb{R} \to \mathbb{R}$ given by $T(\mathbf{v}) = e^{-\mathbf{v}}$. This operator is passive on the positive reals but does not satisfy the α-conic condition in (4.22) for any finite value of α. On the other hand, the twice composed operator $T \circ T(\mathbf{v}) = e^{-e^{-\mathbf{v}}}$ does with $\alpha = \frac{1}{e}$ and has a unique fixed point of $v^\star \approx 0.5671$.

4.3.3 | Passive system operators

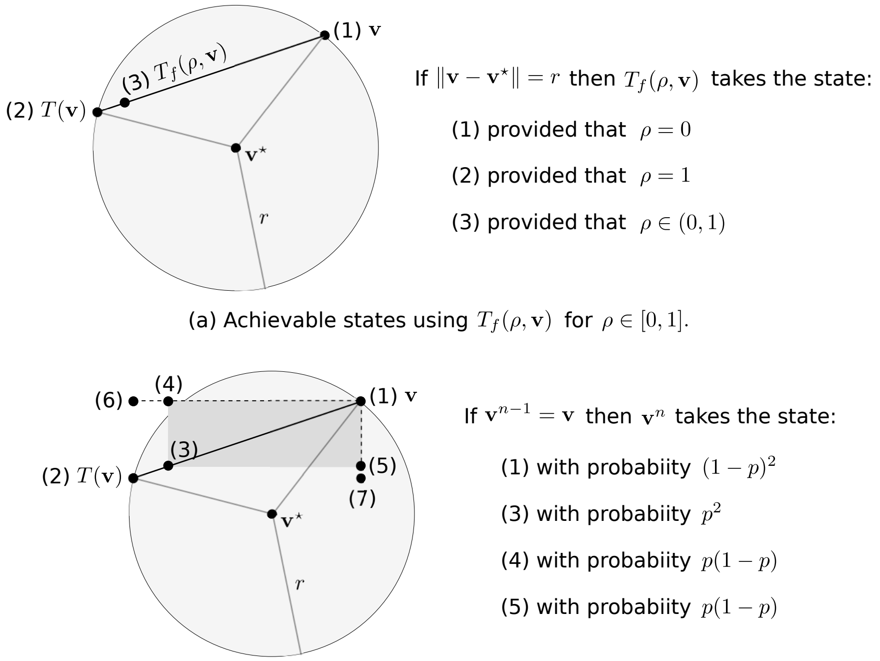
A key category of interconnective systems we address include those whose system operators are passive everywhere. In contrast to the α-dissipative counterparts analyzed in the previous subsection, stability and robustness conditions for passive-everywhere system operators face a variety of important issues. The scalar operator $T \colon \mathbb{R} \to \mathbb{R}$ given by $T(\mathbf{v}) = -\mathbf{v}$ illustrates one such issue: although T has a singleton fixed-point set $\mathcal{F}_T = \{\mathbf{0}\}$, any state sequence initialized with $\mathbf{v}^0 \neq \mathbf{0}$ will oscillate forever and thus T cannot be stable or robust in the sense of (4.32). For canonical-form interconnective structures that have been transformed into scattering form, the functional realization of the interconnecting network as an orthogonal matrix cannot suffer from this issue since it cannot have eigenvalues of -1. The family of scalar operators given by $T(\mathbf{v}) = \mathbf{v} + a$ for any non-zero value of a illustrates a second issue: the fixed-point set $\mathcal{F}_T$ is empty. In light of this, we proceed in this subsection to consider only those passive everywhere system operators that have non-empty fixed-point sets, or equivalently that are associated with well-defined CCSPs. Before analyzing the stability and robustness of passive everywhere system operators under this assumption, we first examine the uniqueness of their fixed points.

The fixed-point set associated with a passive everywhere system operator is, in general, a convex set. This assertion is justified as follows. Let T denote such an operator and let $\mathbf{u}^\star$ and $\mathbf{v}^\star$ denote two distinct fixed points. Observe that the inequality

$$\begin{aligned}
\|\mathbf{u}^\star - \mathbf{v}^\star\| &\leq \|\mathbf{u}^\star - T\left(\gamma\mathbf{v}^\star + (1-\gamma)\mathbf{u}^\star\right)\| + \|T\left(\gamma\mathbf{v}^\star + (1-\gamma)\mathbf{u}^\star\right) - \mathbf{v}^\star\| && (4.67)\\
&\leq \|\gamma\left(\mathbf{u}^\star - \mathbf{v}^\star\right)\| + \|(1-\gamma)\left(\mathbf{u}^\star - \mathbf{v}^\star\right)\| && (4.68)\\
&\leq \|\mathbf{u}^\star - \mathbf{v}^\star\| && (4.69)
\end{aligned}$$

holds with equality for any γ in the interval $[0,1]$. Passivity of the system operator T is specifically used to obtain the second inequality from the first and the implication of the overall equality after use of the triangle inequality is that the elements $T(\gamma\mathbf{v}^\star + (1-\gamma)\mathbf{u}^\star)$ and $\gamma\mathbf{v}^\star + (1-\gamma)\mathbf{u}^\star$ are on the line segment connecting $\mathbf{u}^\star$ to $\mathbf{v}^\star$ and subsequently that $\gamma\mathbf{v}^\star + (1-\gamma)\mathbf{u}^\star \in \mathcal{F}_T$. Hence, this proves that $\mathcal{F}_T$ is a convex set, which, of course, does not preclude it from being empty.

We next analyze the stability and robustness properties associated with asynchronous algorithms that correspond to implementing filtered passive-everywhere system operators. To separate the effects of the filter parameter from the stochastic dynamics attributed to the asynchronous delay modules, we proceed by characterizing the update mechanism described in (3.47) by breaking the iteration into two stages: (i) the deterministic application of the


(1) $\mathbf{v}$
(3) $T_f(\rho, \mathbf{v})$
(2) $T(\mathbf{v})$
$\mathbf{v}^\star$
r
If $\|\mathbf{v} - \mathbf{v}^\star\| = r$ then $T_f(\rho, \mathbf{v})$ takes the state:
(1) provided that $\rho = 0$
(2) provided that $\rho = 1$
(3) provided that $\rho \in (0, 1)$
(6)
(4)
(1) $\mathbf{v}$
(3)
(2) $T(\mathbf{v})$
(5)
(7)
$\mathbf{v}^\star$
r
If $\mathbf{v}^{n-1} = \mathbf{v}$ then $\mathbf{v}^n$ takes the state:
(1) with probabiity $(1-p)^2$
(3) with probabiity p^2
(4) with probabiity $p(1-p)$
(5) with probabiity $p(1-p)$

(a) Achievable states using $T_f(\rho, \mathbf{v})$ for $\rho \in [0, 1]$.

(b) Possible outcomes according to the asynchronous protocol

Figure 4-4: An illustration of (a) the states achievable by application of T_f and (b) the possible outcomes of $\mathbf{v}^n$ for an arbitrary $\mathbf{v}^{n-1}$ according to the stochastic dynamics of (3.47) for a system operator T satisfying the conditions in Theorem 4.3.6.

filtered system operator $T_f(\rho, \cdot)$ to $\mathbf{v}^{n-1}$, and (ii) the stochastic combination of $\mathbf{v}^{n-1}$ and $T_f(\rho, \mathbf{v}^{n-1})$ governed by the random matrix $D^{(p)}$. To avoid cluttering notation in doing this, the previous state $\mathbf{v}^{n-1}$ will simply be denoted as $\mathbf{v}$ in the ensuing discussion and accompanying illustrations.

In discussing the role of the filtering parameter in stage (i), we proceed by specializing the characterization of the squared distance of the filtered system operator $T_f(\rho, \cdot)$ to a fixed-point of the original system operator T as a function of the filtering parameter ρ derived in Section 4.2.3 to the case where T is passive everywhere. To assist in visualizing this discussion, we provide an example in $\mathbb{R}^2$ in Figure 4-4(a) where we have additionally assumed the "worst case" outcome $\|T(\mathbf{v}) - \mathbf{v}^\star\|^2 = \|\mathbf{v} - \mathbf{v}^\star\|^2 = r$. By restricting the filtering parameter ρ to the open unit interval $(0, 1)$, it is straightforward to conclude that the state $T_f(\rho, \mathbf{v})$ is strictly closer to $\mathbf{v}$ than $T(\mathbf{v})$ is. The size of this distance can be arbitrarily small since $T(\mathbf{v})$ an $\mathbf{v}$ can be arbitrarily close together on the boundary of $B(\mathbf{v}^\star, r)$. Algebraically, this observation is justified by the derivation of (4.26).

Focusing next on the dynamics of stage (ii), we proceed to enumerate all possible outcomes that the state $\mathbf{v}^n$ can take, and for each outcome we determine the likelihood that it occurs based upon the coordinatewise stochastic combination of the states $\mathbf{v}$ and $T_f(\rho, \mathbf{v})$. Continuing the two-dimensional example in Figure 4-4(a), these outcomes and their associated likelihoods are provided in Figure 4-4(b). In the general case, the state $\mathbf{v}^n$ can take any state corresponding to one of the 2^K corners of a K-orthotope defined using $\mathbf{v}$ and $T_f(\rho, \mathbf{v})$ as opposite corners, i.e. the hyperrectangle constructed as the Cartesian product of the K intervals $[a_i, b_i]$ where $a_i = \min\{\mathbf{v}_i, T_{f,i}(\rho, \mathbf{v})\}$ and $b_i = \max\{\mathbf{v}_i, T_{f,i}(\rho, \mathbf{v})\}$. Intuitively, each corner represents one of the 2^K ways that i of K asynchronous delay modules can trigger for i inthe range $0 \leq i \leq K$. This characterization is exhaustive since

$$\sum_{i=0}^{K} C(K,i) p^i (1-p)^{K-i} = 1 \tag{4.70}$$

where the number $C(K,i)$ is the binomial coefficient corresponding to the total number of ways i of K delay elements can trigger. Referring again to Figure 4-4(b), the elements labeled (1), (3), (4), and (5) correspond to the four possible outcomes of $\mathbf{v}^n$ for a fixed value of ρ. The line segments with endpoints labeled (1)-(2), (1)-(6), and (1)-(7) correspond to the continuum of possible outcomes as ρ varies between 0 and 1.

From the discussion in this section, it is straightforward to conclude that a subset of possible outcomes for $\mathbf{v}^n$ are strictly further from the fixed-point $\mathbf{v}^\star$ than either $\mathbf{v}$ or $T_f(\rho, \mathbf{v})$ are. Referring again to Figure 4-4(b), the outcome labeled (4) and the continuum of outcomes corresponding to the line segment with endpoints (4)-(6) are examples of this. In general, these outcomes cannot be avoided by simply restricting the range of filter parameters further. The following theorem, however, ensures that filtered system operators are stable in mean square so long as their filtering parameters are chosen in the open unit interval.

Theorem 4.3.6 (Stability in $\mathbb{R}^K$; $\alpha = 1$). *Let $T\colon \mathbb{R}^K \to \mathbb{R}^K$ denote a system operator that is passive everywhere with non-empty fixed-point set $\mathcal{F}_T$. Then the filtered system operator T_f associated with T is stable in mean square, i.e. $\mathbf{v}^n \overset{2}{\to} \mathbf{v}^\star$ for some $\mathbf{v}^\star \in \mathcal{F}_T$, provided $\rho \in (0,1)$.*

A formal proof of Theorem 4.3.6 is deferred to Appendix A. Note, however, that stability and robustness in r-th mean of T is not implied by Theorem 4.3.6 for any value of r since the range of filter parameters does not include $\rho = 1$. This is consistent with the issues discussed earlier for system operators akin to $T(\mathbf{v}) = -\mathbf{v}$.

Moving beyond simple first-order filtering of delay modules to strengthen the stability and robustness of asynchronous algorithms, the numerical continuation schemes discussed in Section 4.2.2 can also be used to solve for fixed-points of a passive everywhere system operator T. To see this, consider the adaptive system operator T_a defined by

$$T_a(\rho, \mathbf{v}) = \rho T(\mathbf{v}). \tag{4.71}$$

This particular adaptive operator corresponds to selecting $T^{(0)}$ to be the everywhere-zero operator in Definition 4.2.3. One approach is to then use a traditional continuation method where the interval $[0, 1]$ is discretized and we solve for a fixed point of T_a for increasing values of the homotopy parameter ρ on that interval, where the fixed point from the previous value of ρ is used as a warm start to solving for a fixed point of the next. A second approach is to use the fact that the operator $T_a(\rho, \cdot)$ is itself ρ-dissipative everywhere for $\rho < 1$. By application of Theorem 4.3.3, the adaptive system operator T_a is stable in mean square, and, therefore, asynchronously implementing it will result in a fixed point. However, the fixed point obtained is not necessarily a fixed point of T, but under reasonable assumptions it can be made arbitrarily close to the fixed point of T by selecting ρ arbitrarily close to 1.

4.3.4 | Expansive system operators

The next category of interconnective systems we address include those whose system operators are α-expansive everywhere with non-empty, convex fixed-point sets. When these system operators additionally satisfy a conic mixing property about their fixed-point sets, then a filtering parameter may be judiciously chosen so that the filtered implementation of the asynchronous algorithm is effectively dissipative everywhere. By doing this, a stability and robustness theorem similar to the one for dissipative everywhere systems in Theorem 4.3.3 can be obtained. The required mixing property and its consequences on stability and robustness are stated formally in the following theorem.

Theorem 4.3.7 (Conic mixing; $\alpha > 1$)**.** *Let $T\colon \mathbb{R}^K \to \mathbb{R}^K$ denote a system operator that is α-expansive everywhere and whose fixed-point set $\mathcal{F}_T$ is non-empty and convex. If T additionally satisfies the conic mixing property*

$$\sup_{\mathbf{v}\notin\mathcal{F}_T} \frac{\langle T(\mathbf{v}) - \mathbf{v}^\star, \mathbf{v} - \mathbf{v}^\star\rangle}{\|T(\mathbf{v}) - \mathbf{v}^\star\| \, \|\mathbf{v} - \mathbf{v}^\star\|} \leq \gamma \tag{4.72}$$

for all $\mathbf{v}^\star \in \mathcal{F}_T$ and some $\gamma \in [-1, 1)$ such that $\alpha\gamma < 1$, then the filtered system operator T_f associated with T is stable in mean square, i.e. $\mathbf{v}^n \xrightarrow{2} \mathbf{v}^\star$ for some $\mathbf{v}^\star \in \mathcal{F}_T$, provided

$$\rho \in \left(0, \frac{2(1-\alpha\gamma)}{1+\alpha^2-2\alpha\gamma}\right). \tag{4.73}$$

Proof: The average squared distance of the state $\mathbf{v}^n$ to any fixed-point $\mathbf{v}^\star$ after a single iteration is upper bounded by

$$\begin{aligned}
\mathbb{E}\left[\|\mathbf{v}^n - \mathbf{v}^\star\|^2\right] &= p\mathbb{E}\left[\left\|T_f(\rho, \mathbf{v}^{n-1}) - \mathbf{v}^\star\right\|^2\right] + (1-p)\mathbb{E}\left[\left\|\mathbf{v}^{n-1} - \mathbf{v}^\star\right\|^2\right] && (4.74)\\
&= p\rho^2\mathbb{E}\left[\left\|T(\mathbf{v}^{n-1}) - T(\mathbf{v}^\star)\right\|^2\right] + \left(1-p+p(1-\rho)^2\right)\mathbb{E}\left[\left\|\mathbf{v}^{n-1} - \mathbf{v}^\star\right\|^2\right] && (4.75)\\
&\quad +2p\rho(1-\rho)\mathbb{E}\left[\left\langle T(\mathbf{v}^{n-1}) - \mathbf{v}^\star, \mathbf{v}^{n-1} - \mathbf{v}^\star\right\rangle\right]\\
&\leq \underbrace{\left(p\rho^2\alpha^2 + 1 - p + p(1-\rho)^2 + 2p\alpha\gamma\rho(1-\rho)\right)}_{\phi(\rho)} \mathbb{E}\left[\left\|\mathbf{v}^{n-1} - \mathbf{v}^\star\right\|^2\right] && (4.76)
\end{aligned}$$

where the second equality is due to the identity $\|\mathbf{v} + \mathbf{u}\|^2 = \langle \mathbf{v} + \mathbf{u}, \mathbf{v} + \mathbf{u}\rangle = \|\mathbf{v}\|^2 + \|\mathbf{u}\|^2 + 2\langle \mathbf{v}, \mathbf{u}\rangle$. Iterating this inequality n times yields

$$\mathbb{E}\left[\|\mathbf{v}^n - \mathbf{v}^\star\|^2\right] \leq \phi(\rho)^n \left\|\mathbf{v}^0 - \mathbf{v}^\star\right\|^2 \tag{4.77}$$

where $\mathbf{v}^0$ is the arbitrarily selected, deterministic initial state and $\phi(\rho)$ is a convex quadratic function in ρ of the form

$$\phi(\rho) = p\left(1 + \alpha^2 - 2\alpha\gamma\right)\rho^2 - 2p\left(1 - \alpha\gamma\right)\rho + 1. \tag{4.78}$$

We next prove that $\phi(\rho) \in (0,1)$ for ρ satisfying (4.73). The lower bound for $\phi(\rho)$ follows from basic properties of quadratic forms, in particular the fact that $\phi(\rho)$ is guaranteed to be strictly positive everywhere since, by assumption that $\gamma < 1$, it cannot have a real root. By solving $\frac{d}{d\rho}\phi(\rho) = 0$ for ρ we find that the minimum of $\phi(\rho)$ occurs at $\rho^\star = \frac{(1-\alpha\gamma)}{1+\alpha^2-2\alpha\gamma}$ corresponding to the midpoint of the interval in (4.73). We derive an upper bound for ρ such that $\phi(\rho)$ is strictly upper bounded by unity by reducing the expression in (4.78) by subtracting constants and dividing through by $p\rho > 0$. This results in the condition

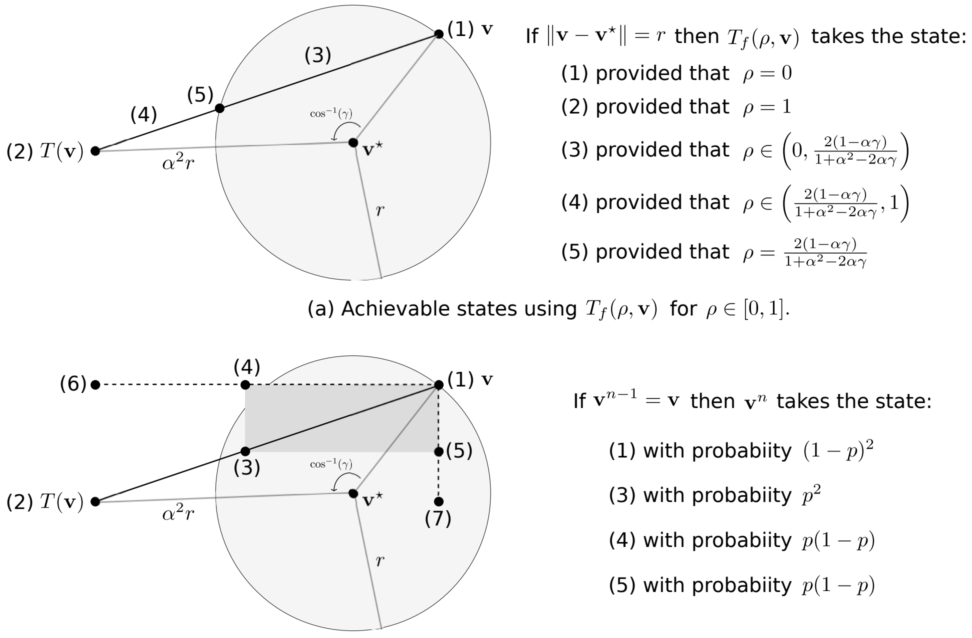
$$\rho(1+\alpha^2-2\alpha\gamma) \quad < \quad 2(1-\alpha\gamma) \tag{4.79}$$

from which we obtain the upper limit of the interval in (4.73). By symmetry, the lower bound for ρ is easily shown to be 0. Therefore, utilizing the fact $\phi(\rho) \in (0,1)$ and taking a limit of the inequality established in (4.77), we obtain the result

$$\begin{aligned} \lim_{n\to\infty} \mathbb{E}\left[\|\mathbf{v}^n - \mathbf{v}^\star\|^2\right] \quad &\leq \quad \left\|\mathbf{v}^0 - \mathbf{v}^\star\right\|^2 \lim_{n\to\infty} \phi(\rho)^n & (4.80)\\ &= \quad 0 & (4.81)\end{aligned}$$

as desired and hence we have shown that $\mathbf{v}^n \xrightarrow{2} \mathbf{v}^\star$ thus concluding the proof. ■

The relationship previously discussed between Theorem 4.3.6 and Figure 4-4 is essentially the same as the relationship between Theorem 4.3.7 and Figure 4-5 with two main caveats. The first caveat is that the restriction of the filtering parameters is to different intervals. More specifically, restricting the filtering parameter to the open interval $(0,1)$ in (4.73) is analogous to restricting the filtering parameter to the open interval $(0,1)$ in Theorem 4.3.6 since the basic strategy in both cases is to ensure that the squared distance between the state $T_f(\rho,\mathbf{v})$ and the fixed-point $\mathbf{v}^\star$ is strictly smaller than the squared distance between the state $\mathbf{v}$ and $\mathbf{v}^\star$. Referring to Figure 4-5, the state labeled (3) corresponds to the state $T_f(\rho,\mathbf{v})$ for $\rho = \frac{2(1-\alpha\gamma)}{1+\alpha^2-2\alpha\gamma}$ and is equidistant to the fixed point with $\mathbf{v}$. The second caveat is that there is a mixing property required of system operators in Theorem 4.3.7. The condition that the conic mixing parameter satisfy $\gamma < 1$ implies that $T_f(\rho,\mathbf{v})$ and $\mathbf{v}$ cannot be arbitrarily close


(1) $\mathbf{v}$
(2) $T(\mathbf{v})$
(3)
(4)
(5)
$\cos^{-1}(\gamma)$
$\mathbf{v}^\star$
$\alpha^2 r$
r
If $\|\mathbf{v}-\mathbf{v}^\star\| = r$ then $T_f(\rho, \mathbf{v})$ takes the state:
(1) provided that $\rho = 0$
(2) provided that $\rho = 1$
(3) provided that $\rho \in \left(0, \frac{2(1-\alpha\gamma)}{1+\alpha^2-2\alpha\gamma}\right)$
(4) provided that $\rho \in \left(\frac{2(1-\alpha\gamma)}{1+\alpha^2-2\alpha\gamma}, 1\right)$
(5) provided that $\rho = \frac{2(1-\alpha\gamma)}{1+\alpha^2-2\alpha\gamma}$
(6)
(4)
(1) $\mathbf{v}$
(3)
(5)
(2) $T(\mathbf{v})$
(7)
$\cos^{-1}(\gamma)$
$\mathbf{v}^\star$
$\alpha^2 r$
r
If $\mathbf{v}^{n-1} = \mathbf{v}$ then $\mathbf{v}^n$ takes the state:
(1) with probabiity $(1-p)^2$
(3) with probabiity p^2
(4) with probabiity $p(1-p)$
(5) with probabiity $p(1-p)$

(a) Achievable states using $T_f(\rho, \mathbf{v})$ for $\rho \in [0, 1]$.

(b) Possible outcomes according to the asynchronous protocol

Figure 4-5: An illustration of (a) the states achievable by application of T_f and (b) the possible outcomes of $\mathbf{v}^n$ for an arbitrary $\mathbf{v}^{n-1}$ according to the stochastic dynamics of (3.47) for a system operator T satisfying the conditions in Theorem 4.3.7.

together on the perimeter of the basin $B(\mathbf{v}^\star, r)$ in Figure 4-5. There is not an analogous condition for passive system operators. Consequently, the state sequence converges linearly to a fixed point and the optimal filtering parameter in the sense of provable convergence rates is $\rho^\star = \frac{1-\alpha\gamma}{1+\alpha^2-2\alpha\gamma}$. The analogous choice of the filtering parameter in Figure 4-4 is $\rho^\star = \frac{1}{2}$. In the respective contexts of Figures 4-4 and 4-5, the optimal filtering parameters correspond to selecting the state $T_f(\rho, \mathbf{v})$ which minimizes the squared distance $\|T_f(\rho, \mathbf{v}) - \mathbf{v}^\star\|^2$.

To conclude this subsection, we briefly comment on that the class of system operators satisfying the conic mixing property (4.72) are distinct from the class of system operators that are chaotic, as one would expect. Specifically, a system operator T is defined as being chaotic if the state sequence produced using a synchronous protocol, e.g. with $p = 1$, satisfies three criteria: (i) hypersensitivity to initial conditions, (ii) topological mixing, and (iii) dense periodic orbits. While system operators that are α-expansive everywhere generally satisfy condition (i), this is not sufficient to be chaotic. For example, the scalar operator $T \colon \mathbb{R} \to \mathbb{R}$ given by $T(\mathbf{v}) = -1.1\mathbf{v}$ satisfies both condition (i) and the conditions of Theorem 4.3.7 but is not chaotic. For this example, hypersensitivity to initial conditions is easily verified by

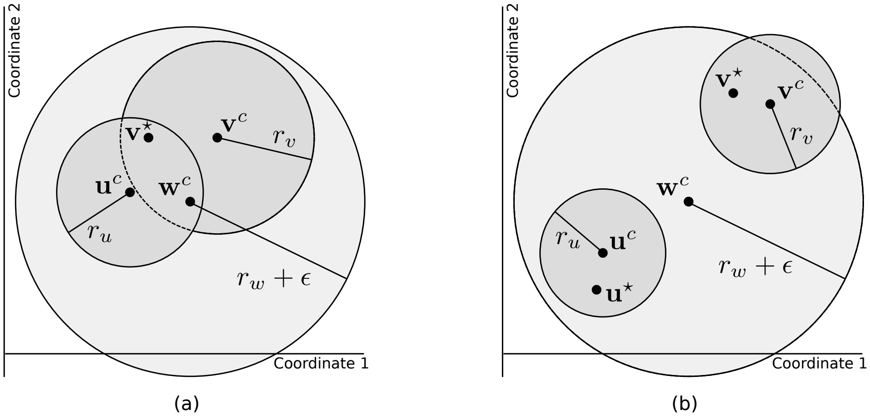


Figure 4-6: An illustration of the possible domains consistent with the system operator discussed in Section 4.3.4 in $\mathbb{R}^2$ demonstrating the utility of Theorems 4.3.2 and 4.3.4 in assembling stability conditions for α-expansive system operators.

defining the state sequences $\{\mathbf{v}^n : n \in \mathbb{N}_0\}$ and $\{\mathbf{u}^n : n \in \mathbb{N}_0\}$ respectively initialized by $\mathbf{v}^0$ and $\mathbf{u}^0 = \mathbf{v}^0 + \delta$ for some $\delta \neq 0$ and observing that

$$\mathbf{v}^n = (-1.1)^n \mathbf{v}^0 \qquad \text{and} \qquad \mathbf{u}^n = (-1.1)^n \left(\mathbf{v}^0 + \delta\right) \tag{4.82}$$

hence the distance $\|\mathbf{v}^n - \mathbf{u}^n\| = (1.1)^n |\delta|$ can be made arbitrarily large by selecting $n \in \mathbb{N}$ appropriately. While Theorem 4.3.7 requires mixing in the sense of (4.72), topological mixing is a stronger condition since the state sequence is required to be hypercyclic about a point, i.e. the state sequence must be dense in its phase space. Theorem 4.3.7 also makes no requirement similar to condition (iii).

4.3.5 | Local stability and robustness properties

Consistent with the remarks preceding Theorem 4.3.2, interconnective structures with α-conic system operators can generally satisfy the local condition (4.22) about many different states with different values of α for each. The stability and robustness of these structures can be understood by collecting together many different local stability and robustness conditions and aggregating their consequences onto one another. Local stability and robustness conditions include those in Theorems 4.3.1, 4.3.2, and 4.3.4.

As a concrete example of analyzing an expansive system operator, consider a system operator $T\colon \mathbb{R}^K \to \mathbb{R}^K$ that is known to be α-expansive about at least one state and additionally satisfies the following properties:

(i) T is α_w-dissipative about $\mathbf{w}^c$ where $\mathbf{w}^c \neq T(\mathbf{w}^c)$ (Theorem 4.3.2);

(ii) T is α_v-dissipative about all $\mathbf{v} \in B(\mathbf{v}^c, r_v)$ for a radius r_v satisfying $\|\mathbf{v}^c - T(\mathbf{v}^c)\| \leq (1 - \alpha_v)r_v$ (Theorem 4.3.4);

(iii) T is α_u-dissipative about all $\mathbf{u} \in B(\mathbf{u}^c, r_u)$ for a radius r_u satisfying $\|\mathbf{u}^c - T(\mathbf{u}^c)\| \leq (1 - \alpha_u)r_u$ (Theorem 4.3.4).

An illustration of the domain of a system operator with dimensionality $K = 2$ and that is consistent with these properties is provided in Figure 4-6. Referring to the figure, from property (i) it follows that within a finite number of iterations a state sequence will enter the basin $B(\mathbf{w}^c, r_w + \epsilon)$ for any $\epsilon > 0$. Combining the consequences from properties (ii) and (iii), scenario (a) depicts the case where the intersection of $B(\mathbf{u}^c, r_u)$ and $B(\mathbf{v}^c, r_v)$ is non-empty and therefore any state sequence that enters either basin will converge to the unique fixed point $\mathbf{v}^\star = \mathbf{u}^\star$ in mean (provided ρ is appropriately selected). Scenario (b) depicts the complementary case where the intersection of the two basins is empty and, therefore, T has at least two distinct fixed points. Furthermore, if a state sequence enters either basin, then it will converge in mean to the unique fixed point inside that basin.

4.4 | Solving SCSPs using convex, tree-based algorithms

In this section, we develop some general purpose processing instructions related to solving SCSPs and the optimization problem in (4.21), and we discuss a straightforward procedure for arranging these instructions into various greedy algorithms. The general strategy is to associated a tree graph with each problem so that the algorithms are interpretable as learning the topology of the tree by starting at the root node and seeking to uncover the furthest leaf node. As such, the algorithms are naturally suited to distributed and parallel computing environments since the processing performed to uncover each node can be performed independent of the processing used to uncover nodes in the same generation of the tree. To avoid confusion with previous nomenclature, we restrict our use of the word *node* in this section to mean the vertex of a graph and reserve the term *vertex* to mean the corner or extreme point of a polytope.

Originally presented in [84], the basic connection between an SCSP and the associated tree follows from defining the set $\mathcal{S}$ to describe the root node. Children of the root node correspond to coordinates of the elements in $\mathcal{S}$ that can be set to zero by producing new feasible elements. Upon receiving these elements, children then determine using the same rule if they have children of their own, but where they can only produce new elements that remain zero in all coordinates that have already been set to zero. The process then repeats itself where various ways of traversing the tree correspond to different algorithms in the same spirit. The processing instructions defined in this section are specifically designed to enforce this behavior so that the structure of the tree relates to the sparsity of the elements produced by each node in two ways: (i) the number of non-zero entries of each element produced by a node decreases with each successive generation of the tree, and (ii) siblings of the same parent node produce sets of feasible elements with different sparsity patterns. However, nodes with different parents may produce elements with the same sparsity pattern.

4.4.1 | Initializing the root node

Recall that trees are defined as graphs where any pair of nodes is connected by exactly one path. To initialize the algorithms under design, the root of the tree is established by defining a set $\mathcal{P}$ consisting of M distinct elements drawn from the feasible set $\mathcal{S}$ in (4.19), i.e. the set $\mathcal{P}$ takes the form

$$\mathcal{P} = \left\{\mathbf{v}^1, \ldots, \mathbf{v}^M \colon \mathbf{v}^m \in \mathcal{S}, m = 1, \ldots, M, \text{ and } \mathbf{v}^m \neq \mathbf{v}^n, \text{for } m \neq n\right\} \tag{4.83}$$

where each element $\mathbf{v}^m$ is allowed to be dense. To facilitate a geometric interpretation of the class of algorithms, note that sets of the form (4.83) can be thought of as encoding the vertex representation of a particular polytope embedded inside $\mathcal{S}$ and consisting of a continuum of feasible elements. In particular, $\mathcal{P}$ describes a closed and convex polytope corresponding to the convex hull of the elements forming its description as demonstrated by

$$\operatorname{convhull}(\mathcal{P}) \quad = \quad \left\{\sum_{m=1}^{M} \alpha_m \mathbf{v}^m \in \mathcal{S} \colon \sum_{m=1}^{M} \alpha_m = 1,\ \alpha_m \geq 0, \text{for } m = 1, \ldots, M\right\}. \tag{4.84}$$

For problems where the feasible set $\mathcal{S}$ has a natural expression through half-space representations, i.e. through a finite collection of linear equality and inequality constraints, standard vertex enumeration algorithms can be used to generate $\mathcal{P}$ [85]. Alternatively, convex optimization problems such as linear programs designed with random or systematically chosen cost functions can target and obtain well-spread vertices of $\mathcal{S}$ or feasible elements with other desirable properties. As will become evident shortly, greedy algorithms formed using the processing instructions developed in this section are unable to obtain sparse solutions contained within $\mathcal{S}$ that are not included in the convex hull of $\mathcal{P}$[2].

A pertinent question at this stage involves how the number M in (4.83) should be selected. To answer this question, we point to a result in [86] that says the following: let a polytope in an N-dimensional space with ν vertices achieve a maximum number of obtainable facets F. Then, the corresponding dual polytope maximizes the number of vertices for a fixed number of facets where ν is on the order of $F^{\lfloor \frac{N}{2} \rfloor}$. This fact, consistent with the complexity of basis exchange algorithms in linear programming, suggests enumerating the vertices in $\mathcal{S}$ is computationally intractable. We defer further comments to the later discussion.

4.4.2 | Incremental sparsity: processing instructions

We are now prepared to define the processing instructions used to form algorithms in the presented class. In particular, after receiving a convex polytope $\mathcal{P}$ in vertex representation from a parent node (or initialization for the root node), the processing performed by each node in the tree corresponds to some instantiation of the following three subroutines: (i) unveilChildren, (ii) vanishCoordinate, and (iii) reduceComplexity. In this subsection, the responsibility of each subroutine to the overall iteration is explained, and example pseudocode is provided for each. To assist with doing this, we will use the terminology that a node *vanishes coordinate d* if the result of its processing is a new polytope $\mathcal{P}'$ consisting entirely of elements that satisfy the following property:

$$\mathbf{v}_d \quad = \quad 0, \qquad \mathbf{v} \in \text{convhull}\left(\mathcal{P}'\right). \tag{4.85}$$

[2] This fact does not include algorithms which successfully transform leaf nodes into parent nodes by drawing additional elements from $\mathcal{S}$, as described in the section on the processing instruction unveilChildren.

Pseudocode 4.4.1 Example pseudocode for the subroutine UNVEILCHILDREN that is used to determine the set $\mathcal{I}$ describing the potential children a given node can produce using the set of feasible solutions $\mathcal{P}$.

```
function UNVEILCHILDREN($\mathcal{P}$)
    $\mathcal{I} \leftarrow \{d \in \{1, \ldots, K\} \colon \exists\, \mathbf{v}^+, \mathbf{v}^- \in \mathcal{P} \text{ s.t. } \mathbf{v}_d^+ > 0 \text{ and } \mathbf{v}_d^- < 0\}$
end function
```

Consequently, every node in the tree will satisfy the following property: for every element $\mathbf{v} \in \text{convhull}(\mathcal{P}')$, $\mathbf{v}_i = 0$ for all coordinates i that have been vanished along the unique path starting from the root node and ending at the node that produced $\mathcal{P}'$. Moreover, every node in the g-th generation of the tree receives a set of feasible elements from their parent where each element satisfies $\|\mathbf{v}\|_0 \leq K - g$. This observation provides a straightforward stopping criterion for solving an SCSP for a prespecified sparsity parameter t in (4.20).

UNVEILCHILDREN

Toward uncovering the structure of the tree, the first processing instruction we define allows a node to determine which, if any, children it can produce. To do this, nodes are equipped with a subroutine named UNVEILCHILDREN that identifies which, if any, coordinates can be vanished in the sense of (4.85) by further processing the polytope $\mathcal{P}$ received from that nodes parent. A simple sufficient condition for a particular node to vanish coordinate d is that the received polytope $\mathcal{P}$ contain at least one element $\mathbf{v}^+$ whose d-th entry is strictly positive and at least one element $\mathbf{v}^-$ whose d-th entry is strictly negative. If this condition is met, the node is said to be able to produce child d. Let $\mathcal{I} \subseteq \{\emptyset, 1, \ldots, K\}$ denote the set of all children that a given node can produce, i.e. $|\mathcal{I}|$ is the maximum number of children that node can produce. The quantity $|\mathcal{I}|$ can assist in determining whether trees or subtrees should be searched depth or breadth first. Pseudocode 4.4.1 provides an example of the subroutine UNVEILCHILDREN where the index set $\mathcal{I}$ is produced for a given polytope $\mathcal{P}$ on the basis of the sufficient condition above. From this perspective, nodes are classified as leaves if they cannot generate any children, i.e. if their index set $\mathcal{I}$ is empty.

The objective of an SCSP in the context of the tree graph is to identify a t-length path starting from the root node and traversing down the tree. This suggests straightforward stopping rules for algorithms using the subroutine UNVEILCHILDREN to include criteria such

Pseudocode 4.4.2 Example pseudocode for the subroutine which generates the set $\mathcal{P}'$ using elements of the set $\mathcal{P}$ by vanishing coordinate d according to (4.86) with $L = 2$.

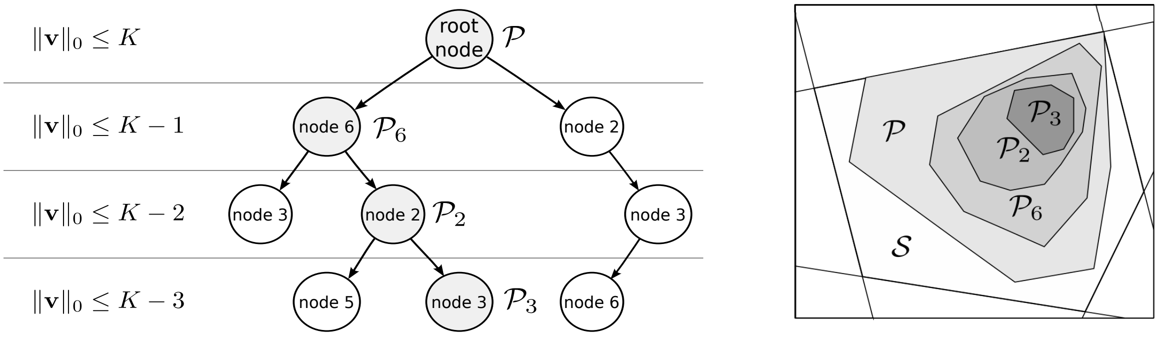
```
function vanishCoordinate($\mathcal{P}, d$)
    for each $\mathbf{v}^+ \in \mathcal{P}$ with $\mathbf{v}_d^+ > 0$ do
        for each $\mathbf{v}^- \in \mathcal{P}$ with $\mathbf{v}_d^- < 0$ do
            $\mathcal{P}' \leftarrow \left(\frac{\mathbf{v}_d^+}{\mathbf{v}_d^+ - \mathbf{v}_d^-}\right)\mathbf{v}^+ + \left(\frac{-\mathbf{v}_d^-}{\mathbf{v}_d^+ - \mathbf{v}_d^-}\right)\mathbf{v}^-$
        end for
    end for
end function
```

as the identification of either the longest or a sufficiently long path starting from the root node and ending at a leaf. The process of transforming a leaf node into one with children, which we shall refer to as *subtree exploration*, is sometimes possible by drawing additional elements from the set $\mathcal{S}$. To do this in practice, however, relies on the computational complexity associated with generating new elements from $\mathcal{S}$. One potential strategy for doing this in practice is to solve a convex optimization problem where the cost function is strategically chosen to target the sign of a particular coordinate while simultaneously imposing additional constraints to ensure the coordinates already vanished along the direct path from the leaf node to the root remain zero. If $\mathcal{S}$ is easily described using half-space representation, this strategy reduces to simply solving a linear programming problem.

vanishCoordinate

The next processing instruction we define allows a given node to produce a new polytope $\mathcal{P}'$ consisting of elements in $\mathcal{S}$ that all have their d-th coordinate vanished starting with the polytope $\mathcal{P}$ received from that nodes parent. To carry out this task, nodes are equipped with a subroutine named vanishCoordinate. To ensure elements in the convex hull of $\mathcal{P}'$ maintain feasibility according to (4.19) while simultaneously satisfying (4.85), we will rely on straightforward properties of convexity. Specifically, consider the following basic property that says a convex combination of L elements $\mathbf{v}^1, \ldots, \mathbf{v}^L$ taken from $\mathcal{P}$ produces a new element $\mathbf{v}'$ that is also an element of $\mathcal{P}$ as is demonstrated by

$$\mathbf{v}' = \sum_{l=1}^{L} \alpha_l \mathbf{v}^l \in \mathcal{P} \quad \text{for} \quad \sum_{l=1}^{L} \alpha_l = 1, \alpha_l \geq 0. \tag{4.86}$$



(a) An example of a partially unveiled tree graph

(b) A cross section of the polytope embeddings

Figure 4-7: An illustration depicting (a) three generations of a partially unveiled tree and (b) a cross-section of the polytope embeddings corresponding to the nodes S, "root node", "node 6", "node 2", and "node 3" on the highlighted walk.

It is straightforward to verify that the conditions required for the coordinate d to be an element of the set $\mathcal{I}$ are sufficient for a set of linear combination coefficients $\{\alpha_1, \ldots, \alpha_L\}$ to exist satisfying (4.86) for at least one collection of L elements $\mathbf{v}^l$ in $\mathcal{P}$. Any such set of coefficients is sufficient to systematically produce elements $\mathbf{v}'$ of $\mathcal{P}'$.

Pseudocode 4.4.2 provides an example of the subroutine vanishCoordinate. This particular example corresponds to using (4.86) with $L = 2$ and provides expressions for the weights α_1 and α_2. In addition, (4.86) can be used to verify that all coordinates previously vanished while generating the set $\mathcal{P}$ remain zero valued during the the process of generating $\mathcal{P}'$ and that the property $\mathcal{P}' \subseteq \mathcal{P}$ is satisfied since $\mathbf{v}_d^+ > 0 > \mathbf{v}_d^-$, i.e. $\mathcal{P}'$ is embedded within $\mathcal{P}$. From the perspective of solving nonlinear systems of equations by iteratively applying projection operators, as mentioned in the discussion surrounding (4.1), the processing instruction vanishCoordinate is easily interpretable as a projection operator since applying it repeatedly to the same set will not yield any additional elements.

Figure 4-7(a) illustrates the first three generations of a partially unveiled tree. Referring to the figure, the four polytopes labeled $\mathcal{P}, \mathcal{P}_6$, $\mathcal{P}_2$, and $\mathcal{P}_3$ indicate sets of elements that are generated along the highlighted path involving the nodes labeled "root node", "node 6", "node 2", and "node 3", respectively. A cross-section of these polytopes demonstrating the embedding property $\mathcal{P}_3 \subseteq \mathcal{P}_2 \subseteq \mathcal{P}_6 \subseteq \mathcal{P} \subseteq \mathcal{S}$ is also provided in Figure 4-7(b). The walk starting from "root node" and proceeding to "node 2" to " node 3" to "node 6" results in a set of elements with sparse vertices that have the same sparsity pattern as those produced by the highlighted path. This observation underscores the redundancy that can occur when selecting which nodes to vanish during an implementation of an algorithm and underlies the computational difficulty of the problem.

reduceComplexity

In response to the amount of computation required to enumerate all vertices that can be used to describe a general convex polytope, the next processing instruction allows nodes to reduce their computation at the expense of approximation. To do this, each node is equipped with a subroutine reduceComplexity that trades off between the thoroughness of a search and complexity, thereby allowing algorithms to remain tractable. More concretely, the complexity associated with Pseudocode 4.4.2 is often high, especially for problems with long root-to-leaf distances. To see this, let $\mathcal{P}$ denote the polytope received by a node and let M_d^+ denote the number of elements $\mathbf{v} \in \mathcal{P}$ whose d-th entry is strictly positive and let M_d^- denote the number of elements $\mathbf{v} \in \mathcal{P}$ whose d-th entry is strictly negative for some $d \in \mathcal{I}$. Also, let $\mathcal{P}'$ denote the result of vanishing coordinate d using the example subroutine vanishCoordinate. The cardinality of $\mathcal{P}'$ is M' and corresponds to

$$M_d' = M_d^+ M_d^- \tag{4.87}$$

possibly repeated vertices. It immediately follows that the total number of vertices generated by a node in the g-th generation of the tree is on the order of M^{2^g} where M is the number of elements used to define the root. By limiting the number of vertices used to describe the embedded polytope $\mathcal{P}'$ while vanishing coordinate d, the problem of tree traversal remains computationally tractable. This is explicitly at the expense of excluding regions of the polytope $\mathcal{P}$ in forming $\mathcal{P}'$ where a maximally sparse solution may reside.

An example of the subroutine reduceComplexity is provided in Pseudocode 4.4.2 to control the amount of computation performed at each node. The example is specifically set up to work by by limiting the total number of vertices used that satisfy $\mathbf{v}_d > 0$ and $\mathbf{v}_d < 0$ to $\widehat{M}_d^+ < M_d^+$ and $\widehat{M}_d^- < M_d^-$, respectively. In our experience with several examples, the number of unique vertices generated by vanishCoordinate is smaller than M_d', especially

Pseudocode 4.4.3 Example psuedocode describing a function that reduces the overall computational complexity required in vanishing the d-th coordinate according to VANISH-COORDINATE.

```
function REDUCECOMPLEXITY($\mathcal{P}, d$)
    $\mathcal{P}' \leftarrow$ at most $\widehat{M}_d^+$ unique vertices from $\mathcal{P}$ that satisfy $\mathbf{v}_d > 0$
    $\mathcal{P}' \leftarrow$ at most $\widehat{M}_d^-$ unique vertices from $\mathcal{P}$ that satisfy $\mathbf{v}_d < 0$
end function
```

in later generations of the tree. This heuristic may help guide the dynamic selection of $\widehat{M}_d^+$ and $\widehat{M}_d^-$ for a particular problem.

4.4.3 | Tree-search protocols

Structure from the description of the set $\mathcal{S}$ can often be used to inform the decision about how an algorithm should be directed to explore the tree using the processing instructions defined in the previous subsection. In this subsection, we discuss some important factors in determining how the algorithm selects which branches of the tree to search and in what order. For example, a key decision involves choosing the tree traversal protocol and is informed by balancing how quickly solutions of a predetermined sparsity level are to be sought after against the thoroughness that the initial polytope $\mathcal{P}$ is to be searched within. In the computer science literature, much work has focused on the development of efficient depth-first and breadth-first tree search protocols for similar problems [87]. Deciding which protocol to use, however, is typically obfuscated at the outset since we do not know the topology of the tree in the present context. In considering breadth-first searches, an important factor relates to the rate at which the trees width grows. For example, if W_g denotes the width of a tree at generation g, i.e. the total number of nodes across generation g in a complete unveiling of the tree, then W_g is bounded using the number of nodes in generation $g-1$ according to

$$W_g \leq W_{g-1}(K - g + 1) \tag{4.88}$$

where $W_0 = 1$. The details of particular problem classes may provide some form of side information that restricts the expansion rate of the trees width making breadth-first protocols seem more computationally attractive than originally implied by (4.88).

In considering depth-first search protocols, the availability of side information or other

Pseudocode 4.4.4 Example pseudocode describing a how to select a single-path pre-walk during runtime by selecting which coordinate to vanish accoridng to the number of possible sparse vertices produced.

```
function selectCoordinate($\mathcal{P}$, $\mathcal{I}$)
    $d' \leftarrow \arg\max_{d \in \mathcal{I}} |\{\mathbf{v} \in \mathcal{P} \colon \mathbf{v}_d > 0\}| \cdot |\{\mathbf{v} \in \mathcal{P} \colon \mathbf{v}_d < 0\}|$
end function
```

heuristics can often be used to choose between a predetermined coordinate vanishing order or the use of a run-time protocol for selecting coordinates to vanish. As an example of the former, the indices of magnitude sorted 1-norm relaxations to (4.21) may result in solutions of higher sparsity or, equivalently, the earlier unveiling of deeper leaves. As an example of the latter, pseudocode for a depth-first search protocol is provided in Pseudocode 4.4.4 where the subroutine selectCoordinate chooses an elimination order in real time that uses no heuristics or side information but aims to select coordinates that when vanished using vanishCoordinate produce embedded polytopes $\mathcal{P}'$ that contain the most possible elements.

4.4.4 | Subtree exploration

It is sometimes possible to employ techniques that are agnostic to the specific problem at hand in order to restart or extend a tree once a leaf node has been found. More generally, these same techniques can be used to identify if more children can be produced than those that can be identified using unveilChildren. We refer to these techniques as *subtree exploration* techniques and, in this subsection, present two examples of how they can easily be incorporated into the design of greedy algorithms by augmenting the processing instructions previously defined. When these techniques are applied once a leaf node has been found or after there are no remaining nodes with unexplored children, they provide direction in generating algorithms that are recurrent in their exploration of the tree rather than single application. For example, one such technique that was mentioned earlier involves transforming a leaf node into a parent node by drawing additional elements from $\mathcal{S}$. The success of this approach depends on the amount of computation associated with generating elements from $\mathcal{S}$ with additional structure imposed on them.

Another subtree exploration technique that can be used to both trim branches and identify unexplored children involves merging the polytopes produced by two nodes that satisfy a certain relationship, and then ending the exploration down one of their paths. This approach is similar to graph algorithms in the branch-cut style. In particular, let node n_1 in generation g have received polytope $\mathcal{P}_{n_1}$ from its parent and let node n_2 denote a second node in the same generation having received polytope $\mathcal{P}_{n_2}$ from a different parent. Further, assume the collection of vanished coordinates from node n_1 to the root is the same as those from node n_2 to the root. Then, we can terminate either node and take the union of the two polytopes as $\mathcal{P}_{n_1 \cup n_2}$ where

$$\mathcal{P}_{n_1 \cup n_2} \quad = \quad \left\{ \mathbf{v} \in \mathcal{S} \colon \mathbf{v} \in \mathcal{P}_{n_1} \text{ or } \mathbf{v} \in \mathcal{P}_{n_2} \right\}. \tag{4.89}$$

This polytope is then assigned to either node n_1 or node n_2 and processing continues down the path of that node, and the exploration down the other node is discontinued. For example, this subtree exploration technique can be used by merging the nodes labeled "node 3" and "node 6" in the third generation of the tree in Figure 4-7(a). This procedure is equivalent to searching for potential children nodes inside regions that are generally larger than the union of the convex hull of the individual polytopes $\mathcal{P}_{n_1}$ and $\mathcal{P}_{n_2}$ since $(\mathcal{P}_{n_1} \cup \mathcal{P}_{n_2}) \subseteq$ convhull$(\mathcal{P}_{n_1 \cup n_2})$.

4.5 | Numerical experiments

In this section, the results from several numerical experiments are presented to illustrate further the algorithms for solving CCSPs and SCSPs developed in this chapter. In the context of solving CCSPs, synchronous and asynchronous scattering algorithms corresponding to filtered implementations of scattering-form interconnective structures are presented to demonstrate the theoretical and practical agreement of the stability and robustness conditions derived in Section 4.3. The particular interconnective structures chosen in this section correspond to system operators that are not accounted for by the examples presented in Chapter 6. The basic setup for these numerical experiments is as follows:

(i) for the particular CCSP at hand, write the problem in interconnective form and then generate the system operator T associated with it;

(ii) identify a fixed-point $\mathbf{v}^\star \in \mathcal{F}_T$ of the system operator found in step (i);

(iii) implement the structure by generating state sequences $\{\mathbf{v}^n \colon n \in \mathbb{N}_0\}$ using filtered system operators T_f for various probability values p in the iteration (3.47) and filtering parameter $\rho = \frac{1}{2}$, and track the size of $\mathbf{v}^n - \mathbf{v}^\star$; and

(iv) repeat step (iii) for many trials where the initial state $\mathbf{v}^0$ is suitably chosen at random for each trial and average the results.

The convergence results of the synchronous and asynchronous scattering algorithms produced by following the procedure above are presented in the first two subsections below as a function of "equivalent (normalized) iterations," which corresponds to an iteration count summarizing the total amount of computation performed. Specifically, a single equivalent iteration under this count corresponds to K scalar delay modules triggering and can be approximated as $\frac{1}{p}$ iterations of (3.47) where each asynchronous delay module triggers with probability p according to the stochastic matrix $D^{(p)}$.

In the context of solving SCSPs, the processing instructions in Section 4.4 are paired with a depth-first tree-search protocol to form an algorithm in the style of a synchronous runloop. This algorithm is then used to design a frequency-selective filter with a sparse impulse response in the final subsection below. The impulse response obtained by this method is then compared with alternative impulse responses obtained using conventional sparse filter design methods from the literature.

4.5.1 | Stability and robustness of passive everywhere system operators

The first numerical experiment we discuss corresponds to solving the CCSP associated with an interconnective structure with a passive everywhere system operator whose representation as a fixed-point problem in (4.15) and (4.16) uses a coordinate-wise absolute value function for the memoryless nonlinearity $m(\cdot)$. For interconnecting networks generated in a certain way, this particular interconnective structure was shown in [88] to be in the same interconnective equivalence class as the optimality conditions associated with linear pro-

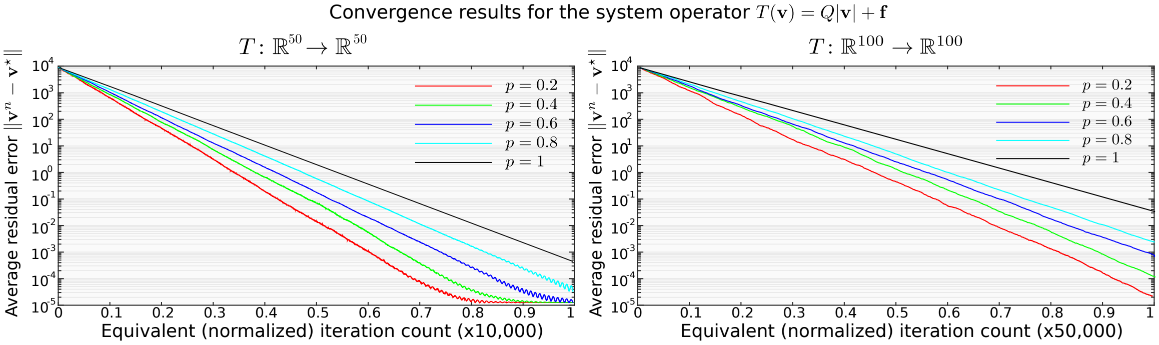


Figure 4-8: Numerical convergence results for a filtered implementation of the passive everywhere system operator in (4.90) with filter parameter $\rho = 0.5$. The depicted results were obtained by averaging over 1000 trials for various values of the stochastic parameter p and dimension sizes $k = 50$ (left) and $k = 100$ (right). The initial system state $\mathbf{v}^0$ is randomly selected on the boundary of $B(\mathbf{v}^\star, 10^4)$ for each trial.

gramming problems. In Chapter 5, we use the quadratic conservation principle inherent to CCSPs to elaborate on this link and to extend the connection to more general classes of optimization problems. In this subsection, we consider a different subset of these CCSPs where Theorem 4.3.6 summarizes the conditions relevant to the stability and robustness of the processing system.

Let $T\colon \mathbb{R}^K \to \mathbb{R}^K$ denote the system operator associated with an orthogonal-form interconnective structure as described above, i.e. the system operator T takes the form

$$T(\mathbf{v}) \quad = \quad Q|\mathbf{v}| + \mathbf{f} \tag{4.90}$$

where $Q\colon \mathbb{R}^K \to \mathbb{R}^K$ is a randomly generated orthogonal matrix obtained by first drawing a $K \times K$ matrix from a Gaussian ensemble and then projecting it to the nearest orthogonal matrix in the Frobenius norm sense, and where $\mathbf{f} \in \mathbb{R}^K$ is a Gaussian random vector. Closed form projection operators and efficient numerical methods for performing the matrix projection used to obtain Q are discussed in [89]. In implementing algorithms derived from (4.90), the state sequences $\mathbf{v}^n$ are generated using filtered realizations of the system operator where the filter parameter ρ is selected to be one half.

The numerical convergence results for this experiment averaged over 1000 trials for problems with dimension sizes $K = 50$ and $K = 100$ and asynchronous delay probabilities $p = 0.2, 0.4, 0.6, 08, 1.0$ are provided in Figure 4-8. Referring to the trends observed in the figure, the state sequences converge to their fixed-points at an approximately linear rate initially and then slow to a sublinear trend once they reach a sufficiently nearby basin to the fixed-point. This trend is consistent with the comments in the proof of Theorem 4.3.6 in Appendix A. Consistent with the theorem, the synchronous algorithm corresponding to $p = 1$ exhibits monotone convergence. Finally, we comment that the synchronous and asynchronous algorithms in this subsection where the identity function replaces the absolute value function provide ways to solve linear systems of the form $(I - Q)\mathbf{v} = \mathbf{f}$ without explicitly decomposing or inverting $(I - Q)$ or Q. By proper selection of the vector $\mathbf{f}$, asynchronous algorithms for determining certain eigenvectors and eigenvalues of Q are provided.

4.5.2 | Stability and robustness of exponential system operators

In the discussion following the proof of Theorem 4.3.5, it was shown that the scalar operator $T(\mathbf{v}) = e^{-\mathbf{v}}$ is not α-conic for any finite value of α, but the composed function $T \circ T$ is with coefficient $\alpha = \frac{1}{e}$. The second numerical experiment we discuss corresponds to solving the CCSP associated with an interconnective structure with a system operator that is the multidimensional generalization of this operator. In particular, the experiment described next illustrates numerically that a synchronous and direct implementation of a system operator satisfying the conditions in Theorem 4.3.5 extends to asynchronous and unfiltered implementations too. This observation is also consistent with the remark that asynchronous algorithms formed by composing dissipative system operators are stable and robust in the sense of Theorem 4.3.3.

Let $T\colon \mathbb{R}^K \to \mathbb{R}^K$ denote the system operator associated with an orthogonal-form interconnective structure as described above, i.e. the system operator T takes the form

$$T(\mathbf{v}) = e^{-Q\mathbf{v}} + \mathbf{f} \tag{4.91}$$

where $Q\colon \mathbb{R}^K \to \mathbb{R}^K$ is an orthogonal matrix with eigenvalues bounded away from -1, $\mathbf{f} \in \mathbb{R}^K$ is drawn from a Gaussian vector distribution, and the exponential is computed coordinatewise. The matrix Q is generated by drawing a $K \times K$ matrix from a Gaussian ensemble and then computing the Cayley transform of its skew-symmetric portion.

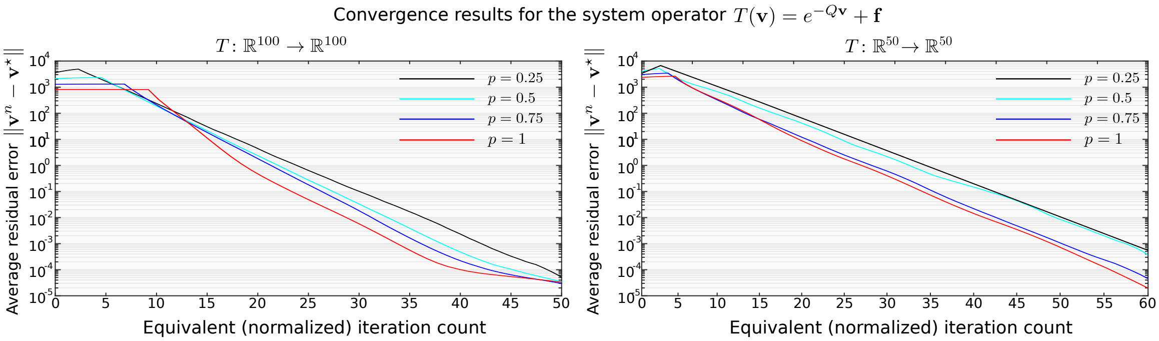


Figure 4-9: Numerical convergence results for a filtered implementation of the transcendental system operator in (4.91) with filter parameter $\rho = 1.0$. The depicted results were obtained by averaging over 1000 trials for various values of stochastic parameter p and dimension sizes $k = 100$ (left) and $k = 50$ (right).

The numerical convergence results for this experiment averaged over 1000 trials for problems with dimension sizes $K = 50$ and $K = 100$ and asynchronous delay probabilities $p = 0.25, 0.5, 0.75, 1.0$ are provided in Figure 4-9. Referring to the trends observed in the figure, the state sequences converge to their fixed-points at an approximately linear rate after the initial transient effects wear off. Note that the observed rates appear to be independent of the delay probability p. Similar to the comments in the previous subsection, we remark in closing that the asynchronous algorithms associated with the fixed-point problem described in this subsection provide ways to solve transcendental exponential equations that have no analytic solutions, i.e. whose solutions cannot be expressed in closed form as polynomial equations.

4.5.3 | Convex, tree based sparse filter design algorithms

The design of sparse impulse responses for linear and time-invariant filtering systems is computationally difficult in general but leads to many practical advantages including reduced hardware area and power consumption on very-large scale integrated (VLSI) systems. Intuitively, the difficulty in the design problem stems from the combinatoric nature of sparsity patterns rather than the design of the filter tap values after having chosen a particular sparsity pattern. Formulated using the SCSP setup, common design specifications that are easily expressed in half-space representation typically do not meet the constraint qualifications that compressive sensing algorithms rely upon to equation 1-norm and 0-norm minimization problems. Many design procedures have been proposed that have a similar flavor to those in compressive sensing, e.g. greedy algorithms that iteratively solve weighted 1-norm linear programs [90]. The formulation of these design problems using non-convex optimization problem statements has also been explored and used to generate sparse impulse responses for a limited number of constraint types [91]. More broadly, alternative formulations that rely on convexity principles have been used to assemble projection onto convex sets style algorithms where initial designs are successively refined until a design satisfies each constraint. These methods often result in final designs that satisfy all of the constraints but are not explicitly optimal with regard to any metric [66, 67].

The setup for a sparse filter design problem is described next. Let $h[n]$ denote an impulse response defined over the support $[0, 2N]$. Furthermore, let Ω_{pb} and Ω_{sb} denote the respective passband and stopband intervals given by

$$\Omega_{pb} \quad = \quad \{\omega_i \in \mathbb{R} \colon |\omega_i| \leq \omega_{pb}, 1 \leq i \leq I\} \tag{4.92}$$

and

$$\Omega_{sb} \quad = \quad \{\omega_i \in \mathbb{R} \colon \omega_{sb} \leq |\omega_i| \leq \pi, 1 \leq i \leq I\} \tag{4.93}$$

where the passband and stopband edges ω_{pb} and ω_{sb} satisfy $0 < \omega_{pb} < \omega_{sb} < \pi$, and the integer I is chosen to be large enough that the frequency axis is sufficiently sampled with respect to the size of the filter support. Over these intervals, frequency-domain attenuation constraints are imposed so that a candidate impulse response is said to be feasible if it's Fourier transform amplitudes deviate no more than δ_{pb} and δ_{sb} from the ideal amplitude response over the passband and stopband intervals, respectively. Let $D(\omega)$ denote the ideal amplitude response and, for the example in this subsection, be defined as unity on the interval Ω_{pb} and zero on the interval Ω_{sb}. The definition of the feasible set $\mathcal{S}$ in (4.19) is

Algorithm 4.1 A single-path, depth-first algorithm with randomized vertex reduction and a run-time order selection subroutine.

```
$\mathcal{I} \leftarrow$ unveilChildren($\mathcal{P}$)
while $\mathcal{I} \neq \emptyset$ do
    $d \leftarrow$ selectCoordinate($\mathcal{P}, \mathcal{I}$)
    $\mathcal{P} \leftarrow$ reduceComplexity($\mathcal{P}, d$)
    $\mathcal{P} \leftarrow$ vanishCoordinate($\mathcal{P}, d$)
    $\mathcal{I} \leftarrow$ unveilChildren($\mathcal{P}$)
end while
```

written in terms of a zero-phase impulse response according to

$$\mathcal{S} = \{\mathbf{v} \in \mathbb{R}^N : \quad |T(\omega, \mathbf{v}) - D(\omega)| \leq \delta_{pb},\ \omega \in \Omega_{pb}, \tag{4.94}$$

$$|T(\omega, \mathbf{v}) - D(\omega)| \leq \delta_{sb},\ \omega \in \Omega_{sb}\} \tag{4.95}$$

where

$$T(\omega, \mathbf{v}) = \sum_{k=1}^{N} \mathbf{v}_k \cos(\omega(k-1)) \tag{4.96}$$

and $\mathbf{v}_1 = h[0]$ and $\mathbf{v}_k = 2h[k-1]$ for $2 \leq k \leq N$. The design example presented so far can easily be modified to describe other classes of filters and to incorporate many additional design constraints. For example, the half-space representation of many common filter design constraints is shown in [66] to be a closed and convex set. The values of the design specifications for the example in this subsection are chosen to be the same as those in [92] to allow for comparison and are: passband cutoff frequency $\omega_{pb} = 0.20\pi$, stopband cutoff frequency $\omega_{sb} = 0.25\pi$, passband ripple $\delta_{pb} = 0.01$ (linear scale), stopband ripple $\delta_{sb} = 0.1$ (linear scale), and support parameter $N = 31$.

Pseudocode for an example sparse filter design algorithm is provided in Algorithm 4.1 by assembling the processing instructions developed in Section 4.4 into a runloop. Note that the processing instructions are guided by the single-path depth-first coordinate selection procedure that is described by the subroutine selectCoordinate in Pseudocode 4.4.4. To apply Algorithm 4.1 to the sparse filter design problem, the root node is equipped with an initial polytope $\mathcal{P}$ populated using $M = 500$ vertices drawn from $\mathcal{S}$ where each vertex corresponds to a non-sparse filter that meets the design specifications. These initial elements are, in particular, selected by solving a sequence of linear programs using a basis exchange

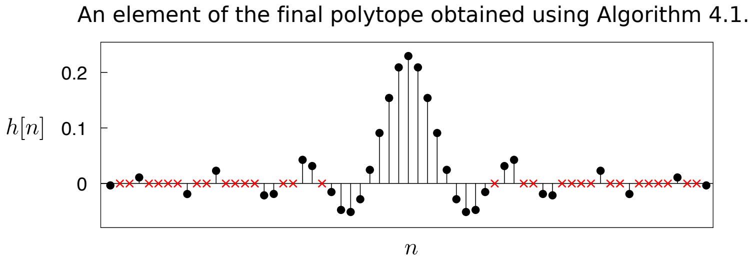


Figure 4-10: An impulse response corresponding to one sparse solution generated using Algorithm 4.1. Zero valued coefficients are marked with red x's.

method where the coefficients of the objective vector are drawn uniformly at random from the interval $[-1, 1]$. To retain tractability, the subroutine reduceComplexity is applied in each iteration with parameters $\widehat{M}^{+} = \widehat{M}^{-} = 500$, i.e. the polytope $\mathcal{P}$ generated by each node cannot exceed $250,000$ elements at any stage. As described, the algorithm does not make any attempt at subtree exploration, i.e. no additional processing is used to draw additional samples to transform a leaf node into a parent node. Therefore, the algorithm naturally terminates upon the discovery of a leaf.

A randomly selected element belonging to the leaf node discovered by running Algorithm 4.1 has been transformed into an impulse response $h[n]$ and is depicted in Figure 4-10. This particular solution has an equal sparsity level to the solution generated in [92] by solving a sequence of linear programming problems but has a different sparsity pattern. The length of the walk from the root node to the leaf node for the solution depicted is 15 and directly relates to the sparsity seen in the impulse response. Variations and extensions to Algorithm 4.1 follow in a straightforward way, e.g. by switching to differnet order selection rules depending on the current sparsity level or pattern. Of course, a natural alternative is to select the fixed ordering before runtime corresponding to the solution to the 1-norm relaxation of (4.21). In comparing this path selection rule with the one used above for many trials and different specifications, we comment that the rule in Pseudocode 4.4.4 tends to result in sparser filters. The well-known fact that the 1-norm solution does not degrade gracefully in the frequency-domain with respect to the specifications in part validates this observation. By using subtree extension techniques and other variations of Algorithm 4.1, impulse responses of different sparsity levels and patterns than the one depicted in Figure 4-10 are obtainable.

Chapter 5

Scattering structures for solving optimization problems

There are a variety of places where mathematical optimization problems manifest themselves in signal processing applications, including as design tools for optimal parameter selection and as processing stages in the signal chain itself [93]. The goal of this chapter is to unite these applications by developing a class of signal processing systems whose equilibrium states or steady-state variable values satisfy a variational property, and, therefore, have an interpretation as solutions to associated optimization problems. Drawing upon the interconnective framework and the capability of scattering algorithms to solve a wide range of CCSPs, the tools developed in this chapter more broadly facilitate the design and implementation of asynchronous, distributed optimization algorithms realized as scattering-form signal processing systems, and therefore provide straightforward methods for solving a broad class of convex and non-convex optimization problems.

This chapter begins with a review of some known conditions relating the stationarity principles underlying a class of optimization problems and the orthogonality principles pertaining to conservative vector spaces. Using these conditions, a primal and dual pair of optimization problems are proposed whose joint feasibility conditions serve as sufficient conditions for stationarity of the associated problems. The interconnective description of the stationarity conditions is shown to reduce to a CCSP, and the results in Chapter 4 suggest a variety of structures and sufficient conditions for stabilization under which the primal and dual optimization problems can be solved asynchronously and simultaneously.

To allow scattering-form signal processing systems to be assembled from inspection of optimization problem statements, several constitutive modules are derived that correspond to common feasibility constraints and objective functions, and straightforward procedures are provided for deriving additional modules. The scattering algorithms that result from attaching these modules to centralized or decentralized interconnecting networks specifically operate by passing around algorithm variables that are linear combinations of the primal and dual decision variables, and the scattering coordinate transforms describe these combinations. A numerical method for decentralizing the interconnecting networks in a system's interconnective description is presented, and an interpretation is discussed of the solutions produced by scattering algorithms that have been terminated early. To conclude the chapter, connections are made between the proposed class of scattering algorithms and existing methods from the optimization literature, and we find that while scattering algorithms are distinct, they are also in the same interconnective equivalence class as many of their gradient-based and proximal counterparts.

5.1 | A class of conservative optimization problems

In this section, a class of *conservative optimization problems* is defined whose structure is consistent with the class of problems discussed in [88, 94]. The formulation of this class is motivated by connections between stationarity and orthogonality principles in conservative vector spaces, and these connections allow for variational interpretations of signal processing systems organized into canonical form. Optimization problems in the presented class will generally be described in two equivalent ways. The first description, referred to as *canonical form*, is suited to the characterization of the stationarity conditions. As the naming convention suggests, the interconnective description of the stationarity conditions associated with problems written this way is a canonical-form structure. The second description, referred to as *reduced form*, only exists for a subset of problems written in canonical form and is useful in making connections with common optimization problem classes.

The strategy we follow in this section to define the class of conservative optimization problems is specifically to define canonical and reduced forms of a primal optimization problem as special cases of the general form

$$\underset{(\mathbf{x}_1,\ldots,\mathbf{x}_M)}{\text{minimize}} \quad Q(\mathbf{x}) \tag{5.1}$$

$$\text{s.t.} \quad f(\mathbf{x}) \in \mathcal{V}_A \tag{5.2}$$

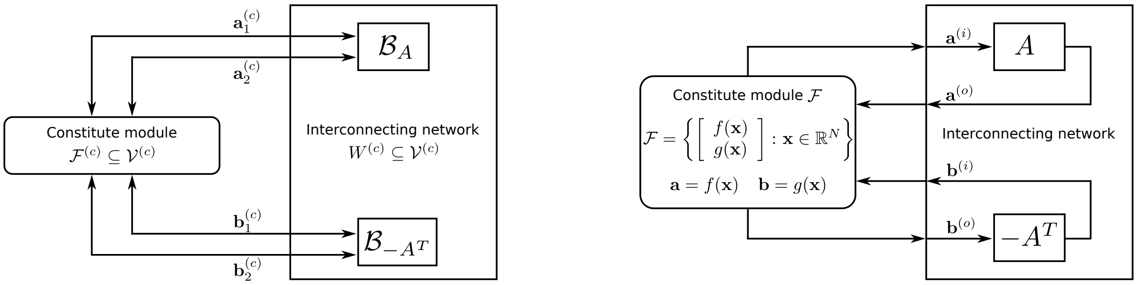
where Q denotes the primal cost functional, $\mathbf{x}$ denotes the decision vector, f denotes a vector-valued function defined on $\mathbf{x}$, and $\mathcal{V}_A$ denotes a vector space. Similarly, we define the canonical and reduced forms of the corresponding “dual” optimization problem as special cases of the general form

$$\underset{(\mathbf{x}_1,\ldots,\mathbf{x}_M)}{\text{maximize}} \quad -R(\mathbf{x}) \tag{5.3}$$

$$\text{s.t.} \quad g(\mathbf{x}) \in \mathcal{V}_B. \tag{5.4}$$

where R denotes the dual cost functional, g denotes a vector-valued function defined on $\mathbf{x}$, and $\mathcal{V}_B$ denotes a vector space that is orthogonal to $\mathcal{V}_A$. As will be explained shortly, the stationarity conditions impose a set of relationships between the functions f, g, Q, and R so that the primal and dual feasibility conditions (5.2) and (5.4) collectively serve as sufficient conditions for stationarity of the cost functionals (5.1) and (5.3) about any small perturbations of the shared decision vector $\mathbf{x}$ for which $f(\mathbf{x})$ and $g(\mathbf{x})$ remain feasible, i.e. for which $f(\mathbf{x})$ remains in the vector space $\mathcal{V}_A$ and $g(\mathbf{x})$ remains in the vector space $\mathcal{V}_B$.

The interconnective description of the primal and dual feasibility conditions can be described as a canonical-form signal processing system where the relationships required between f, g, Q, and R are enforced using constitutive modules and the orthogonal subspaces $\mathcal{V}_A$ and $\mathcal{V}_B$ are generated using interconnecting networks. The interconnective structures depicted in Figure 5-1 illustrate this. Specifically, the structure in (b) corresponds to the special case of the general canonical-form structure in (a) where the constitutive module $\mathcal{F}$ imposes the conditions $\mathbf{a} = f(\mathbf{x})$ and $\mathbf{b} = g(\mathbf{x})$ for appropriately chosen functions f and g, and the interconnecting network enforces that $\mathbf{a}$ and $\mathbf{b}$ belong to orthogonal subspaces.


$\mathbf{a}_1^{(c)}$
$\mathcal{B}_A$
$\mathbf{a}_2^{(c)}$
Constitute module
$\mathcal{F}^{(c)} \subseteq \mathcal{V}^{(c)}$
Interconnecting network
$W^{(c)} \subseteq \mathcal{V}^{(c)}$
$\mathbf{b}_1^{(c)}$
$\mathcal{B}_{-A^T}$
$\mathbf{b}_2^{(c)}$
$\mathbf{a}^{(i)}$
A
Constitute module $\mathcal{F}$
$\mathbf{a}^{(o)}$
$\mathcal{F} = \left\{ \begin{bmatrix} f(\mathbf{x}) \\ g(\mathbf{x}) \end{bmatrix} : \mathbf{x} \in \mathbb{R}^N \right\}$
Interconnecting network
$\mathbf{a} = f(\mathbf{x}) \quad \mathbf{b} = g(\mathbf{x})$
$\mathbf{b}^{(i)}$
$\mathbf{b}^{(o)}$
$-A^T$

(a) A general canonical-form structure

(b) Canonical-form structure for stationarity conditions

Figure 5-1: Canonical-form structures illustrating the notation used to state the stationarity principle. (a) A general canonical-form structure. (b) The specific form of the stationarity conditions.

5.1.1 | A stationarity principle for orthogonal vector spaces

In this subsection, we review the particular set of stationarity conditions that pertain to the pair of optimization problems described by (5.1) through (5.2) and (5.3) through (5.4), where the conditions enforce the vector-valued functions f and g to give rise to a stationarity principle involving the cost functionals Q and R whenever the functions f and g evaluate to vectors in orthogonal vector spaces. A general discussion of the relationship between these conditions and their role surrounding the concepts of stationary content and co-content in electrical network theory, as originally presented in references such as [9,95], is omitted here and can be found in [11, Theorem 5.1].

To begin, consider the inner product space $(\mathbb{R}^N, \langle\cdot,\cdot\rangle)$ where $\langle\cdot,\cdot\rangle$ denotes the standard inner product and let $\mathcal{V}_A \subseteq \mathbb{R}^N$ and $\mathcal{V}_B \subseteq \mathbb{R}^N$ denote two orthogonal vector spaces. The subspaces $\mathcal{V}_A$ and $\mathcal{V}_B$ are not required to uniquely decompose $\mathbb{R}^N$ but must satisfy pairwise orthogonality between their elements as demonstrated by

$$\langle \mathbf{a}, \mathbf{b} \rangle = 0, \qquad \mathbf{a} \in \mathcal{V}_A, \mathbf{b} \in \mathcal{V}_B. \tag{5.5}$$

With this setting established, we proceed to discuss the roles played by the functions f and g and functionals Q and R while listing their requisite assumptions. First, let $f\colon \mathbb{R}^M \to \mathbb{R}^N$ and $g\colon \mathbb{R}^M \to \mathbb{R}^N$ denote functions whose respective Jacobians are assumed to exist at certain points. Writing this out to make the notation clear, let $J_f\colon \mathbb{R}^M \to \mathbb{R}^N$ denote the

Jacobian of the function f evaluated at one such point $\mathbf{x}^\star \in \mathbb{R}^M$ according to

$$J_f(\mathbf{x}^\star) \quad = \quad \left(\nabla f_1^T(\mathbf{x}^\star),\ \nabla f_2^T(\mathbf{x}^\star),\ \ldots,\ \nabla f_N^T(\mathbf{x}^\star)\right) \tag{5.6}$$

where the notation $f_n(\mathbf{x}^\star)$ indicates the n-th entry of f evaluated at $\mathbf{x}^\star$, and where the gradient $\nabla f_n(\mathbf{x}^\star)$ is given by

$$\nabla f_n(\mathbf{x}^\star) \quad = \quad \left(\frac{\partial f_n}{\partial \mathbf{x}_1}(\mathbf{x}^\star),\ \frac{\partial f_n}{\partial \mathbf{x}_2}(\mathbf{x}^\star),\ \ldots,\ \frac{\partial f_n}{\partial \mathbf{x}_M}(\mathbf{x}^\star)\right), \qquad n = 1, \ldots, N. \tag{5.7}$$

Similarly, let $J_g \colon \mathbb{R}^M \to \mathbb{R}^N$ denote the Jacobian of the function g evaluated at the point $\mathbf{x}^\star \in \mathbb{R}^M$ according to

$$J_g(\mathbf{x}^\star) \quad = \quad \left(\nabla g_1^T(\mathbf{x}^\star),\ \nabla g_2^T(\mathbf{x}^\star),\ \ldots,\ \nabla g_N^T(\mathbf{x}^\star)\right) \tag{5.8}$$

where the notation $g_n(\mathbf{x}^\star)$ indicates the n-th entry of g evaluated at $\mathbf{x}^\star$, and where the gradient $\nabla g_n(\mathbf{x}^\star)$ is given by

$$\nabla g_n(\mathbf{x}^\star) \quad = \quad \left(\frac{\partial g_n}{\partial \mathbf{x}_1}(\mathbf{x}^\star),\ \frac{\partial g_n}{\partial \mathbf{x}_2}(\mathbf{x}^\star),\ \ldots,\ \frac{\partial g_n}{\partial \mathbf{x}_M}(\mathbf{x}^\star)\right), \qquad n = 1, \ldots, N. \tag{5.9}$$

The stationarity principle will be stated in terms of the functionals $Q \colon \mathbb{R}^M \to \mathbb{R}$ and $R \colon \mathbb{R}^M \to \mathbb{R}$, and these functionals will each be decomposed as the sum of several functionals that the stationarity conditions will be written in terms of. In particular, the functional Q is decomposed into the sum of N functionals $Q_n \colon \mathbb{R}^M \to \mathbb{R}$, for $n = 1, \ldots, N$, according to

$$Q(\mathbf{x}) = \sum_{n=1}^{N} Q_n(\mathbf{x}). \tag{5.10}$$

Similarly, the functional R is decomposed into the sum of N functionals $R_n \colon \mathbb{R}^M \to \mathbb{R}$, for $n = 1, \ldots, N$, according to

$$R(\mathbf{x}) = \sum_{n=1}^{N} R_n(\mathbf{x}). \tag{5.11}$$

To state the stationarity principle, the following relationships are imposed between the functions f and g and the functionals Q_n and R_n, for $n = 1, \ldots, N$:

$$\begin{aligned} \nabla Q_n(\mathbf{x}) &= g_n(\mathbf{x}) \nabla f_n(\mathbf{x}), \qquad n = 1, \ldots, N && (5.12) \\ \nabla R_n(\mathbf{x}) &= f_n(\mathbf{x}) \nabla g_n(\mathbf{x}), \qquad n = 1, \ldots, N && (5.13) \\ Q_n(\mathbf{x}) + R_n(\mathbf{x}) &= f_n(\mathbf{x}) g_n(\mathbf{x}), \qquad n = 1, \ldots, N. && (5.14) \end{aligned}$$

The following theorem summarizes the stationarity principle that arises due to any element $\mathbf{x}^\star \in \mathbb{R}^M$ for which $f(\mathbf{x}^\star)$ evaluates to a vector in $\mathcal{V}_A$ and $g(\mathbf{x}^\star)$ evaluates to a vector in $\mathcal{V}_B$. It is precisely at such elements $\mathbf{x}^\star$ that a stationarity principle holds, and is an immediate consequence of the relationships between the functions and functionals described by (5.12) through (5.14). In the statement of the theorem, the notation $D_{\mathbf{u}}f(\mathbf{x})$ is used to indicate the standard definition of a directional derivative of the function f evaluated at the point $\mathbf{x}$ in the direction $\mathbf{u}$, and this directional derivative $D_{\mathbf{u}}f(\mathbf{x})$ can be related to the Jacobian of f evaluated at $\mathbf{x}$ using the notation defined above according to

$$D_{\mathbf{u}}f(\mathbf{x}) = J_f(\mathbf{x})\mathbf{u}. \qquad (5.15)$$

Theorem 5.1.1. [Baran's stationarity principle [11]]**.** *This theorem pertains to two orthogonal subspaces $\mathcal{V}_A \subseteq \mathbb{R}^N$ and $\mathcal{V}_B \subseteq \mathbb{R}^N$ and two functions $f\colon \mathbb{R}^M \to \mathbb{R}^N$ and $g\colon \mathbb{R}^M \to \mathbb{R}^N$, as well as a point $\mathbf{x}^\star$ for which $f(\mathbf{x}^\star) \in \mathcal{V}_A$ and $g(\mathbf{x}^\star) \in \mathcal{V}_B$, and for which $J_f(\mathbf{x}^\star)$ and $J_g(\mathbf{x}^\star)$ exist. At any such point, the total content Q is stationary with respect to small variations taken in any direction $\mathbf{u}$ for which $D_{\mathbf{u}}f(\mathbf{x}^\star) \in \mathcal{V}_A$. Likewise, the total co-content R is stationary with respect to small variations taken in any direction $\mathbf{w}$ for which $D_{\mathbf{w}}g(\mathbf{x}^\star) \in \mathcal{V}_B$. Furthermore, $Q(\mathbf{x}^\star) = R(\mathbf{x}^\star)$ at any such point.*

An argument that essentially proves this theorem in the context of the the interconnective description of the feasibility conditions associated with the canonical-form primal and dual conservative optimization problems is provided in Section 5.1.3. An interpretation of this theorem is also provided in Section 5.1.5 for the reduced-form problem descriptions developed in Section 5.1.4 to assist with connecting the stationarity principle with alternative descriptions of optimality conditions such as the KKT conditions discussed in Section 2.5.1.

5.1.2 | Notational conventions for describing the stationarity conditions

The purpose of this subsection is to clearly define the notational conventions that will be used to address the individual features of the canonical-form description of the primal and dual conservative optimization problems defined in the next subsection. This notation will also assist with addressing the individual constitutive modules and interconnects that are critical to designing distributed and asynchrnous scattering algorithms from the interconnective description of the joint primal and dual feasibility conditions associated with the problem at hand. To state this notation formally, we turn to the terminal vector partitioning schemes outlined by (3.9) through (3.11) and (3.32), which we proceed to restate in the present context for clarity.

In particular, to assist with addressing the individual components of the feasibility conditions associated with canonical-form primal problems, we shall refer to subvectors of the shared decision vector $\mathbf{x}$ of the form

$$(\mathbf{x}_1, \ldots, \mathbf{x}_N) \quad = \quad (\mathbf{x}_1^{(CR)}, \ldots, \mathbf{x}_K^{(CR)}) \tag{5.16}$$

as well as subvectors of the primal decision vector $\mathbf{a}$ of the form

$$\begin{aligned}(\mathbf{a}_1, \ldots, \mathbf{a}_N) \quad &= \quad (\mathbf{a}_1^{(CR)}, \ldots, \mathbf{a}_K^{(CR)}) &(5.17)\\ &= \quad (\mathbf{a}_1^{(LI)}, \ldots, \mathbf{a}_L^{(LI)}) &(5.18)\end{aligned}$$

where each of the subvectors $\mathbf{x}_k^{(CR)}$ and $\mathbf{a}_k^{(CR)}$, for $k = 1, \ldots K$, and $\mathbf{a}_l^{(LI)}$, for $l = 1, \ldots L$, additionally satisfies

$$\begin{aligned}\mathbf{a}_k^{(CR)} \quad &= \quad f_k(\mathbf{x}_k^{(CR)}), \qquad k = 1, \ldots, K &(5.19)\\ \mathbf{a}_l^{(LI)} \quad &= \quad (\mathbf{a}_l^{(i)}, \mathbf{a}_l^{(o)}), \qquad l = 1, \ldots, L &(5.20)\end{aligned}$$

and where the functions $f_k \colon \mathbb{R}^{N_k^{(CR)}} \to \mathbb{R}^{N_k^{(CR)}}$, for $k = 1, \ldots, K$, map the shared decision subvector $\mathbf{x}_k^{(CR)}$ to the primal decision subvector $\mathbf{a}_k^{(CR)}$. Similarly, to assist with addressing the individual components of the feasibility conditions associated with canonical dual prob-

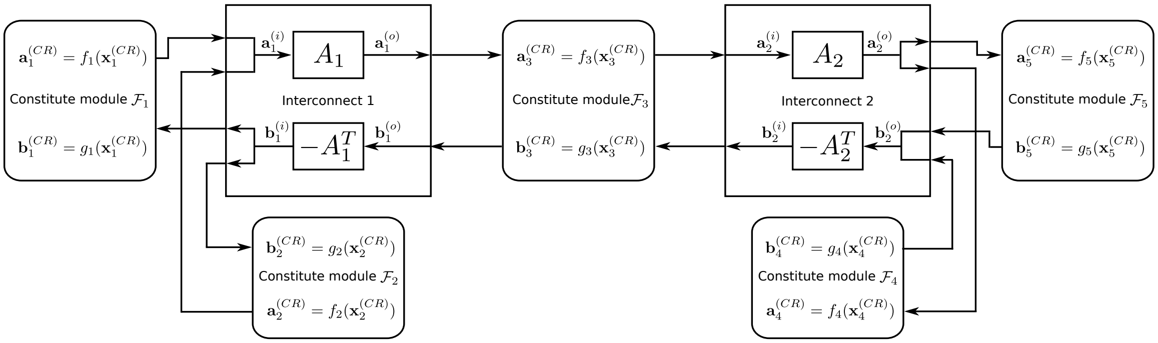

$\mathbf{a}_1^{(CR)} = f_1(\mathbf{x}_1^{(CR)})$
Constitute module $\mathcal{F}_1$
$\mathbf{b}_1^{(CR)} = g_1(\mathbf{x}_1^{(CR)})$
$\mathbf{a}_1^{(i)}$
A_1
$\mathbf{a}_1^{(o)}$
Interconnect 1
$\mathbf{b}_1^{(i)}$
$-A_1^T$
$\mathbf{b}_1^{(o)}$
$\mathbf{a}_3^{(CR)} = f_3(\mathbf{x}_3^{(CR)})$
Constitute module$\mathcal{F}_3$
$\mathbf{b}_3^{(CR)} = g_3(\mathbf{x}_3^{(CR)})$
$\mathbf{a}_2^{(i)}$
A_2
$\mathbf{a}_2^{(o)}$
Interconnect 2
$\mathbf{b}_2^{(i)}$
$-A_2^T$
$\mathbf{b}_2^{(o)}$
$\mathbf{a}_5^{(CR)} = f_5(\mathbf{x}_5^{(CR)})$
Constitute module $\mathcal{F}_5$
$\mathbf{b}_5^{(CR)} = g_5(\mathbf{x}_5^{(CR)})$
$\mathbf{b}_2^{(CR)} = g_2(\mathbf{x}_2^{(CR)})$
Constitute module $\mathcal{F}_2$
$\mathbf{a}_2^{(CR)} = f_2(\mathbf{x}_2^{(CR)})$
$\mathbf{b}_4^{(CR)} = g_4(\mathbf{x}_4^{(CR)})$
Constitute module $\mathcal{F}_4$
$\mathbf{a}_4^{(CR)} = f_4(\mathbf{x}_4^{(CR)})$

Figure 5-2: The canonical-form structure in Figure 5-1(b) illustrating the notation used to address individual features of the stationarity conditions for canonical-form primal and dual conservative optimization problems.

lems, we shall refer to the subvectors of $\mathbf{x}$ in (5.16) as well as subvectors of the dual decision vector $\mathbf{b}$ of the form

$$(\mathbf{b}_1, \dots \mathbf{b}_N) = (\mathbf{b}_1^{(CR)}, \dots, \mathbf{b}_K^{(CR)}) \tag{5.21}$$

$$= (\mathbf{b}_1^{(LI)}, \dots, \mathbf{b}_L^{(LI)}) \tag{5.22}$$

where each of the subvectors $\mathbf{x}_k^{(CR)}$ and $\mathbf{b}_k^{(CR)}$, for $k = 1, \dots, K$, and $\mathbf{b}_l^{(LI)}$, for $l = 1, \dots, L$, additionally satisfies

$$\mathbf{b}_k^{(CR)} = g_k(\mathbf{x}_k^{(CR)}), \qquad k = 1, \dots, K \tag{5.23}$$

$$\mathbf{b}_l^{(LI)} = (\mathbf{b}_l^{(i)}, \mathbf{b}_l^{(o)}), \qquad l = 1, \dots, L \tag{5.24}$$

and where the functions $g_k \colon \mathbb{R}^{N_k^{(CR)}} \to \mathbb{R}^{N_k^{(CR)}}$, for $k = 1, \dots, K$, map the shared decision subvector $\mathbf{x}_k^{(CR)}$ to the dual decision subvector $\mathbf{b}_k^{(CR)}$.

An example interconnective structure illustrating the notational conventions presented above is depicted in Figure 5-2 with $K = 5$ constitutive modules and $L = 2$ interconnects. This particular structure can equivalently be described using the structure in Figure 5-1(b) where the functions f and g are constructed by stacking the individual functions f_k and g_k according to $f = (f_1, \dots, f_5)$ and $g = (g_1, \dots, g_5)$.

To assist with addressing a term of the cost function associated with each constitutive module, the primal objective function will be separated into the sum of K functionals

$Q_k\colon \mathbb{R}^{N_k^{(CR)}} \to \mathbb{R}$, for $k = 1, \ldots, K$, and, similarly, the dual objective function will be separated into the sum of K functionals $R_k\colon \mathbb{R}^{N_k^{(CR)}} \to \mathbb{R}$, for $k = 1, \ldots, K$. Specifically, the following conditions are required to hold between the functions f_k and g_k in (5.19) and (5.23) and the functionals Q_k and R_k for optimization problems written in canonical form:

$$\begin{aligned} \nabla Q_k(\mathbf{x}_k^{(CR)}) &= J_{f_k}^T(\mathbf{x}_k^{(CR)}) g_k(\mathbf{x}_k^{(CR)}), \qquad k = 1, \ldots, K \qquad (5.25) \\ \nabla R_k(\mathbf{x}_k^{(CR)}) &= J_{g_k}^T(\mathbf{x}_k^{(CR)}) f_k(\mathbf{x}_k^{(CR)}), \qquad k = 1, \ldots, K \qquad (5.26) \\ Q_k(\mathbf{x}_k^{(CR)}) + R_k(\mathbf{x}_k^{(CR)}) &= \langle f_k(\mathbf{x}_k^{(CR)}), g_k(\mathbf{x}_k^{(CR)}) \rangle, \qquad k = 1, \ldots, K \qquad (5.27) \end{aligned}$$

where the size of the inner products in (5.27) is implied by the length of its arguments. To avoid being pedantic in the remainder of this chapter, we do not state these conditions explicitly but assume they hold and freely reference them in the course of proving various properties whenever the notation f_k, g_k, Q_k, and R_k is used.

5.1.3 | Canonical-form problems

Consistent with the strategy presented at the beginning of this section, the canonical-form description of primal and dual conservative optimization problems is defined in this subsection as a special case of the problems (5.1)-(5.2) and (5.3)-(5.4). Once these problem definitions have been established, the notational conventions outlined in the previous subsection are used to justify that the joint primal and dual feasibility conditions are in fact sufficient conditions for stationarity of the associated cost functions with respect to all feasible perturbations of the decision variables. With the claim of stationarity justified, the issue of a duality gap is discussed and we find that no duality gap exists between the canonical-form primal and dual cost functions for any solution $\mathbf{x}^\star$ to the joint feasibility conditions.

Before proceeding onwards, we note that convexity of the canonical-form cost functions and feasible sets is not explicitly required by the analysis used to establish the stationarity and strong duality results in this subsection. As will be discussed in Section 5.1.5, the strong duality principle presented in this subsection is consistent with well-established duality principles for convex problems, and connections to this follow from the relationship between canonical-form and reduced-form problem descriptions. The role of convexity in the context of canonical-form problems is explored in detail in Section 5.2.3.

Drawing on the notational conventions and conditions discussed in the previous subsection, we are prepared to define the canonical-form description of a conservative optimization problem. Specifically, a primal conservative optimization problem is formally described in canonical form according to

$$\begin{aligned}
\underset{\substack{(\mathbf{x}_1,\ldots,\mathbf{x}_N)\\(\mathbf{a}_1,\ldots,\mathbf{a}_N)}}{\text{minimize}} \quad & \sum_{k=1}^{K} Q_k(\mathbf{x}_k^{(CR)}) && && (5.28)\\
\text{s.t.} \quad & \mathbf{a}_k^{(CR)} = f_k(\mathbf{x}_k^{(CR)}), && k = 1,\ldots,K && (5.29)\\
& A_l\mathbf{a}_l^{(i)} = \mathbf{a}_l^{(o)}, && l = 1,\ldots,L && (5.30)
\end{aligned}$$

where $A_l\colon \mathbb{R}^{N_l^{(i)}} \to \mathbb{R}^{N_l^{(o)}}$, for $l = 1,\ldots L$, denotes a matrix for each l. For primal conservative optimization problems written in canonical form, we associate a conservative dual optimization problem with it. The dual optimization problem is formally described in canonical form according to

$$\begin{aligned}
\underset{\substack{(\mathbf{x}_1,\ldots,\mathbf{x}_N)\\(\mathbf{b}_1,\ldots,\mathbf{b}_N)}}{\text{maximize}} \quad & -\sum_{k=1}^{K} R_k(\mathbf{x}_k^{(CR)}) && && (5.31)\\
\text{s.t.} \quad & \mathbf{b}_k^{(CR)} = g_k(\mathbf{x}_k^{(CR)}), && k = 1,\ldots,K && (5.32)\\
& \mathbf{b}_l^{(i)} = -A_l^T\mathbf{b}_l^{(o)}, && l = 1,\ldots,L && (5.33)
\end{aligned}$$

where the matrices A_l, for $l = 1,\ldots,L$, are the same as the matrices in (5.30).

Equipped with the canonical-form problem definitions (5.28) through (5.30) and (5.31) through (5.33), the joint feasibility conditions (5.29) through (5.30) and (5.32) through (5.33) are sufficient conditions for stationarity of the objective functionals (5.28) and (5.31) with respect to small perturbations of the vector of shared decision variables $\mathbf{x}$ for which the vectors $\mathbf{a} = f(\mathbf{x})$ and $\mathbf{b} = g(\mathbf{x})$ remain feasible. The remainder of this section is spent proving this claim since the design of scattering algorithms to solve the joint feasibility conditions associated with a pair of primal and dual conservative optimization problems is motivated primarily by this fact. The outline of the proof is as follows: first, we define two vector spaces through the respective linear constraints (5.30) and (5.33) and show that they are orthogonal subspaces of $\mathbb{R}^N$. Next, using the orthogonality of these subspaces, we prove that the conditions (5.25) through (5.27) give rise to a stationarity property using (5.28) through (5.29) and (5.31) through (5.32) from the canonical-form problem definitions.

Let $\mathcal{V}_A$ denote the vector subspace of $\mathbb{R}^N$ that consists of all configurations of the primal decision vector $\mathbf{a}$ that are consistent with the linear constraints in (5.30). Using the variable ordering described in (5.20), the vector space $\mathcal{V}_A$ is written as

$$\mathcal{V}_A \quad = \quad \text{range}\left(\begin{bmatrix} I_{N_1^{(i)}} & \cdots & 0 \\ A_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & I_{N_L^{(i)}} \\ 0 & \cdots & A_L \end{bmatrix}\right). \tag{5.34}$$

Similarly, let $\mathcal{V}_B$ denote the vector subspace of $\mathbb{R}^N$ that consists of all configurations of the dual decision vector $\mathbf{b}$ that are consistent with the linear constraints in (5.33). Using the variable ordering described in (5.24), the vector space $\mathcal{V}_B$ is written as

$$\mathcal{V}_B \quad = \quad \text{range}\left(\begin{bmatrix} -A_1^T & \cdots & 0 \\ I_{N_1^{(o)}} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & -A_L^T \\ 0 & \cdots & I_{N_L^{(o)}} \end{bmatrix}\right). \tag{5.35}$$

By inspection of these vector space descriptions and direct application of the rank-nullity theorem [34, Theorem 3.4], we obtain that the sum of dimensionalities of the vector spaces in (5.34) and (5.35) satisfy

$$N \quad = \quad \underbrace{N_1^{(i)} + \cdots + N_L^{(i)}}_{\dim(\mathcal{V}_A)} + \underbrace{N_1^{(o)} + \cdots + N_L^{(o)}}_{\dim(\mathcal{V}_B)} \tag{5.36}$$

independent of the values taken by the matrices A_l. Therefore, as will become important to our analysis later, the vector spaces $\mathcal{V}_A$ and $\mathcal{V}_B$ uniquely decompose $\mathbb{R}^N$ according to $\mathbb{R}^N = \mathcal{V}_A \oplus \mathcal{V}_B$.

We next verify that the skew-symmetric relationships between the matrices A_l and $-A_l^T$, for $l = 1, \dots, L$, appearing in the primal and dual linear feasibility constraints (5.30) and (5.33) is sufficient to imply that the vector spaces $\mathcal{V}_A$ and $\mathcal{V}_B$ form an orthogonal direct sum decomposition of $\mathbb{R}^N$. To do this, let $\mathbf{a} \in \mathcal{V}_A$ and $\mathbf{b} \in \ V_B$. Substituting this into the pairwise orthogonality condition (5.5) and utilizing the interconnect subvector notation yields

$$\begin{aligned}
\langle \mathbf{a}, \mathbf{b} \rangle &= \sum_{l=1}^{L} \langle \mathbf{a}_l^{(LI)}, \mathbf{b}_l^{(LI)} \rangle && (5.37)\\
&= \sum_{l=1}^{L} \left\langle \begin{bmatrix} \mathbf{a}_l^{(i)} \\ \mathbf{a}_l^{(o)} \end{bmatrix}, \begin{bmatrix} \mathbf{b}_l^{(i)} \\ \mathbf{b}_l^{(o)} \end{bmatrix} \right\rangle && (5.38)\\
&= \sum_{l=1}^{L} \left(\langle \mathbf{a}_l^{(i)}, \mathbf{b}_l^{(i)} \rangle + \langle \mathbf{a}_l^{(o)}, \mathbf{b}_l^{(o)} \rangle \right) && (5.39)\\
&= \sum_{l=1}^{L} \left(\langle \mathbf{a}_l^{(i)}, -A_l^T \mathbf{b}_l^{(o)} \rangle + \langle A_l \mathbf{a}_l^{(i)}, \mathbf{b}_l^{(o)} \rangle \right) && (5.40)\\
&= 0 && (5.41)
\end{aligned}$$

as desired. Note that the final equality above follows from application of the definition of an adjoint, thereby meaning the vector spaces $\mathcal{V}_A$ and $\mathcal{V}_B$ will always be orthogonal to one another independent of the choice of matrices A_l, for $l = 1, \dots, L$.

With orthogonality between the vector subspaces $\mathcal{V}_A$ and $\mathcal{V}_B$ established, we are prepared to prove the stationarity principle for conservative optimization problems written in canonical form. Specifically, we first show that for any element $\mathbf{x}^\star \in \mathbb{R}^N$ for which $\mathbf{a} = f(\mathbf{x}^\star) \in \mathcal{V}_A$ and $\mathbf{b} = g(\mathbf{x}^\star) \in \mathcal{V}_B$, the primal objective functional (5.28) is stationary about any small perturbation in a direction for which (5.30) remains satisfied. To see this, let $\mathbf{u} \in \mathbb{R}^N$ denote any direction such that $D_{\mathbf{u}} f(\mathbf{x}^\star) = J_f(\mathbf{x}^\star)\mathbf{u} \in \mathcal{V}_A$. Then, at the point $\mathbf{x}^\star$ the primal objective function $Q\colon \mathbb{R}^N \to \mathbb{R}$ can be written succinctly as

$$Q(\mathbf{x}) \quad = \quad \sum_{k=1}^{K} Q_k(\mathbf{x}_k^{(CR)}) \tag{5.42}$$

and satisfies the stationarity condition

$$\begin{aligned}
D_{\mathbf{u}}Q(\mathbf{x}^\star) \quad &= \quad (\nabla Q_1(\mathbf{x}_1^{\star(CR)}), \ldots, \nabla Q_K(\mathbf{x}_K^{\star(CR)}))^T \mathbf{u} && (5.43)\\
&= \quad \sum_{k=1}^{K} \langle \nabla Q_k(\mathbf{x}_k^{\star(CR)}), \mathbf{u}_k^{(CR)} \rangle && (5.44)\\
&= \quad \sum_{k=1}^{K} \langle J_{f_k}^T(\mathbf{x}_k^{\star(CR)}) g_k(\mathbf{x}_k^{\star(CR)}), \mathbf{u}_k^{(CR)} \rangle && (5.45)\\
&= \quad \sum_{k=1}^{K} \langle g_k(\mathbf{x}_k^{\star(CR)}), J_{f_k}(\mathbf{x}_k^{\star(CR)}) \mathbf{u}_k^{(CR)} \rangle && (5.46)\\
&= \quad \langle \underbrace{g(\mathbf{x}^\star)}_{\in \mathcal{V}_B}, \underbrace{J_f(\mathbf{x}^\star)\mathbf{u}}_{\in \mathcal{V}_A} \rangle && (5.47)\\
&= \quad 0 && (5.48)
\end{aligned}$$

as desired, where the third equality follows immediately from from substituting the relationships described in (5.25) and the final equality follows from the fact that $\mathcal{V}_A$ and $\mathcal{V}_B$ are orthogonal subspaces.

The canonical-form description of conservative primal and dual optimization problems, and the relationship of this description to the stationarity conditions in particular, possess many symmetries that can be exploited for various reasons. In fact, the roles played by f and g and by Q and R can be swapped without breaking any of the relationships required to hold since asymmetrical conditions are not imposed between these objects. Using this symmetry, the analysis above also proves that the dual objective function will be stationarity to perturbations for which the dual solution remains feasible. For completeness, we present this argument in its entirety next.

Continuing on, we show that for the same element $\mathbf{x}^\star \in \mathbb{R}^N$ for which the stationarity of the primal cost functional Q was shown above, i.e. for which $\mathbf{a} = f(\mathbf{x}^\star) \in \mathcal{V}_A$ and $\mathbf{b} = g(\mathbf{x}^\star) \in \mathcal{V}_B$, the dual objective functional (5.31) is also stationary about any small perturbation in a direction for which (5.33) remains satisfied. To see this, let $\mathbf{w} \in \mathbb{R}^N$

denote any direction such that $D_{\mathbf{w}}g(\mathbf{x}^\star) = J_g(\mathbf{x}^\star)\mathbf{w} \in \mathcal{V}_B$. Then, at the point $\mathbf{x}^\star$ the dual objective function $R\colon \mathbb{R}^N \to \mathbb{R}$ can be written succinctly as

$$R(\mathbf{x}) \;=\; \sum_{k=1}^{K} R_k(\mathbf{x}_k^{(CR)}) \tag{5.49}$$

and satisfies the stationarity condition

$$\begin{aligned}
D_{\mathbf{w}}R(\mathbf{x}^\star) &= (\nabla R_1(\mathbf{x}_1^{\star(CR)}), \ldots, \nabla R_K(\mathbf{x}_K^{\star(CR)}))^T\mathbf{w} && (5.50)\\
&= \sum_{k=1}^{K}\langle \nabla R_k(\mathbf{x}_k^{\star(CR)}), \mathbf{w}_k^{(CR)}\rangle && (5.51)\\
&= \sum_{k=1}^{K}\langle J_{g_k}^T(\mathbf{x}_k^{\star(CR)}) f_k(\mathbf{x}_k^{\star(CR)}), \mathbf{w}_k^{(CR)}\rangle && (5.52)\\
&= \sum_{k=1}^{K}\langle f_k(\mathbf{x}_k^{\star(CR)}), J_{g_k}(\mathbf{x}_k^{\star(CR)})\mathbf{w}_k^{(CR)}\rangle && (5.53)\\
&= \langle \underbrace{f(\mathbf{x}^\star)}_{\in\mathcal{V}_A}, \underbrace{J_g(\mathbf{x}^\star)\mathbf{w}}_{\in\mathcal{V}_B}\rangle && (5.54)\\
&= 0 && (5.55)
\end{aligned}$$

as desired, where the third equality follows immediately from from substituting the relationships in (5.26) and the final equality follows from the fact that $\mathcal{V}_A$ and $\mathcal{V}_B$ are orthogonal subspaces.

To summarize the stationarity result implied by the the primal and dual feasibility conditions thus far, the analysis in (5.43) through (5.48) and (5.50) through (5.55) justifies the claim that any solution $\mathbf{x}^\star$ that produces a primal decision vector $\mathbf{a}$ and dual decision vector $\mathbf{b}$ that satisfy the primal and dual feasibility conditions is itself a stationary point of both the primal and dual objective functionals, where stationarity is specifically meant in all directions that maintain feasibility, i.e. for which the linear constraints (5.30) and (5.33) remain satisfied. Subsequently, the collective feasibility constraints described by (5.29) through (5.30) and (5.32) through (5.33) will be referred to in the remainder of the thesis as the *joint feasibility* or *stationarity conditions*. Consider the interconnective structure in Figure 5-1(b). Any elements $(\mathbf{a}^\star, \mathbf{b}^\star)$ consistent with the stationarity conditions depicted by the structure can be associated with a solution to primal and dual optimization problems with cost functions consistent with the functions f and g.

Before concluding this subsection, we address the issue of a duality gap between the primal and dual objective functionals (5.42) and (5.49). To do this, consider any element $\mathbf{x}^\star \in \mathbb{R}^N$ for which $\mathbf{a} = f(\mathbf{x}^\star) \in \mathcal{V}_A$ and $\mathbf{b} = g(\mathbf{x}^\star) \in \mathcal{V}_B$, i.e. $\mathbf{x}^\star$ denotes a solution to the stationarity conditions. Then, from the relationships in (5.27), the sum of the canonical primal and dual objectives at any such solution satisfies

$$\begin{aligned} Q(\mathbf{x}^\star) + R(\mathbf{x}^\star) &= \sum_{k=1}^{K} \Big(Q_k(\mathbf{x}_k^{\star(CR)}) + R_k(\mathbf{x}_k^{\star(CR)}) \Big) && (5.56)\\ &= \sum_{k=1}^{K} \langle f_k(\mathbf{x}_k^{\star(CR)}), g_k(\mathbf{x}_k^{\star(CR)}) \rangle && (5.57)\\ &= \langle \underbrace{f(\mathbf{x}^\star)}_{\in \mathcal{V}_A}, \underbrace{g(\mathbf{x}^\star)}_{\in \mathcal{V}_B} \rangle && (5.58)\\ &= 0 && (5.59) \end{aligned}$$

where we again have used the fact that $\mathcal{V}_A$ and $\mathcal{V}_B$ are orthogonal subspaces. In conclusion, we have shown that for any feasible solution there is no duality gap between the primal and dual objective functionals since $Q(\mathbf{x}^\star) = -R(\mathbf{x}^\star)$. Moreover, from the symmetry in the relationships (5.25) through (5.27) and the skew-symmetry in the linear equality constraints (5.30) and (5.33), it is straightforward to verify that generating the dual problem associated with the dual problem associated with an original primal problem returns the original primal problem, i.e. the dual of the dual is the primal.

5.1.4 | Reduced-form problems

In this subsection, the reduced-form description of a particular subset of conservative optimization problems written in canonical form is defined. The goal in establishing a reduced-form description is to be able to make connections with common classes of optimization problems where the decision variables in the primal and dual problems are different. The basic idea in defining the reduced-form description is to simplify a canonical-form description by selecting the functions f and g and the functionals Q and R so that the explicit dependence on the shared decision vector $\mathbf{x}$ can be replaced by a direct dependence of the primal problem on the primal decision vector $\mathbf{a}$ and a direct dependence of the dual problem on the dual decision vector $\mathbf{b}$. When reduced-form problem descriptions exist, they allow straightforward connections to be made between conservative duality as in Theorem 5.1.1 and Lagrangian duality, as will be discussed in Section 5.1.5.

For primal optimization problems written in canonical form, the existence of a reduced-form description depends on whether the parametric representation of the behavior of a set of K functionals $\widehat{Q}_k\colon \mathbb{R}^{N_k^{(CR)}} \to \mathbb{R}$, for $k = 1, \ldots, K$, exists such that the set of all elements generated by concatenating $Q_k(\mathbf{x}_k^{(CR)})$ and $f_k(\mathbf{x}_k^{(CR)})$ into a vector for each value of $\mathbf{x}_k^{(CR)} \in \mathbb{R}^{N_k^{(CR)}}$ is equal to the behavior of $\widehat{Q}_k$ as a function of the primal decision subvector $\mathbf{a}_k^{(CR)} = f_k(\mathbf{x}_k^{(CR)})$ for each value of k. Stated more formally, the k-th set-valued relationship required for a reduced-form primal description to exist is

$$\left\{ \begin{bmatrix} Q_k(\mathbf{x}_k^{(CR)}) \\ f_k(\mathbf{x}_k^{(CR)}) \end{bmatrix} \in \mathbb{R}^{N_k^{(CR)}+1} \colon \mathbf{x}_k^{(CR)} \in \mathbb{R}^{N_k^{(CR)}} \right\}$$
$$= \left\{ \begin{bmatrix} \widehat{Q}_k(\mathbf{a}_k^{(CR)}) \\ \mathbf{a}_k^{(CR)} \end{bmatrix} \in \mathbb{R}^{N_k^{(CR)}+1} \colon \mathbf{a}_k^{(CR)} \in \mathcal{A}_k \right\} \tag{5.60}$$

for an appropriately defined domain $\mathcal{A}_k \subseteq \mathbb{R}^{N_k^{(CR)}}$. When the condition in (5.60) holds for each k, $k = 1, \ldots, K$, then the reduced-form description of the associated canonical-form primal problem is written according to

$$\underset{(\mathbf{a}_1, \ldots, \mathbf{a}_N)}{\text{minimize}} \quad \sum_{k=1}^{K} \widehat{Q}_k(\mathbf{a}_k^{(CR)}) \tag{5.61}$$
$$\text{s.t.} \quad \mathbf{a}_k^{(CR)} \in \mathcal{A}_k, \qquad k = 1, \ldots, K \tag{5.62}$$
$$A_l \mathbf{a}_l^{(i)} = \mathbf{a}_l^{(o)}, \qquad l = 1, \ldots, L \tag{5.63}$$

where the matrices A_l, for $l = 1, \ldots, L$, are the same as the matrices appearing in the canonical-form primal problem.

Similarly, for dual optimization problems written in canonical form, the existence of a reduced-form description depends on whether the parametric representation of the behavior of a set of K functionals $\widehat{R}_k\colon \mathbb{R}^{N_k^{(CR)}} \to \mathbb{R}$, for $k = 1, \ldots, K$, exists such that the set of all elements generated by concatenating $R_k(\mathbf{x}_k^{(CR)})$ and $g_k(\mathbf{x}_k^{(CR)})$ into a vector for each value of $\mathbf{x}_k^{(CR)} \in \mathbb{R}^{N_k^{(CR)}}$ is equal to the behavior of $\widehat{R}_k$ as a function of the dual decision subvector $\mathbf{b}_k^{(CR)} = g_k(\mathbf{x}_k^{(CR)})$ for each value of k. Stated more formally, the k-th set-valued relationship required for a reduced-form dual description to exist is

$$\begin{aligned}\left\{ \begin{bmatrix} R_k(\mathbf{x}_k^{(CR)}) \\ g_k(\mathbf{x}_k^{(CR)}) \end{bmatrix} \in \mathbb{R}^{N_k^{(CR)}+1} \colon \mathbf{x}_k^{(CR)} \in \mathbb{R}^{N_k^{(CR)}} \right\} \\ = \left\{ \begin{bmatrix} \widehat{R}_k(\mathbf{b}_k^{(CR)}) \\ \mathbf{b}_k^{(CR)} \end{bmatrix} \in \mathbb{R}^{N_k^{(CR)}+1} \colon \mathbf{b}_k^{(CR)} \in \mathcal{B}_k \right\} \qquad (5.64)\end{aligned}$$

for an appropriately defined domain $\mathcal{B}_k \subseteq \mathbb{R}^{N_k^{(CR)}}$. When the condition in (5.64) holds for each k, $k = 1, \ldots, K$, then the reduced-form description of the associated canonical-form dual problem is written according to

$$\begin{aligned} \underset{(\mathbf{b}_1,\ldots,\mathbf{b}_N)}{\text{maximize}} \quad & -\sum_{k=1}^{K} \widehat{R}_k(\mathbf{b}_k^{(CR)}) && && (5.65) \\ \text{s.t.} \quad & \mathbf{b}_k^{(CR)} \in \mathcal{B}_k, && k = 1, \ldots, K && (5.66) \\ & \mathbf{b}_l^{(i)} = -A_l^T \mathbf{b}_l^{(o)}, && l = 1, \ldots, L && (5.67) \end{aligned}$$

where the matrices A_l, for $l = 1, \ldots, L$, are the same as the matrices appearing in the canonical dual problem. When they exist, the sum of the reduced-form mapping objects $\widehat{Q}_k$ and $\widehat{R}_k$ are respectively referred to as the reduced-form primal and dual cost functions.

Before concluding this subsection, we make two important remarks. First, in contrast with the conditions in (5.25) through (5.26) describing the relationships between canonical-form mapping objects, the conditions (5.60) and (5.64) do not require the reduced-form cost functions to be differentiable with respect to the primal and dual decision variables. This observation will play an important role in designing scattering algorithms to solve optimization problems with non-smooth cost functions. Second, the requirements in the set-valued relationships (5.60) and (5.64) are symmetric in many ways. Despite these symmetries, the existence of a reduced-form description of a primal problem is not necessary or sufficient for the canonical-form dual problem to have a reduced-form description, nor does the canonical-form dual problem having a reduced-form description imply whether or not the canonical-form problem does. In the interest of making connections and solving common classes of optimization problems appearing in the signal processing community, the focus in the remainder of this chapter is on canonical-form primal problems that have reduced-form descriptions regardless of whether or not the associated conservative dual problems do.

5.1.5 | Reduced-form interpretation of Theorem 5.1.1

For problems with reduced-form descriptions, the stationarity principle in Theorem 5.1.1 can often be interpreted directly in terms of the components of a reduced-form problem description, rather than requiring interpretation through a combination of the problem's canonical-form description and the stationarity conditions relating the mapping objects f, g, Q, and R. The purpose of this subsection is to present this interpretation, thereby allowing straightforward connections to other well-known optimality conditions. Although many of the coming arguments can be extended to handle non-differentiable reduced-form cost functions, we proceed assuming these cost functions are differentiable with respect to their arguments to avoid technical digressions. For example, straightforward use of subdifferential operators can be used to extend the results in this subsection for general convex reduced-form problems since $\partial\widehat{Q}$ and $\partial\widehat{R}$ are well-defined. In addition, for the purpose of clarity, we proceed without the subvector notation since the primary focus in this subsection is to interpret the mapping objects rather than interpret their arguments.

Recall that reduced-form primal problems satisfy the set constraint in (5.60), meaning the canonical-form cost function $Q(\mathbf{x})$ is equal to $\widehat{Q}(f(\mathbf{x}))$ for every value of $\mathbf{x}$. Therefore, applying the same linear operator $\nabla_{\mathbf{x}}$ to each expression gives the following relationships

$$\begin{aligned}\nabla_{\mathbf{x}}Q(\mathbf{x}) &= \nabla_{\mathbf{x}}\widehat{Q}(f(\mathbf{x})) &(5.68)\\ &= J_f(\mathbf{x})^T\nabla_f\widehat{Q}(f(\mathbf{x})) &(5.69)\\ &= J_f(\mathbf{x})^T g(\mathbf{x}) &(5.70)\end{aligned}$$

where the final equality follows from substitution of the relationship in (5.25). Assuming further that the Jacobian of f with respect to $\mathbf{x}$ is non-singular, we have that this relationship reduces to the expression

$$\nabla_f \widehat{Q}(f(\mathbf{x})) = g(\mathbf{x}) \tag{5.71}$$

or equivalently that the dual variables are generated according to

$$\mathbf{b} \quad = \quad \nabla_{\mathbf{a}} \widehat{Q}(\mathbf{a}). \tag{5.72}$$

This result essentially says the corresponding constitutive module has an $(\mathbf{a}, \mathbf{b})$ relationship where $\mathbf{b}$ is the (sub)gradient of the reduced-form primal objective function, and suggests how to design constitutive modules to solve a particular optimization problem.

Using symmetry between the constraints (5.60) and (5.64) and symmetry in the relationships (5.25) and (5.26), the same analysis holds for dual conservative optimization problems with reduced-form descriptions. Therefore, the resulting relationship is that the recued-form cost function $\widehat{R}$ is related to the functions f and g according to

$$\nabla_g \widehat{R}(g(\mathbf{x})) = f(\mathbf{x}) \tag{5.73}$$

or equivalently that the primal variables are generated according to

$$\mathbf{a} \quad = \quad \nabla_b \widehat{R}(\mathbf{b}). \tag{5.74}$$

As we will see in Section 5.3, a variety of constitutive modules used to describe differentiable and non-differentiable, convex reduced-form primal cost functions can be generated by setting the dual decision variables $\mathbf{b}$ equal to the subderivative of the reduced-form primal cost function $\widehat{Q}$. Formalizing this fact requires additional assumptions to be made in order to apply an appropriate notion of a chain rule [96]. As an alternative to this approach, we proceed in this thesis to work with the parametric version of the stationarity conditions relying simply on straightforward differential relationships and variational calculations.

5.2 | Scattering-form description of the stationarity conditions

Motivated by the close relationship between feasibility and stationarity established in the previous section, we proceed in this section to focus on determining stationary points of conservative optimization problems by recasting the associated primal and dual feasibility conditions as a CCSP. The key connection making this possible is the observation that the quadratic form representing the conservation principle inherent to the stationarity conditions, which also underlies the orthogonality of the vector spaces $\mathcal{V}_A$ and $\mathcal{V}_B$ used to establish the stationarity principle, is isomorphic to the quadratic form representing the conservation principle associated with the vector space W used in the description of an interconnecting network for a CCSP in (4.12). Essentially the same observation was used in Section 3.4.4 to derive the scattering coordinate transform between canonical and orthogonal form interconnective structures with maximal partition decompositions.

To solve a prespecified pair of primal and dual conservative optimization problems written in either canonical or reduced form, the general strategy we use in this thesis is to:

(i) create a CCSP by performing a scattering coordinate transform to the canonical-form interconnective description of the joint feasibility conditions,

(ii) solve the corresponding CCSP using synchronous or asynchronous scattering algorithms, potentially informed by the stability and robustness conditions developed in Chapter 4, and

(iii) put the solution obtained through the inverse scattering transform to obtain a stationary point of both the original primal and dual optimization problems.

The mechanics associated with solving the CCSP using scattering algorithms in step (ii), in particular the connection between scattering-form structures and their associated CCSPs, has already been discussed in Chapter 4. In this section, the focus is on the mechanics associated with carrying out steps (i) and (iii). Specifically, we discuss the procedure used to transform the stationarity conditions into scattering-form, and then develop some algebraic reduction techniques that are able to reduce the associated system operator from being passive everywhere to dissipative everywhere in some cases. To facilitate assembling scattering-form structures directly from optimization problem statements, the general approach to designing the constitutive modules and interconnects is presented and an illustrative example of the strategy is provided. To conclude the section, the role of convexity in the present context is discussed where the focus is on the effect convexity has on the stability of scattering-form structures and on the existence of functional realizations of the modules and interconnects appearing in the structures.

5.2.1 | Transforming the stationarity conditions into a CCSP

To describe the stationarity conditions associated with a pair of primal and dual conservative optimization problems as a scattering-form interconnective structure, we first organize the conditions into a canonical-form structure and then apply the scattering coordinate transform derived in Section 3.4.4. Referring to the notation in (3.59) and (3.63), a canonical-form description of the joint feasibility conditions (5.29) through (5.30) and (5.32) through (5.33) is achieved by organizing the primal and dual decision vectors $\mathbf{a}$ and $\mathbf{b}$ according to the following variable assignments:

$$\mathbf{a}_1^{(c)} = (\mathbf{a}_1^{(i)}, \ldots, \mathbf{a}_L^{(i)}) \tag{5.75}$$

$$\mathbf{b}_1^{(c)} = (\mathbf{b}_1^{(i)}, \ldots, \mathbf{b}_L^{(i)}) \tag{5.76}$$

$$\mathbf{a}_2^{(c)} = (\mathbf{a}_1^{(o)}, \ldots, \mathbf{a}_L^{(o)}) \tag{5.77}$$

$$\mathbf{b}_2^{(c)} = (\mathbf{b}_1^{(o)}, \ldots, \mathbf{b}_L^{(o)}) \tag{5.78}$$

where the dimensions associated with the description of the stationarity conditions are related to the dimensions associated with the canonical-form interconnective structure according to $N = R^{(c)} + L^{(c)}$ where $R^{(c)}$ and $L^{(c)}$ are given by

$$R^{(c)} = N_1^{(i)} + \cdots + N_L^{(i)} \tag{5.79}$$

$$L^{(c)} = N_1^{(o)} + \cdots + N_L^{(o)}. \tag{5.80}$$

To compete the canonical-form description of the stationarity conditions, we need only to state the canonical interconnecting network $W^{(c)}$ and constitutive relation $\mathcal{F}^{(c)}$. With regard to the interconnecting network, the vector space is generated according to (3.63) where the $R^{(c)} \times L^{(c)}$-dimensional matrix A is given by

$$A = \begin{bmatrix} A_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & A_L \end{bmatrix} \tag{5.81}$$

where the submatrices A_l are the same as the matrices appearing in (5.30) and (5.33). The canonical constitutive relation $\mathcal{F}^{(c)}$ is formed by aggregating the individual constitutive relations $\mathcal{F}_k$ derived using the parametric functions f_k and g_k according to

$$\mathcal{F}^{(c)} = \mathcal{F}_1 \times \cdots \times \mathcal{F}_K \tag{5.82}$$

where the k-th constitutive module is described using the relation $\mathcal{F}_k$ and can be generated by sweeping the parametric variable $\mathbf{x}_k^{(CR)}$ as

$$\mathcal{F}_k = \left\{ \begin{bmatrix} f_k(\mathbf{x}_k^{(CR)}) \\ g_k(\mathbf{x}_k^{(CR)}) \end{bmatrix} \in \mathbb{R}^{2N_k^{(CR)}} : \mathbf{x}_k^{(CR)} \in \mathbb{R}^{N_k^{(CR)}} \right\}, \qquad k = 1, \ldots, K. \tag{5.83}$$

Recall that the primal and dual objective functionals Q_k and R_k do not appear in the stationarity conditions except implicitly through the conditions (5.25) through (5.27) and therefore do not explicitly appear in the canonical-form description of the feasibility conditions either.

It is a straightforward exercise to verify that the interconnective description of the stationarity conditions provided above does in fact satisfy the requirements of a canonical-form structure. For example, the fact that the canonical-form interconnecting network is conservative with regard to the canonical-form organization $\mathcal{O}^{(c)}$ in (3.65) was implicitly used to prove the stationarity principle for conservative optimization problems written in either canonical or reduced forms. In the notation of the canonical comparison space, the analysis in (5.37) through (5.41) essentially justifies the condition (3.70) where the vector subspaces $\mathcal{V}_A$ and $\mathcal{V}_B$ correspond to the conjugate subspaces and the conjugate maps $M_A^{(c)}$ and $M_B^{(c)}$ are the same as those in (3.68) and (3.69), respectively.

To provide the scattering-form description of the stationarity conditions, we next apply the scattering coordinate transform from Section 3.4.4 to the canonical-form description of the problem where we assume without loss of generality that the canonical coordinate matrix $M^{(c)}$ is the identity matrix. Indeed, the scattering-form description is immediately obtained by applying the variable transform in (3.88) and then reflecting that transformation to $W^{(c)}$ and $\mathcal{F}^{(c)}$. For the purpose of solving conservative optimization problems, we proceed to write this transformation in terms of the decision variables $\mathbf{a}$ and $\mathbf{b}$. Specifically, the transformed or scattering variables $\mathbf{c}$ and $\mathbf{d}$ are defined according to the 2×2 scattering transforms

$$\begin{bmatrix} \mathbf{c}_n \\ \mathbf{d}_n \end{bmatrix} = M_n \begin{bmatrix} \mathbf{a}_n \\ \mathbf{b}_n \end{bmatrix}, \qquad n = 1, \dots, N \tag{5.84}$$

where the matrix $M^{(i)}$ in (3.90) takes the place of M_n if $(\mathbf{a}_n, \mathbf{b}_n)$ appears in $(\mathbf{a}_1^{(c)}, \mathbf{b}_1^{(c)})$ and the matrix $M^{(o)}$ in (3.91) takes the place of M_n if $(\mathbf{a}_n, \mathbf{b}_n)$ appears in $(\mathbf{a}_2^{(c)}, \mathbf{b}_2^{(c)})$. Said another way, the matrix $M^{(i)}$ is used when the primal variable $\mathbf{a}_n$ correspond to input to the interconnecting network and the matrix $M^{(o)}$ is used otherwise. Consistent with the input-output configuration P in (3.85), when a realization of the scattering-form description of the stationarity conditions exists, we associate a solution $(\mathbf{c}^\star, \mathbf{d}^\star) \in \mathbb{R}^{2N}$ to the corresponding CCSP as a solution to the nonlinear system of equations

$$\mathbf{d}^\star = G\mathbf{c}^\star \tag{5.85}$$

$$\mathbf{c}^\star = m(\mathbf{d}^\star) \tag{5.86}$$

where the orthogonal matrix G and memoryless nonlinearity m are, in particular, the realizations of the scattering-form interconnecting network $W^{(s)}$ and constitutive module $\mathcal{F}^{(s)}$. In the sequel, we shall refer to the (5.85) and (5.86) as the *transformed stationarity conditions* and the variables $\mathbf{c}$ and $\mathbf{d}$ as the transformed or scattering variables. A closed-form expression for the matrix G, which is always guaranteed to exist, is provided shortly. We defer stating the conditions under which a realization of the form (5.85) and (5.86) exists to the end of this subsection.

In anticipation of designing and assembling constitutive modules to directly produce scattering-form interconnective structures from inspection of a conservative optimization problem statement, we proceed to define some additional coordinate transform matrices that directly transform the constitutive modules and interconnects used in the interconnective description of the untransformed stationarity conditions. In particular, for the interconnect subvectors $\mathbf{a}_l^{(LI)}$ and $\mathbf{b}_l^{(LI)}$, we define the l-th scattering transform matrix $M_l^{(LI)}\colon \mathbb{R}^{2N_l^{(LI)}} \to \mathbb{R}^{2N_l^{(LI)}}$ according to

$$\begin{bmatrix} \mathbf{c}_l^{(LI)} \\ \mathbf{d}_l^{(LI)} \end{bmatrix} = M_l^{(LI)} \begin{bmatrix} \mathbf{a}_l^{(LI)} \\ \mathbf{b}_l^{(LI)} \end{bmatrix}, \qquad l = 1, \ldots, L. \tag{5.87}$$

Similarly, for the constitutive module subvectors $\mathbf{a}_k^{(CR)}$ and $\mathbf{b}_k^{(CR)}$, we define the k-th scattering transform matrix $M_k^{(CR)}\colon \mathbb{R}^{2N_k^{(CR)}} \to \mathbb{R}^{2N_k^{(CR)}}$ according to

$$\begin{bmatrix} \mathbf{c}_k^{(CR)} \\ \mathbf{d}_k^{(CR)} \end{bmatrix} = M_k^{(CR)} \begin{bmatrix} \mathbf{a}_k^{(CR)} \\ \mathbf{b}_k^{(CR)} \end{bmatrix}, \qquad k = 1, \ldots, K. \tag{5.88}$$

The numerical entries in the scattering matrices $M_l^{(LI)}$ and $M_k^{(CR)}$ are completely determined by the entries of the N 2×2-dimensional matrices M_n in (5.84). Using this notation, an equivalent description of the transformed stationarity conditions (5.85) through (5.86) using the subvector partitioning schemes in (5.87) and (5.88) is given by

$$\mathbf{d}_l^{\star(LI)} = G_l \mathbf{c}_l^{\star(LI)}, \qquad l = 1, \ldots, L \tag{5.89}$$

$$\mathbf{c}_k^{\star(CR)} = m_k(\mathbf{d}_k^{\star(CR)}), \qquad k = 1, \ldots, K. \tag{5.90}$$

The equivalence between these two formulations of the transformed stationarity conditions follows from two observations. First, from the block-diagonal structure of A in (5.81), it follows that the transformed interconnect can take advantage of this block-diagonal structure in $W^{(c)}$. By combining the result in (3.94) with the subvector ordering in (5.87), it is straightforward to verify that

$$G_l = \left(I_{N_l^{(LI)}} + \begin{bmatrix} 0 & -A_l^T \\ A_l & 0 \end{bmatrix}\right)\left(I_{N_l^{(LI)}} + \begin{bmatrix} 0 & A_l^T \\ -A_l & 0 \end{bmatrix}\right)^{-1}, \quad l = 1, \ldots L. \tag{5.91}$$

Second, the direct product decomposition of the constitutive module $\mathcal{F}^{(c)}$ in (5.82) implies that the memoryless nonlinearity m is also decomposable into K mutually independent functions m_k.

A pertinent question at this point deals with understanding conditions under which the matrices G_l and memoryless nonlinearities m_k are well-defined, thereby certifying that a functional realization of the interconnective structure describing the transformed stationarity conditions exists. To address this, we state several conditions using the scattering matrices $M_l^{(LI)}$ and $M_k^{(CR)}$ defined above. Specifically, the orthogonal matrices G_l, for $l = 1, \ldots, L$, are well-defined as long as the following conditions hold:

$$\left\{ M_l^{(LI)} \begin{bmatrix} \mathbf{a}_l^{(i)} \\ A_l \mathbf{a}_l^{(i)} \\ -A_l^T \mathbf{b}_l^{(o)} \\ \mathbf{b}_l^{(o)} \end{bmatrix} \in \mathbb{R}^{2N_l^{(LI)}} : \begin{bmatrix} \mathbf{a}_l^{(i)} \\ \mathbf{b}_l^{(o)} \end{bmatrix} \in \mathbb{R}^{N_l^{(LI)}} \right\}$$
$$= \left\{ \begin{bmatrix} \mathbf{c}_l^{(LI)} \\ G_l \mathbf{c}_l^{(LI)} \end{bmatrix} \in \mathbb{R}^{2N_l^{(LI)}} : \mathbf{c}_l^{(LI)} \in \mathbb{R}^{N_l^{(LI)}} \right\}, \quad l = 1, \ldots, L. \tag{5.92}$$

As was previously mentioned, these conditions will always be met. Similarly, the memoryless nonlinearities m_k, for $k = 1, \ldots, K$, in (5.90) are well-defined as long as the following conditions hold:

$$\left\{ M_k^{(CR)} \begin{bmatrix} f_k(\mathbf{x}_k^{(CR)}) \\ g_k(\mathbf{x}_k^{(CR)}) \end{bmatrix} \in \mathbb{R}^{2N_k^{(CR)}} : \mathbf{x}_k^{(CR)} \in \mathbb{R}^{N_k^{(CR)}} \right\}$$
$$= \left\{ \begin{bmatrix} m_k(\mathbf{d}_k^{(CR)}) \\ \mathbf{d}_k^{(CR)} \end{bmatrix} \in \mathbb{R}^{2N_k^{(CR)}} : \mathbf{d}_k^{(CR)} \in \mathbb{R}^{N_k^{(CR)}} \right\}, \quad k = 1, \ldots, K. \tag{5.93}$$

Unfortunately, the existence of the functions m_k cannot always be guaranteed. Since the ability to solve the transformed stationarity conditions by implementing a scattering algorithm hinges solely on their existence, we solve for these functions corresponding to many objective functions and feasibility conditions that frequently appear in optimization problems of interest to the signal processing community in Section 5.3. To provide guidance in this task, we first discuss the role convexity of the objective function plays in generating these functions in Section 5.2.3.

When a functional realization of the scattering-form structure describing the transformed stationarity conditions exists, the methods for solving CCSPs developed in Chapter 4 can be used to solve them. Once a solution $(\mathbf{c}^\star, \mathbf{d}^\star)$ is identified, taking the inverse scattering coordinate transform provides a stationary point $\mathbf{a}^\star$ and $\mathbf{b}^\star$ of the primal and dual optimization problems, respectively. In fact, solving the CCSP described by (5.89) through (5.90) not only solves the original primal and dual optimization problems, but may provide a solution to any optimization problem corresponding to an interconnect invariant coordinate transform of the canonical-form description of the stationarity conditions.

5.2.2 | A modular approach to designing scattering-form structures

Consistent with the general strategy discussed at the beginning of this section, the goal of this subsection is to present a modular approach to designing scattering-form structures from inspection of an optimization problem statement, where the structure describes the transformed stationarity conditions associated with the problem. To facilitate doing this, the general form of constitutive modules and interconnects in canonical and scattering coordinates are defined by relating features of an optimization problem with functions used in the implementation of a scattering algorithm. By connecting these modules together, signal processing structures in scattering form can be assembled so that fixed-points of the structure correspond to solutions of the optimization problem to within a change of coordinates. Certain constitutive modules can be eliminated using straightforward algebra, resulting in scattering structures with fewer variables and whose corresponding system operator can be reduced from passive everywhere to dissipative everywhere in some cases. Solutions to the CCSP associated with the smaller structure are related to solutions to the CCSP associated

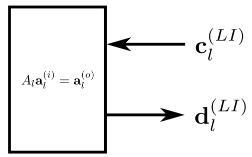
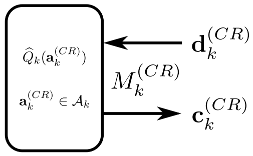
Graph symbol	Canonical behavior	Reduced-form primal components	Reduced-form dual components	Realization		α- connicity
 $A_l\mathbf{a}_l^{(i)} = \mathbf{a}_l^{(o)}$ $\mathbf{c}_l^{(LI)}$ $\mathbf{d}_l^{(LI)}$	$\begin{bmatrix}\mathbf{a}_l^{(i)}\\ \mathbf{a}_l^{(o)}\\ \mathbf{b}_l^{(i)}\\ \mathbf{b}_l^{(o)}\end{bmatrix} \in \text{range}\left(\begin{bmatrix} I_{N_l^{(i)}} & 0\\ A_l & 0\\ 0 & -A_l^T\\ 0 & I_{N_l^{(o)}}\end{bmatrix}\right)$	$A_l\mathbf{a}_l^{(i)} = \mathbf{a}_l^{(o)}$	$\mathbf{b}_l^{(i)} = -A^T\mathbf{b}_l^{(o)}$	$\mathbf{d}_l^{(LI)} = G_l\mathbf{c}_l^{(LI)}$ $G_l = \begin{bmatrix} I & -A_l^T\\ A_l & I\end{bmatrix}\begin{bmatrix} I & A_l^T\\ -A_l & I\end{bmatrix}^{-1}$		passive everywhere
 $\widehat{Q}_k(\mathbf{a}_k^{(CR)})$ $\mathbf{a}_k^{(CR)} \in \mathcal{A}_k$ $M_k^{(CR)}$ $\mathbf{d}_k^{(CR)}$ $\mathbf{c}_k^{(CR)}$	$\left\{\begin{bmatrix} f_k^{(CR)}(\mathbf{x}_k^{(CR)})\\ g_k^{(CR)}(\mathbf{x}_k^{(CR)})\end{bmatrix} : \mathbf{x}_k^{(CR)} \in \mathbb{R}^{N_k^{(CR)}}\right\}$	$\widehat{Q}_k(\mathbf{a}_k^{(CR)})$ $\mathbf{a}_k^{(CR)} \in \mathcal{A}_k$	$\widehat{R}_k(\mathbf{b}_k^{(CR)})$ $\mathbf{b}_k^{(CR)} \in \mathcal{B}_k$	$M^{(i)}$	$\mathbf{c}_k^{(CR)} = m_k^{(i)}(\mathbf{d}_k^{(CR)})$	case by case
				$M^{(o)}$	$\mathbf{c}_k^{(CR)} = m_k^{(o)}(\mathbf{d}_k^{(CR)})$	

Figure 5-3: An illustration of the two types of modules consistent with assembling an interconnective graphs to solve the transformed stationarity conditions as a CCSP. The top row depicts an interconnect module and the bottom row depicts a constitutive relation module.

$$\min_{(\mathbf{a}_1,\ldots,\mathbf{a}_N)} \sum_{k=1}^{K} \widehat{Q}_k(\mathbf{a}_k^{(CR)})$$

$$\text{s.t.}\quad A_l\mathbf{a}_l^{(i)} = \mathbf{a}_l^{(o)} \quad l = 1,\ldots,L$$

$$\mathbf{a}_k^{(CR)} \in \mathcal{A}_k \quad k = 1,\ldots,K$$

(a)

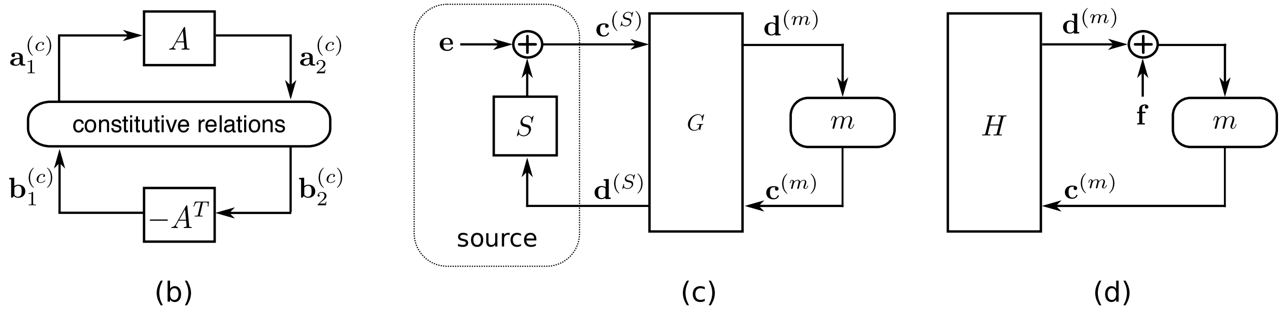


Figure 5-4: An illustration of (a) a reduced-form primal optimization problem, (b) the canonical-form structure representing the stationarity conditions, (c) the scattering-form structure representing the transformed stationarity conditions, and (d) a smaller scattering-form structure obtained by algebraically eliminating the source elements in (c).

with the original structure through a closed-form expression. Therefore, the smaller structures can solve conservative optimization problems as well. To conclude this subsection, an example illustrating the modular design of an interconnective structure corresponding to a classic regression problem in signal processing is provided. This example highlights the ease by which distributed, asynchronous scattering algorithms for solving optimization problems in the presented class can be assembled, and also illustrates the flexibility in moving between closely related optimization problems by simply changing a small number of the modules.

The first module we define corresponds to an interconnect and describes the linear constraints appearing in a primal optimization problem; a complete description of this module is provided in the top row of Figure 5-3. Interconnects defined in this way are implemented using the matrices G_l in (5.92) and operate on the scattering variables $\mathbf{c}_l^{(LI)}$ and $\mathbf{d}_l^{(LI)}$. The second module we define corresponds to a constitutive module and describes the cost function and set constraints appearing in a primal optimization problem; a complete description of this module is provided in the bottom row of Figure 5-3. The functional realizations of constitutive modules do not always exist for particular cost functions and constraint sets, but when they do they are implemented using the memoryless nonlinearities m_k in (5.93) and depend on the specific scattering transform used to produce the relationship between the transformed variables $\mathbf{c}_k^{(CR)}$ and $\mathbf{d}_k^{(CR)}$. Motivated by the importance of these modules existing, the primary focus of Section 5.3 is to derive example modules of this type.

The general procedure for connecting constitutive modules and interconnects together to form the scattering-form description of the transformed stationarity conditions associated with a conservative optimization problem is illustrated in Figure 5-4. In particular, the interconnective structure depicted in Figure 5-4(c) provides a complete description of the transformed stationarity conditions associated with the primal problem statement in (a) whose stationarity conditions are depicted as a canonical-form interconnective structure in (b). Referring to Figure 5-4(c), the individual interconnects are denoted in aggregate by the interconnect labeled G and the individual constitutive modules are denoted using a generic memoryless nonlinearity labeled $m(\cdot)$ and a second constitutive module labeled “source”. The distinction in assigning the constitutive modules to either the source element or the function $m(\cdot)$ can be made using the following definition.

Definition 5.2.1 (Source modules)**.** *The functional realization of a constitutive module denoted by* $m\colon \mathbb{R}^K \to \mathbb{R}^K$ *is said to be a* source *provided that it has the algebraic form*

$$m(\mathbf{v}) \quad = \quad S\mathbf{v} + \mathbf{e} \tag{5.94}$$

where $S\colon \mathbb{R}^K \to \mathbb{R}^K$ *is a linear, passive everywhere map and* $\mathbf{e} \in \mathbb{R}^K$ *is a constant.*

With the definition of a source module in place, we next show that the scattering form structure in Figure 5-4(c) can be reduced to the scattering-form structure in (d) by algebraically eliminating the source module from the structure. Indeed, interconnective structures after such transformations have been applied will be referred to as being *source-free.*

Let the scattering variables $\mathbf{c}$ and $\mathbf{d}$ depicted in the interconnective structure in Figure 5-4(c) be partitioned according to

$$\begin{bmatrix} \mathbf{d}^{(m)} \\ \mathbf{d}^{(S)} \end{bmatrix} \quad = \quad \underbrace{\begin{bmatrix} G_{mm} & G_{mS} \\ G_{Sm} & G_{SS} \end{bmatrix}}_{G} \begin{bmatrix} \mathbf{c}^{(m)} \\ \mathbf{c}^{(S)} \end{bmatrix}. \tag{5.95}$$

where the realization of the constitutive relation module corresponding to $\mathbf{c}^{(S)}$ and $\mathbf{d}^{(S)}$ is a source element of the form $\mathbf{c}^{(S)} = S\mathbf{d}^{(S)} + \mathbf{e}$. After some straightforward manipulations, we obtain that the elements of the source-free interconnective structure in Figure 5-4(d) can be written in terms of (5.95) as

$$\mathbf{d}^{(m)} \quad = \quad \underbrace{\left(G_{mm} + G_{mS}\left(I - SG_{SS}\right)^{-1} SG_{Sm}\right)}_{H} \mathbf{c}^{(m)} + \underbrace{G_{mS}\left(I - SG_{SS}\right)^{-1} \mathbf{e}}_{\mathbf{f}}. \tag{5.96}$$

The linear map denoted by H in (5.96) is passive everywhere. This point follows directly from substituting the constitutive relationship $\mathbf{c}^{(S)} = S\mathbf{d}^{(S)} + \mathbf{e}$ and (5.95) into the definition of α-connicity and utilizing the fact that the squared 2-norm is separable over its entries into the sum of squared 2-norms. As such, the source-free interconnective structure in Figure 5-4(d) is also a scattering-form structure and therefore has an associated CCSP.

Given a solution $(\mathbf{c}^{\star(m)}, \mathbf{d}^{\star(m)})$ to the CCSP associated with the source-free interconnective structure defined above, a solution $(\mathbf{c}^{\star}, \mathbf{d}^{\star})$ to the CCSP associated with the original

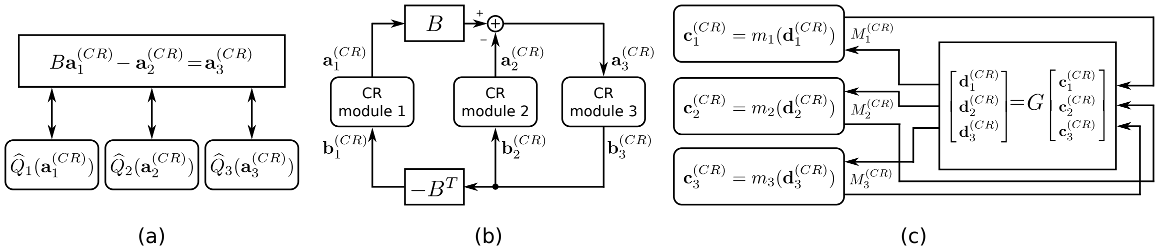

$B\mathbf{a}_1^{(CR)} - \mathbf{a}_2^{(CR)} = \mathbf{a}_3^{(CR)}$
$\widehat{Q}_1(\mathbf{a}_1^{(CR)})$
$\widehat{Q}_2(\mathbf{a}_2^{(CR)})$
$\widehat{Q}_3(\mathbf{a}_3^{(CR)})$
(a)
B
+
−
$\mathbf{a}_1^{(CR)}$
$\mathbf{a}_2^{(CR)}$
$\mathbf{a}_3^{(CR)}$
CR module 1
CR module 2
CR module 3
$\mathbf{b}_1^{(CR)}$
$\mathbf{b}_2^{(CR)}$
$\mathbf{b}_3^{(CR)}$
$-B^T$
(b)
$\mathbf{c}_1^{(CR)} = m_1(\mathbf{d}_1^{(CR)})$
$M_1^{(CR)}$
$\mathbf{c}_2^{(CR)} = m_2(\mathbf{d}_2^{(CR)})$
$M_2^{(CR)}$
$\mathbf{c}_3^{(CR)} = m_3(\mathbf{d}_3^{(CR)})$
$M_3^{(CR)}$
$\begin{bmatrix}\mathbf{d}_1^{(CR)}\\ \mathbf{d}_2^{(CR)}\\ \mathbf{d}_3^{(CR)}\end{bmatrix} = G \begin{bmatrix}\mathbf{c}_1^{(CR)}\\ \mathbf{c}_2^{(CR)}\\ \mathbf{c}_3^{(CR)}\end{bmatrix}$
(c)

Figure 5-5: An illustration of (a) a shorthand graphical representation of the stationarity conditions using reduced-form primal components, (b) the full graphical representation of the stationarity conditions, and (c) the transformed stationarity conditions associated with several different regularization problems.

interconnective structure can be recovered by concatenating $\mathbf{c}^{\star(S)}$ to $\mathbf{c}^{\star(m)}$ and $\mathbf{d}^{\star(S)}$ to $\mathbf{d}^{\star(m)}$ where $\mathbf{d}^{\star(S)}$ can be generated by

$$\mathbf{d}^{\star(S)} = (I - G_{SS}S)^{-1}\left(G_{Sm}\mathbf{c}^{\star(m)} + G_{ss}\mathbf{e}\right) \tag{5.97}$$

and $\mathbf{c}^{\star(S)}$ can be generated by $\mathbf{c}^{\star(S)} = S\mathbf{d}^{\star(S)} + \mathbf{e}$. As will often be the case, the equivalence between solutions to the CCSPs described by the interconnective structures in Figures 5-4(c) and (d) is particularly useful when the constitutive function $m(\cdot)$ is α-dissipative everywhere. When this occurs, the system operator associated with the interconnective structure in (c) is passive everywhere while the system operator associated with the source-free interconnective structure in (d) is α-dissipative everywhere. In terms of solving the CCSPs, the stability and robustness properties of the structures in (c) and (d) are summarized by Theorems 4.3.6 and 4.3.3, respectively. Note that solving the problem described by the structure in (c) by asynchronously implementing the system requires the use of filtered delay modules while solving the problem described by the structure in (d) does not.

We next present an example of the modular design process used to assemble canonical-form and scattering-form structures from an optimization problem statement. Consider the standard non-negative least squares (NNLS) problem given by

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \tfrac{\lambda}{2}\|B\mathbf{x} - \mathbf{y}\|_2^2 \\ \text{s.t.} \quad & \mathbf{x} \geq \mathbf{0} \end{aligned} \tag{5.98}$$

where B is a coefficient matrix and $\mathbf{y}$ is a vector of measurements. Written in reduced form, the corresponding primal (P) and dual (D) conservative problems are given by

$$\text{(P)}\quad \begin{array}{rl} \underset{\mathbf{a}}{\text{minimize}} & \frac{\lambda}{2}\|\mathbf{a}_3^{(CR)}\|_2^2 \\ \text{s.t.} & B\mathbf{a}_1^{(CR)} - \mathbf{a}_2^{(CR)} = \mathbf{a}_3^{(CR)} \\ & \mathbf{a}_2^{(CR)} = \mathbf{y} \\ & \mathbf{a}_1^{(CR)} \geq \mathbf{0} \end{array} \qquad \text{(D)}\quad \begin{array}{rl} \underset{\mathbf{b}}{\text{maximize}} & -\frac{1}{2\lambda}\|\mathbf{b}_3^{(CR)}\|_2^2 - \mathbf{y}^T\mathbf{b}_2^{(CR)} \\ \text{s.t.} & -B^T\mathbf{b}_3^{(CR)} = \mathbf{b}_1^{(CR)} \\ & \mathbf{b}_3^{(CR)} = \mathbf{b}_2^{(CR)} \\ & \mathbf{b}_1^{(CR)} \leq \mathbf{0} \end{array} \qquad (5.99)$$

where the primal decision vectors $(\mathbf{a}_1^{(CR)}, \mathbf{a}_2^{(CR)}, \mathbf{a}_3^{(CR)})$ correspond to the vectors $(\mathbf{x}, \mathbf{y}, B\mathbf{x} - \mathbf{y})$. Observe that while the problem formulations in (5.98) and (5.99)(P) are clearly equivalent, they are not the same problem. The stationarity conditions associated with the conservative formulation are depicted as a canonical-form interconnective structure in Figure 5-5(a)-(b) where the constitutive relation modules are unspecified. Note that the structures in (a) and (b) are equivalent to one another, where the structure in (b) additionally shows the portion of the stationarity conditions implied by the dual feasibility constraints that correspond to the primal feasibility conditions depicted in (a). The scattering-form structure corresponding to the transformed stationarity conditions is depicted in Figure 5-5(c). An asynchronous, distributed scattering algorithm for solving both primal and dual problems correspond to implementing the scattering-form structure by inserting delay elements to break the delay-free loops in (c). The details in doing this, in particular for effectively distributing the computation in a variety of settings, is discussed in Chapter 6.

We conclude this subsection and the example presented above with two remarks. First, as will be discussed later, source-free structures for the NNLS problem can be constructed by eliminating all variables except those related to module 1. Second, by modifying the memoryless nonlinearities associated with the constitutive modules, the stationarity conditions can be appropriately modified to solve a variety of related problems including variants of least squares such as unconstrained and bounded-value, regularization problems such as basis pursuit denoising, ridge regression, total variation, the Dantzig selection, maximum likelihood and MAP estimation in exponential families, and training of support vector machines. The modules required for these modifications are derived in the next section.

5.2.3 | The role of convexity

A celebrated result in convex analysis states that if an unconstrained convex function has a stationary point then that point is a global minimizer, and the collection of all such minimizers forms a convex set [40]. Moreover, if the function is strictly convex, then the minimizer is unique. In this subsection, the role of convexity is explored in the context of obtaining well-defined functional realizations of the constitutive modules used in the implementation of a scattering algorithm. The main result is that convexity of the reduced-form primal cost function is sufficient but not necessary for these functions to exist, and, moreover, that convexity implies they are α-conic with parameters α no larger than unity.

Consider the general form of the constitutive module depicted in Figure 5-3 where we proceed without the subvector notation for clarity. The scattering coordinate transforms $M^{(i)}$ and $M^{(o)}$ are applied to the relations describing the primal and dual decision variables $\mathbf{a}$ and $\mathbf{b}$ depending on whether $\mathbf{a}$ corresponds to an input or output of the interconnecting network that the constitutive module is attached to, respectively. Using the standard assignments $\mathbf{a} = f(\mathbf{x})$ and $\mathbf{b} = g(\mathbf{x})$, the scattering variables $\mathbf{c}$ and $\mathbf{d}$ are related to one another using the transform $M^{(i)}$ when $\mathbf{a}$ denotes an output of the module according to

$$\begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix} = \underbrace{\begin{bmatrix} I & -I \\ I & I \end{bmatrix}}_{M^{(i)}} \begin{bmatrix} f(\mathbf{x}) \\ g(\mathbf{x}) \end{bmatrix}. \tag{5.100}$$

The functional mapping from $\mathbf{d}$ to $\mathbf{c}$, when it exists, can thus be expressed as

$$\mathbf{c} \;=\; (f-g)(f+g)^{-1}(\mathbf{d}). \tag{5.101}$$

Similarly, the scattering variables $\mathbf{c}$ and $\mathbf{d}$ are related to the functions f and g when $\mathbf{a}$ denotes an input to the module and the coordinate transform $M^{(o)}$ is used according to

$$\begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix} = \underbrace{\begin{bmatrix} -I & I \\ I & I \end{bmatrix}}_{M^{(o)}} \begin{bmatrix} f(\mathbf{x}) \\ g(\mathbf{x}) \end{bmatrix}. \tag{5.102}$$

The functional mapping from $\mathbf{d}$ to $\mathbf{c}$ for this case, when it exists, can be expressed as

$$\mathbf{c} \;=\; (g-f)(f+g)^{-1}(\mathbf{d}). \tag{5.103}$$

There is a symmetry between the expressions (5.101) and (5.103), and arguments for the existence of these functional relationships rely solely on the invertibility of $\mathbf{d} = (f+g)(\mathbf{x})$. Therefore, the existence of one function is sufficient for the existence of the other, so we proceed by focusing the discussion to the existence and properties of the function in (5.101) with the understanding that a similar treatment for the function in (5.103) follows analogously.

Drawing on the reduced-form interpretation of the stationarity conditions in Section 5.1.5, the relation between the variables $\mathbf{a}$ and $\mathbf{b}$ used to define a constitutive module reduces to the subdifferential relationship $\mathbf{b} = \partial\widehat{Q}(\mathbf{a})$ where the reduced-form cost function $\widehat{Q}$ subsumes the set constraint $\mathcal{A}$ by taking values of infinity for any value of $\mathbf{a} \notin \mathcal{A}$. For this case, the function in (5.101) can be written as

$$\mathbf{c} \;=\; (I-\partial\widehat{Q})(I+\partial\widehat{Q})^{-1}(\mathbf{d}). \tag{5.104}$$

A sufficient condition for invertibility of the mapping $\mathbf{d} = (I+\partial\widehat{Q})(\mathbf{x})$ is convexity of the reduced-form cost function $\widehat{Q}$. For example, in the case where $\mathbf{c}$ and $\mathbf{d}$ are scalar-valued variables, the function $I+\partial\widehat{Q}$ is composed of the sum of a strictly increasing function and a non-decreasing relation, and is subsequently invertible since it is itself a strictly increasing function. It is clear in this case that the term $\partial\widehat{Q}$ is not required to be non-decreasing for the relationship to be invertible, hence convexity of $\widehat{Q}$ is a sufficient but not necessary condition. The argument used to extend this result to the multi-dimensional setting is similar to the argument in the scalar case. Indeed, from convexity of $\widehat{Q}$, a well-known result in convex analysis is that the relation $\partial\widehat{Q}(\mathbf{x})$ is monotone, i.e. $\partial\widehat{Q}$ satisfies the condition

$$\langle \partial\widehat{Q}(\mathbf{x}) - \partial\widehat{Q}(\mathbf{y}), \mathbf{x}-\mathbf{y}\rangle \;\geq\; 0, \qquad \mathbf{x},\mathbf{y}\in\mathbb{R}^N, \tag{5.105}$$

thus the operator $I+\partial\widehat{Q}$ is strongly monotone since it is the sum of a monotone and a strongly monotone function. Invertibility of $I+\partial\widehat{Q}$ then follows from application of the theorem of Browder-Minty on monotone operators [40]. Moreover, the operator $(I+\partial\widehat{Q})^{-1}$ appears in many optimization algorithms and is the so-called proximal operator associated with the function $\widehat{Q}$. Connections between proximal operators and algorithms and scattering operators and algorithms are made in Section 5.6.3.

Having established that convexity of the reduced-form cost function $\widehat{Q}$ is sufficient for the functional realization of the associated constitutive module in a scattering-form structure to be well-defined, we proceed to justify that the functional realization is, in the worst case, passive everywhere. Recall that the value of α corresponding to an α-conic operator of the form $T = T_1 \circ T_2$ is given by the product of the values of α for the individual operators T_1 and T_2 when both operators are α-conic themselves. For the form of the functional realization in (5.104), the operator $T_1 = (I - \partial\widehat{Q})$ and $T_2 = (I + \partial\widehat{Q})^{-1}$. The value of α satisfied by the operator T_1 is upper bounded by 1, which follows from straightforward properties of Lipschitz functions and monotone operators. The value of α satisfied by the operator T_2 is also 1, and this fact follows from well-known results about the Lipschitz continuity of proximal operators [97]. Therefore, the functional realization in (5.104) is α-conic with a value of α no larger than one for convex reduced-form cost functions $\widehat{Q}$.

5.3 | Constitutive modules

To assist with assembling scattering algorithms for solving conservative optimization problems by connecting basic modules together by inspection of an optimization problem statement, we present a general strategy in this section for deriving constitutive modules pertaining to particular cost functions and feasibility constraints and then proceed to generate several example modules by following it. Referring to the general form of a constitutive module in the bottom row of Figure 5-3, the goal in doing this is specifically to relate the reduced-form primal components $\widehat{Q}$ and $\mathcal{A}$ in column three to the functional realization of the module m in column five. To do this, we first describe a procedure to go from a functional realization to the reduced-form primal components followed by a complementary procedure to go from the reduced-form primal components to the functional realization. Defining modules by using the latter procedure is not generally a well-posed task.

For the purpose of clarity in the coming presentation, we proceed without the subvector notation used in Figure 5-3 since the primary objects of interest in this context are the mapping objects themselves rather than their arguments. Furthermore, the coordinate transforms referenced throughout this section are specifically scattering coordinate transforms and relate canonical-form and scattering-form interconnective structures. The modules appearing in this section, which may be used to solve a broad class of convex and non-convex conservative optimization problems, as well as several others are listed in a table in Appendix B for convenience.

5.3.1 | General strategies for deriving constitutive modules

Consider first the task of determining the reduced-form primal components $\widehat{Q}$ and $\mathcal{A}$ from a prespecified function m relating the scattering variables in the transformed stationarity conditions according to $\mathbf{c} = m(\mathbf{d})$. The following steps outline the general approach we take to determining these components:

(i) Generate the set-valued relations forming the correspondence between the canonical decision variables $\mathbf{a}$ and $\mathbf{b}$ by applying the inverse scattering transform to the relation forming the correspondence between the transformed variables $\mathbf{c}$ and $\mathbf{d}$. This can be done, for example, by applying the inverse matrix $[M_k^{(CR)}]^{-1}$ to each of the set-valued relationships appearing in (5.93) and assigning the relationships $\mathbf{a} = f_k(\mathbf{x}_k^{(CR)})$ and $\mathbf{b} = g_k(\mathbf{x}_k^{(CR)})$.

(ii) Determine the canonical functions $f(\mathbf{x})$ and $g(\mathbf{x})$ and the functionals $Q(\mathbf{x})$ and $R(\mathbf{x})$ that are consistent with the set-valued relationships generated in step (i) and that also satisfy the stationarity conditions (5.25) through (5.27).

(iii) Determine the reduced-form primal components $\widehat{Q}(\mathbf{a})$ and $\mathcal{A}$ that satisfy the set-valued relationship in (5.60) using the function $f(\mathbf{x})$ and functional $Q(\mathbf{x})$ found in step (ii).

A sufficient condition for the functions $f(\mathbf{x})$ and $g(\mathbf{x})$ identified in step (ii) to exist is Lipschitz continuity of the function m. The task associated with finding them is equivalent to a so-called function chase or curve parameterization and can readily make use of standard methods from differential geometry. As was mentioned previously, the task associated with step (iii) cannot always be accomplished since the reduced-form components do not generally exist. Furthermore, when the reduced-form dual components $\widehat{R}(\mathbf{b})$ and $\mathcal{B}$ are also desired and exist, they may be found in the same manner as step (iii) using (5.64) and the functions $g(\mathbf{x})$ and $R(\mathbf{x})$ from step (ii).

Consider next the complementary task of defining a constitutive relation module by starting with reduced-form primal components $\widehat{Q}$ and $\mathcal{A}$ and ending with a scattering function m satisfying $\mathbf{c} = m(\mathbf{d})$. The following steps outline the general approach we take to determining these functions:

(i) Determine the canonical-form primal components $Q(\mathbf{x})$ and $f(\mathbf{x})$ that satisfy the set-valued relationship in (5.60) using the provided components $\widehat{Q}$ and $\mathcal{A}$.

(ii) Determine the canonical-form dual components $R(\mathbf{x})$ and $g(\mathbf{x})$ that are consistent with the function $f(\mathbf{x})$ and functional $Q(\mathbf{x})$ found in step (i) and satisfy the stationarity conditions (5.25) (5.27).

(iii) Generate the set-valued relation constraining the scattering variables $\mathbf{d}$ and $\mathbf{c}$ by first generating the set-valued relation constraining the canonical variables $\mathbf{a} = f(\mathbf{x})$ and $\mathbf{b} = g(\mathbf{x})$ according to (5.93).

(iv) Determine a functional relationship mapping the scattering vector $\mathbf{d}$ to $\mathbf{c}$. When a differentiable realization is known to exist, this may be done by solving for the primitive function of the expression in (5.101) or (5.103).

Aside from the previous connection between the objective function Q determined in step (i) and the function m sought after in step (iv), the relation generated in step (iii) may not have a functional realization for the desired input-output configuration. For a complete description of a constitutive module, the instructions for step (iv) need to be repeated for each scattering coordinate transform of interest. In the next subsection, we primarily follow the second strategy above since we are interested in all of the functions m associated with particular reduced-form objective functions and constraints.

5.3.2 | Example constitutive modules

Guided by the strategies for deriving constitutive modules discussed in the previous subsection, some example modules are derived in this subsection and commentary is presented on useful analytic and numerical tools that can be used to derive a variety of additional modules. A recurring theme throughout this process is to take advantage of various symmetries, both in terms of the canonical-form and reduced-form problem descriptions as well as in the stationarity conditions, to produce several constitutive modules at once. The modules derived in this section are used in Chapter 6 to solve various optimization problems arising in signal processing contexts by attaching the modules to the interconnect described in the top row of Figure 5-3 using asynchronous delay elements to break delay-free loops, possibly after eliminating source modules.

The first module we define, which will be used to generate several related modules through straightforward transformations, is associated with reduced-form components that have the following behavior: a primal decision variable $\mathbf{a}$ is constrained to a closed interval and does not contribute to the primal cost function while the dual decision variable $\mathbf{b}$ is unconstrained and contributes to the dual cost function through a piece-wise linear function. Indeed, to restrict the primal decision variable $\mathbf{a} = f(\mathbf{x})$ to a simple closed interval $[l, u] \subset \mathbb{R}$ for arbitrary constants l and u satisfying $l \leq u$, we define the scalar-valued function $f\colon \mathbb{R} \to \mathbb{R}$ according to

$$f(\mathbf{x}) \quad = \quad \begin{cases} u, & \mathbf{x} > u \\ \frac{u-l}{2} \sin\left(\frac{\pi}{u-l}\mathbf{x} + \frac{\pi}{2} - \frac{\pi u}{u-l}\right) + \frac{u+l}{2}, & l \leq \mathbf{x} \leq u \\ l, & \mathbf{x} < l \end{cases} . \tag{5.106}$$

Similarly, to allow the dual decision variable $\mathbf{b} = g(\mathbf{x})$ to be unconstrained, we define the function $g\colon \mathbb{R} \to \mathbb{R}$ according to

$$g(\mathbf{x}) \quad = \quad \begin{cases} (\mathbf{x} - u)^2, & \mathbf{x} > u \\ 0, & l \leq \mathbf{x} \leq u \\ -(\mathbf{x} - l)^2, & \mathbf{x} < l \end{cases} . \tag{5.107}$$

With these canonical functions in place, it is straightforward to verify that a valid choice of canonical-form primal and dual objective functions consistent with the stationarity conditions (5.25) and (5.26) is

$$Q(\mathbf{x}) = 0 \tag{5.108}$$

$$R(\mathbf{x}) = \begin{cases} u(\mathbf{x}-u)^2, & \mathbf{x} > u \\ 0, & l \leq \mathbf{x} \leq u \\ -l(\mathbf{x}-l)^2, & \mathbf{x} < l \end{cases} . \tag{5.109}$$

The example module consistent with the description above is provided in the top row of Figure 5-6 where the functional realizations in column five correspond to having solved for $\mathbf{c}$ as a function of $\mathbf{d}$ for the scattering coordinate transforms listed. Note that the quadratic nature of R does not imply a quadratic form of $\widehat{R}$ since $\widehat{R}$ only depends on $\mathbf{x}$ through g. The reduced-form components listed in columns three and four were generated by solving (5.60) and (5.64). Referring still to Figure 5-6, the two modules listed in the second and third rows respectively correspond to constraining the primal decision variable to left-closed and right-closed intervals and may be obtained by proper manipulation of the two-sided inequality. Specifically, the module in the middle row is obtained by selecting $l = 0$ and taking u to infinity while the module in the bottom row is obtained by selecting $u = 0$ and taking l to negative infinity. Modules for arbitrary single-sided inequalities follow from a similar treatment of l and u. Referring to the function f in (5.106), the selection of a sinusoid over the interval $l \leq \mathbf{x} \leq u$ is not critical to the reduced-form components, i.e. any function that is antisymmetric about $\frac{l+u}{2}$ and appropriately maintains continuity of f and its first derivative over this interval results in essentially the same module. Subsequently, in presenting the remaining modules we will often omit the particular functions f and g for brevity with an understanding that they may all be defined over such invariant regions using standard parameterization tools such as Hermite interpolation polynomials.

Beyond imposing inequality constraints on primal and dual decision variables, many additional example modules of interest follow immediately from the selection of f and g in (5.106) and (5.107). The next category of modules we develop are associated with reduced-

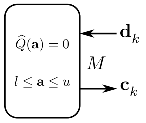
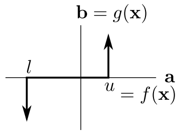
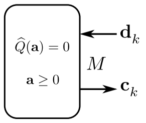
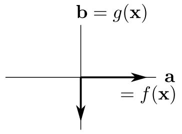
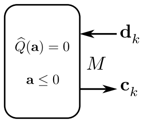
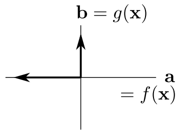
Graph symbol	Canonical behavior	Reduced-form primal components	Reduced-form dual components	Realization		α- connicity
 $\widehat{Q}(\mathbf{a})=0$ $l \le \mathbf{a} \le u$ M $\mathbf{d}_k$ $\mathbf{c}_k$	 $\mathbf{b}=g(\mathbf{x})$ l u $\mathbf{a}$ $= f(\mathbf{x})$	$\widehat{Q}(\mathbf{a})=0$ $l \le \mathbf{a} \le u$	$\widehat{R}(\mathbf{b}) = \begin{cases} u\mathbf{b}, & \mathbf{b} \ge 0 \\ l\mathbf{b}, & \mathbf{b} < 0 \end{cases}$ $\mathbf{b} \in \mathbb{R}$	$M^{(i)}$	$\mathbf{c} = \begin{cases} \mathbf{d}, & l \le \mathbf{d} \le u \\ 2u - \mathbf{d}, & \mathbf{d} > u \\ 2l - \mathbf{d}, & \mathbf{d} < l \end{cases}$	passive everywhere
				$M^{(o)}$	$\mathbf{c} = \begin{cases} -\mathbf{d}, & l \le \mathbf{d} \le u \\ \mathbf{d} - 2u, & \mathbf{d} > u \\ \mathbf{d} - 2l, & \mathbf{d} < l \end{cases}$	
 $\widehat{Q}(\mathbf{a})=0$ $\mathbf{a} \ge 0$ M $\mathbf{d}_k$ $\mathbf{c}_k$	 $\mathbf{b}=g(\mathbf{x})$ $\mathbf{a}$ $= f(\mathbf{x})$	$\widehat{Q}(\mathbf{a})=0$ $\mathbf{a} \ge 0$	$\widehat{R}(\mathbf{b})=0$ $\mathbf{b} \le 0$	$M^{(i)}$	$\mathbf{c} = \|\mathbf{d}\|$	passive everywhere
				$M^{(o)}$	$\mathbf{c} = -\|\mathbf{d}\|$	
 $\widehat{Q}(\mathbf{a})=0$ $\mathbf{a} \le 0$ M $\mathbf{d}_k$ $\mathbf{c}_k$	 $\mathbf{b}=g(\mathbf{x})$ $\mathbf{a}$ $= f(\mathbf{x})$	$\widehat{Q}(\mathbf{a})=0$ $\mathbf{a} \le 0$	$\widehat{R}(\mathbf{b})=0$ $\mathbf{b} \ge 0$	$M^{(i)}$	$\mathbf{c} = -\|\mathbf{d}\|$	passive everywhere
				$M^{(o)}$	$\mathbf{c} = \|\mathbf{d}\|$	

Figure 5-6: Example constitutive modules that are associated with reduced-form primal and dual components that have zero-valued cost functions and enforce inequality constraints.

form components satisfying the following behavior: a decision variable in one reduced-form problem is set to a fixed-value while the decision variable in the dual problem is linearly penalized and vice versa. Modules of this type will play an important role in optimization problems for which data in the form of a measurement vector is to be incorporated into the problem statement. The first module of this type is completely determined using (5.106) and (5.107) by selecting a degenerate interval $[l, u] = [\rho, \rho]$ for some $\rho \in \mathbb{R}$. We summarize the effect of this selection on the canonical functions and objective functions according to

$$\left.\begin{array}{rcl} f(\mathbf{x}) & = & \rho \\ g(\mathbf{x}) & = & \begin{cases} (\mathbf{x}-\rho)^2, & \mathbf{x} \ge \rho \\ -(\mathbf{x}-\rho)^2, & \mathbf{x} < \rho \end{cases} \end{array}\right\} \implies \left\{\begin{array}{rcl} Q(\mathbf{x}) & = & 0 \\ R(\mathbf{x}) & = & \begin{cases} \rho(\mathbf{x}-\rho)^2, & \mathbf{x} \ge \rho \\ -\rho(\mathbf{x}-\rho)^2, & \mathbf{x} < \rho \end{cases} \end{array}\right. . \quad (5.110)$$

An example module consistent with the description above and the module that, by symmetry, swaps the roles of the primal and dual decision variables are illustrated in Figure 5-7. Note that the functional realizations of the modules are precisely in a form that is a source module.

We proceed to establish some example modules that have norm and norm-like reduced-form primal objective functions. Once again, we begin by drawing upon another symmetry

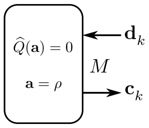
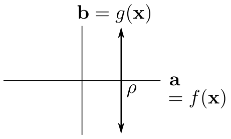
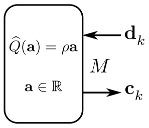
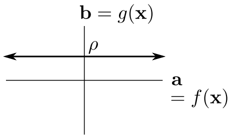
Graph symbol	Canonical behavior	Reduced-form primal components	Reduced-form dual components	Realization		$\alpha-$ connicity
 $\widehat{Q}(\mathbf{a})=0$ $\mathbf{a}=\rho$ $\mathbf{d}_k$ M $\mathbf{c}_k$		$\widehat{Q}(\mathbf{a})=0$ $\mathbf{a}=\rho$	$\widehat{R}(\mathbf{b})=\rho\mathbf{b}$ $\mathbf{b}\in\mathbb{R}$	$M^{(i)}$	$\mathbf{c}=-\mathbf{d}+2\rho$	passive everywhere
				$M^{(o)}$	$\mathbf{c}=\mathbf{d}-2\rho$	
 $\widehat{Q}(\mathbf{a})=\rho\mathbf{a}$ $\mathbf{a}\in\mathbb{R}$ $\mathbf{d}_k$ M $\mathbf{c}_k$		$\widehat{Q}(\mathbf{a})=\rho\mathbf{a}$ $\mathbf{a}\in\mathbb{R}$	$\widehat{R}(\mathbf{b})=0$ $\mathbf{b}=\rho$	$M^{(i)}$	$\mathbf{c}=\mathbf{d}-2\rho$	passive everywhere
				$M^{(o)}$	$\mathbf{c}=-\mathbf{d}+2\rho$	

Figure 5-7: Example constitutive modules that are associated with reduced-form primal and dual components satisfying the structural conditions of source modules.

of the two-sided inequality module defined using (5.106) and (5.107) to establish a module with a scaled absolute value function as its reduced-form primal objective. Indeed, by selecting $-l=u=\rho$ and swapping the roles of f and g, we have effectively transformed the two-sided inequality module into a weighted 1-norm module. The result of carrying out this transformation is depicted in the top row of Figure 5-8. A similar module, provided in the middle row, illustrates a common approximation to the 1-norm that corresponds to using a quadratic approximation near 0 so that the module's reduced-form objective function $\widehat{Q}$ is differentiable everywhere. To establish p-norm modules with $p>1$, we write the canonical relationship $\mathbf{a}=f(\mathbf{x})$ and $\mathbf{b}=g(\mathbf{x})$ directly as $\mathbf{b}=\operatorname{sgn}(\mathbf{a})|\mathbf{a}|^{p-1}$. Generally speaking, explicitly writing $\mathbf{c}$ as a function of $\mathbf{d}$ for this relationship is difficult analytically. However, it is straightforward to produce an approximation via a table lookup for arbitrary values of p. The module associated with $p=2$, for which the relationship reduces to a linear system of implicit equations, is depicted in the bottom row of Figure 5-8 where we have additionally parameterized the module so that the quadratic penalty can be scaled differently for positive and negative values of $\mathbf{a}$.

The final norm-related example module we derive corresponds to a canonical-form primal cost function $Q\colon\mathbb{R}^K\to\mathbb{R}$ given by $Q(\mathbf{x})=\frac{1}{2}\mathbf{x}^TA\mathbf{x}$ where A is without loss of generality a symmetric matrix that is also assumed not to have any eigenvalues of -1. By selecting $f(\mathbf{x})=\mathbf{x}$ and $g(\mathbf{x})=\nabla Q(\mathbf{x})$, we can immediately solve the condition (5.101) to obtain a functional realization $m\colon\mathbb{R}^K\to\mathbb{R}^K$ that produces the scattering variable $\mathbf{c}$ as a function of

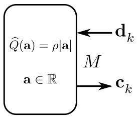
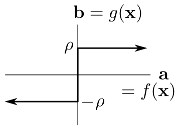
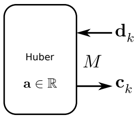
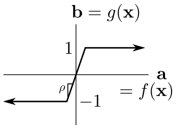
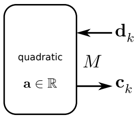
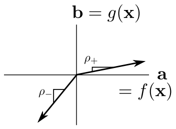
Graph symbol	Canonical behavior	Reduced-form primal components	Reduced-form dual components	Realization		α- connicity
 $\widehat{Q}(\mathbf{a})=\rho\|\mathbf{a}\|$ $\mathbf{a}\in\mathbb{R}$ M $\mathbf{d}_k$ $\mathbf{c}_k$		$\widehat{Q}(\mathbf{a})=\rho\|\mathbf{a}\|$ $\mathbf{a}\in\mathbb{R}$	$\widehat{R}(\mathbf{b})=0$ $-\rho\leq\mathbf{b}\leq\rho$	$M^{(i)}$	$\mathbf{c}=\begin{cases}-\mathbf{d}, & \|\mathbf{d}\|\leq\rho\\ \mathbf{d}-2\rho\text{sgn}(\mathbf{d}), & \|\mathbf{d}\|>\rho\end{cases}$	passive everywhere
				$M^{(o)}$	$\mathbf{c}=\begin{cases}\mathbf{d}, & \|\mathbf{d}\|\leq\rho\\ 2\rho\text{sgn}(\mathbf{d})-\mathbf{d}, & \|\mathbf{d}\|>\rho\end{cases}$	
 Huber $\mathbf{a}\in\mathbb{R}$ M $\mathbf{d}_k$ $\mathbf{c}_k$		$\widehat{Q}(\mathbf{a})=\begin{cases}\|\mathbf{a}\|, & \|\mathbf{a}\|\geq\frac{1}{\rho}\\ \frac{\rho}{2}\mathbf{a}^2+\frac{1}{2\rho}, & \|\mathbf{a}\|<\frac{1}{\rho}\end{cases}$ $\mathbf{a}\in\mathbb{R}$	$\widehat{R}(\mathbf{b})=\frac{1}{2\rho}(\mathbf{b}^2-1)$	$M^{(i)}$	$\mathbf{c}=\begin{cases}\frac{1-\rho}{1+\rho}\mathbf{d} & \|\mathbf{d}\|\leq\frac{1}{\rho}+1\\ \mathbf{d}-2\text{sgn}(\mathbf{d}), & \|\mathbf{d}\|>\frac{1}{\rho}+1\end{cases}$	passive everywhere
				$M^{(o)}$	$\mathbf{c}=\begin{cases}\frac{\rho-1}{\rho+1}\mathbf{d} & \|\mathbf{d}\|\leq\frac{1}{\rho}+1\\ -\mathbf{d}+2\text{sgn}(\mathbf{d}), & \|\mathbf{d}\|>\frac{1}{\rho}+1\end{cases}$	dissipative everywhere if $\|\mathbf{d}\|\leq\frac{1}{\rho}+1$ $0<\rho<\infty$
 quadratic $\mathbf{a}\in\mathbb{R}$ M $\mathbf{d}_k$ $\mathbf{c}_k$		$\widehat{Q}(\mathbf{a})=\begin{cases}\frac{\rho_+}{2}\mathbf{a}^2, & \mathbf{a}\geq0\\ \frac{\rho_-}{2}\mathbf{a}^2, & \mathbf{a}<0\end{cases}$ $\mathbf{a}\in\mathbb{R}$	$\widehat{R}(\mathbf{b})=\begin{cases}\frac{1}{2\rho_+}\mathbf{b}^2, & \mathbf{b}\geq0\\ \frac{1}{2\rho_-}\mathbf{b}^2, & \mathbf{b}<0\end{cases}$ $\mathbf{b}\in\mathbb{R}$	$M^{(i)}$	$\mathbf{c}=\begin{cases}\frac{1-\rho_+}{1+\rho_+}\mathbf{d}, & \mathbf{d}\geq0\\ \frac{1-\rho_-}{1+\rho_-}\mathbf{d}, & \mathbf{d}<0\end{cases}$	dissipative everywhere if $0<\rho_-<\infty$ $0<\rho_+<\infty$
				$M^{(o)}$	$\mathbf{c}=\begin{cases}\frac{\rho_+-1}{\rho_++1}\mathbf{d}, & \mathbf{d}\geq0\\ \frac{\rho_--1}{\rho_-+1}\mathbf{d}, & \mathbf{d}<0\end{cases}$	passive everywhere if $\rho_+=\rho_-=0$

Figure 5-8: Example constitutive modules that are associated with reduced-form primal and dual components that have norm and norm-like cost functions with unconstrained domains.

the scattering variable $\mathbf{d}$ and takes the form

$$m(\mathbf{d}) \;=\; (I-A)\,(I+A)^{-1}\,\mathbf{d}. \tag{5.111}$$

The particular functional realization in (5.111) corresponds to a scattering coordinate transform consistent with the decision vector $\mathbf{a}=f(\mathbf{x})$ being an output of the interconnecting network and an input to the constitutive module. A similar expression for alternative coordinate transforms can be obtained by following the same steps above. The following proposition summarizes the system operator properties for (5.111) as they pertain to spectral properties of the matrix A.

Proposition 5.3.1 (α-conicity of quadratic objective functions and definiteness)**.** *Let A denote a symmetric $K\times K$-dimensional matrix without eigenvalues of -1. Then the function $m\colon\mathbb{R}^K\to\mathbb{R}^K$ in* (5.111) *is α-conic everywhere and is categorized as:*

(i) α-dissipative everywhere if and only if A is positive definite;

(ii) passive everywhere if and only if A is positive semidefinite;

(iii) α-expansive otherwise.

Proof. We proceed to use the condition (4.23) and the fact that m is linear to reduce solving for the value of α to solving for an operator norm. Let $\mathbf{u}\in\mathbb{R}^K$ denote a non-zero vector

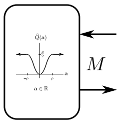
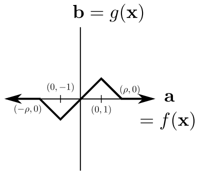
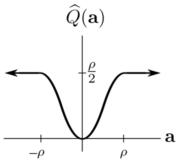
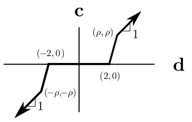
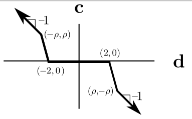
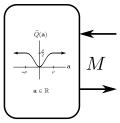
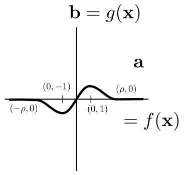
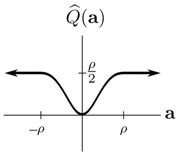
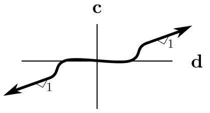
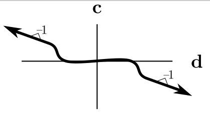
Graph symbol	Canonical behavior	Reduced-form primal components	Realization		α- connicity
		 $\mathbf{a} \in \mathbb{R}$	$M^{(i)}$		passive about $\mathbf{d} = 0$
			$M^{(o)}$		
		 $\mathbf{a} \in \mathbb{R}$	$M^{(i)}$		passive about $\mathbf{d} = 0$
			$M^{(o)}$		

Figure 5-9: Example constitutive modules that are associated with reduced-form primal components that have non-convex cost functions and unconstrained domains. The reduced-form dual components for these modules do not exist.

and set $\mathbf{v} = (I_K + A)^{-1}\mathbf{u}$. Then,

$$\|m\| = \sup_{\mathbf{u}\neq\mathbf{0}} \frac{\|m(\mathbf{u})\|}{\|\mathbf{u}\|} = \sup_{\mathbf{u}\neq\mathbf{0}} \frac{\|(I_K - A)(I_K + A)^{-1}\mathbf{u}\|}{\|\mathbf{u}\|} = \sup_{\mathbf{v}\neq\mathbf{0}} \frac{\|(I_K - A)\mathbf{v}\|}{\|(I_K + A)\mathbf{v}\|} \tag{5.112}$$

which is consistent with α being non-negative. Substituting the inner-product notation for the 2-norm, we obtain

$$\|m\|_2 = \sup_{\mathbf{v}\neq\mathbf{0}} \sqrt{\frac{\|\mathbf{v}\|^2 - \mathbf{v}^T(A^T + A)\mathbf{v} + \|A\mathbf{v}\|^2}{\|\mathbf{v}\|^2 + \mathbf{v}^T(A^T + A)\mathbf{v} + \|A\mathbf{v}\|^2}} = \sqrt{1 - \inf_{\mathbf{v}\neq\mathbf{0}} \frac{2\mathbf{v}^T A\mathbf{v}}{\|(I_K + A)\mathbf{v}\|^2}}. \tag{5.113}$$

Since $\|(I_K + A)\mathbf{v}\|^2 > 0$ for all $\mathbf{v} \neq \mathbf{0}$ the value of $\|m\|$ is determined by the sign of $\inf_{\mathbf{v}\neq\mathbf{0}} \mathbf{v}^T A\mathbf{v}$ as required. The conditions in (i) - (iii) follow immediately. ■

The next group of example constitutive modules we derive correspond to non-convex reduced-form primal objective functions. The two modules we define are similar to one another and are useful in statistical regularization problems where insensitivity to outliers is desired. To derive these modules, we begin with the desired reduced-form primal cost function $\widehat{Q}(\mathbf{a})$ using the free parameter $\rho > 1$ to parameterize the cost function according to

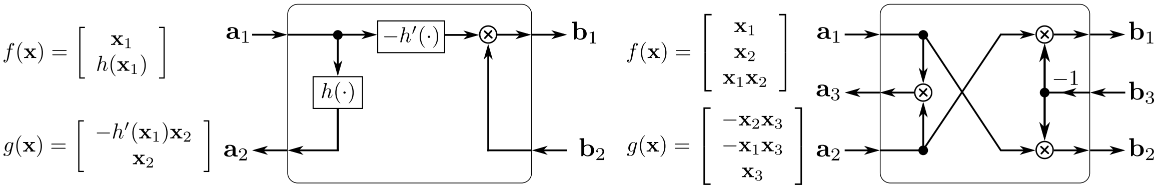
$$\widehat{Q}(\mathbf{a}) = \begin{cases} \frac{\rho}{2}, & |\mathbf{a}| > \rho \\ \frac{1}{2}\mathbf{a}^2, & |\mathbf{a}| < 1 \\ \frac{1}{2(1-\rho)}\mathbf{a}^2 - \operatorname{sgn}(\mathbf{a})\frac{\rho}{1-\rho}\mathbf{a} + \frac{\rho}{2(1-\rho)}, & 1 < |\mathbf{a}| \leq \rho \end{cases} . \tag{5.114}$$

Using the second procedure discussed in the previous subsection to derive the module, the function m mapping the scattering vector $\mathbf{d}$ to $\mathbf{c}$ for both coordinate transforms $M^{(i)}$ and $M^{(o)}$ is provided in column four of the top row of Figure 5-9, and these functions are well-defined as long as $\rho > 2$. This module additionally serves as the first example module presented with a well-defined realization but for which the reduced-form dual components do not exist, i.e. for which the dual cost function $\widehat{R}$ cannot be written as a function of the dual decision vector $\mathbf{b} = g(\mathbf{x})$ while simultaneously satisfying the condition in (5.64). To demonstrate an example module that is generated via table lookup, consider a trigonometric approximation of the reduced-form objective function (5.114) generated by selecting $\mathbf{a} = f(\mathbf{x}) = \mathbf{x}$ and $\mathbf{b} = g(\mathbf{x})$ where g is given by

$$g(\mathbf{x}) \quad = \quad \begin{cases} 0, & |\mathbf{x}| > \rho \\ \sin\left(\frac{\pi}{2}\mathbf{x}\right), & |\mathbf{x}| < 1 \\ \frac{\operatorname{sgn}(\mathbf{x})}{2}\left(\cos\left(\frac{\pi}{\rho-1}(1 - \operatorname{sgn}(\mathbf{x})\mathbf{x})\right) + 1\right), & 1 \leq |\mathbf{x}| \leq \rho \end{cases} . \tag{5.115}$$

Note that (5.115) is an approximation of the derivative of (5.114). A constitutive module associated with this description is illustrated in the bottom row of Figure 5-9 where the functional realizations in column four are generated numerically by evaluating (5.93) for $\mathbf{x} \in [-5, 5]$ with $\rho = 3.5$ and also result in valid functions as long as $\rho > 2$.

We conclude this section by discussing the canonical-form description of two modules associated with nonlinear constraints between primal decision variables where the module does not contribute to the primal objective function. The first module, illustrated in Figure 5-10(a), enforces the relationship $\mathbf{a}_2 = h(\mathbf{a}_1)$ between the primal decision variables $\mathbf{a}_1$ and $\mathbf{a}_2$ where $h\colon \mathbb{R} \to \mathbb{R}$ is an arbitrary, twice differentiable nonlinearity. The second module, illustrated in Figure 5-10(b), enforces the multiplicative relationship $\mathbf{a}_3 = \mathbf{a}_2\mathbf{a}_1$ between



(a) canonical nonlinear relationship

(b) canonical multiplicative constraint

Figure 5-10: Example constitutive modules that are associated with nonlinear canonical-form and reduced-form constraints.

the primal decision variables $\mathbf{a}_1$, $\mathbf{a}_2$, and $\mathbf{a}_3$. For both of these modules, an analytic description between the transformed variables $\mathbf{c}$ and $\mathbf{d}$ is currently unknown but can be generated numerically for use in practice as previously discussed.

5.4 | Decentralizing interconnecting networks using quadratic coupling modules

In this section, we develop a general purpose method for decentralizing interconnecting networks where the basic approach is to split the interconnecting network into many pieces, and to connect these pieces together by introducing auxiliary variables that communicate through quadratic coupling constitutive modules. These particular constitutive modules are designed with a free parameter that, when interpreted as the adaptive parameter in an adaptive signal processing system, suggest a numerical continuation scheme to be applied during the implementation of the structure in the scattering coordinate system. The details associated with setting up and executing scattering algorithms using the continuation scheme and quadratic coupling modules developed in this section are presented in Section 6.1.1.

To elaborate on the approach, let A denote a matrix that corresponds to, for example, the aggregate realization of all primal linear constraints in an optimization problem statement or the matrix that realizes an independent block of those constraints. In contexts where the rows, columns, or subblocks of A are formed using geographically distributed data sets, the method in this section enables the design and implementation of scattering algorithms where the subcomputation associated with executing or manipulating A are not required to be performed on any one of the individual computing resources available. This includes the

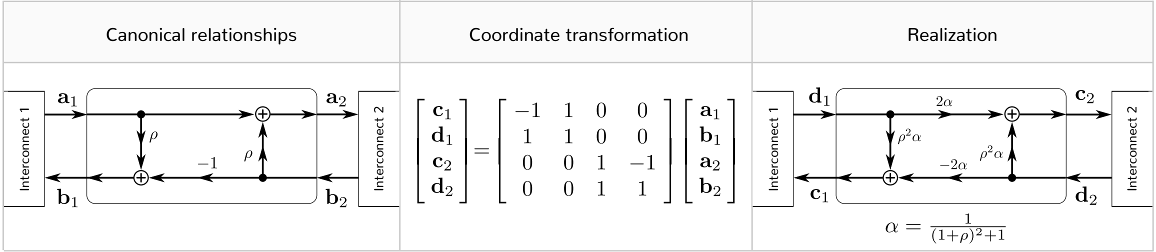

Canonical relationships
Coordinate transformation
Realization
Interconnect 1
Interconnect 2
$\mathbf{a}_1$
$\mathbf{a}_2$
$\mathbf{b}_1$
$\mathbf{b}_2$
ρ
-1
ρ
$\begin{bmatrix} \mathbf{c}_1 \\ \mathbf{d}_1 \\ \mathbf{c}_2 \\ \mathbf{d}_2 \end{bmatrix} = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{b}_1 \\ \mathbf{a}_2 \\ \mathbf{b}_2 \end{bmatrix}$
Interconnect 1
Interconnect 2
$\mathbf{d}_1$
$\mathbf{c}_2$
$\mathbf{c}_1$
$\mathbf{d}_2$
2α
$\rho^2\alpha$
-2α
$\rho^2\alpha$
$\alpha = \frac{1}{(1+\rho)^2+1}$

Figure 5-11: An illustration of the constitutive module that is used decentralize linear equations.

computation associated with forming the scattering-form interconnect matrix G associated with A. The approach is specifically to allow arbitrary blocks of A to be assigned to different processors so that the processors can be coupled together using the quadratic coupling constitutive module where agreement in satisfying the overall linear constraints is achieved by driving the adaptive parameter to zero. When the adaptive parameter is non-zero, the module acts as a quadratic penalty on the disagreement and has a dissipative everywhere functional realization in the transformed coordinate system.

We begin by deriving the quadratic coupling module consistent with the action described above. To accommodate coupling together subblocks of A of arbitrary dimensions, we define the module to couple together a single scalar variable from two neighboring interconnects. Coupling vectors is then accomplished from repeated application of the module. Indeed, consider the canonical functions $f\colon \mathbb{R}^2 \to \mathbb{R}^2$ and $g\colon \mathbb{R}^2 \to \mathbb{R}^2$ given by

$$f(\mathbf{x}) = (\mathbf{x}_1,\ \mathbf{x}_1 + \rho\mathbf{x}_2) \tag{5.116}$$

$$g(\mathbf{x}) = (\rho\mathbf{x}_1 - \mathbf{x}_2,\ \mathbf{x}_2). \tag{5.117}$$

where the free parameter $\rho \geq 0$ will play the role of the adaptive parameter. By solving the stationarity conditions (5.25) and (5.26), we obtain that the canonical-form primal and dual objective functionals $Q\colon \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ and $R\colon \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ are equal to one another and take the sum-of-squares form

$$Q(\mathbf{x}_1, \mathbf{x}_2) = R(\mathbf{x}_1, \mathbf{x}_2) \tag{5.118}$$

$$= \frac{1}{2}\rho\mathbf{x}_1^2 + \frac{1}{2}\rho\mathbf{x}_2^2. \tag{5.119}$$

Next, we make the standard variable assignments: $\mathbf{a} = f(\mathbf{x})$ and $\mathbf{b} = g(\mathbf{x})$. The relationships between the primal and dual decision variables are depicted in Figure 5-11 on the left. Using these variable assignments and solving for the reduced-form canonical components $\widehat{Q} \colon \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ and $\mathcal{A} \subseteq \mathbb{R}^2$ in (5.60), we find that

$$\widehat{Q}\left(\mathbf{a}_1, \mathbf{a}_2\right) \quad = \quad \frac{1}{2}\rho \mathbf{a}_1^2 + \frac{1}{2\rho}(\mathbf{a}_1 - \mathbf{a}_2)^2, \qquad \mathbf{a} \in \mathcal{A} = \mathbb{R}^2 \tag{5.120}$$

Observe that following the same procedure to this point but having chosen $\rho = 0$ results in everywhere zero canonical and reduced form primal objective functions and the set $\mathcal{A}$ constrains $\mathbf{a}_1 = \mathbf{a}_2$ as desired. Similarly, solving (5.64) for the reduced-form dual components $\widehat{R} \colon \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ and $\mathcal{B} \subseteq \mathbb{R}^2$ yields

$$\widehat{R}\left(\mathbf{b}_1, \mathbf{b}_2\right) \quad = \quad \frac{1}{2}\rho \mathbf{b}_2^2 + \frac{1}{2\rho}(\mathbf{b}_1 + \mathbf{b}_2)^2, \qquad \mathbf{b} \in \mathcal{B} = \mathbb{R}^2. \tag{5.121}$$

Finally, we apply the scattering coordinate transforms associated with the variable $\mathbf{a}_1$ being an input and the variable $\mathbf{a}_2$ being an output of the module under design, i.e. $\mathbf{a}_1$ and $\mathbf{a}_2$ are respectively the outputs and inputs of the interconnecting network, and then solve for the functional realization of the module that produces the transformed vector $\mathbf{c}$ from the transformed vector $\mathbf{c}$. In particular, the coordinate transformation is provided in Figure 5-11 in the center and the resulting function is given by

$$\begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix} \quad = \quad \frac{1}{(1+\rho)^2 + 1} \begin{bmatrix} \rho^2 & -2 \\ 2 & \rho^2 \end{bmatrix} \begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{bmatrix} \tag{5.122}$$

and is depicted graphically on the right of the figure. It is straightforward to verify by substituting (5.122) into the definition of a function being α-conic everywhere that (5.122) is α-dissipative everywhere for $\rho > 0$ and passive everywhere for $\rho = 0$. As the reduced form primal objective function alludes to, the interpretation of the module for non-zero values of the parameter ρ motivates referring to this module as a quadratic coupling module.

Consider a primal optimization problem statement whose aggregate linear constraints are of the form $A\mathbf{x} = \mathbf{y}$. We discuss the partitioning of A first by row and then by column.

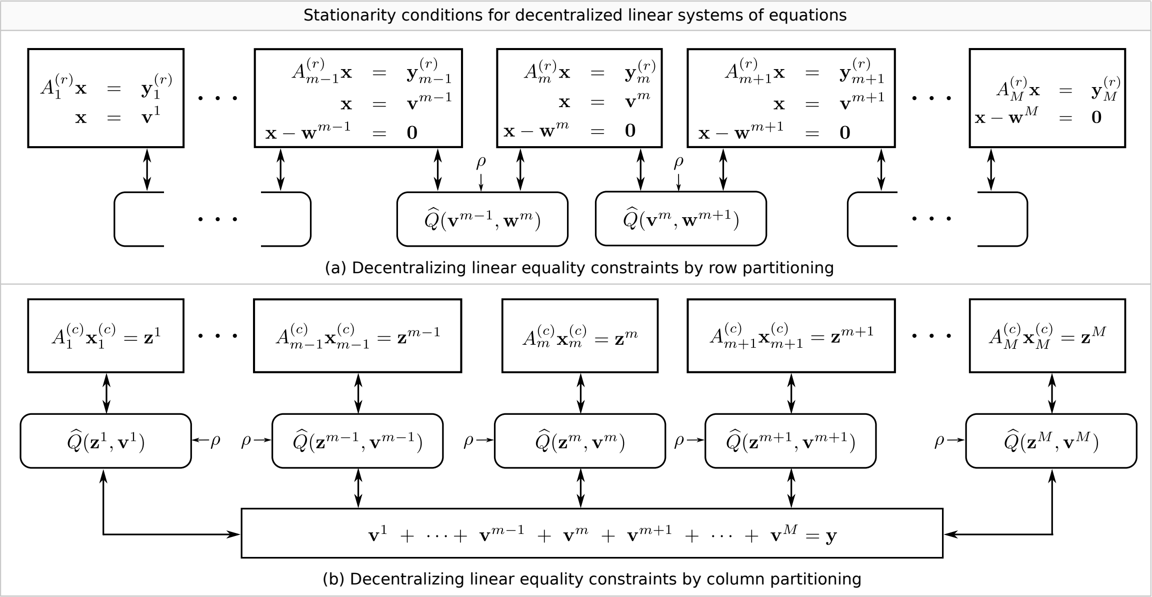
For either case, the general approach is to assign a local copy of some of the variables on neighboring processors and then to use the quadratic coupling module to ensure that the copies on different machines agree in steady state. Figure 5-12(a) depicts the general scheme associated with row partitioning the matrix A into M blocks $A_m^{(r)}$ for $m = 1, \ldots, M$. Referring to the figure, the constraints are distributed according to the notation

$$\begin{aligned} A_m^{(r)}\mathbf{x} &= \mathbf{y}_m^{(r)}, \qquad m = 1, \ldots, M && (5.123) \\ \mathbf{x} &= \mathbf{v}^m \qquad m = 1, \ldots, M-1 && (5.124) \\ \mathbf{x} - \mathbf{w}^m &= \mathbf{0} \qquad m = 2, \ldots, M && (5.125) \end{aligned}$$

where a copy of the vector $\mathbf{x}$ is private on each interconnect, $\mathbf{y}_m^{(r)}$ is the subvector of $\mathbf{y}$ corresponding to the rows of A in $A_m^{(r)}$, and $\mathbf{v}^m$ and $\mathbf{w}^m$ are vector sequences with the same number of entries as $\mathbf{x}$. The quadratic coupling modules labeled $\widehat{Q}(\mathbf{v}^m, \mathbf{w}^{m+1})$ for $m = 1, \ldots, M-1$ are distributed between interconnects as indicated by the figure. Observe that setting $\rho = 0$ enforces the aggregate equality constraints to be met with the same value of $\mathbf{x}$ at each interconnect. Figure 5-12(b) depicts the general scheme associated with column partitioning the matrix A into M blocks $A_m^{(c)}$, for $m = 1, \ldots, M$. Referring to the figure, the constraints are distributed according to the notation

$$\begin{aligned} A_m^{(c)}\mathbf{x}_m^{(c)} &= \mathbf{z}^m, \qquad m = 1, \ldots, M && (5.126) \\ \sum_{m=1}^{M} \mathbf{v}^m &= \mathbf{y} && (5.127) \end{aligned}$$

where $\mathbf{x}_m^{(c)}$ is the subvector of $\mathbf{x}$ corresponding to the columns of A in $A_m^{(c)}$ and $\mathbf{z}^m$ and $\mathbf{v}^m$ are vector sequences with the same number of entries as $\mathbf{y}$. The quadratic coupling modules labeled $\widehat{Q}(\mathbf{z}^m, \mathbf{v}^m)$ for $m = 1, \ldots, M$ are distributed between interconnects as indicated by the figure where an additional interconnect has been used to enforce (5.127). Observe that setting $\rho = 0$ enforces the aggregate equality constraints to be met. Of course, row and column partitionings may be combined by applying the decomposition in (b) to any interconnect in (a) and vice versa. For both partitioning schemes, the overall precompute



Stationarity conditions for decentralized linear systems of equations

$A_1^{(r)}\mathbf{x} = \mathbf{y}_1^{(r)}$
$\mathbf{x} = \mathbf{v}^1$
$A_{m-1}^{(r)}\mathbf{x} = \mathbf{y}_{m-1}^{(r)}$
$\mathbf{x} = \mathbf{v}^{m-1}$
$\mathbf{x} - \mathbf{w}^{m-1} = \mathbf{0}$
$A_m^{(r)}\mathbf{x} = \mathbf{y}_m^{(r)}$
$\mathbf{x} = \mathbf{v}^m$
$\mathbf{x} - \mathbf{w}^m = \mathbf{0}$
$A_{m+1}^{(r)}\mathbf{x} = \mathbf{y}_{m+1}^{(r)}$
$\mathbf{x} = \mathbf{v}^{m+1}$
$\mathbf{x} - \mathbf{w}^{m+1} = \mathbf{0}$
$A_M^{(r)}\mathbf{x} = \mathbf{y}_M^{(r)}$
$\mathbf{x} - \mathbf{w}^M = \mathbf{0}$
ρ
ρ
$\widehat{Q}(\mathbf{v}^{m-1}, \mathbf{w}^m)$
$\widehat{Q}(\mathbf{v}^m, \mathbf{w}^{m+1})$

(a) Decentralizing linear equality constraints by row partitioning

$A_1^{(c)}\mathbf{x}_1^{(c)} = \mathbf{z}^1$
$A_{m-1}^{(c)}\mathbf{x}_{m-1}^{(c)} = \mathbf{z}^{m-1}$
$A_m^{(c)}\mathbf{x}_m^{(c)} = \mathbf{z}^m$
$A_{m+1}^{(c)}\mathbf{x}_{m+1}^{(c)} = \mathbf{z}^{m+1}$
$A_M^{(c)}\mathbf{x}_M^{(c)} = \mathbf{z}^M$
$\widehat{Q}(\mathbf{z}^1, \mathbf{v}^1)$
ρ
ρ
$\widehat{Q}(\mathbf{z}^{m-1}, \mathbf{v}^{m-1})$
ρ
$\widehat{Q}(\mathbf{z}^m, \mathbf{v}^m)$
ρ
$\widehat{Q}(\mathbf{z}^{m+1}, \mathbf{v}^{m+1})$
ρ
$\widehat{Q}(\mathbf{z}^M, \mathbf{v}^M)$
$\mathbf{v}^1 + \cdots + \mathbf{v}^{m-1} + \mathbf{v}^m + \mathbf{v}^{m+1} + \cdots + \mathbf{v}^M = \mathbf{y}$

(b) Decentralizing linear equality constraints by column partitioning

Figure 5-12: An illustration of the use of the quadratic coupling module in decentralizing linear systems of equations. (a) By row partitioning a matrix. (b) By column partitioning a matrix.

associated with realizing the interconnect for any submatrix of A according to (5.91) may be significantly reduced using this scheme since the Cayley transform of a much smaller matrix is required. The overall precompute associated with generating the Cayley transform of the bottom interconnect in (b) requires very little computation since the linear system corresponds to concatenated identity matrices.

Consider the scattering-form structure produced by decentralizing A using the quadratic coupling module defined above and then performing the scattering coordinate transform. The associated continuation scheme refers to dynamically changing the realization of each quadratic coupling module during runtime by incrementally decreasing ρ to zero in (5.122). From the stability and robustness results in Chapter 4, when the other constitutive modules in the graph are dissipative everywhere, the transformed variables converge linearly to a fixed-point if $\rho > 0$. However, this fixed-point corresponds to a primal objective function consisting of the original cost function in addition to the sum of the quadratic coupling penalties described by (5.120). By setting $\rho = 0$, a fixed-point of the system corresponds to a solution to the original problem. Allowing the system to initially process with small ρ provides fast convergence to a neighborhood near the desired solution where sublinear convergence is guaranteed for $\rho = 0$ when the delay modules are appropriately filtered.

5.5 | Early termination of scattering algorithms

So far in this chapter, we have defined the class of conservative optimization problems and developed a modular approach to designing scattering algorithms to solve them. In particular, the approach involves assembling and implementing a signal processing system in scattering-form whose fixed-points correspond to solutions to the transformed stationarity conditions associated with the primal and dual optimization problems at hand. As such, questions pertaining to the convergence of the scattering algorithms are immediately resolved using the stability and robustness conditions derived in Section 4.3 for general implementations of interconnective structures in the context of solving CCSPs. In this section, we discuss the relationship between the intermediary values taken by the scattering algorithm's variables and the values taken by the corresponding decision variables as the algorithm converges to a solution. Said another way, the focus in this section is on interpreting the partial solution obtained when a scattering algorithm is terminated early.

To illustrate the relationships mentioned above in the context of an example, we continue with the non-negative least squares problem in (5.99). Referring to the canonical-form description of the stationarity conditions in Figure 5-5(a), the constitutive modules are assigned as follows:

module 1: the second row of Figure 5-6 with scattering transform $M^{(i)}$,

module 2: the first row of Figure 5-7 with scattering transform $M^{(i)}$,

module 3: the third row of Figure 5-8 with scattering transform $M^{(o)}$.

With these assignments, the scattering algorithm corresponds to asynchronously implementing the listed realizations after delays have been inserted between the constitutive modules and interconnecting networks. Results from implementing this structure for an instantiation of the problem are provided in Figure 5-13 where the asynchronous delay parameter p was chosen to be 1 to remove the influence of the stochastic dynamics of the system from the present discussion, i.e. the scattering algorithm was implemented synchronously. Referring to the figure, the sequences of scattering vectors $\mathbf{c}^n$ and $\mathbf{d}^n$ denote the values the algorithm

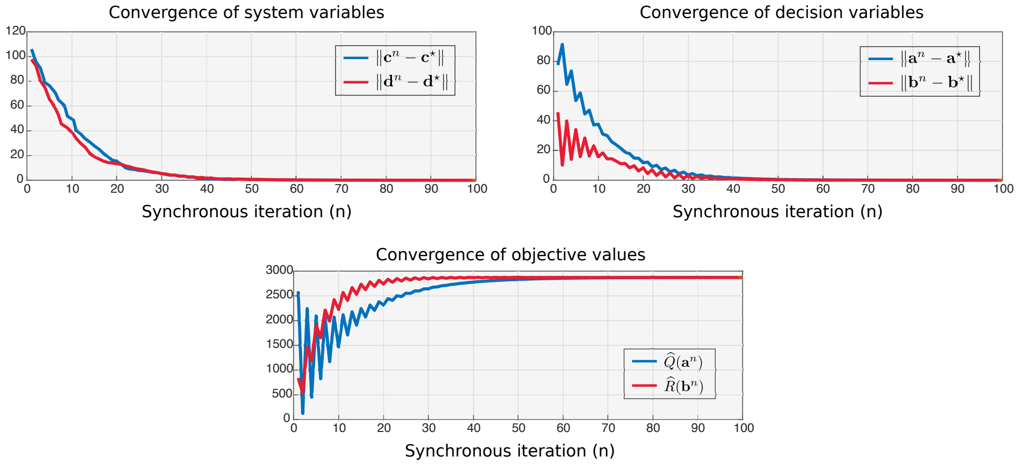


Figure 5-13: Illustration of the stability for a synchronous algorithm for solving a non-negative least squares problem demonstrating the behavior of the reduced-form primal and dual objective functions as well as the canonical and transformed system variables.

variables take at each iteration and the sequences $\mathbf{a}^n$ and $\mathbf{b}^n$ of canonical vectors denote the corresponding values taken by the primal and dual decision vectors. Observe that $\mathbf{c}^n$ and $\mathbf{d}^n$ monotonically converge to their respective fixed-points, as expected.

The first observation we call attention to is that monotonicity of the scattering variables to their respective fixed-points does not necessarily imply monotonicity in any form of the decision variables to theirs. This observation is demonstrated by the trends in the top two plots in Figure 5-13. For the non-negative least squares problem, the modules associated with the primal decision variables $\mathbf{a}_2^{(CR)}$ and $\mathbf{a}_3^{(CR)}$ are sources and therefore can be algebraically eliminated using (5.95) through (5.96). Doing so results in an implementation with only scattering variables corresponding to the first module. Monotonicity of the decision variables and the primal and dual cost values for such source-free structures is also not guaranteed.

The second observation we call attention to is that monotonicity of the scattering variables does not necessarily imply monotonicity of the canonical-form or reduced-form primal and dual cost functions. This observation is demonstrated by the trends in the bottommost plot in Figure 5-13. This feature is in contrast with steepest-descent like algorithms where the primal cost function is non-increasing with the iteration count for appropriately chosen step sizes. Furthermore, the scalar-valued sequences $\widehat{Q}(\mathbf{a}^n)$ and $\widehat{R}(\mathbf{b}^n)$ do not necessarily upper or lower bound one another as a function of n, as may be found with primal-dual algorithms relying on Lagrangian duality. Similar to these methods, the fact that $\widehat{Q}(\mathbf{a}^n)$ can take values lower than $\widehat{Q}(\mathbf{a}^{\star})$ is due to the strong relationship imposed between feasibility and stationarity. Specifically, the duality principle invoked here ensures that if $\widehat{Q}(\mathbf{a}^{n})$ is less than $\widehat{Q}(\mathbf{a}^{\star})$, then $\mathbf{a}^{n}$ must be infeasible, i.e. does not satisfy the primal feasibility conditions.

5.6 | Connections with existing optimization methods

In this section, connections between the class of scattering algorithms developed in this chapter and existing algorithm classes in the optimization literature are presented. These connections are specifically made between scattering algorithms and a variety of steepest-descent and proximal methods, including the stochastic gradient descent and ADMM algorithms discussed in Section 2.5.2. In particular, connections are made by identifying correspondences between the stationarity conditions each algorithm class is designed to solve, and we focus on this basis for comparison since it allows for a large number of specific algorithms belonging to these classes to be handled at once. The key result in this section is that the interconnective description of steepest-descent, proximal, and scattering algorithms all belong to the same interconnective equivalence class. Therefore, these algorithms can all be generated using the modular framework developed in this chapter by starting with the same interconnective description of the stationarity conditions associated with a problem, where the different algorithms correspond to applying different coordinate transforms before inserting memory to produce well-defined iterations.

5.6.1 | Interconnective description of reduced-form stationarity conditions

The purpose of this subsection is to describe the notation that will be used to make the connections discussed above. In casting the stationarity conditions associated with common optimization problems as fixed-point problems, recall that the interconnective description of the reduced-form stationarity conditions corresponds to the coupling of an interconnecting network and constitutive modules where the interconnecting network describes the linear feasibility conditions and the constitutive modules characterize the cost functions and nonlinear set-valued feasibility constraints. In particular, the constitutive modules are described through the set constraint $\mathcal{F}$ acting on the entire collection of primal and dual decision variables $\mathbf{a}$ and $\mathbf{b}$, where the direct product decomposition $\mathcal{F} = \mathcal{F}_1 \times \cdots \times \mathcal{F}_K$ corresponds to defining individual constitutive modules for each separable term in the cost function, i.e. the constraint $\mathcal{F}_k$ couples the variables $(\mathbf{a}_k^{(CR)}, \mathbf{b}_k^{(CR)})$ for $k = 1, \ldots, K$. From a given optimization problem written in reduced-form, the set constraint $\mathcal{F}_k$ is defined using an intermediary functional $p_k \colon \mathbb{R}^{N_k^{(CR)}} \to \mathbb{R}$ that is equal to the cost term $\widehat{Q}_k$ associated with the variables $\mathbf{a}_k^{(CR)}$ over the feasible set $\mathcal{A}_k$ and is equal to infinity elsewhere. For example, if $\mathbf{a}_k^{(CR)}$ is constrained to the set $\mathcal{A}_k \subseteq \mathbb{R}^{N_k^{(CR)}}$ with cost function $\widehat{Q}_k(\mathbf{a}_k^{(CR)})$, then the functional p_k is given by

$$p_k(\mathbf{a}_k^{(CR)}) \quad = \quad \begin{cases} \widehat{Q}_k(\mathbf{a}_k^{(CR)}), & \mathbf{a}_k^{(CR)} \in \mathcal{A}_k \\ \infty, & \text{otherwise} \end{cases} . \tag{5.128}$$

For this setup, the individual constitutive modules $\mathcal{F}_k$ are generated using the intermediary functions p_k according to

$$\mathcal{F}_k \quad = \quad \left\{ \begin{bmatrix} \mathbf{a}_k^{(CR)} \\ \mathbf{b}_k^{(CR)} \end{bmatrix} \in \mathbb{R}^{2N_k^{(CR)}} : \mathbf{b}_k^{(CR)} = \partial p_k(\mathbf{a}_k^{(CR)}) \right\}, \qquad k = 1, \ldots, K. \tag{5.129}$$

The stationarity conditions are completed by writing the linear feasibility conditions as a vector space constraint in exactly the same way as presented in Section 5.1.

5.6.2 | Steepest descent algorithms

Steepest descent algorithms form a common class of first-order optimization methods for finding local and global extrema of smooth functions, and generally work by iteratively moving in directions loosely aligned with the negative of the gradient of the function. For example, consider the problem of minimizing an unconstrained, convex, differential functional $Q \colon \mathbb{R}^N \to \mathbb{R}$ formulated as

$$\underset{\mathbf{x}}{\text{minimize}} \quad Q(\mathbf{x}) \ . \tag{5.130}$$

The standard gradient-descent algorithm associated with solving the first-order optimality condition $\nabla Q(\mathbf{x}) = \mathbf{0}$ is given by

$$\mathbf{x}^{n+1} \;=\; \mathbf{x}^n - \rho \nabla Q(\mathbf{x}^n), \qquad n \in \mathbb{N} \tag{5.131}$$

where ρ is a step size parameter and can be selected to ensure convergence of the sequence $\mathbf{x}^n$ to a solution $\mathbf{x}^\star$. Toward making connections between this iteration and the scattering algorithms in this thesis, we proceed by recasting (5.130) into a form that resembles a conservative optimization problem according to

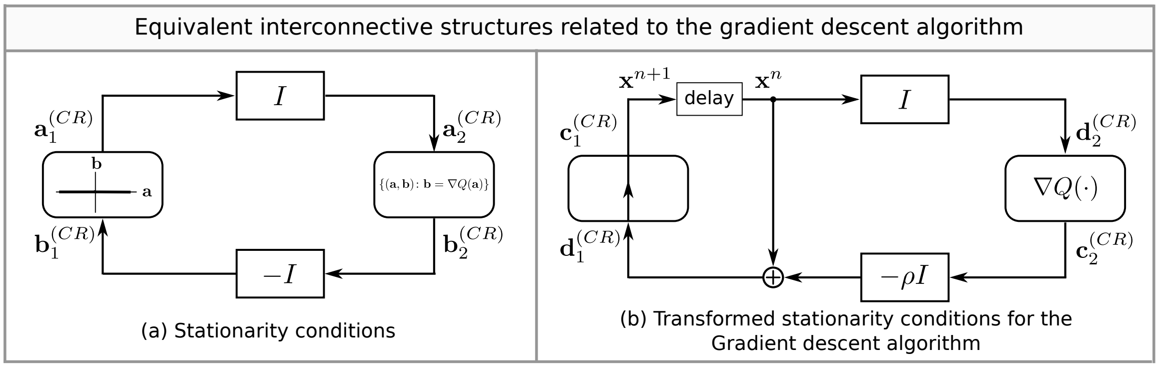
$$\begin{aligned} \underset{\mathbf{x},\,\mathbf{z}}{\text{minimize}} \quad & Q(\mathbf{z}) \\ \text{s.t.} \quad & \mathbf{x} = \mathbf{z}. \end{aligned} \tag{5.132}$$

The purpose of reformulating unconstrained problems as constrained problems is to assist with defining the portion of the interconnective description of the stationarity conditions associated with the interconnecting network, and the choice between assigning the cost function to $\mathbf{x}$ or $\mathbf{z}$ in (5.132) is non-essential. In this case, the formulations in (5.130) and (5.132) are clearly equivalent to one another in the sense that their solution sets are equal.

An interconnective description of the stationarity conditions associated with the constrained problem (5.132) is provided in Figure 5-14(a). The interconnecting network enforces the constraints $\mathbf{a}_1^{(CR)} - \mathbf{a}_2^{(CR)} = \mathbf{0}$ and $\mathbf{b}_1^{(CR)} + \mathbf{b}_2^{(CR)} = \mathbf{0}$, which we recast as the vector space constraint

$$\begin{bmatrix} \mathbf{a}_1^{(CR)} \\ \mathbf{b}_1^{(CR)} \\ \mathbf{a}_2^{(CR)} \\ \mathbf{b}_2^{(CR)} \end{bmatrix} \;\in\; \text{range}\left(\begin{bmatrix} I & 0 \\ 0 & -I \\ I & 0 \\ 0 & I \end{bmatrix} \right). \tag{5.133}$$

The constitutive modules $\mathcal{F}_1$ and $\mathcal{F}_2$ can be defined using the general approach described



(a) Stationarity conditions

(b) Transformed stationarity conditions for the Gradient descent algorithm

Figure 5-14: Interconnective structures illustrating the coordinate transform used to obtain gradient-descent algorithms from the stationarity conditions associated with (5.132).

in Section 5.6.1 where the relevant intermediary functionals are given by

$$p_1(\mathbf{a}_1^{(CR)}) \quad = \quad 0 \tag{5.134}$$

$$p_2(\mathbf{a}_2^{(CR)}) \quad = \quad Q(\mathbf{a}_2^{(CR)}). \tag{5.135}$$

Note that the constitutive module $\mathcal{F}_1$ formed using (5.134) corresponds to the module in the second row of Figure 5-7.

The scattering algorithm for (5.132) is formed by applying the scattering coordinate transform to the interconnective description of the problem's stationarity conditions and then inserting memory into the resulting structure. To connect the scattering algorithm to the gradient-descent algorithm, we next perform a coordinate transform to the interconnective description of the stationarity conditions associated with the problem (5.132) to obtain an interconnective structure resembling the gradient-descent algorithm. In particular, the gradient-descent coordinate transform M_{GD} is provided in the following expression:

$$\begin{bmatrix} \mathbf{c}_1^{(CR)} \\ \mathbf{d}_1^{(CR)} \\ \mathbf{c}_2^{(CR)} \\ \mathbf{d}_2^{(CR)} \end{bmatrix} = \underbrace{\begin{bmatrix} I & 0 & 0 & 0 \\ I & \rho I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix}}_{\triangleq M_{GD}} \begin{bmatrix} \mathbf{a}_1^{(CR)} \\ \mathbf{b}_1^{(CR)} \\ \mathbf{a}_2^{(CR)} \\ \mathbf{b}_2^{(CR)} \end{bmatrix}. \tag{5.136}$$

The interconnective structure corresponding to this coordinate transform is depicted in

Figure 5-14(b), from which the iteration in (5.131) can be obtained by assigning the variable $\mathbf{x} = \mathbf{c}_1^{(CR)}$ and inserting a delay module as illustrated in the figure. To see this more clearly, we first transform the interconnecting network associated with (5.132) by the coordinate matrix M_{GD} as

$$\begin{bmatrix} \mathbf{c}_1^{(CR)} \\ \mathbf{c}_1^{(CR)} \\ \mathbf{d}_1^{(CR)} \\ \mathbf{d}_2^{(CR)} \end{bmatrix} \in \text{range}\left(\underbrace{\begin{bmatrix} I & 0 & 0 & 0 \\ 0 & 0 & 0 & I \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \end{bmatrix}}_{\text{input-output config.}} \underbrace{\begin{bmatrix} I & 0 & 0 & 0 \\ I & \rho I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix}}_{M_{GD}} \underbrace{\begin{bmatrix} I & 0 \\ 0 & -I \\ I & 0 \\ 0 & I \end{bmatrix}}_{W \text{ for (5.132)}} \right) \tag{5.137}$$

$$\in \text{range}\left(\begin{bmatrix} I & 0 \\ 0 & I \\ I & -\rho I \\ I & 0 \end{bmatrix} \right) \tag{5.138}$$

where the input-output configuration has been appropriately selected to be consistent with the standard notation of the **c** and **d** variables respectively denoting inputs and outputs of the interconnecting network. Reflecting the coordinate transform onto the constitutive modules yields functional realizations of the modules given by

$$\mathbf{c}_1^{(CR)} = \mathbf{d}_1^{(CR)} \tag{5.139}$$

$$\mathbf{c}_2^{(CR)} = \nabla Q(\mathbf{d}_2^{(CR)}), \tag{5.140}$$

and completes the derivation of the transformed interconnective structure used to define (5.131) as desired. The action of the coordinate transform M_{GD} on the first constitutive module is not arbitrary. In fact, this particular transform is shown in the next section to be closely related to the proximal operator associated with an indicator function defined on the dual decision vector $\mathbf{b}_1^{(CR)}$. The action of the coordinate transform M_{GD} on the second constitutive module is to simply rearrange the system variables to use the gradient operator $\nabla Q(\cdot)$ in a way that is consistent with the chosen input-output configuration. The stochastic gradient descent algorithm discussed in Section 2.5.2 corresponds to implementing the interconnective structure in Figure 5-14(b) using an asynchronous delay module.

A variety of steepest descent methods that are closely related to the gradient-descent iteration have been designed to handle various modifications of the problem (5.130) where the feasible set includes linear equality and inequality constraints [98]. For example, consider the modified problem

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & Q(\mathbf{x}) \\ \text{s.t.} \quad & A\mathbf{x} \leq \mathbf{b}. \end{aligned} \tag{5.141}$$

The class of projected gradient-descent methods can be used to solve problems of this form, and these methods can be derived using the same procedure as the gradient-descent iteration for unconstrained problems with the following modifications. The variable assignment for the interconnective description of the stationarity conditions becomes $(\mathbf{a}_1^{(CR)}, \mathbf{a}_2^{(CR)}) = (\mathbf{x}, A\mathbf{x})$ and the constitutive modules are formed using the intermediary functionals

$$p_1(\mathbf{a}_1^{(CR)}) = Q(\mathbf{a}_1^{(CR)}) \tag{5.142}$$

$$p_2(\mathbf{a}_2^{(CR)}) = \begin{cases} 0, & \mathbf{a}_2^{(CR)} \leq \mathbf{b} \\ \infty, & \text{otherwise} \end{cases}. \tag{5.143}$$

The interconnecting network is modified to incorporate the matrix A according to

$$\begin{bmatrix} \mathbf{a}_1^{(CR)} \\ \mathbf{b}_1^{(CR)} \\ \mathbf{a}_2^{(CR)} \\ \mathbf{b}_2^{(CR)} \end{bmatrix} \in \text{range}\left(\begin{bmatrix} I & 0 \\ 0 & -A^T \\ A & 0 \\ 0 & I \end{bmatrix} \right) \tag{5.144}$$

and the coordinate transform and input-output configuration are chosen to preserve the first module and apply the proximal transform defined in the next subsection to the second module, similar to how the first module in the unconstrained gradient-descent algorithm was transformed to obtain the iteration in (5.131).

5.6.3 | Proximal algorithms

Proximal methods are commonly used for solving non-smooth, convex optimization problems where gradient-descent methods may not apply due to the lack of a well-defined derivative of the cost function. Proximal algorithms are generally distinct from their gradient-based counterparts in the sense that they iterate proximal operators rather than, for example, derivatives or subgradients [46]. This feature is similar to the scattering algorithms established in this thesis where the functions composing the transformed stationarity conditions are often related to subderivatives through scattering coordinate transformations.

Mathematically, a proximal algorithm for minimizing a convex function $Q\colon \mathbb{R}^N \rightarrow \mathbb{R} \cup \{\infty\}$ takes the general form

$$\mathbf{x}^{n+1} \quad = \quad \operatorname{prox}_{\rho Q}(\mathbf{x}^n), \qquad n \in \mathbb{N} \tag{5.145}$$

where ρ is the so-called scaling parameter, Q is allowed to be non-differentiable and take values of infinity, and the scaled proximal operator $\operatorname{prox}_{\rho Q}\colon \mathbb{R}^N \rightarrow \mathbb{R}^N$ is an operator defined according to the expression

$$\operatorname{prox}_{\rho Q}(\mathbf{x}) \quad \triangleq \quad \arg\min_{\mathbf{v}} Q(\mathbf{v}) + \frac{1}{2\rho}\|\mathbf{v}-\mathbf{x}\|^2. \tag{5.146}$$

By allowing Q to take values of infinity, it is straightforward to express equality and inequality constraints in the cost function. Despite the fact that evaluating the definition of a proximal operator involves solving an optimization problem involving the original cost function to be minimized, closed-form expressions for many functions Q commonly used in practice are known [99]. Convergence analysis for proximal algorithms often relies on properties of monotone and firmly non-expansive operators in a similar way to how the stability of a system operator in this thesis relies on α-connicitiy and passivity properties [47].

To begin, we first show that the scaled proximal operator defined by (5.146) corresponds to a functional realization of the transformed behavior of the gradient or subderivative of the function Q. This is analogous to the characterization of scattering operators in Section 5.2.3 using reduced-form, convex cost functions. In particular, the scaled proximal

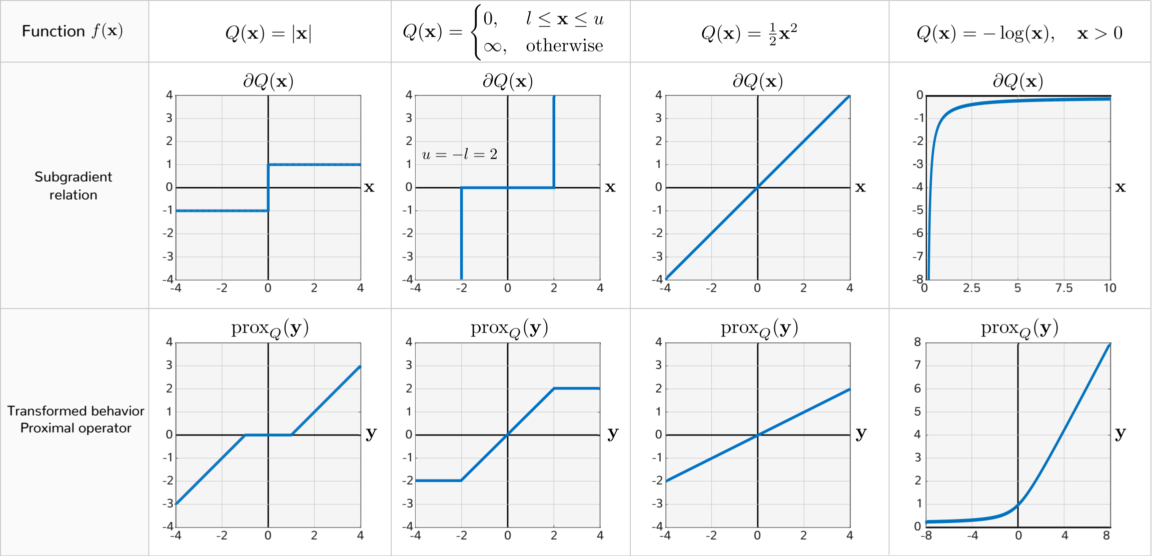


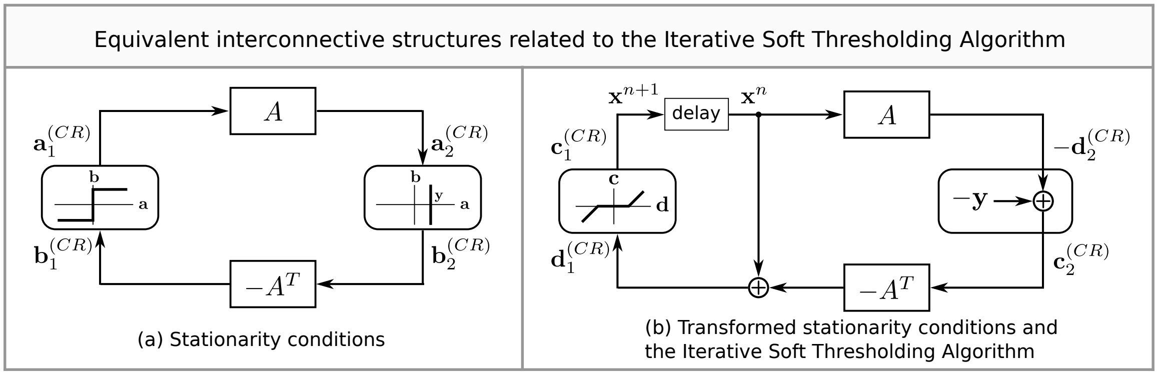
Figure 5-15: Transformed behaviors illustrating the relationship between subderivatives and proximal operators using the proximal coordinate transform in (5.147).

operator $\text{prox}_{\rho Q}(\cdot)$ is related to the (sub)differential relation ∂Q according to

$$\begin{bmatrix} \text{prox}_{\rho Q}(\mathbf{y}) \\ \mathbf{y} \end{bmatrix} = \underbrace{\begin{bmatrix} I & 0 \\ I & \rho I \end{bmatrix}}_{\triangleq M_{prox}} \begin{bmatrix} \mathbf{x} \\ \partial Q(\mathbf{x}) \end{bmatrix}. \tag{5.147}$$

Proving that this relationship holds requires the relationship $\mathbf{y} = (I + \rho\partial Q)(\text{prox}_{\rho Q}(\mathbf{y}))$ to be true. To show this, let $\mathbf{z}^\star = \text{prox}_{\rho Q}(\mathbf{y})$ and consider the function $g(\mathbf{z}) = Q(\mathbf{z}) + \frac{1}{2\rho}\|\mathbf{z} - \mathbf{y}\|^2$. The condition $\partial g(\mathbf{z}^\star) = 0$ then yields the constraint $\rho\partial Q(\mathbf{z}^\star) + \mathbf{z}^\star = \mathbf{y}$ which is precisely the relationship in (5.147). We discuss this relationship further using an example next.

Four example scalar-valued proximal operators demonstrating the relationship (5.147) with $\rho = 1$ are provided in Figure 5-15. The subdifferential relation in the first example is specifically the special case of the relationship between the primal and dual decision variables for the scaled absolute value cost function in the second column of the first row of Figure 5-8 with the choices $\mathbf{a} = \mathbf{x}$ and $\mathbf{b} = \partial Q(\mathbf{x})$ and with $\rho = 1$. Recall that this relationship coincides with the subdifferential relation corresponding to the absolute value function given in (2-1), also with $\rho = 1$. The functional realization of the scaled proximal



(a) Stationarity conditions

(b) Transformed stationarity conditions and the Iterative Soft Thresholding Algorithm

Figure 5-16: Interconnective structures illustrating the coordinate transform used to obtain the Iterative Soft Thresholding Algorithm from the stationarity conditions associated with (5.149).

operator for this example is the so-called soft thresholding operator and is given by

$$\mathrm{prox}_{\rho|\cdot|}(\mathbf{x}) \quad = \quad \begin{cases} 0, & |\mathbf{x}| \leq \rho \\ \mathbf{x} - \rho\mathrm{sgn}(\mathbf{x}), & |\mathbf{x}| > \rho \end{cases}. \tag{5.148}$$

The soft-thresholding operator is also depicted in the third column of Figure 2-1. The relationship in (5.147) suggests one way that proximal operators can be generated numerically for a given function Q and forms the key observation used to relate proximal algorithms with the class scattering algorithms discussed in this chapter.

To illustrate the difference between the coordinate transforms used to produce proximal algorithms and the scattering coordinate transforms used to produce scattering algorithms, consider the example of a non-smooth optimization problem given by

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \|\mathbf{x}\|_1 \\ \text{s.t.} \quad & A\mathbf{x} = \mathbf{y}. \end{aligned} \tag{5.149}$$

The standard proximal algorithm associated with solving this problem is the so-called iterative soft thresholding algorithm (ISTA) and corresponds to the iteration [100]

$$\mathbf{x}^{n+1} \quad = \quad \mathrm{prox}_{\rho|\cdot|}(\mathbf{x}^n + \rho A^T(\mathbf{y} - A\mathbf{x}^n)), \qquad n \in \mathbb{N}_0 \tag{5.150}$$

where ρ serves to assist with the stability of the iteration.

An interconnective description of the stationarity conditions associated with (5.149) is provided in Figure 5-16(a). The formal description of this problem as a conservative optimization problem and the scattering algorithm produced by appropriately transforming the problem's stationarity conditions is presented in Section 6.1.2. The intent in this subsection is to connect the proximal algorithm for solving this problem to the problem's stationarity conditions, and to do so using interconnective equivalence classes. Referring to Figure 5-16(a), the interconnecting network enforces the constraints $A\mathbf{a}_1^{(CR)} - \mathbf{a}_2^{(CR)} = \mathbf{0}$ and $\mathbf{b}_1^{(CR)} + A^T\mathbf{b}_2^{(CR)} = \mathbf{0}$, which we recast as the vector space constraint

$$\begin{bmatrix} \mathbf{a}_1^{(CR)} \\ \mathbf{b}_1^{(CR)} \\ \mathbf{a}_2^{(CR)} \\ \mathbf{b}_2^{(CR)} \end{bmatrix} \in \text{range}\left(\begin{bmatrix} I & 0 \\ 0 & -A^T \\ A & 0 \\ 0 & I \end{bmatrix}\right). \tag{5.151}$$

The constitutive modules $\mathcal{F}_1$ and $\mathcal{F}_2$ can be defined using the general approach described in Section 5.6.1 where the relevant intermediary functionals are given by

$$p_1(\mathbf{a}_1^{(CR)}) = \|\mathbf{a}_1^{(CR)}\|_1 \tag{5.152}$$

$$p_2(\mathbf{a}_2^{(CR)}) = \begin{cases} 0, & \mathbf{a}_2^{(CR)} = \mathbf{y} \\ \infty, & \text{otherwise} \end{cases}. \tag{5.153}$$

We next perform a proximal coordinate transform to the interconnective description of the stationarity conditions associated with the problem (5.149) to obtain an interconnective structure resembling the ISTA algorithm. In particular, we apply M_{prox} to the first constitutive module and a similar characterization of the proximal coordinate transform to the second constitutive module. This coordinate transform is described by

$$\begin{bmatrix} \mathbf{c}_1^{(CR)} \\ \mathbf{d}_1^{(CR)} \\ \mathbf{c}_2^{(CR)} \\ \mathbf{d}_2^{(CR)} \end{bmatrix} = \begin{bmatrix} I & 0 & 0 & 0 \\ I & \rho I & 0 & 0 \\ 0 & 0 & -I & 0 \\ 0 & 0 & I & I \end{bmatrix} \begin{bmatrix} \mathbf{a}_1^{(CR)} \\ \mathbf{b}_1^{(CR)} \\ \mathbf{a}_2^{(CR)} \\ \mathbf{b}_2^{(CR)} \end{bmatrix}. \tag{5.154}$$

The result of applying this coordinate transform is depicted in Figure 5-16(b) from which the iteration in (5.150) can be obtained by assigning the variable $\mathbf{x} = \mathbf{c}_1^{(CR)}$ and inserting a delay module as illustrated in the figure. To see this, we apply the coordinate transform (5.154) to the interconnecting network according to

$$\begin{bmatrix} \mathbf{c}_1^{(CR)} \\ \mathbf{c}_2^{(CR)} \\ \mathbf{d}_1^{(CR)} \\ \mathbf{d}_2^{(CR)} \end{bmatrix} \in \text{range}\left(\underbrace{\begin{bmatrix} I & 0 & 0 & 0 \\ 0 & 0 & 0 & I \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \end{bmatrix}}_{\text{input-output config.}} \underbrace{\begin{bmatrix} I & 0 & 0 & 0 \\ I & \rho I & 0 & 0 \\ 0 & 0 & -I & 0 \\ 0 & 0 & I & I \end{bmatrix}}_{M \text{ in (5.154)}} \underbrace{\begin{bmatrix} I & 0 \\ 0 & -A^T \\ A & 0 \\ 0 & I \end{bmatrix}}_{W \text{ for (5.149)}} \right) \tag{5.155}$$

$$\in \quad \text{range}\left(\begin{bmatrix} I & 0 \\ A & I \\ I & -\rho A^T \\ -A & 0 \end{bmatrix} \right) \tag{5.156}$$

where the input-output configuration has been appropriately selected to be consistent with the standard notation of the $\mathbf{c}$ and $\mathbf{d}$ variables respectively denoting inputs and outputs of the interconnecting network. Reflecting the coordinate transform onto the constitutive modules yields a functional realization of the first constitutive module given by (5.148).

Observe that the scattering decision vector $\mathbf{d}_1^{(CR)}$ produced by the proximal coordinate transform in (5.154) and the scattering coordinate transform in (3.90) are the same. Therefore, the argument presented in Section 5.2.3 showing that the realization of transformed constitutive modules for convex cost functions is well-defined also holds for the realization of the proximal function generated according to M_{prox} when the cost function Q is convex. The treatment in this subsection used to obtain the proximal algorithm for (5.149) did not use anything specific to the 1-norm cost function. We end this subsection by commenting that, as a result, general proximal algorithms can also be produced using this coordinate transform and will also be in the same interconnective equivalence class as the scattering algorithm produced by coupling together the modules in Section 5.3.

5.6.4 | Alternating Direction Method of Multipliers

A common class of optimization problems and an associated class of proximal methods were introduced in Section 2.5.2, where problems in the general form of (2.37) are solved using various instantiations of the ADMM iteration in (2.39). A key feature of the ADMM algorithm is the ability to jointly optimize over multiple non-smooth objective functions while maintaining the computational advantages of Gauss-Seidel like iterations. To elaborate on this further, consider the class of optimization problems that can be formulated as

$$\underset{\mathbf{x}}{\text{minimize}} \quad f(\mathbf{x}) + g(\mathbf{x}) \tag{5.157}$$

where the cost functions $f\colon \mathbb{R}^N \to \mathbb{R} \cup \{\infty\}$ and $g\colon \mathbb{R}^N \to \mathbb{R} \cup \{\infty\}$ are convex and allowed to take values of infinity. This problem setup is equivalent to the class of problems described by (2.37) since, for example, selecting g to take the form in the second column of Figure 5-15 is sufficient to enforce equality and inequality constraints. The ADMM iteration used to solve problems in the present class notationally simplifies from the form of the iteration presented in (2.39) by using proximal operators to the iteration

$$\mathbf{x}^{n+1} = \operatorname{prox}_{\rho f}(\mathbf{z}^n - \boldsymbol{\pi}^n) \tag{5.158}$$

$$\mathbf{z}^{n+1} = \operatorname{prox}_{\rho g}(\mathbf{x}^{n+1} + \boldsymbol{\pi}^n) \tag{5.159}$$

$$\boldsymbol{\pi}^{n+1} = \boldsymbol{\pi}^n + \rho(\mathbf{x}^{n+1} - \mathbf{z}^{n+1}). \tag{5.160}$$

The quadratic penalty term in the definition of a proximal operator naturally accounts for the augmented Lagrangian that the ADMM algorithm is derived from. In the remainder of this subsection, we show that the ADMM algorithm above can be generated by transforming the stationarity conditions associated with a reformulation of (5.157) and then inserting memory into the resulting structure to break delay-free loops. In this sense, the key result in this subsection is that the ADMM algorithm is in the same interconnective equivalence class as the scattering algorithm for the problem where the ADMM coordinate transform corresponds to selecting a coordinate transform that is not $\mathcal{D}_p$-invariant.

Toward making connections between the iteration in (5.158) through (5.160) and the scattering algorithms in this thesis, we proceed by recasting (5.157) into a form that resembles a conservative optimization problem according to

$$\begin{aligned} \underset{\mathbf{x},\mathbf{z}}{\text{minimize}} \quad & f(\mathbf{x}) + g(\mathbf{z}) \\ \text{s.t.} \quad & \mathbf{x} - \mathbf{z} = \mathbf{0} \end{aligned} \tag{5.161}$$

where a solution to either (5.157) or (5.161) clearly solves both problems. An interconnective description of the stationarity conditions associated with (5.161) is provided in Figure 5-17(a) where the variables $\mathbf{a} = (\mathbf{a}_1^{(CR)}, \mathbf{a}_2^{(CR)}, \mathbf{a}_3^{(CR)})$ correspond to $(\mathbf{x}, \mathbf{0}, \mathbf{z})$. The interconnecting network enforces the constraints $\mathbf{a}_1^{(CR)} - \mathbf{a}_2^{(CR)} - \mathbf{a}_2^{(CR)} = \mathbf{0}$, $\mathbf{b}_1^{(CR)} + \mathbf{b}_2^{(CR)} = \mathbf{0}$ and $\mathbf{b}_2^{(CR)} - \mathbf{b}_3^{(CR)} = \mathbf{0}$, which we recast as the vector space constraint

$$\begin{bmatrix} \mathbf{a}_1^{(CR)} \\ \mathbf{b}_1^{(CR)} \\ \mathbf{a}_2^{(CR)} \\ \mathbf{b}_2^{(CR)} \\ \mathbf{a}_3^{(CR)} \\ \mathbf{b}_3^{(CR)} \end{bmatrix} \in \text{range}\left(\begin{bmatrix} I & 0 & 0 \\ 0 & -I & 0 \\ I & 0 & -I \\ 0 & I & 0 \\ 0 & 0 & I \\ 0 & I & 0 \end{bmatrix} \right). \tag{5.162}$$

The constitutive modules can be formed by using the general approach described in the introduction to this section where the relevant intermediary functionals are given by

$$p_1(\mathbf{a}_1^{(CR)}) = f(\mathbf{a}_1^{(CR)}) \tag{5.163}$$

$$p_2(\mathbf{a}_2^{(CR)}) = \begin{cases} 0, & \mathbf{a}_2^{(CR)} = \mathbf{0} \\ \infty, & \text{otherwise} \end{cases} \tag{5.164}$$

$$p_3(\mathbf{a}_3^{(CR)}) = g(\mathbf{a}_3^{(CR)}). \tag{5.165}$$

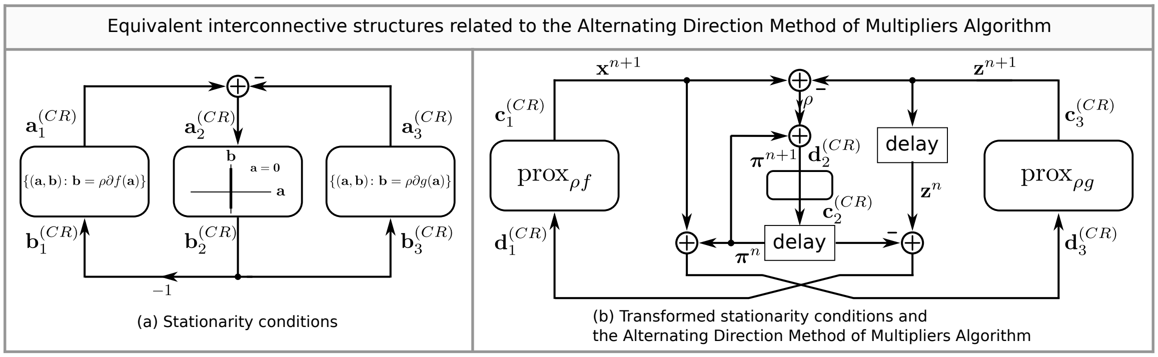
The scattering algorithm for (5.161) is formed by applying the scattering coordinate transform to the interconnective description of the problem's stationarity conditions and then inserting memory into the resulting structure. To connect the scattering algorithm to the ADMM algorithm, we next perform a different coordinate transform to produce an interconnective structure resembling the ADMM iteration. In particular, the coordinate transform is described by the coordinate matrix M_{ADMM} in the expression

$$\begin{bmatrix} \mathbf{c}_1^{(CR)} \\ \mathbf{d}_1^{(CR)} \\ \mathbf{c}_2^{(CR)} \\ \mathbf{d}_2^{(CR)} \\ \mathbf{c}_3^{(CR)} \\ \mathbf{d}_3^{(CR)} \end{bmatrix} = \underbrace{\begin{bmatrix} I & 0 & 0 & 0 & 0 & 0 \\ 0 & \rho I & 0 & 0 & I & 0 \\ 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & I & I & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 \\ I & 0 & 0 & 0 & 0 & \rho I \end{bmatrix}}_{M_{ADMM}} \begin{bmatrix} \mathbf{a}_1^{(CR)} \\ \mathbf{b}_1^{(CR)} \\ \mathbf{a}_2^{(CR)} \\ \mathbf{b}_2^{(CR)} \\ \mathbf{a}_3^{(CR)} \\ \mathbf{b}_3^{(CR)} \end{bmatrix}. \tag{5.166}$$

The result of applying this coordinate transform is depicted in Figure 5-17(b) from which the iteration in (5.158) through (5.160) can be obtained by assigning the variables $\mathbf{x} = \mathbf{c}_1^{(CR)}$, $\pi = \mathbf{c}_2^{(CR)}$, and $\mathbf{z} = \mathbf{c}_3^{(CR)}$ and inserting delay modules as illustrated in the figure. To see this, we first transform the interconnecting network by the coordinate matrix M_{ADMM} as

$$\begin{bmatrix} \mathbf{c}_1^{(CR)} \\ \mathbf{c}_2^{(CR)} \\ \mathbf{c}_3^{(CR)} \\ \mathbf{d}_1^{(CR)} \\ \mathbf{d}_2^{(CR)} \\ \mathbf{d}_3^{(CR)} \end{bmatrix} \in \text{range} \left(\underbrace{\begin{bmatrix} I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 \\ 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I \end{bmatrix}}_{\text{Input-output config.}} \underbrace{\begin{bmatrix} I & 0 & 0 & 0 & 0 & 0 \\ 0 & \rho I & 0 & 0 & I & 0 \\ 0 & 0 & I & I & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 \\ I & 0 & 0 & 0 & 0 & \rho I \end{bmatrix}}_{M_{ADMM}} \underbrace{\begin{bmatrix} I & 0 & 0 \\ 0 & -I & 0 \\ I & 0 & -I \\ 0 & I & 0 \\ 0 & 0 & I \\ 0 & I & 0 \end{bmatrix}}_{W \text{ for (5.161)}} \right) \tag{5.167}$$

$$\in \quad \text{range} \left(\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \\ 0 & -\rho I & I \\ I & I & -I \\ I & \rho I & 0 \end{bmatrix} \right) \tag{5.168}$$



(a) Stationarity conditions

(b) Transformed stationarity conditions and the Alternating Direction Method of Multipliers Algorithm

Figure 5-17: Interconnective structures illustrating the coordinate transform used to obtain the Alternating Direction Method of Multipliers Algorithm from the stationarity conditions associated with (5.161).

where the input-output configuration has been appropriately selected to be consistent with the standard notation of the $\mathbf{c}$ and $\mathbf{d}$ variables respectively denoting inputs and outputs of the interconnecting network. Reflecting the coordinate transform onto the constitutive modules yields functional realizations as depicted in the figure.

Note that the effect of the coordinate transform in (5.166) on the partition decomposition of the system is that the second constitutive module is transformed independent of the first and third modules to produce the functional realization $\mathbf{c}_2^{(CR)} = \mathbf{d}_2^{(CR)}$ while the first and third constitutive modules get mixed together. We next justify that the coordinate transform does in fact produce the proximal operators associated with the functions f and g despite not using the coordinate matrix M_{prox} directly on those modules. To see this, let $(\mathbf{a}_1^{(CR)}, \mathbf{b}_1^{(CR)}\mathbf{a}_3^{(CR)}, \mathbf{b}_3^{(CR)}) = (\mathbf{x}, \partial f(\mathbf{x}), \mathbf{z}, \partial g(\mathbf{z}))$ and $(\mathbf{c}_1^{(CR)}, \mathbf{d}_1^{(CR)}\mathbf{c}_3^{(CR)}, \mathbf{d}_3^{(CR)}) = (\mathrm{prox}_{\rho f}(\mathbf{u}), \mathbf{u}, \mathrm{prox}_{\rho g}(\mathbf{v}), \mathbf{v})$. Using this notation, the coordinate transform in (5.167) applied to the first and third modules is equivalent to the smaller coordinate transform

$$\begin{bmatrix} \mathrm{prox}_{\rho f}(\mathbf{u}) \\ \mathbf{u} \\ \mathrm{prox}_{\rho g}(\mathbf{v}) \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & \rho I & I & 0 \\ 0 & 0 & I & 0 \\ I & 0 & 0 & \rho I \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \partial f(\mathbf{x}) \\ \mathbf{z} \\ \partial g(\mathbf{z}) \end{bmatrix}. \tag{5.169}$$

Note that this particular formulation still does not make use of M_{prox} twice except in the sense of an alternating cross term. The conditions in (5.169) are equivalent to the the system of equations

$$\mathbf{u} = \partial f(\mathrm{prox}_{\rho f}(\mathbf{u})) + \mathrm{prox}_{\rho g}(\mathbf{v}) \tag{5.170}$$

$$\mathbf{v} = \partial g(\mathrm{prox}_{\rho g}(\mathbf{v})) + \mathrm{prox}_{\rho f}(\mathbf{u}). \tag{5.171}$$

These equations are generally not consistent with the definition of a proximal operator except in the case that $\mathrm{prox}_{\rho g}(\mathbf{v}) = \mathrm{prox}_{\rho f}(\mathbf{u})$. Said another way, (5.169) is a valid representation of the proximal operators so long as $\mathbf{x} = \mathbf{z}$. This condition is guaranteed at a solution to the stationarity conditions since it is equivalent to the constraint $\mathbf{x} - \mathbf{z} = \mathbf{0}$ in (5.161).

5.6.5 | Forward-Backward algorithms

Another common class of optimization algorithms we make connections to pertain to solving unconstrained convex optimization problems of the form

$$\underset{x}{\text{minimize}} \quad f(\mathbf{x}) + g(\mathbf{x}) \tag{5.172}$$

where the cost functions f and g are convex and allowed to take values of infinity, and where the function g is also required to be differentiable. The class of *forward-backward splitting* algorithms attempt to determine solutions $\mathbf{x}^\star$ by producing sequences of vectors $\mathbf{x}^n$ according to variations of the basic iteration

$$\mathbf{x}^{n+1} = \underbrace{\mathrm{prox}_{\rho f}}_{\text{backward step}} (\underbrace{\mathbf{x}^n - \rho \nabla g(\mathbf{x}^n)}_{\text{forward step}}), \tag{5.173}$$

where convergence of the iteration can be guaranteed for appropriate selection rules of the step-size parameter ρ [99, Proposition 10.4]. The forward-backward algorithm in (5.173) encapsulates many algorithms previously discussed in this section. For example, the iteration in (5.173) reduces to the gradient-descent algorithm in (5.131) if the cost term $f = 0$ and reduces to the proximal minimization algorithm in (5.145) if the cost term $g = 0$. Note that the ADMM algorithm can also be used to solve problems described by (5.172).

Consider the special case of the cost function in (5.172) where the terms f and g are equal, and thus both differentiable. Substituting the relationship that $\text{prox}_{\rho f} = (I+\rho\nabla f)^{-1}$, the forward-backward iteration simplifies to the form

$$\mathbf{x}^{n+1} = (I + \rho\nabla f)^{-1}(I - \rho\nabla f)(\mathbf{x}^n). \tag{5.174}$$

Comparing (5.174) to the functional realization in (5.104) used to execute a scattering algorithm, the scattering operator is interpretable as simultaneously applying the forward and backward steps of the update in (5.173), except with the backward step being enforced before the forward step and with the input being the scattering variable **d** rather than the primal decision variable **a**. It is straightforward to verify that the scattering and forward-backward operators are not related through linear behavioral transformations, thus scattering algorithms are not generally in the same interconnective equivalence class as forward-backward algorithms, though the two are closely related in the sense described above.

5.6.6 | Comments on alternative coordinate transforms

The key point emphasized by the connections presented in this section is that steepest-descent and proximal algorithms are closely related to scattering algorithms in straightforward and fundamental ways. In particular, algorithms belonging to these classes can all be produced by starting with the same interconnective description of the stationarity conditions associated with an optimization problem and inserting memory into the interconnective structure after performing different coordinate transforms. This observation suggests additional opportunity in designing alternative coordinate transform matrices beyond those discussed in this thesis to produce algorithm classes tailored to certain computing environments. Understanding how coordinate matrices can be designed so that the resulting algorithms have desirable properties forms an important direction for future research. In the remainder of this section, we summarize some of the desirable properties the class of scattering algorithms possess in the context of solving large-scale optimization problems using decentralized signal processing systems, and we provide comments on directions of potentially interesting future developments.

The interconnective description of many real world systems satisfies the requirements of a canonical-form structure, e.g. systems involving electrical, biological, chemical, financial, and transport networks. In the context of manipulating canonical-form interconnective structures, the scattering coordinate transforms used in this thesis were chosen specifically to preserve maximal partition decompositions. As a consequence, the locality of the individual subsystems composing the structure is preserved in the transformation, and this fact applies to both the locality of the interconnecting networks and constitutive modules. In the context of variational properties exhibited by canonical-form structures, this separability is instrumental to the modular design of fully asynchronous and uncoordinated scattering algorithms for solving optimization problems using the same topology or underlying graph associated with the problem at hand. In deriving the ADMM algorithm, for example, the main differentiating feature between it and proximal algorithms is that the coordinate transform M_{ADMM} does not preserve the locality or separability of the partition decomposition, but still results in distributable algorithms that are known to possess desirable convergence properties [45]. The alternating cross term between the first and third constitutive modules constrained the associated primal and dual variables to be equal to one another by relying on the constraints imposed by the interconnecting network. This type of strategy could be applied to the scattering coordinate transform as well for problem classes where these types of relationships exist.

Another important property scattering coordinate transforms provide is that the interconnecting networks are realized by orthogonal matrices which are perfectly conditioned and passive everywhere. In addition, the interplay between convexity and the scattering transform provides the ability to use local, independent certificates to certify global stability and robustness properties of the system. For example, convexity of reduced-form cost functions provide well-defined functional realizations of the constitutive modules, and these functions are passive everywhere in the worst case. As such, convergence guarantees for scattering algorithms follow immediately without relying on problem dependent tuning or step-size parameters. In addition, the convergence for many strictly convex problems result in modules with contractive functional realizations, resulting in linear convergence of the scattering algorithm, meaning tuning or step-size parameters are unnecessary but can be used to improve the rate of convergence.

Chapter 6

Examples of interconnective structures and scattering algorithms

The formal methods in this thesis for designing and implementing large-scale signal processing systems are naturally-suited to solving broad classes of fixed-point and optimization problems, and the interconnective framework provides a straightforward way to take advantage of a variety of computing resources in doing so. For example, by describing solutions to a problem as the behavior of a signal processing system, the interconnective description of the system focuses on issues of separability and robustness that are closely related to issues of distributability and synchronization in many distributed and embedded computing environments. The implementation of the system is then able take advantage of the particular strengths of the individual computing resources that are available, and this motivates a concrete discussion of these strengths in distributed and embedded contexts.

In this chapter, several scattering algorithms are presented using the modules from Chapter 5 to solve common classes of fixed-point and optimization problems appearing in signal processing applications, and the details associated with implementing these scattering systems as distributed and embedded signal processing systems are also presented. Specifically, scattering algorithms implemented as web services are formed by organizing the associated interconnective structures into relational databases so that the implementations inherit the scalability and security of the particular databases used. In addition, two embedded systems architectures are discussed that take advantage of the stability of interconnective structures to reduce inefficiencies associated with synchronization and communication by desynchronizing individual processors and simplifying the handshaking protocols used to communicate between processors. The chapter concludes with a brief summary of the contributions of the thesis as well as indicates some directions for future research.

6.1 | Example scattering algorithms

In this section, scattering algorithms are presented for solving a variety of constraint satisfaction and optimization problems of interest to the signal processing community. The general approach we take to discussing these algorithms is to first present the scattering algorithms for solving linear systems of equations using a decentralized network of processors. These decentralization techniques can then be applied to further separate the linear system of equations appearing in the stationarity conditions of a given optimization problem. For each class of optimization problems considered in this section, the general form of the Lagrangian primal and dual problems are stated. Then, the equivalent reduced-form conservative primal and dual problems are stated and the associated canonical-form and scattering-form interconnective structures respectively describing the untransformed and transformed stationarity conditions are derived. Numerical results demonstrating the convergence of the scattering algorithms presented in this section are provided where the results are generated by implementing the algorithms as distributed and embedded signal processing systems. The specific details associated with implementing these algorithms are provided in the two sections that follow.

The scattering algorithms formed using the modular framework developed in this thesis appear to be novel. As such, the examples presented in this section have specifically been chosen to highlight the stationarity conditions and iterations, both synchronous and asynchronous, for well-known signal processing problems. From this perspective, these examples serve as concluding remarks for the thesis, and suggest future work in two directions. The first direction involves making connections between scattering and existing optimization algorithms beyond those discussed in Section 5.6 by identifying correspondences between the iterations used to solve a problem and the duality principles used to establish the two iterations. The second direction, discussed in Section 5.3, pertains to the procedure associated with producing additional modules and involves designing modules to target problems outside the scope of those that can be addressed using the modules derived in this thesis. The modules listed in Appendix B can be used to solve a large class of convex and non-convex problems, as demonstrated by the examples in this section.

6.1.1 | Decentralized scattering algorithms for solving linear equations

Numerical linear algebra subroutines provide a powerful set of tools with which to perform computations involving matrices and vectors. The ability to solve linear systems of equations is fundamental to many fields, and solving equations of the form

$$A\mathbf{x} = \mathbf{y} \tag{6.1}$$

using a decentralized network of processors is important in signal processing applications where the coefficient matrix A is either too large to be stored or manipulated by a single processor in the network or cannot be assembled using centralized means for other reasons, e.g. the privacy and security of medical and personal data. In Section 5.4, the stationarity conditions associated with decentralizing linear equations were derived where the coefficient matrix was not required to satisfy any particular spectral properties or possess any special structure. The general strategy was to partition the matrix A into blocks of arbitrary size and assign each block to a different processor. Then, each block was taken through a scattering transform and certain of the processors were coupled together using quadratic coupling modules, as in Figure 5-11, operating on auxiliary variables introduced on those processors that are connected. This strategy is illustrated in Figure 5-12. The quadratic coupling module carries with it an adaptive or homotopy parameter that suggests a numerical scheme for slowly adapting the processing system in the transformed domain from an initial well-behaved system to the desired system. In this subsection, we present the interconnective description of the transformed stationarity conditions associated with separately partitioning the coefficient matrix A by column and by row and present some numerical results corresponding to implementing the slowly-adapting system. The ability to partition the coefficient matrix A into arbitrarily-sized subblocks follows immediately from this presentation by applying row partitioning to the column partitioning scheme or vice versa.

The interconnective structure illustrating the transformed stationarity conditions associated with partitioning the coefficient matrix A in (6.1) into a total of M mutually-exclusive and collectively-exhaustive groups of its rows is depicted in Figure 6-1, where the notation and variable ordering correspond to the formulation of this problem in (5.123) through (5.125). Similarly, the interconnective structure illustrating the transformed stationarity conditions associated with partitioning A into a total of M mutually-exclusive and collectively-exhaustive groups of its columns is depicted in Figure 6-2, where the notation and variable ordering correspond to the formulation of this problem in (5.126) and (5.127). Referring to the interconnective structures depicted in both figures, the interprocessor structures depict the functional realization of the quadratic coupling module in Figure 5-11, where the parameter $\alpha = \frac{1}{(1+\rho)^2+1}$ and the intraprocessor structures depict the transformed stationarity conditions that are private to each processor and make use of the constitutive relation modules in Figure 5-7. Numerical convergence results for the state sequence of primal decision variables $\mathbf{x}^n$ for asynchronous implementations of the transformed stationarity conditions are also provided for three of $M = 2500$ virtual processors used to solve a decentralized linear problem where the coefficient matrix A has about 1.5 billion non-zero entries. The implementations used filtered asynchronous delay elements with the filtering parameter $\rho = \frac{1}{2}$ and delay probability $p = \frac{1}{2}$. Also provided in the numerical results is the schedule of the quadratic coupling parameter ρ_{qc}. We note that for non-zero values of this parameter the system variables converge linearly, which manifests itself through the observed linear convergence of $\mathbf{x}^n$. The matrix G_{M+1} used in the column partitioning example is only non-zero in less than 0.1% of its entries for the depicted problem size. An explicit form for G_{M+1} is also provided as an alternative to generating it using (5.91).

The ability to solve linear equations is critical to solving the transformed stationarity conditions associated with any conservative optimization problem since the matrices associated with the canonical-form interconnecting network are used to generate the orthogonal subspaces that the stationarity condition was derived from. Referring to the intraprocessor

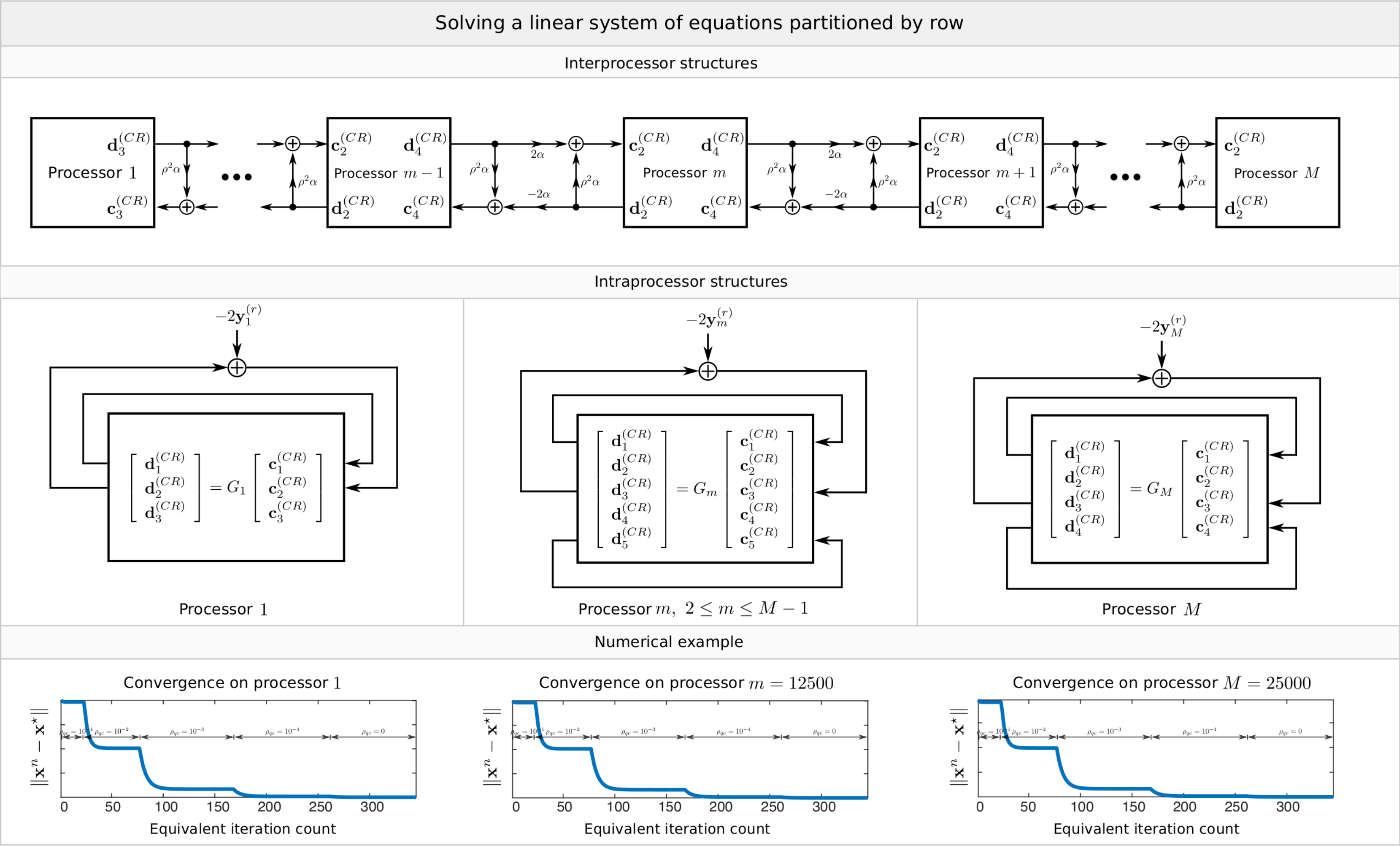

Solving a linear system of equations partitioned by row
Interprocessor structures
Processor 1
$\mathbf{d}_3^{(CR)}$
$\mathbf{c}_3^{(CR)}$
Processor $m-1$
Processor m
Processor $m+1$
Processor M
$\mathbf{c}_2^{(CR)}$
$\mathbf{d}_4^{(CR)}$
$\mathbf{d}_2^{(CR)}$
$\mathbf{c}_4^{(CR)}$
$\rho^2\alpha$
2α
-2α
Intraprocessor structures
$-2\mathbf{y}_1^{(r)}$
$\begin{bmatrix}\mathbf{d}_1^{(CR)}\\ \mathbf{d}_2^{(CR)}\\ \mathbf{d}_3^{(CR)}\end{bmatrix} = G_1 \begin{bmatrix}\mathbf{c}_1^{(CR)}\\ \mathbf{c}_2^{(CR)}\\ \mathbf{c}_3^{(CR)}\end{bmatrix}$
Processor 1
$-2\mathbf{y}_m^{(r)}$
$\begin{bmatrix}\mathbf{d}_1^{(CR)}\\ \mathbf{d}_2^{(CR)}\\ \mathbf{d}_3^{(CR)}\\ \mathbf{d}_4^{(CR)}\\ \mathbf{d}_5^{(CR)}\end{bmatrix} = G_m \begin{bmatrix}\mathbf{c}_1^{(CR)}\\ \mathbf{c}_2^{(CR)}\\ \mathbf{c}_3^{(CR)}\\ \mathbf{c}_4^{(CR)}\\ \mathbf{c}_5^{(CR)}\end{bmatrix}$
Processor m, $2 \le m \le M-1$
$-2\mathbf{y}_M^{(r)}$
$\begin{bmatrix}\mathbf{d}_1^{(CR)}\\ \mathbf{d}_2^{(CR)}\\ \mathbf{d}_3^{(CR)}\\ \mathbf{d}_4^{(CR)}\end{bmatrix} = G_M \begin{bmatrix}\mathbf{c}_1^{(CR)}\\ \mathbf{c}_2^{(CR)}\\ \mathbf{c}_3^{(CR)}\\ \mathbf{c}_4^{(CR)}\end{bmatrix}$
Processor M
Numerical example
Convergence on processor 1
Convergence on processor $m = 12500$
Convergence on processor $M = 25000$
$\|\mathbf{x}^n - \mathbf{x}^\star\|$
$\rho_{qc} = 10^{-1}$
$\rho_{qc} = 10^{-2}$
$\rho_{qc} = 10^{-3}$
$\rho_{qc} = 10^{-4}$
$\rho_{qc} = 0$
0
50
100
150
200
250
300
Equivalent iteration count

Figure 6-1: Example interconnective structures used to solve linear systems of equations partitioned by row using decentralized processors. The numerical results were obtained using a continuation scheme with 500 processors where the parameter ρ_{cr} is the tuning parameter for the quadratic coupling module in Figure 5-11.

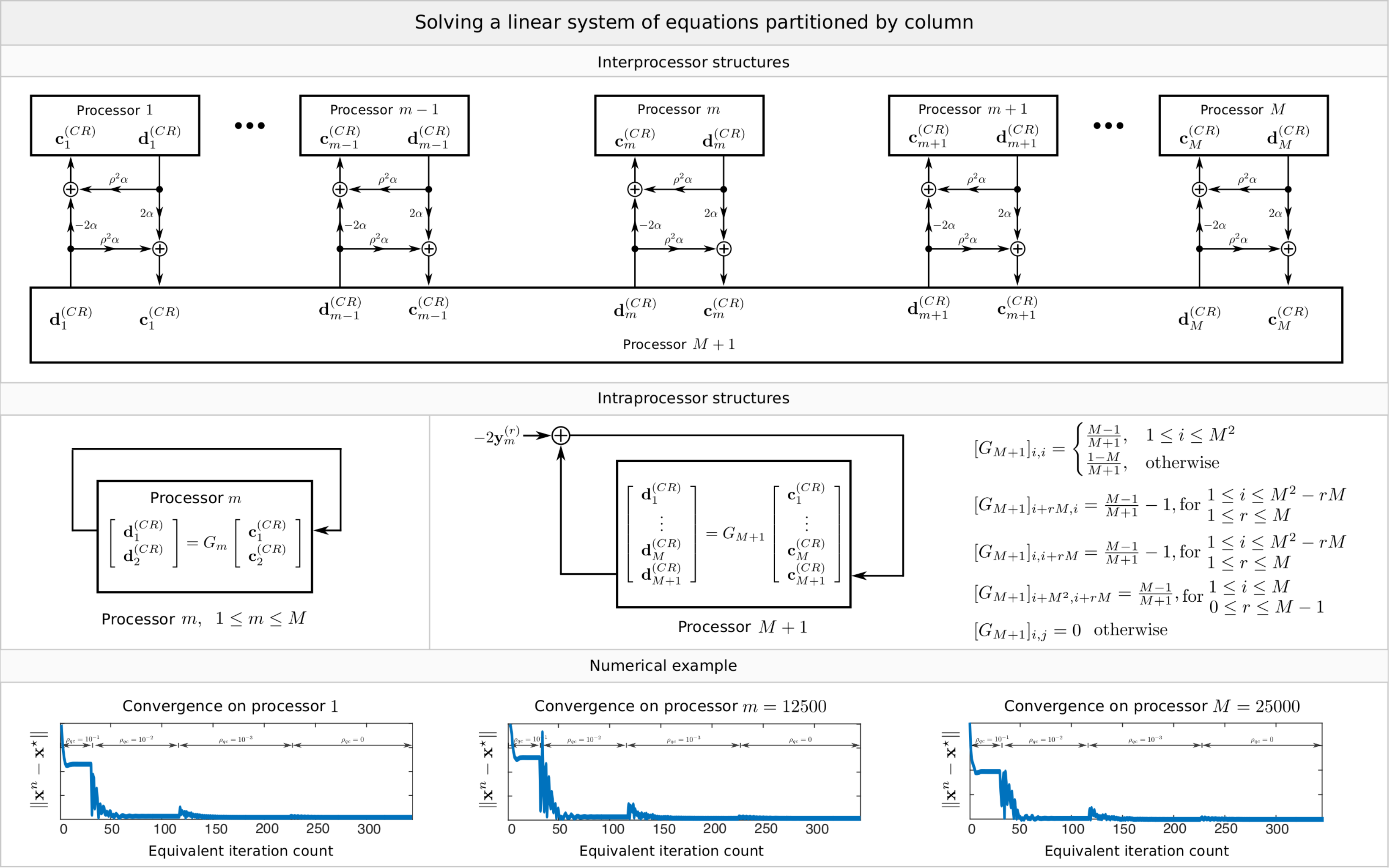

Solving a linear system of equations partitioned by column
Interprocessor structures
Processor 1
$\mathbf{c}_1^{(CR)}$ $\mathbf{d}_1^{(CR)}$
Processor $m-1$
$\mathbf{c}_{m-1}^{(CR)}$ $\mathbf{d}_{m-1}^{(CR)}$
Processor m
$\mathbf{c}_m^{(CR)}$ $\mathbf{d}_m^{(CR)}$
Processor $m+1$
$\mathbf{c}_{m+1}^{(CR)}$ $\mathbf{d}_{m+1}^{(CR)}$
Processor M
$\mathbf{c}_M^{(CR)}$ $\mathbf{d}_M^{(CR)}$
$\rho^2\alpha$
2α
-2α
$\rho^2\alpha$
$\mathbf{d}_1^{(CR)}$ $\mathbf{c}_1^{(CR)}$
$\mathbf{d}_{m-1}^{(CR)}$ $\mathbf{c}_{m-1}^{(CR)}$
$\mathbf{d}_m^{(CR)}$ $\mathbf{c}_m^{(CR)}$
$\mathbf{d}_{m+1}^{(CR)}$ $\mathbf{c}_{m+1}^{(CR)}$
$\mathbf{d}_M^{(CR)}$ $\mathbf{c}_M^{(CR)}$
Processor $M+1$
Intraprocessor structures
Processor m
$\begin{bmatrix} \mathbf{d}_1^{(CR)} \\ \mathbf{d}_2^{(CR)} \end{bmatrix} = G_m \begin{bmatrix} \mathbf{c}_1^{(CR)} \\ \mathbf{c}_2^{(CR)} \end{bmatrix}$
Processor $m,\ 1 \le m \le M$
$-2\mathbf{y}_m^{(r)}$
$\begin{bmatrix} \mathbf{d}_1^{(CR)} \\ \vdots \\ \mathbf{d}_M^{(CR)} \\ \mathbf{d}_{M+1}^{(CR)} \end{bmatrix} = G_{M+1} \begin{bmatrix} \mathbf{c}_1^{(CR)} \\ \vdots \\ \mathbf{c}_M^{(CR)} \\ \mathbf{c}_{M+1}^{(CR)} \end{bmatrix}$
Processor $M+1$
$[G_{M+1}]_{i,i} = \begin{cases} \frac{M-1}{M+1}, & 1 \le i \le M^2 \\ \frac{1-M}{M+1}, & \text{otherwise} \end{cases}$
$[G_{M+1}]_{i+rM,i} = \frac{M-1}{M+1} - 1, \text{for } \begin{matrix} 1 \le i \le M^2 - rM \\ 1 \le r \le M \end{matrix}$
$[G_{M+1}]_{i,i+rM} = \frac{M-1}{M+1} - 1, \text{for } \begin{matrix} 1 \le i \le M^2 - rM \\ 1 \le r \le M \end{matrix}$
$[G_{M+1}]_{i+M^2,i+rM} = \frac{M-1}{M+1}, \text{for } \begin{matrix} 1 \le i \le M \\ 0 \le r \le M-1 \end{matrix}$
$[G_{M+1}]_{i,j} = 0 \quad \text{otherwise}$
Numerical example
Convergence on processor 1
Convergence on processor $m = 12500$
Convergence on processor $M = 25000$
$\|\mathbf{x}^n - \mathbf{x}^\star\|$
$\rho_{qc} = 10^{-1}$
$\rho_{qc} = 10^{-2}$
$\rho_{qc} = 10^{-3}$
$\rho_{qc} = 0$
0
50
100
150
200
250
300
Equivalent iteration count

Figure 6-2: Example interconnective structures used to solve linear systems of equations partitioned by column using decentralized processors. The numerical results were obtained using a continuation scheme with 500 processors where the parameter ρ_{cr} is the tuning parameter for the quadratic coupling module in Figure 5-11.

portion of the transformed stationarity conditions for both row and column partitionings of the coefficient matrix, an objective function can be defined on each entry or copy of the decision vector $\mathbf{x}$ by simply replacing the identity function used to generate $\mathbf{c}_1^{(CR)}$ from $\mathbf{d}_1^{(CR)}$ on each processor with the function corresponding to the desired cost term. Using straightforward manipulations and module replacements such as this, all of interconnective structures used to depict transformed stationarity conditions in the remainder of this section can be decentralized using the approach in this section, or for already decentralized structures the individual interconnects can be further decentralized. However, to avoid overly complicated notation that might obscure the simplicity of the algorithms for solving various optimization problems arising in signal processing contexts, we state the transformed stationarity conditions using centralized formulations with an understanding that decentralizing those formulations is straightforward using the techniques in this subsection.

6.1.2 | Linear programs

The problem of minimizing a linear cost function subject to linear equality and inequality constraints is referred to as a *linear program* and appears in a variety of signal processing applications such as optimum filter and array design [101], sparse signal recovery [3], spectral estimation [102], and error correction in transform coding [103]. The Lagrangian primal and dual formulations of a linear program are written in standard form according to

$$\text{(LP)}\quad \begin{array}{ll} \underset{\mathbf{x}}{\text{minimize}} & \mathbf{q}^T\mathbf{x} \\ \text{s.t.} & A\mathbf{x} \leq \mathbf{r} \\ & \mathbf{x} \geq \mathbf{0} \end{array} \qquad\qquad \text{(LD)}\quad \begin{array}{ll} \underset{\mathbf{y}}{\text{maximize}} & \mathbf{r}^T\mathbf{y} \\ \text{s.t.} & A^T\mathbf{y} \geq \mathbf{q} \\ & \mathbf{y} \geq \mathbf{0} \end{array} \tag{6.2}$$

where the vectors of primal and dual decision variables are respectively denoted $\mathbf{x}$ and $\mathbf{y}$, the primal cost vector is denoted $\mathbf{q}$, the primal inequality vector is denoted $\mathbf{r}$, and the linear coefficient matrix is denoted A. As is widely known, the feasible set of a linear program is a convex polytope in the non-negative orthant, and solutions to a linear program are always on the boundary of this set. Basis exchange algorithms, a typical approach to solving linear programs, iteratively update a single solution by traversing the vertices of the feasible set

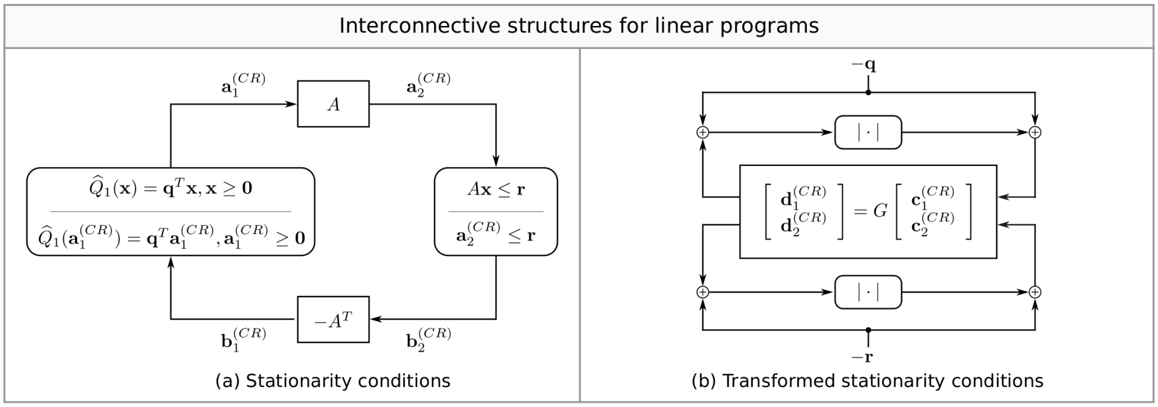

Interconnective structures for linear programs
$\mathbf{a}_1^{(CR)}$
A
$\mathbf{a}_2^{(CR)}$
$\widehat{Q}_1(\mathbf{x}) = \mathbf{q}^T\mathbf{x}, \mathbf{x} \geq \mathbf{0}$
$\widehat{Q}_1(\mathbf{a}_1^{(CR)}) = \mathbf{q}^T\mathbf{a}_1^{(CR)}, \mathbf{a}_1^{(CR)} \geq \mathbf{0}$
$A\mathbf{x} \leq \mathbf{r}$
$\mathbf{a}_2^{(CR)} \leq \mathbf{r}$
$-A^T$
$\mathbf{b}_1^{(CR)}$
$\mathbf{b}_2^{(CR)}$
$-\mathbf{q}$
$|\cdot|$
$\begin{bmatrix} \mathbf{d}_1^{(CR)} \\ \mathbf{d}_2^{(CR)} \end{bmatrix} = G \begin{bmatrix} \mathbf{c}_1^{(CR)} \\ \mathbf{c}_2^{(CR)} \end{bmatrix}$
$|\cdot|$
$-\mathbf{r}$
(a) Stationarity conditions
(b) Transformed stationarity conditions

Figure 6-3: The interconnective structures used to solve linear programs in standard form. (a) The stationarity conditions in a canonical-form structure. (b) The transformed stationarity conditions in a scattering-form structure.

until the cost function cannot be decreased further. Alternatively, interior point and barrier methods identify a solution by moving through the interior of the feasible set until settling at a vertex corresponding to a solution [104].

The standard form of the linear program (6.2)(LP) can be reformulated in accordance with the notation used to describe conservative optimization problems. In particular, the reduced-form primal (P) and corresponding dual (D) problems are written according to

$$\text{(P)} \quad \begin{array}{rl} \underset{\mathbf{a}}{\text{minimize}} & \mathbf{q}^T\mathbf{a}_1^{(CR)} \\ \text{s.t.} & A\mathbf{a}_1^{(CR)} = \mathbf{a}_2^{(CR)} \\ & \mathbf{a}_1^{(CR)} \geq \mathbf{0} \\ & \mathbf{a}_2^{(CR)} \leq \mathbf{r} \end{array} \qquad \text{(D)} \quad \begin{array}{rl} \underset{\mathbf{b}}{\text{maximize}} & -\mathbf{r}^T\mathbf{b}_2^{(CR)} \\ \text{s.t.} & -A^T\mathbf{b}_2^{(CR)} = \mathbf{b}_1^{(CR)} \\ & \mathbf{b}_1^{(CR)} \leq \mathbf{q} \\ & \mathbf{b}_2^{(CR)} \geq \mathbf{0} \end{array} \tag{6.3}$$

where the primal decision variables $(\mathbf{a}_1^{(CR)}, \mathbf{a}_2^{(CR)})$ correspond to $(\mathbf{x}, A\mathbf{x})$. Observe that the conservative dual problem, as discussed in Section 5.1.5, can be reduced using straightforward manipulations into the form of the Lagrangian dual problem (LD). An interconnective description of the stationarity conditions associated with a linear program is provided in Figure 6-3(a). An equivalent description of the transformed stationarity conditions formulated as a scattering-form interconnective structure is illustrated in Figure 6-3(b). Referring to the transformed structure, the matrix G is generated from the coefficient matrix A according to (5.91) and the constitutive modules are implemented using the functions

$$\mathbf{c}_1^{(CR)} = |\mathbf{d}_1^{(CR)} - \mathbf{q}| - \mathbf{q} \tag{6.4}$$

$$\mathbf{c}_2^{(CR)} = |\mathbf{d}_2^{(CR)} - \mathbf{r}| - \mathbf{r}. \tag{6.5}$$

Note that the expression in (6.4) combines the second module in Figure 5-7 with the first module in Figure 5-8, and the expression in (6.5) combines the third module in Figure 5-6 with the first module in Figure 5-7. Synchronous and asynchronous algorithms then correspond to implementing the interconnective structure representing the transformed stationarity conditions after inserting a valid configuration of state and selecting a protocol for state exchange, as was discussed in Section 3.3. The system operator associated with these implementations is passive everywhere, hence convergence is guaranteed when the delay modules are filtered with an appropriately chosen filtering parameter as is discussed in Theorem 4.3.6.

A variety of problems appearing in signal processing contexts and elsewhere that are naturally described using nonlinear features have been shown to be equivalent to linear programming problems. In response to this, many transformations and manipulations have been developed to assist in recasting an initial problem description into the standard form of (6.2), e.g. see [105, Ch. 6]. Drawing upon the modular tools developed in this thesis, these same manipulations can be applied, and then corresponding algorithms can be generated, by casting the problem's stationarity conditions into the interconnective structures provided in Figure 6-3. Alternatively, specialized algorithms can be assembled for these problems by using modules that directly represent the nonlinear features.

For example, a common approach used to generate a sparse solution to an underdetermined linear system of equations satisfying certain spectral properties is to solve the basis pursuit problem [3] given by

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \|\mathbf{x}\|_1 \\ \text{s.t.} \quad & A\mathbf{x} = \mathbf{r}. \end{aligned} \tag{6.6}$$

Rather than reformulating this problem to have a linear cost function by introducing auxiliary variables and constraints, an algorithm can be designed through the modular framework developed in this thesis by using an appropriate constitutive module to account for the non-linear behavior of the objective function. The interconnective structures associated with doing so are depicted along the top of Figure 6-4. Referring to the transformed stationarity conditions, the matrix G is generated from the coefficient matrix A according to (5.91) and the constitutive modules are implemented using the functions

$$\mathbf{c}_1^{(CR)} = \begin{cases} -\mathbf{d}_1^{(CR)}, & |\mathbf{d}_1^{(CR)}| \leq 1 \\ \mathbf{d}_1^{(CR)} - 2\mathrm{sgn}(\mathbf{d}_1^{(CR)}), & |\mathbf{d}_1^{(CR)}| > 1 \end{cases} \tag{6.7}$$

$$\mathbf{c}_2^{(CR)} = \mathbf{d}_2^{(CR)} - 2\mathbf{r}. \tag{6.8}$$

Note that the expression in (6.7) is the first module in Figure 5-8, and the expression in (6.8) is the first module in Figure 5-7. Numerical convergence results and the intermediary values taken by the primal objective function as well as the final solution obtained for both synchronous and asynchronous implementations of the transformed stationarity conditions are also provided. Both implementations used filtered delay elements with parameter $\rho = \frac{1}{2}$ and the asynchronous implementation used a delay probability of $p = \frac{3}{4}$.

As a second example, consider the problem of designing a minimax optimal, Type I impulse response [35]. As is well-known, this design problem can be written as a linear programming problem according to either of the following equivalent formulations:

$$\underset{\mathbf{h}}{\text{minimize}} \quad \max_{\omega \in \Omega} |T(\omega, \mathbf{h}) - \mathbf{d}| \qquad \begin{array}{ll} \underset{\delta, \mathbf{h}}{\text{minimize}} & \delta \\ \text{s.t.} & |T(\omega, \mathbf{h}) - \mathbf{d}|, \leq \delta, \quad \omega \in \Omega \end{array} \tag{6.9}$$

where the vector of impulse response values is denoted $\mathbf{h}$, Ω is a discretized set of frequency values in $[0, \pi]$ that the desired frequency response amplitude $\mathbf{d}$ is defined over, the maximum deviation of the designed frequency response from $\mathbf{d}$ on Ω is denoted δ, and the linear operator $T(\omega, \mathbf{h})$ is from (4.96). The interconnective structures used to solve this problem are depicted on the bottom of Figure 6-4. Referring to the transformed stationarity conditions, the matrix G is generated from the coefficient matrix appearing in the stationarity conditions

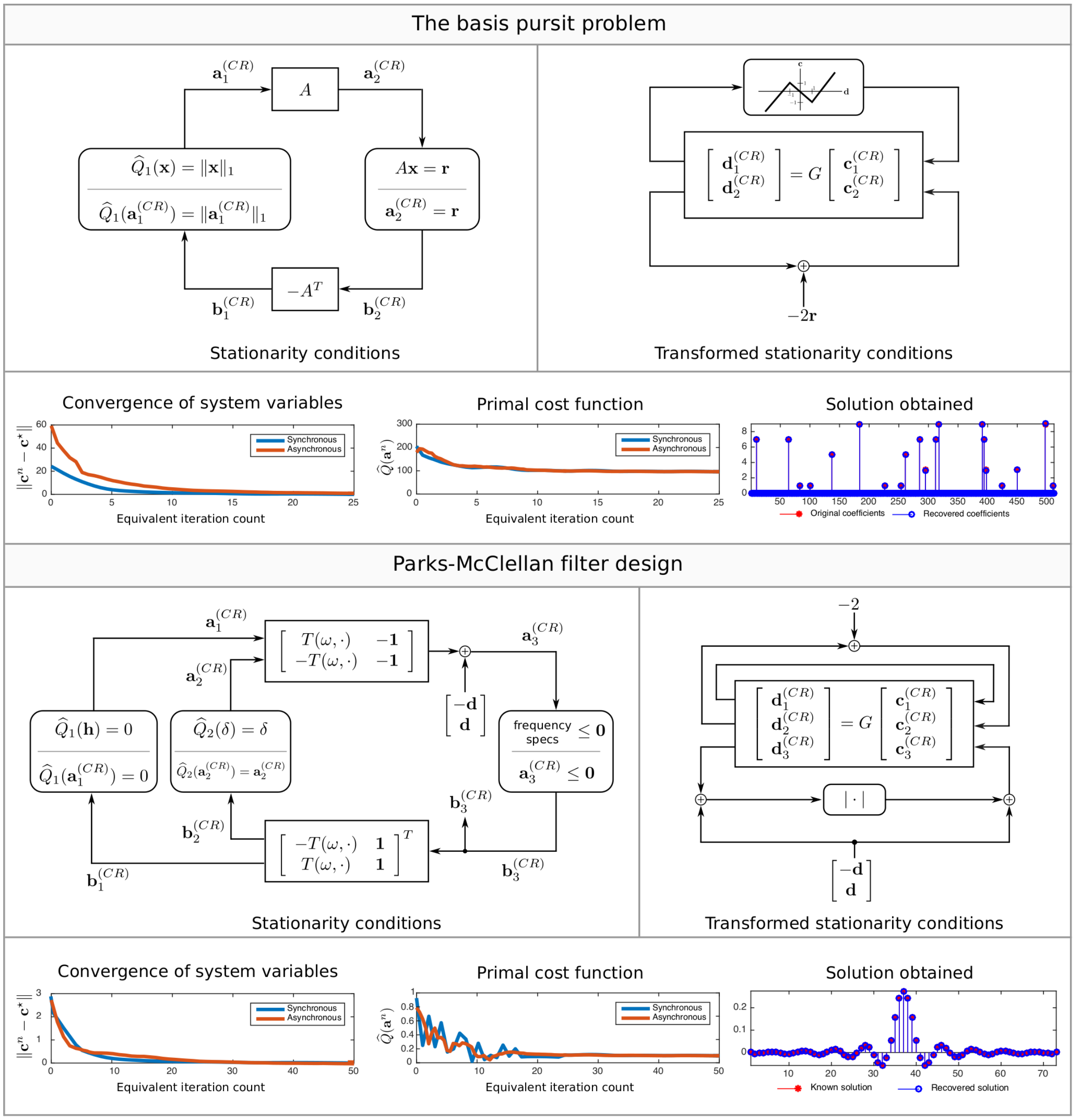

The basis pursit problem
$\mathbf{a}_1^{(CR)}$
A
$\mathbf{a}_2^{(CR)}$
$\widehat{Q}_1(\mathbf{x}) = \|\mathbf{x}\|_1$
$\widehat{Q}_1(\mathbf{a}_1^{(CR)}) = \|\mathbf{a}_1^{(CR)}\|_1$
$A\mathbf{x} = \mathbf{r}$
$\mathbf{a}_2^{(CR)} = \mathbf{r}$
$-A^T$
$\mathbf{b}_1^{(CR)}$
$\mathbf{b}_2^{(CR)}$
Stationarity conditions
c
d
$\begin{bmatrix} \mathbf{d}_1^{(CR)} \\ \mathbf{d}_2^{(CR)} \end{bmatrix} = G \begin{bmatrix} \mathbf{c}_1^{(CR)} \\ \mathbf{c}_2^{(CR)} \end{bmatrix}$
$-2\mathbf{r}$
Transformed stationarity conditions
Convergence of system variables
$\|\mathbf{c}^n - \mathbf{c}^\star\|$
Synchronous
Asynchronous
Equivalent iteration count
Primal cost function
$\widehat{Q}(\mathbf{a}^n)$
Synchronous
Asynchronous
Equivalent iteration count
Solution obtained
Original coefficients
Recovered coefficients
Parks-McClellan filter design
$\mathbf{a}_1^{(CR)}$
$\begin{bmatrix} T(\omega,\cdot) & -\mathbf{1} \\ -T(\omega,\cdot) & -\mathbf{1} \end{bmatrix}$
$\mathbf{a}_2^{(CR)}$
$\mathbf{a}_3^{(CR)}$
$\begin{bmatrix} -\mathbf{d} \\ \mathbf{d} \end{bmatrix}$
$\widehat{Q}_1(\mathbf{h}) = 0$
$\widehat{Q}_1(\mathbf{a}_1^{(CR)}) = 0$
$\widehat{Q}_2(\delta) = \delta$
$\widehat{Q}_2(\mathbf{a}_2^{(CR)}) = \mathbf{a}_2^{(CR)}$
frequency specs $\leq \mathbf{0}$
$\mathbf{a}_3^{(CR)} \leq \mathbf{0}$
$\mathbf{b}_3^{(CR)}$
$\mathbf{b}_2^{(CR)}$
$\begin{bmatrix} -T(\omega,\cdot) & \mathbf{1} \\ T(\omega,\cdot) & \mathbf{1} \end{bmatrix}^T$
$\mathbf{b}_1^{(CR)}$
$\mathbf{b}_3^{(CR)}$
Stationarity conditions
-2
$\begin{bmatrix} \mathbf{d}_1^{(CR)} \\ \mathbf{d}_2^{(CR)} \\ \mathbf{d}_3^{(CR)} \end{bmatrix} = G \begin{bmatrix} \mathbf{c}_1^{(CR)} \\ \mathbf{c}_2^{(CR)} \\ \mathbf{c}_3^{(CR)} \end{bmatrix}$
$|\cdot|$
$\begin{bmatrix} -\mathbf{d} \\ \mathbf{d} \end{bmatrix}$
Transformed stationarity conditions
Convergence of system variables
$\|\mathbf{c}^n - \mathbf{c}^\star\|$
Synchronous
Asynchronous
Equivalent iteration count
Primal cost function
$\widehat{Q}(\mathbf{a}^n)$
Synchronous
Asynchronous
Equivalent iteration count
Solution obtained
Known solution
Recovered solution

Figure 6-4: Example structures used to generate algorithms for solving two linear programs.

using (5.91) and the constitutive modules are implemented using the functions

$$\mathbf{c}_1^{(CR)} = \mathbf{d}_1^{(CR)} \tag{6.10}$$

$$\mathbf{c}_2^{(CR)} = \mathbf{d}_2^{(CR)} - 2 \tag{6.11}$$

$$\mathbf{c}_3^{(CR)} = \left| \mathbf{d}_3^{(CR)} + \begin{bmatrix} -\mathbf{d} \\ \mathbf{d} \end{bmatrix} \right| + \begin{bmatrix} -\mathbf{d} \\ \mathbf{d} \end{bmatrix}. \tag{6.12}$$

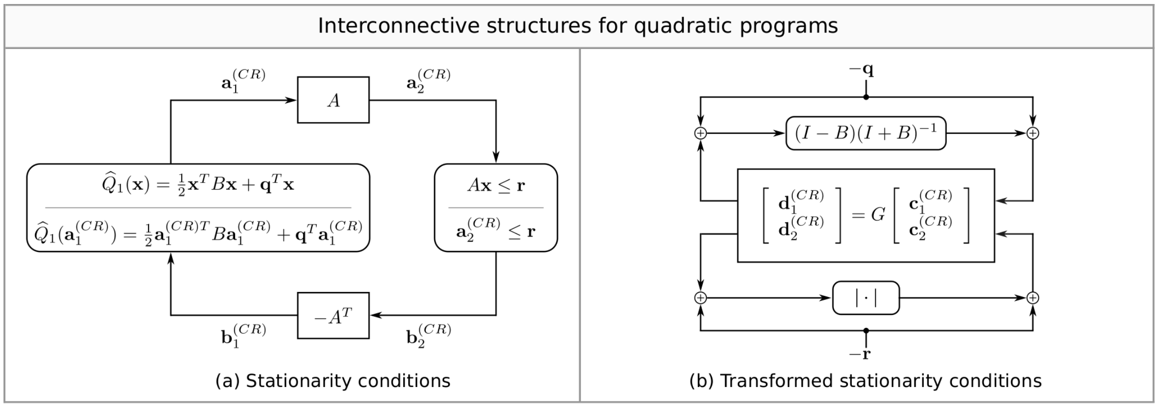
Note that the expression in (6.10) is the second module in Figure 5-7 with $\rho = 0$, the expression in (6.11) is the second module in Figure 5-7 with $\rho = 1$, and the expression in (6.12) combines the third module in Figure 5-6 with the first module in Figure 5-7. Numerical convergence results and the intermediary values taken by the primal objective function as well as the final solution obtained for both synchronous and asynchronous implementations of the transformed stationarity conditions are also provided. The implementations used filtered delay elements with parameter $\rho = \frac{1}{2}$ and the asynchronous implementation used delay probability $p = \frac{3}{4}$. Additional scattering algorithms that are consistent with coupling constitutive modules together to represent the transformed stationarity conditions associated with the Chebyshev center problem and to perform error correction decoding in transform coding applications can be found in [88] and [106], respectively.

6.1.3 | Quadratic programs

The problem of minimizing a quadratic cost function subject to linear equality and inequality constraints is referred to as a *quadratic program* and appears in a variety of signal processing applications such as sparse signal recovery [107], robust Kalman filtering [93], and variants of least squares including non-negative and bounded value [108]. The Lagrangian primal and dual formulations of a quadratic program are written in standard form according to

$$\text{(QP)}\quad \begin{array}{rl} \underset{\mathbf{x}}{\text{minimize}} & \frac{1}{2}\mathbf{x}^T B\mathbf{x} + \mathbf{q}^T\mathbf{x} \\ \text{s.t.} & A\mathbf{x} \leq \mathbf{r} \end{array} \qquad \text{(QD)}\quad \begin{array}{rl} \underset{\mathbf{y}}{\text{maximize}} & -\frac{1}{2}(\mathbf{y}-\mathbf{q})^T AB^{-1}A^T(\mathbf{y}-\mathbf{q}) - \mathbf{r}^T\mathbf{y} \\ \text{s.t.} & \mathbf{y} \geq \mathbf{0} \end{array} \tag{6.13}$$

where the vectors of primal and dual decision variables are respectively denoted $\mathbf{x}$ and $\mathbf{y}$, the coefficient matrix used to define the quadratic cost term is denoted B, the primal cost vector is denoted $\mathbf{q}$, the primal inequality vector is denoted $\mathbf{r}$, and the linear coefficient matrix is denoted A. In writing the dual problem above, we also assume that the quadratic cost matrix B is positive definite for simplicity, thus the problem (QP) is strictly convex and the objective function in (QD) is well-defined. Common algorithms for solving quadratic programs typically make use of interior point or active set techniques. When the inequality constraints all hold with equality, a closed form solution to (6.13)(QP) is possible by directly


Interconnective structures for quadratic programs
$\mathbf{a}_1^{(CR)}$
A
$\mathbf{a}_2^{(CR)}$
$\widehat{Q}_1(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T B\mathbf{x} + \mathbf{q}^T\mathbf{x}$
$\widehat{Q}_1(\mathbf{a}_1^{(CR)}) = \frac{1}{2}\mathbf{a}_1^{(CR)T} B\mathbf{a}_1^{(CR)} + \mathbf{q}^T\mathbf{a}_1^{(CR)}$
$A\mathbf{x} \leq \mathbf{r}$
$\mathbf{a}_2^{(CR)} \leq \mathbf{r}$
$-A^T$
$\mathbf{b}_1^{(CR)}$
$\mathbf{b}_2^{(CR)}$
$-\mathbf{q}$
$(I-B)(I+B)^{-1}$
$\begin{bmatrix} \mathbf{d}_1^{(CR)} \\ \mathbf{d}_2^{(CR)} \end{bmatrix} = G \begin{bmatrix} \mathbf{c}_1^{(CR)} \\ \mathbf{c}_2^{(CR)} \end{bmatrix}$
$|\cdot|$
$-\mathbf{r}$

(a) Stationarity conditions

(b) Transformed stationarity conditions

Figure 6-5: The interconnective structures used to solve quadratic programs in standard form. (a) The stationarity conditions in a canonical-form structure. (b) The transformed stationarity conditions in a scattering-form structure.

solving the KKT conditions since they reduce to a linear system of equations. A quadratic program where B is indefinite is an NP-hard problem, even if B only has a single negative eigenvalue [109].

The standard form of the quadratic program (6.13)(QP) can be reformulated in accordance with the notation used to describe conservative optimization problems. In particular, the reduced-form primal (P) and corresponding dual (D) problems are written according to

$$\begin{array}{llll} & \underset{\mathbf{a}}{\text{minimize}} & \frac{1}{2}\mathbf{a}_1^{(CR)T} B\mathbf{a}_1^{(CR)} + \mathbf{q}^T\mathbf{a}_1^{(CR)} & & \underset{\mathbf{b}}{\text{maximize}} & -\frac{1}{2}(\mathbf{b}_1^{(CR)} - \mathbf{q})^T B^{-1}(\mathbf{b}_1^{(CR)} - \mathbf{q}) - \mathbf{r}^T\mathbf{b}_2^{(CR)} \\ \text{(P)} & \text{s.t.} & A\mathbf{a}_1^{(CR)} = \mathbf{a}_2^{(CR)} & \text{(D)} & \text{s.t.} & -A^T\mathbf{b}_2^{(CR)} = \mathbf{b}_1^{(CR)} \\ & & \mathbf{a}_2^{(CR)} \leq \mathbf{r} & & & \mathbf{b}_2^{(CR)} \geq \mathbf{0} \end{array} \qquad (6.14)$$

where the primal decision variables $(\mathbf{a}_1^{(CR)}, \mathbf{a}_2^{(CR)})$ correspond to $(\mathbf{x}, A\mathbf{x})$. Observe that the conservative dual problem can be reduced using straightforward manipulations into the form of the Lagrangian dual problem (QD). An interconnective description of the stationarity conditions associated with a quadratic program is provided in Figure 6-5(a). An equivalent description of the transformed stationarity conditions formulated as a scattering-form interconnective structure is illustrated in Figure 6-5(b). Referring to the transformed structure, the matrix G is generated from the coefficient matrix A according to (5.91) and the constitutive modules are implemented using the functions

$$\mathbf{c}_1^{(CR)} = (I-B)(I+B)^{-1}(\mathbf{d}_1^{(CR)}-\mathbf{q})-\mathbf{q} \tag{6.15}$$

$$\mathbf{c}_2^{(CR)} = |\mathbf{d}_2^{(CR)}-\mathbf{r}|-\mathbf{r}. \tag{6.16}$$

Note that the expression in (6.15) combines the module established by (5.111) with the second module in Figure 5-7, and the expression in (6.16) combines the third module in Figure 5-6 with the first module in Figure 5-7. Synchronous and asynchronous algorithms then correspond to implementing the interconnective structure representing the transformed stationarity conditions after inserting a valid configuration of state and selecting a protocol for state exchange, as was discussed in Section 3.3. The conic properties of a system operator associated with a quadratic program will generally depend upon the spectral properties of the quadratic cost matrix B. For example, the first constitutive module implemented using (6.15) is α-conic with a value of α that is summarized by Proposition 5.3.1. If B is positive semidefinite then the system operator corresponding to the overall interconnective system is passive everywhere, thus convergence is guaranteed when the delay modules are filtered with an appropriately chosen filtering parameter as is discussed in Theorem 4.3.6.

Quadratic programs appear in many signal processing applications, e.g., through various forms of regularization used in solving ill-conditioned linear inverse problems including Tikhonov regularization, Wiener filtering, total variation denoising in image processing, and the regularization of beamforming patterns [108]. In addition, the basis pursuit problem has been extended using straightforward regularization in the sparse signal recovery context to handle the case where the measurement vector $\mathbf{r}$ is noisy and $A\mathbf{x}$ is only required to be close to $\mathbf{r}$. To enforce this approximation, the basis pursuit denoising problem is formulated by

$$\underset{\mathbf{x}}{\text{minimize}} \quad \tfrac{1}{2}\|A\mathbf{x}-\mathbf{r}\|_2^2+\lambda\|\mathbf{x}\|_1 \tag{6.17}$$

where λ is a tuning parameter that balances the absolute size of the solution with the desired agreement of $A\mathbf{x}$ and $\mathbf{r}$.

The interconnective structures associated with solving the problem in (6.17) are depicted

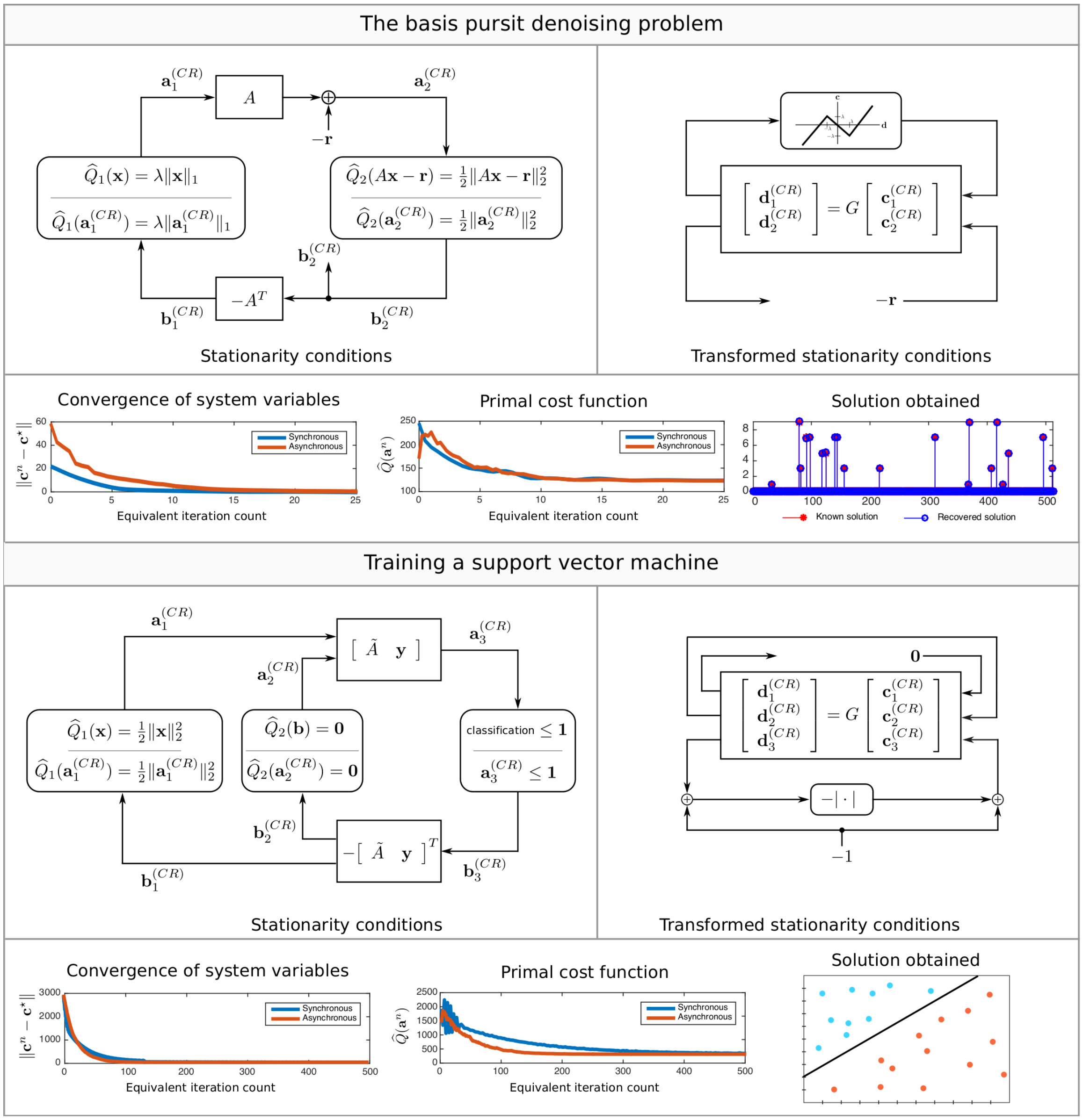

The basis pursit denoising problem
$\mathbf{a}_1^{(CR)}$
A
$\mathbf{a}_2^{(CR)}$
$-\mathbf{r}$
$\widehat{Q}_1(\mathbf{x}) = \lambda\|\mathbf{x}\|_1$
$\widehat{Q}_1(\mathbf{a}_1^{(CR)}) = \lambda\|\mathbf{a}_1^{(CR)}\|_1$
$\widehat{Q}_2(A\mathbf{x}-\mathbf{r}) = \frac{1}{2}\|A\mathbf{x}-\mathbf{r}\|_2^2$
$\widehat{Q}_2(\mathbf{a}_2^{(CR)}) = \frac{1}{2}\|\mathbf{a}_2^{(CR)}\|_2^2$
$\mathbf{b}_2^{(CR)}$
$-A^T$
$\mathbf{b}_1^{(CR)}$
$\mathbf{b}_2^{(CR)}$
Stationarity conditions
c
d
$\begin{bmatrix} \mathbf{d}_1^{(CR)} \\ \mathbf{d}_2^{(CR)} \end{bmatrix} = G \begin{bmatrix} \mathbf{c}_1^{(CR)} \\ \mathbf{c}_2^{(CR)} \end{bmatrix}$
$-\mathbf{r}$
Transformed stationarity conditions
Convergence of system variables
$\|\mathbf{c}^n - \mathbf{c}^\star\|$
0
20
40
60
Synchronous
Asynchronous
0
5
10
15
20
25
Equivalent iteration count
Primal cost function
$\widehat{Q}(\mathbf{a}^n)$
100
150
200
250
Synchronous
Asynchronous
0
5
10
15
20
25
Equivalent iteration count
Solution obtained
0
2
4
6
8
0
100
200
300
400
500
Known solution
Recovered solution
Training a support vector machine
$\mathbf{a}_1^{(CR)}$
$[\ \tilde{A}\ \ \mathbf{y}\ ]$
$\mathbf{a}_3^{(CR)}$
$\mathbf{a}_2^{(CR)}$
$\widehat{Q}_1(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_2^2$
$\widehat{Q}_1(\mathbf{a}_1^{(CR)}) = \frac{1}{2}\|\mathbf{a}_1^{(CR)}\|_2^2$
$\widehat{Q}_2(\mathbf{b}) = \mathbf{0}$
$\widehat{Q}_2(\mathbf{a}_2^{(CR)}) = \mathbf{0}$
classification $\leq \mathbf{1}$
$\mathbf{a}_3^{(CR)} \leq \mathbf{1}$
$\mathbf{b}_2^{(CR)}$
$-[\ \tilde{A}\ \ \mathbf{y}\ ]^T$
$\mathbf{b}_3^{(CR)}$
$\mathbf{b}_1^{(CR)}$
Stationarity conditions
$\mathbf{0}$
$\begin{bmatrix} \mathbf{d}_1^{(CR)} \\ \mathbf{d}_2^{(CR)} \\ \mathbf{d}_3^{(CR)} \end{bmatrix} = G \begin{bmatrix} \mathbf{c}_1^{(CR)} \\ \mathbf{c}_2^{(CR)} \\ \mathbf{c}_3^{(CR)} \end{bmatrix}$
$-|\cdot|$
-1
Transformed stationarity conditions
Convergence of system variables
$\|\mathbf{c}^n - \mathbf{c}^\star\|$
0
1000
2000
3000
Synchronous
Asynchronous
0
100
200
300
400
500
Equivalent iteration count
Primal cost function
$\widehat{Q}(\mathbf{a}^n)$
0
500
1000
1500
2000
2500
Synchronous
Asynchronous
0
100
200
300
400
500
Equivalent iteration count
Solution obtained

Figure 6-6: Example structures used to generate algorithms for solving two quadratic programs.

along the top of Figure 6-7. Referring to the transformed stationarity conditions, the matrix G is generated from the coefficient matrix A according to (5.91) and the constitutive modules are implemented using the functions

$$\mathbf{c}_1^{(CR)} = \begin{cases} -\mathbf{d}_1^{(CR)}, & |\mathbf{d}_1^{(CR)}| \leq \lambda \\ \mathbf{d}_1^{(CR)} - 2\lambda \mathrm{sgn}(\mathbf{d}_1^{(CR)}), & |\mathbf{d}_1^{(CR)}| > \lambda \end{cases} \tag{6.18}$$

$$\mathbf{c}_2^{(CR)} = -\mathbf{r}. \tag{6.19}$$

Note that the expression in (6.18) is the first module in Figure 5-8 with $\rho = \lambda$, and the expression in (6.19) combines the second module in Figure 5-7 with the third module in Figure 5-8. Numerical convergence results and the intermediary values taken by the primal objective function as well as the final solution obtained for both synchronous and asynchronous implementations of the transformed stationarity conditions are also provided. Both implementations used filtered delay elements with parameter $\rho = \frac{1}{2}$ and the asynchronous implementation used a delay probability of $p = 0.75$.

As a second example of a quadratic programming problem, consider the problem of determining an optimal linear classifier given a set of labeled training data that is known to be linearly separable. A popular approach to solving this problem is to train a support vector machine by solving the quadratic program

$$\begin{aligned} \underset{\mathbf{x},\mathbf{b}}{\text{minimize}} \quad & \frac{1}{2}\|\mathbf{x}\|_2^2 \\ \text{s.t.} \quad & \mathbf{y}_n(\mathbf{a}_{(n)}\mathbf{x} + \mathbf{b}) \geq 1, \qquad n = 1, \ldots, N \end{aligned} \tag{6.20}$$

where the binary vector of labels is denoted $\mathbf{y}$ and uses values ±1 to distinguish between two classes, the normal to the hyperplane under design is denoted $\mathbf{x}$ and has an offset denoted $\mathbf{b}$, and the n-th feature vector is denoted $\mathbf{a}_{(n)}$ [110].

The interconnective structures illustrating the transformed and untransformed stationarity conditions associated with (6.20) are depicted in Figure 6-7 where the notation $\tilde{A}$ denotes a matrix whose n-th row corresponds to $\mathbf{y}_n\mathbf{a}_{(n)}$. Referring to the transformed conditions, the matrix G is generated from the coefficient matrix appearing in the stationarity conditions using (5.91) and the constitutive modules are implemented using the functions

$$\mathbf{c}_1^{(CR)} = \mathbf{0} \tag{6.21}$$

$$\mathbf{c}_2^{(CR)} = \mathbf{d}_2^{(CR)} \tag{6.22}$$

$$\mathbf{c}_3^{(CR)} = -|\mathbf{d}_3^{(CR)} - \mathbf{1}| - \mathbf{1}. \tag{6.23}$$

Note that the expression in (6.21) is the third module in Figure 5-8 with $\rho_+ = \rho_- = 1$, the expression in (6.22) is the second module in Figure 5-7 with $\rho = 0$, and the expression in

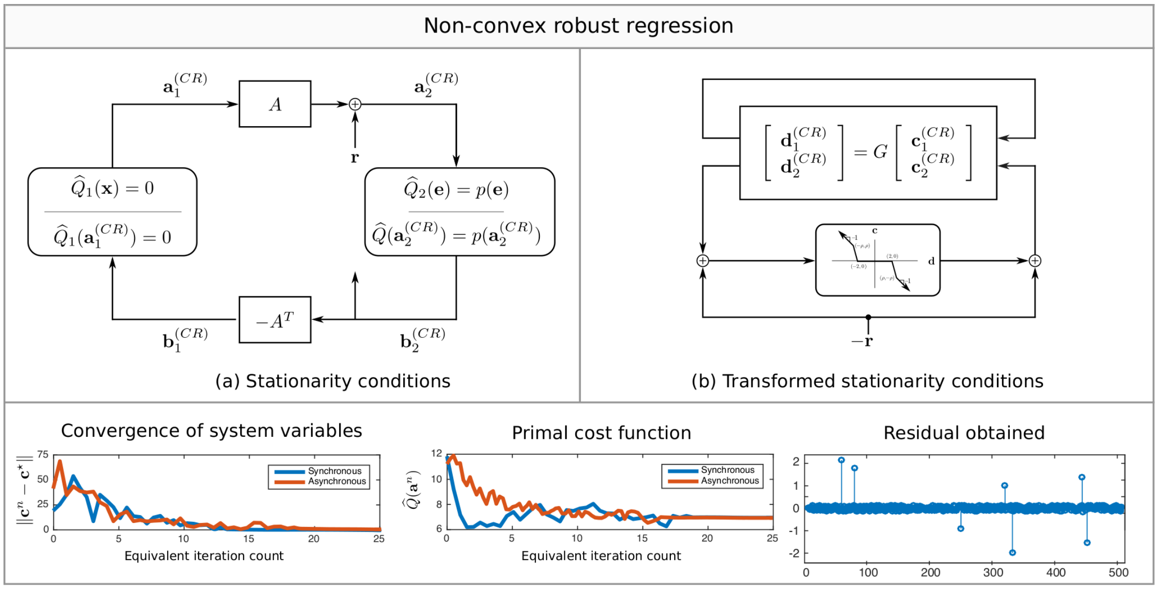
(6.23) was derived as a special case of the first module in Figure 5-6. Numerical convergence results and the intermediary values taken by the primal objective function as well as the final solution obtained for both synchronous and asynchronous implementations of the transformed stationarity conditions are also provided. Both implementations used filtered delay elements with parameter $\rho = \frac{1}{2}$ and the asynchronous implementation used a delay probability of $p = 0.75$. When the training data set is not linearly separable, a soft-margin support vector machine can be formulated in solved in a straightforward way by adding one additional constitutive relation module and augmenting the linear system to reflect the soft-margin variables. A scattering algorithm based on a different formulation of the stationarity conditions for solving the support vector machine problem using decentralized processing nodes arranged into ring graphs can be found in [111].

6.1.4 | Nonconvex approximation problems

Signal approximation problems appear in many signal processing applications, including some of the applications discussed in the previous two subsections. Generally speaking, the problem of identifying the optimal approximation of a signal $\mathbf{r}$ as a linear combination of other signals arranged into the columns of a matrix A can be determined by solving an optimization problem of the form

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \sum_i p(\mathbf{e}_i) \\ \text{s.t.} \quad & A\mathbf{x} - \mathbf{r} = \mathbf{e} \end{aligned} \tag{6.24}$$

where the function $p\colon \mathbb{R} \to \mathbb{R}$ determines the sense in which the approximation is optimal. More concretely, selecting a meaningful approximation penalty function p to assign a cost to the disagreement between the approximation of $\mathbf{r}$ by $A\mathbf{x}$ is especially important when $\mathbf{r}$ is not in the range of A, meaning an inverse or pseudoinverse cannot be used to recover $\mathbf{x}^\star$ and the formulation (6.24) is warranted. In the context of regression problems, the columns of A are the regressors and the vector $A\mathbf{x}^\star$ is the optimal regression of $\mathbf{r}$. The basis pursuit denoising problem (6.17) with $\lambda = 0$ is a standard least squares regression problem where the penalty function p is convex quadratic. In statistical estimation contexts, the problem


Non-convex robust regression
$\mathbf{a}_1^{(CR)}$
A
$\mathbf{a}_2^{(CR)}$
$\mathbf{r}$
$\widehat{Q}_1(\mathbf{x}) = 0$
$\widehat{Q}_1(\mathbf{a}_1^{(CR)}) = 0$
$\widehat{Q}_2(\mathbf{e}) = p(\mathbf{e})$
$\widehat{Q}(\mathbf{a}_2^{(CR)}) = p(\mathbf{a}_2^{(CR)})$
$-A^T$
$\mathbf{b}_1^{(CR)}$
$\mathbf{b}_2^{(CR)}$
$\begin{bmatrix} \mathbf{d}_1^{(CR)} \\ \mathbf{d}_2^{(CR)} \end{bmatrix} = G \begin{bmatrix} \mathbf{c}_1^{(CR)} \\ \mathbf{c}_2^{(CR)} \end{bmatrix}$
$\mathbf{c}$
$\mathbf{d}$
-1
$(-\rho,\rho)$
$(2,0)$
$(-2,0)$
$(\rho,-\rho)$
-1
$-\mathbf{r}$
Convergence of system variables
$\|\mathbf{c}^n - \mathbf{c}^\star\|$
75
50
25
0
0
5
10
15
20
25
Equivalent iteration count
Synchronous
Asynchronous
Primal cost function
$\widehat{Q}(\mathbf{a}^n)$
12
10
8
6
0
5
10
15
20
25
Equivalent iteration count
Synchronous
Asynchronous
Residual obtained
2
1
0
-1
-2
0
100
200
300
400
500

(a) Stationarity conditions

(b) Transformed stationarity conditions

Figure 6-7: Example interconnective structures used to solve nonconvex regression problems.

(6.24) is interpreted as estimating a vector $\mathbf{x}$ where the vector $\mathbf{r}$ was generated as a noisy version of $A\mathbf{x}$ and p penalizes the size of the noise.

Selecting a convex penalty function p corresponds to penalizing large noise or error values greater than smaller ones, and the discussion in Section 5.2.3 ensures that the functional realization of the transformed constitutive module for convex penalty functions will exist. When the noise or errors in $\mathbf{r}$ consist of a few outliers, it may be desirable to discard them in the approximation so as not to overfit to a particular vector $\mathbf{r}$. However, penalizing large errors with no cost would result in solutions that form poor approximations, so a balance must be reached. To do this, a non-convex penalty function can be designed that increasingly penalizes small errors and assigns a constant penalty to any error above a certain threshold. In line with this, the constitutive modules defined in Figure 5-9 have this behavior where the small errors are penalized quadratically, and the transformed stationarity conditions associated with the problem (6.24) are illustrated in Figure 5-9 using the module in the top row.

Referring to the transformed stationarity conditions, the matrix G is generated from the coefficient matrix A according to (5.91) and the constitutive modules are implemented using the functions

$$\mathbf{c}_1^{(CR)} = \mathbf{d}_1^{(CR)} \tag{6.25}$$

$$\mathbf{c}_2^{(CR)} = m(\mathbf{d}_2^{(CR)} - \mathbf{r}) - \mathbf{r} \tag{6.26}$$

where m is given by the fourth column in Figure 5-9. Numerical convergence results and the intermediary values taken by the primal objective function as well as the final residual or approximation error obtained for both synchronous and asynchronous implementations of the transformed stationarity conditions are also provided. Observe that the residual is essentially uniformly distributed with a few outliers. Both implementations used filtered delay elements with parameter $\rho = \frac{1}{2}$ and the asynchronous implementation used a delay probability of $p = \frac{3}{4}$. Note that by changing the function in (6.25) to correspond to a regularization penalty on $\mathbf{x}$ we can obtain a broad range of non-convex regularization problems from the signal approximation problem formulated in (6.24).

6.2 | Scattering algorithms implemented as web services

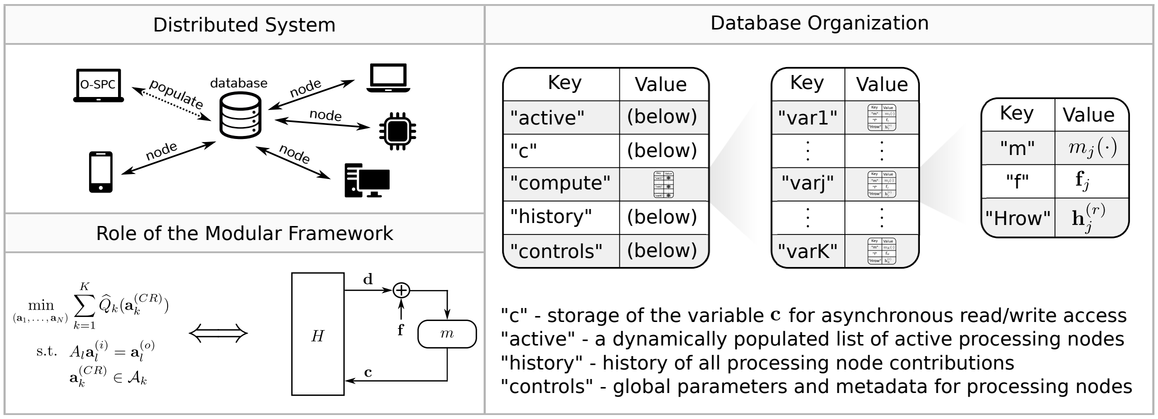
The assignment of processing instructions and distribution of state are perhaps two of the most important tasks in effectively distributing an algorithm, especially onto heterogeneous networks of processors. In response to this observation, the authors in [51] advocate for design patterns in which an algorithm is first specified using a declarative language and second, after selecting a protocol for distributing and exchanging state, implemented on processors using imperative languages. Consistent with this approach, the interconnective framework, in particular the state-free design of signal processing structures by connecting constitutive modules and interconnecting networks together, can be interpreted as one such declarative language for which the issues related to distributing state have already been addressed. When paired with asynchronous communication protocols for state or memory exchange between processors, such as the communication protocol in Definition 3.3.1, the resulting algorithms can be viewed as ensembles of distributable, asynchronous programs realized as signal processing systems.

In this section, we discuss the implementation of scattering algorithms as web services by organizing scattering-form interconnective structures into processing tasks that can be stored and served using databases. In what follows, we use the term *node* to refer to any processing resource that is able to access the database and execute the received instructions, e.g. nodes may correspond to virtual or physical processors. A key benefit to designing signal processing systems as web services is that properties of the database can be used to generate scalable implementations in the sense that the processing tasks can be dynamically assigned to processing nodes where the availability, synchronization, and guaranteed response time of any particular node is unnecessary. For the purpose of illustration, the focus in this section is to describe the web-service implementation that appears in [106]. This service, which suggests opportunity in designing future signal processing systems in a similar way, is publicly available at `http://optimization.spconservation.org` and will be referred to herein as “O-SPC” [112]. As opposed to requiring specialized servers, the philosophy behind O-SPC is to make use of a commodity database back-end as a central resource for exchanging state, as might be used to serve data for websites with large numbers of concurrent users. Subsequently, the implementations in this section inherit the scalability, resilience, and security of these database systems.

By focusing on the connection between databases as storage systems and asynchronous delay modules in interconnective structures, the implementations described in this section can effectively make use of a variety of available computing resources while being able to adapt to a variety of distributed computing issues. For example, the implementations are able to adapt in real-time to changing processing demands, time-varying network congestion, resource outages, and similar disturbances that occur in heterogeneous networks that lack continuous connectivity. Also, the algorithms that fit into this framework do not require the network to have failure recovery mechanisms or only fully functional processing nodes, as is common in realistic models of the Internet.

6.2.1 | Database organizations of interconnective structures

As was discussed previously, solving a large class of convex and nonconvex optimization problems can be recast as solving a fixed-point problem described as a CCSP where the


Distributed System
O-SPC
populate
database
node
node
node
node
Role of the Modular Framework
$\min_{(\mathbf{a}_1,\dots,\mathbf{a}_N)} \sum_{k=1}^{K} \widehat{Q}_k(\mathbf{a}_k^{(CR)})$
s.t. $A_l\mathbf{a}_l^{(i)} = \mathbf{a}_l^{(o)}$
$\mathbf{a}_k^{(CR)} \in \mathcal{A}_k$
$\Longleftrightarrow$
H
$\mathbf{d}$
$\mathbf{f}$
m
$\mathbf{c}$
Database Organization

Key	Value
"active"	(below)
"c"	(below)
"compute"	
"history"	(below)
"controls"	(below)

Key	Value
"var1"	
⋮	⋮
"varj"	
⋮	⋮
"varK"	

Key	Value
"m"	$m_j(\cdot)$
"f"	$\mathbf{f}_j$
"Hrow"	$\mathbf{h}_j^{(r)}$

"c" - storage of the variable $\mathbf{c}$ for asynchronous read/write access
"active" - a dynamically populated list of active processing nodes
"history" - history of all processing node contributions
"controls" - global parameters and metadata for processing nodes

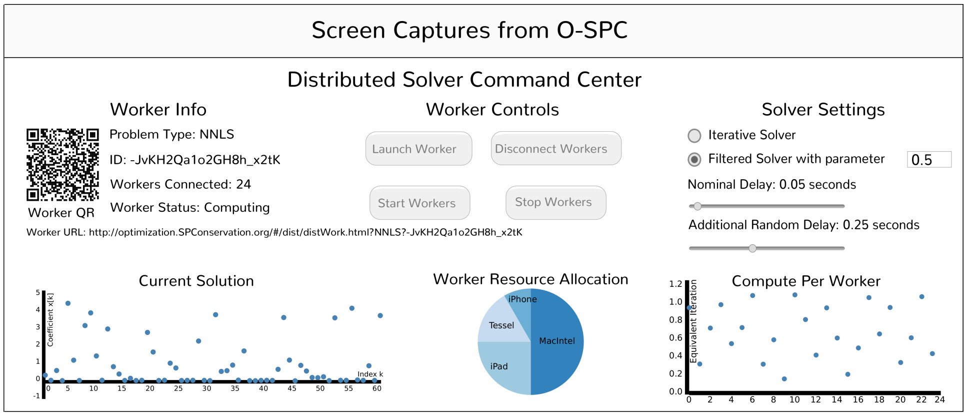
Figure 6-8: Left: the distributed signal processing system utilized by O-SPC to solve optimization problems whose transformed stationarity conditions are organized into the depicted source-free structure using the modular framework. Right: a qualitative description of the organization scheme of the processing instructions and the signals to be processed into a generic database.

overall goal is specifically to identify a solution $(\mathbf{c}^\star, \mathbf{d}^\star)$ to a nonlinear system of equations taking the form

$$\mathbf{c}^\star = m(\mathbf{d}^\star) \quad \text{and} \quad \mathbf{d}^\star = H\mathbf{c}^\star + \mathbf{f}. \tag{6.27}$$

Figure 6-8 illustrates the interconnective description of (6.27) into a generic key-value store, i.e. a non-relational database, for implementation on a distributed system similar to the one depicted on the top of the left pane. Implementations of this distributed system are paired with a communication or state exchange protocol consisting of the processing nodes asynchronously accessing the database to retrieve a subset of the processing instructions and the associated signals to be processed, processing these signals, and asynchronously writing the results back into the database. This operating principle can be viewed as a form of object-oriented signal processing where the objects contain data in the form of the signals to be processed and methods in the form of the processing instructions.

The O-SPC service provides an interactive dashboard interface through which problem descriptions can be provided and metaparameters for the problem can be set. This input is then used to generate a corresponding uniform resource locator (URL) through which processing nodes can attach to the problem instance to perform computation. For devices with integrated cameras, a quick response (QR) code is also dynamically generated. In terms



Screen Captures from O-SPC

Distributed Solver Command Center

Worker Info

Problem Type: NNLS

ID: -JvKH2Qa1o2GH8h_x2tK

Workers Connected: 24

Worker Status: Computing

Worker URL: http://optimization.SPConservation.org/#/dist/distWork.html?NNLS?-JvKH2Qa1o2GH8h_x2tK

Solver Settings

○ Iterative Solver

● Filtered Solver with parameter 0.5

Nominal Delay: 0.05 seconds

Additional Random Delay: 0.25 seconds

Figure 6-9: Screen captures from O-SPC illustrate the global controller and computed solution of a non-negative least squares problem obtained using 24 distributed processing nodes (referred to as workers) implementing an iterative filtered solver with parameter $\rho = 0.5$. A breakdown of the workers computational platform allocation and individual contributions to the overall solution is also depicted.

of using a variety of available computing resources, any computational resource equipped with network access and a basic JavaScript engine can be utilized as a processing node by O-SPC. For example, a heterogeneous set of processing nodes might include modern web browsers on mobile, tablet and desktop machines as well as special-purpose JavaScript-enabled embedded systems [113, 114].

Screen captures from the O-SPC application interface are provided in Figure 6-9 for the non-negative least squares problem (5.99) and depict the dashboard through which the distributed processing nodes can be controlled. The particular solution depicted in Figure 6-9 was obtained using 24 processing nodes and analytics regarding the computational platforms of the nodes as well as their individual contributions to the overall optimization progress are provided via dynamically-generated graphs.

The centralized data store that O-SPC in particular is built upon is Firebase [115]. O-SPC utilizes this service primarily as a high-performance back-end for asynchronous state transfer between browser-based clients, e.g. as opposed to as a centralized resource for coordinating data processing as with [116]. In this sense, O-SPC represents an example of how the interconnective framework can be used to create a performant system operating in the somewhat extreme case where all numerical computation is executed by the extremities of the computing graph. The considerations about O-SPC described in this section would similarly apply to developing web-based optimization services using alternative key-value store systems, e.g. MongoDB [117] or Redis [118], or any number of relational database systems. In each of these cases, the processing instructions would be able to draw upon the particular strengths of the data store being utilized.

6.2.2 | Processing instructions for distributed implementations

In this subsection, the processing instructions used to execute the distributed system discussed in the previous subsection are developed to obtain a solution $(\mathbf{c}^\star, \mathbf{d}^\star)$ to (6.27) once a problem instance has been organized into a database as demonstrated by Figure 6-8. In discussing these processing instructions, we shall refer to an *iterative implementation* as a list of processing instructions that directly produce new state values to be asynchronously written into the database and overwrite the values currently stored there. This is in contrast to an *incremental implementation*, which is discussed in the following subsection, where the processing instructions produce an increment to be added to the state values currently in the data store. We comment upfront that the specific form of the implementations, both iterative and incremental, presented in this section differ from their actual implementation on O-SPC in that they have been adapted here for the purpose of clarity rather than computational efficiency. For the purpose of illustration, we additionally assume that the memoryless nonlinearity $m(\cdot)$ in (6.27) operates coordinatewise on its argument; this assumption holds for most of the modules derived in Chapter 5. The modifications needed to handle general nonlinearities follow in a straightforward way where the separability of $m(\cdot)$ is a consequence of the separability implied by an associated partition decomposition.

The processing instructions and corresponding initialization procedures associated with filtered and unfiltered versions of a distributed, iterative implementation are summarized in Algorithm 6.1. Specifically, each processing node, independent of any and all other processing nodes, performs the following steps until a suitable stopping criterion is met in order to implement the filtered version of the iterative implementation:

(1) generate a random integer $j \in \{1, \dots, K\}$ corresponding to the state variables $\mathbf{c}_j$ and $\mathbf{d}_j$ to be processed;

Algorithm 6.1 The initialization procedure for individual processing nodes and the corresponding processing instructions associated with filtered and unfiltered versions of an iterative implementation.

```
Processing node initialization procedure:
  (1) Read metadata from the database
  (2) Set flags for asynchronous database access
  (3) Listen for the start processing signal
Processing instructions for Filtered and Unfiltered versions of an Iterative
implementation:
  (1) Select a random integer $j \in \{1, \ldots, K\}$
  (2) Request "c" and "varj" from the database and remain idle until they are received
  (3) Compute $\mathbf{d}_j \leftarrow \mathbf{f}_j + \langle \mathbf{h}_j^{(r)}, \mathbf{c} \rangle$
  (4) Compute $\mathbf{c}_j \leftarrow \rho m_j(\mathbf{d}_j) + (1 - \rho)\mathbf{c}_j$
  (5) Send the update $\mathbf{c}_j$ to "c" in the database
  (6) Go to (1)
```

(2) read from the database the current state of the vector $\mathbf{c}$ as well as the object `varj` consisting of a characterization of the nonlinearity m_j labeled `m`, the value of $\mathbf{f}_j$ labeled `f`, and the row vector $\mathbf{h}_j^{(r)}$ corresponding to the j-th row of H labeled `Hrow`;

(3) generate the intermediary state value $\mathbf{d}_j$ according to

$$\mathbf{d}_j \leftarrow \mathbf{h}_1^{(j)}\mathbf{c}_1 + \cdots + \mathbf{h}_K^{(j)}\mathbf{c}_K + \mathbf{f}_j; \tag{6.28}$$

(4) generate the new state value $\mathbf{c}_j$ according to

$$\mathbf{c}_j \leftarrow \rho m_j\left(\mathbf{d}_j\right) + (1 - \rho)\mathbf{c}_j \tag{6.29}$$

where the filter parameter ρ is a metaparameter obtained during the initialization phase;

(5) asynchronously write the new state value $\mathbf{c}_j$ into the j-th position of $\mathbf{c}$ in the database.

Note that the state vector $\mathbf{d}$ is not required to be explicitly stored by the database to execute either version of the iterative implementation. Once the partial solution $\mathbf{c}^\star$ has been identified, the remainder of the full solution can be generated using (6.27) thereby effectively solving the optimization or constraint satisfaction problem. Referring again to Algorithm 6.1, the processing instructions for an unfiltered realization of the iterative implementation correspond to modifying the instructions above by setting $\rho = 1$ in (6.29).

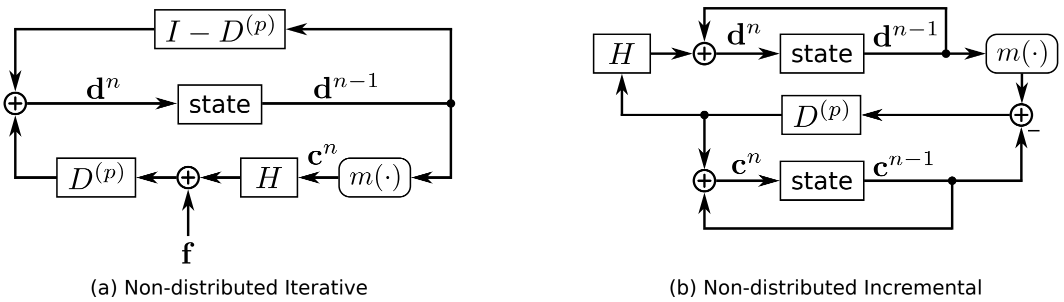
We call special attention to the fact that the processing instructions above and the uncoordinated communication protocol used by processing nodes to access the database system make no attempt to regulate global task allocation nor to enforce concurrency rules of any form. Specifically, the data requests and updates are respectively executed using asynchronous read and write operations with no explicitly specified concept of precedent or preference between the various processing nodes. Specifically, the data requests and updates are respectively executed using asynchronous read and write operations with no concept of precedent or preference between the various processing nodes. For example, if multiple nodes request data associated with the same state variable $\mathbf{c}_j$ and each experience a different latency (and thus each possibly retrieves different state vectors $\mathbf{c}$), then the database records the updates in the order they are received independent of the order of the read operations. The stability and robustness analysis performed in Chapter 4 is in agreement with this protocol by augmenting the system operator to include a noise source introduced at the input to the state elements.

Referring to Figure 6-8, the database where state is transferred and interconnective structures are stored might simultaneously contain numerous active problem instances. This includes the possibility of many unrelated problems or several related problems where, for example, the interconnecting network is the same but the constitutive relation modules are different. The latter might be warranted in settings where the solution to numerous instances of the same problem with different tuning parameters or regularization terms is desired. Processing nodes can be added or removed from any problem instance at any time, including changing between problems, without causing any failures or necessitating coordination since nodes are never assigned responsibility for any particular share of the workload. In this sense, O-SPC is well-suited for computing environments where processing nodes have other primary tasks or are frequently interrupted or reboot. Another advantage to using the presented approach to implement large-scale optimization algorithms in practice is the ability to update the portion of the database, and by extension the interconnective structure as well, associated with measurements and observations as new data becomes available. The response of the distributed system is to naturally transform the state of the database associated with the current solution toward the new fixed-point or invariant state corresponding to the new solution. Consequently, the distributed processing instructions summarized by Algorithm 6.1 are sufficient to solve a broad class of optimization problems over delay or disruption tolerant networks and do not rely on the availability or synchronization of any particular processing nodes.

6.2.3 | Processing instructions for non-distributed implementations

The toolset in O-SPC also provides support for iterative and incremental implementations of the scattering algorithms associated with (6.27) as non-distributed systems. These non-distributed implementations are formed by organizing and implementing the interconnective description of the problem using a single-threaded JavaScript engine as the sole processing node. This computing environment is essentially mathematically equivalent to parallel and distributed computing environments with dedicated and synchronized processing resources. For example, if the processing nodes in a distributed system all communicate with each other and coordinate their computations at every time instance, the processing system progresses at the rate of the slowest processor and communication link in the same fashion a non-distributed system would by progressing after an entire iteration has completed. Compared to the distributed implementations discussed in the previous subsection, highly synchronized distributed systems can be carefully tuned to produce a significant acceleration in absolute convergence time and decrease in overall communication bandwidth as compared to the setting discussed previously. This is possible in part by the ability of the highly synchronized system to assigning collectively-exhaustive subsets of the objects stored in `compute` in Figure 6-8 to the available processing nodes during the initialization phase. Then, the only communication required for implementing the algorithm involves the state vector $\mathbf{c}$ and possibly the state vector $\mathbf{d}$. In the remainder of this subsection, we define the processing instructions for use in non-distributed settings which include by proxy highly-synchronized and distributed systems as well.

The interconnective structures describing the iterative and incremental implementations of an asynchronous scattering algorithm are depicted in Figure 6-10. More formally, the



(a) Non-distributed Iterative

(b) Non-distributed Incremental

Figure 6-10: The interconnective structures associated with the (a) iterative and (b) incremental non-distributed implementations. Filtered realizations correspond to simply replacing the depicted state elements with filtered state elements.

processing used to directly produce the state sequence $\{\mathbf{d}^n \in \mathbb{R}^K : n \in \mathbb{N}_0\}$ given by

$$\mathbf{d}^n = D^{(p)} \left(Hm \left(\mathbf{d}^{n-1} \right) + \mathbf{f} \right) + (I_K - D^{(p)})\mathbf{d}^{n-1}, \qquad n \in \mathbb{N}, \tag{6.30}$$

corresponds to an iterative implementation and instructions are provided as an interconnective structure in Figure 6-10(a) where $D^{(p)}$ is the stochastic matrix from Definition 3.3.1 that models the coordinate-wise discrete-time sample-and-hold state elements triggered by discrete-time Bernoulli processes. Reorganizing this iteration and modifying the initial conditions such that the first difference of the signals rather than the signals themselves are being processed as mentioned earlier in the chapter on CCSPs results in an incremental implementation where the state sequence $\mathbf{c}^n$ is generated according to $\mathbf{c}^n = m(\mathbf{d}^{n-1})$ and the state sequence $\mathbf{d}^n$ is generated according to

$$\mathbf{d}^n = \mathbf{d}^{n-1} + HD^{(p)} \left(m \left(\mathbf{d}^{n-1} \right) - \mathbf{c}^{n-1} \right), \qquad n \in \mathbb{N}. \tag{6.31}$$

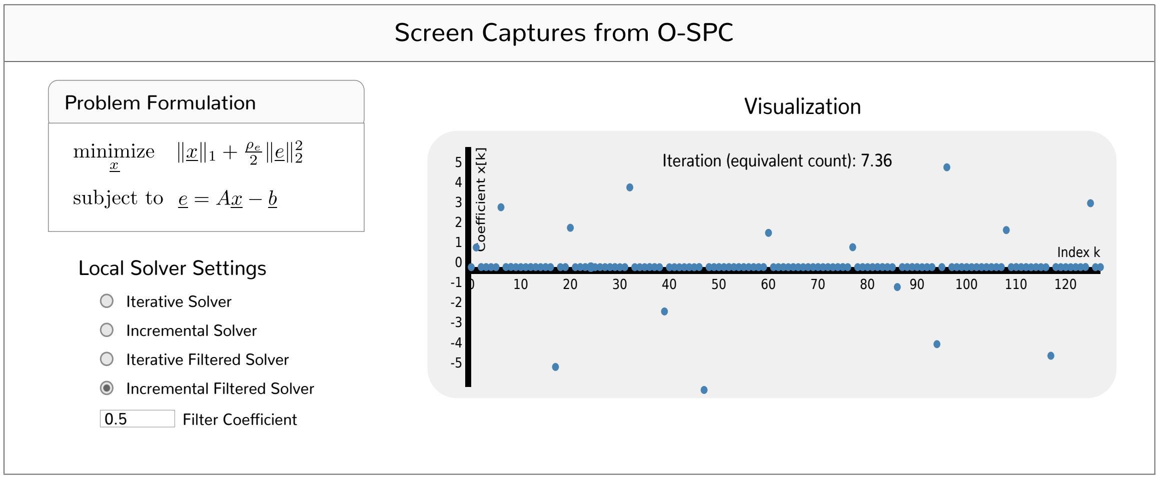
The system initialization procedure and corresponding processing instructions for these non-distributed implementations and their filtered counterparts are summarized in Algorithm 6.2. These processing instructions in particular form the basis for the embedded implementations discussed in the next section.

Screen captures from the non-distributed O-SPC application interface are provided in Figure 6-11 for a LASSO or basis pursuit denoising problem, which differs from the non-negative least squares problem discussed in Section 5.2 by changing a single constitutive

Algorithm 6.2 The procedures for system initialization and processing associated with four non-distributed implementations.

```
Implementation type:  Iterative and Iterative Filtered
Processing node initialization:
    (1) $\mathbf{d}^0 \leftarrow \mathbf{0}$
Processing instructions:
    (1) Compute $\mathbf{c}^n \leftarrow m(\mathbf{d}^{n-1})$
    (2) Compute $\mathbf{d}^n \leftarrow D^{(p)}(\rho(H\mathbf{c}^n + \mathbf{f}) + (1-\rho)\mathbf{d}^{n-1}) + (I_K - D^{(p)})\mathbf{d}^{n-1}$

Implementation type:  Incremental and Incremental Filtered
Processing node initialization:
    (1) $\mathbf{d}^{-1} = \mathbf{d}^0 \leftarrow \mathbf{f}$
Processing instructions:
    (1) Compute $\mathbf{c}^n \leftarrow m(\mathbf{d}^{n-1})$
    (2) Compute $\mathbf{d}^n \leftarrow \mathbf{d}^{n-1} + \rho H(\mathbf{c}^n - \mathbf{c}^{n-1}) + (1-\rho)(\mathbf{d}^{n-1} - \mathbf{d}^{n-2})$
```



Screen Captures from O-SPC

Problem Formulation

$$\underset{\underline{x}}{\text{minimize}} \quad \|\underline{x}\|_1 + \frac{\rho_e}{2}\|\underline{e}\|_2^2$$

$$\text{subject to} \quad \underline{e} = A\underline{x} - \underline{b}$$

Local Solver Settings

○ Iterative Solver

○ Incremental Solver

○ Iterative Filtered Solver

● Incremental Filtered Solver

0.5 Filter Coefficient

Figure 6-11: Screen captures from the O-SPC application interface illustrate the solution to a LASSO or basis pursuit denoising problem [3] obtained by running an incremental filtered solver with filter parameter $\rho = 0.5$ and asynchronous delay probability $p = 0.25$.

module. Through this interface, the type of implementation, value of the filtering parameter, and value of the asynchronous delay probability p may be dynamically selected. The particular solution depicted corresponds to an incremental filtered implementation with filtering parameter $\rho = 0.5$ and delay probability $p = 0.25$. Extensions of the incremental implementations to the distributed setting follow in a straightforward way when the database storing the signals to be processed possesses increment operations in addition to the write operations available in [115].

6.3 | Scattering algorithms implemented as embedded systems

Asynchronous designs in the context of embedded and hardware systems have experienced continuously growing interest throughout the last two decades, especially as engineers tackle the broad range of challenging trends that have arisen during the current "late-Moore" era [119]. In the field of semiconductor design, the asynchronous design paradigm specifically refers to the blending of synchronous and asynchronous modules through handshaking networks in order to communicate and synchronize data exchanges between the modules. Asynchronous designs are typically modular and extensible with on-demand operation and straightforward power management control [120]. These advantages help designers to address increasing process variability, power and thermal bottlenecks, high fault rates, deteriorated performance due to aging, and scalability in densely packed integrated circuits as processes technologies continue to shrink. Cryptographic applications such as the ability to uniquely identify individual devices are enabled by the lack of a coherent power or electromagnetic emission signature due to the asynchronous nature of these systems.

In this section, we discuss some of the key concerns related to the design of hardware architectures as they pertain to the design and implementation of scattering algorithms realized as embedded signal processing systems. In particular, we first review the foundations of communication protocols and handshaking networks that are important to the performance of two types of embedded systems: multi-synchronous and Globally-Asynchronous Locally-Synchronous (GALS) systems. Multi-synchronous systems distribute a globally shared clock to each module so that the individual modules can subsample the clock to optimize for power and performance [121] while GALS like systems allow each module to possess their own clock, thereby allowing each module to independently optimize for power and performance [122]. After discussing the relevant background, the focus turns to how the stability and robustness properties of interconnective structures and scattering algorithms can be used to inform design choices in the hardware design process. To discuss this, we illustrate the key points using two hardware architectures that take advantage of these properties to reduce their overall communication overheard in terms of performance and hardware complexity.

6.3.1 | Foundations of handshaking networks and communication protocols

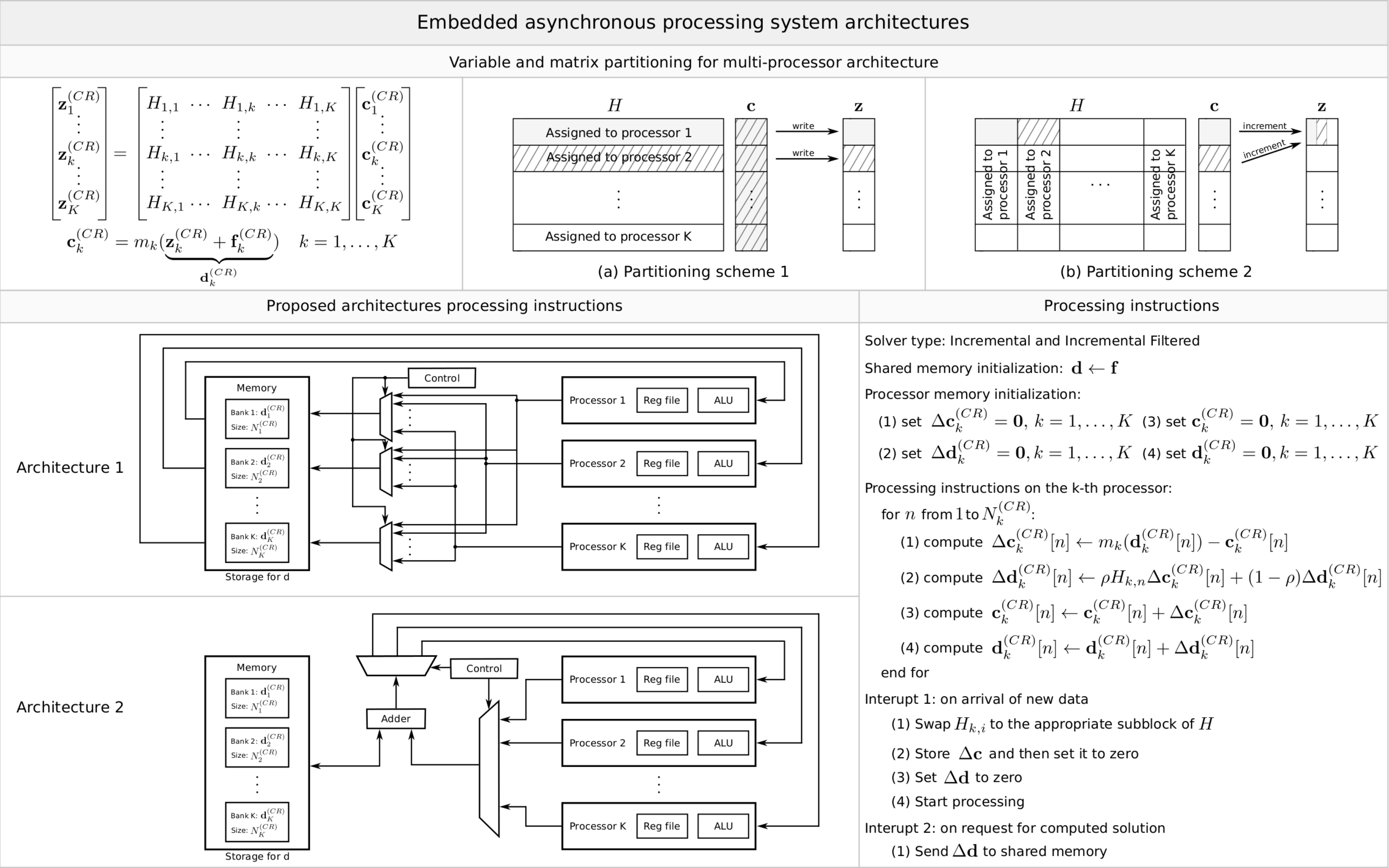
Asynchronous hardware systems are typically assembled by connecting a basic set of components or modules together where the modules exchange data with one another through a handshaking network. Handshaking networks consist of a collection of communication channels going between the modules that need to exchange data, and, generally speaking, there are two key parameters to consider in the design of each channel: (i) a communication protocol, and (ii) a data encoding scheme. The communication protocol is used to communicate request and acknowledgment messages between the sending and receiving modules, and typically consist of two- or four-phase message exchanges. Two modules connected by an asynchronous channel must also agree upon an encoding scheme so that their exchanged signal waveforms can be correctly interpreted. For the purpose of discussing embedded computing systems in this thesis, we focus only on the communication protocol and assume standard data encoding schemes can be used. Once these parameters have been selected, an asynchronous channel between two modules can be pipelined to increase the communication efficiency between the modules.

To implement a large and asynchronous embedded system, there are two additional issues to consider and are commonly referred to as synchronization and arbitration. The issue of synchronization deals with how two modules running at different clock rates choose to transfer data to one another and can be dealt with, for example, by using standard rate conversion systems to align the two clock domains, thereby avoiding setup time violations that may result in metastable operations or outright communication failure. The issue of arbitration deals with how multiple modules compete with one another for shared resources like access to centralized memory. This particular issue is generally solved by using some form of locking or mutex control where the processors who are not granted access remain idle until the shared resource becomes available again. For many types of processing tasks, inefficiencies attributed to arbitration form a major component of the total inefficiency of an implementation. Common solutions to reducing the penalties associated with power consumption on battery-operated devices is to power down idle modules during these periods. For the purpose of embedded signal processing systems in this thesis, we focus only on rules for arbitration and assume a variety of standard data synchronization schemes can readily be used between modules.

6.3.2 | Hardware architectures for embedded scattering algorithms

In this subsection, we discuss the two hardware architectures depicted in Figure 6-12 for asynchronously implementing the scattering algorithms used to solve (6.27). The key features underlying the architectures permit multiple processors to efficiently contribute to the overall task without excessively coordinating the shared memory resources. This is possible since each processor in the architectures is not required to exchange data with the other processors by certain times. Also, processors are able to continuously perform computation without waiting for send, receive, or acknowledge signals, thus the processors do not need to remain idle at any time. This also eliminates inefficiencies when some of the processors can perform computation faster than others, e.g. when some processors have other primary tasks. In this sense, the implementation benefits of GALS-like systems can be achieved by multi-synchronous systems by relying on the simple communication protocols allowed between modules that result from the use of a global clock.

The approach to achieving the computational efficiencies discussed above is to partition the matrix H in (6.27) onto various processors since multiplying by the full matrix H might otherwise be a computationally and memory intensive part of an iteration, especially if H is too large to fit into the available low-latency memory. The algorithmic mapping of the problem in (6.27) onto the two presented architectures is schematically illustrated in the top row of Figure 6-12. Selecting between the two hardware architectures then boils down to issues related to problem size and whether an iterative or incremental implementation of the scattering algorithm is desired. Referring to the two partitioning schemes in the top row of the figure, the schemes labeled (a) and (b) highlight the assignment of matrix-vector product computations for row and column partitioning of the matrix H onto a total of K processors. Specifically, each processor in scheme (a) requires much more hardware than in scheme (b) due to the need to access and compute with the full vector $\mathbf{c}$. To facilitate solving optimization problems with large numbers of decision variables, we proceed focused on scheme (b).



Embedded asynchronous processing system architectures

Variable and matrix partitioning for multi-processor architecture

$$\begin{bmatrix} \mathbf{z}_1^{(CR)} \\ \vdots \\ \mathbf{z}_k^{(CR)} \\ \vdots \\ \mathbf{z}_K^{(CR)} \end{bmatrix} = \begin{bmatrix} H_{1,1} & \cdots & H_{1,k} & \cdots & H_{1,K} \\ \vdots & & \vdots & & \vdots \\ H_{k,1} & \cdots & H_{k,k} & \cdots & H_{k,K} \\ \vdots & & \vdots & & \vdots \\ H_{K,1} & \cdots & H_{K,k} & \cdots & H_{K,K} \end{bmatrix} \begin{bmatrix} \mathbf{c}_1^{(CR)} \\ \vdots \\ \mathbf{c}_k^{(CR)} \\ \vdots \\ \mathbf{c}_K^{(CR)} \end{bmatrix}$$

$$\mathbf{c}_k^{(CR)} = m_k(\underbrace{\mathbf{z}_k^{(CR)} + \mathbf{f}_k^{(CR)}}_{\mathbf{d}_k^{(CR)}}) \quad k = 1, \ldots, K$$

(a) Partitioning scheme 1

(b) Partitioning scheme 2

Proposed architectures processing instructions

Architecture 1
Memory
Bank 1: $\mathbf{d}_1^{(CR)}$ Size: $N_1^{(CR)}$
Bank 2: $\mathbf{d}_2^{(CR)}$ Size: $N_2^{(CR)}$
Bank K: $\mathbf{d}_K^{(CR)}$ Size: $N_K^{(CR)}$
Storage for d
Control
Processor 1
Reg file
ALU
Processor 2
Reg file
ALU
Processor K
Reg file
ALU
Architecture 2
Memory
Bank 1: $\mathbf{d}_1^{(CR)}$ Size: $N_1^{(CR)}$
Bank 2: $\mathbf{d}_2^{(CR)}$ Size: $N_2^{(CR)}$
Bank K: $\mathbf{d}_K^{(CR)}$ Size: $N_K^{(CR)}$
Storage for d
Adder
Control
Processor 1
Reg file
ALU
Processor 2
Reg file
ALU
Processor K
Reg file
ALU

Processing instructions

Solver type: Incremental and Incremental Filtered

Shared memory initialization: $\mathbf{d} \leftarrow \mathbf{f}$

Processor memory initialization:

(1) set $\Delta\mathbf{c}_k^{(CR)} = \mathbf{0},\ k = 1, \ldots, K$ (3) set $\mathbf{c}_k^{(CR)} = \mathbf{0},\ k = 1, \ldots, K$

(2) set $\Delta\mathbf{d}_k^{(CR)} = \mathbf{0}, k = 1, \ldots, K$ (4) set $\mathbf{d}_k^{(CR)} = \mathbf{0}, k = 1, \ldots, K$

Processing instructions on the k-th processor:

```
for $n$ from 1 to $N_k^{(CR)}$:
    (1) compute $\Delta\mathbf{c}_k^{(CR)}[n] \leftarrow m_k(\mathbf{d}_k^{(CR)}[n]) - \mathbf{c}_k^{(CR)}[n]$
    (2) compute $\Delta\mathbf{d}_k^{(CR)}[n] \leftarrow \rho H_{k,n} \Delta\mathbf{c}_k^{(CR)}[n] + (1-\rho)\Delta\mathbf{d}_k^{(CR)}[n]$
    (3) compute $\mathbf{c}_k^{(CR)}[n] \leftarrow \mathbf{c}_k^{(CR)}[n] + \Delta\mathbf{c}_k^{(CR)}[n]$
    (4) compute $\mathbf{d}_k^{(CR)}[n] \leftarrow \mathbf{d}_k^{(CR)}[n] + \Delta\mathbf{d}_k^{(CR)}[n]$
end for
```

Interupt 1: on arrival of new data

(1) Swap $H_{k,i}$ to the appropriate subblock of H

(2) Store $\Delta\mathbf{c}$ and then set it to zero

(3) Set $\Delta\mathbf{d}$ to zero

(4) Start processing

Interupt 2: on request for computed solution

(1) Send $\Delta\mathbf{d}$ to shared memory

Figure 6-12: The two embedded hardware architectures used to solve the problem (6.27) using iterative and incremental implementations of the associated scattering algorithms with a focus on the partitioning of the matrix H since multiplying by it can be computationally expensive.

The matrix partitioning formulation described by scheme (b) is well-suited to implementing an incremental implementation as indicated previously in Figure 6.2(b). From the perspective of hardware design, incremental subcomputations in the asynchronous paradigm can be implicitly handled by the coordination logic between the memory module and the individual processors thus simplifying the control logic needed in their interface. Taking advantage of this, the hardware architectures in Figure 6-12 do not require a complex communication protocol or sophisticated arbitration rules. Referring again to Figure 6-12, each processor implements the processing instructions provided for each of the architectures. The state vector $\mathbf{c}$ is stored in globally accessible system memory implemented using, for example, registers for Architecture 1 and Static Random Access Memory (SRAM) for Architecture 2 depending on the size of the problem. The control logic for both architectures is to use the depicted multiplexers to implement the mapping implied by scheme (b) by cycling through to produce the K increments whose summation forms $\mathbf{z}_k^{(CR)}$, for $k = 1, \ldots, K$, in any way that is collectively exhaustive, i.e. produces each increment of each subvector of $\mathbf{z}$ after a reasonable number of iterations.

6.4 | Summary, conclusions, and future directions

In this thesis, an interconnective framework for describing and manipulating behavioral models of large-scale, decentralized signal processing systems was developed, with emphasis on the stability, robustness, and variational properties of these systems. The framework was also used to establish a class of distributed, asynchronous algorithms, referred to as scattering algorithms, for solving fixed-point and optimization problems. These algorithms are realizable as signal processing systems by connecting basic modules together in a plug-and-play fashion, where mean-square convergence for convex optimization problems is guaranteed without requiring problem-specific tuning or step-size parameters. In this section, we summarize the main contributions of this thesis and review some of the directions for future research that naturally arose during their development.

The interconnective description of a signal processing system was defined in Chapter 3 with a focus on aspects of connectivity and separability that are pertinent to assembling and implementing large-scale systems. In particular, interconnective systems are formed by attaching constitutive modules and interconnecting networks together, and canonical and scattering classes of systems were defined by using interconnecting networks that obey certain conservation principles. Correspondences between the organizations of interconnective systems with fundamentally related behaviors led to the establishment of interconnective equivalence classes, and these correspondences were used to define coordinate transforms between canonical-form and scattering-form systems. Procedures for generating synchronous and asynchronous algorithms from these equivalence classes were provided where these procedures specifically had to deal with issues of computability, delay-free loop reduction, precedence, and scheduling.

The primary goal of Chapter 4 was to explore the stability and robustness of synchronous and asynchronous implementations of interconnective systems, and of scattering algorithms in particular. Equilibrium states of scattering-form systems were connected with solutions to a broad class of fixed-point and constraint satisfaction problems, and local conditions on the individual modules that are easy to certify in practice were derived so that synchronously or asynchronously implementing the system produces a sequence of system states that tends toward solutions to these problems. By allowing for first-order filtering of the delay modules in scattering systems, the stability and robustness properties were shown to strengthen and the class of problems that can be solved by implementing these systems was expanded. In the context of solving convex optimization problems, these conditions ensure mean-square convergence of scattering algorithms without the use of problem-specific parameters.

Variational properties of conservative vector spaces were used in Chapter 5 to link canonical-form signal processing systems with stationarity conditions underlying a broad class of convex and non-convex optimization problems. Drawing upon the correspondences established in Chapter 3, variational interpretations of canonical-form systems were used to derive constitutive modules pertaining to cost functions and constraints appearing in common optimization problems, and the modules were then modified so that they could be implemented in equivalent scattering-form systems to solve the associated problems. These modules were then used to realize asynchronous, distributed optimization algorithms as robust large-scale signal processing systems, and connections between these algorithms and their existing gradient-based and proximal counterparts were made. Examples of the modular approach to designing these systems were presented in Chapter 6 in the context of solving optimization problems by implementing signal processing systems as real-time web services and decentralized processor networks.

Several directions for future research naturally arose during the development of the contributions outlined above. A recurring and important theme that was emphasized throughout the thesis was that the organization of a system for which a useful property manifests itself is not necessarily the only organization that can take advantage of that property in practice. More broadly, the interconnective description of a signal processing system allows a straightforward way to identify connections between systems with closely related behaviors as well as design implementations in different coordinate systems where certain properties are more advantageous. This idea is exemplified by both the proximal and scattering optimization algorithms discussed in this thesis. In light of this, the interconnective model of a system was informally extended in Chapter 3 to handle signal models and their acquisition systems too. This extension suggests opportunity in understanding various properties of signal acquisition systems or in designing acquisition systems in different coordinate systems that are useful in some context.

The discussion of stability and robustness in Chapter 4 for expansive everywhere system operators relied on the system operator exhibiting some amount of natural rotation and an appropriate amount of filtering. Scattering-form structures produced by performing scattering coordinate transforms to canonical-form structures possess interconnecting networks realized by orthogonal matrices that are actually special orthogonal, thus they always exhibit some rotation in the sense that the matrices cannot have eigenvalues of -1. In addition, expansive everywhere system operators typically correspond to optimization problems which have non-convex cost functions, such as with the example in Section 6.1.4. These observations together suggest two important directions for future research. First, there may be a class of expansive operators that preserve this rotation so that convergence can be guaranteed using theorems like Theorem 4.3.7. Second, understanding what non-convex cost functions these expansive operators correspond to could lead to a useful class of non-convex cost functions that can be used as surrogates for non-convex relaxations of non-convex problems, rather than relying on convex functions to produce convex relaxations of non-convex problems.

In Chapter 5, the parametric representation of the stationarity conditions allows for a strong duality principle to hold without the direct assumption of convexity. This suggests investigating the limits to which the non-parametric interpretation of these conditions can be used to design the functions used in (5.25) through (5.27). In particular, the parametric representation provides a fairy straightforward method to tackle writing the stationarity conditions for non-convex optimization problems to which coordinate transforms beyond the scattering ones may be useful in producing iterative algorithms. Finally, the constitutive modules derived in this thesis form only a small subset of those needed to tackle important and interesting large-scale optimization problems in practice, and, of course, this suggests deriving them for these problems using the strategies discussed in Section 5.3.

Appendix A

Stability and robustness of passive everywhere system operators

In this appendix, we provide a proof of Theorem 4.3.6 which summarizes the stability and robustness properties of synchronous and asynchronous implementations of interconnective systems with associated passive everywhere system operators. The theorem, stated in terms of the filtered system operator T_f, is restated below for convenience.

Theorem A.0.1 (Stability in $\mathbb{R}^K$; $\alpha = 1$)**.** *Let $T\colon \mathbb{R}^K \to \mathbb{R}^K$ denote a system operator that is passive everywhere with non-empty fixed-point set $\mathcal{F}_T$. Then the filtered system operator T_f associated with T is stable in mean square, i.e. $\mathbf{v}^n \overset{2}{\to} \mathbf{v}^\star$ for some $\mathbf{v}^\star \in \mathcal{F}_T$, provided $\rho \in (0,1)$.*

Before proving the theorem we collect together two anticipatory lemmata.

Lemma A.0.1. *Let $\{\mathbf{a}_k\colon k \in \mathbb{N}\}$ denote a bounded, non-negative sequence of real-valued scalars which additionally satisfy $\lim_{k\to\infty} k\mathbf{a}_k > 0$, then $\sum_{k\in\mathbb{N}} \mathbf{a}_k = \infty$.*

Proof: Let $\mathbf{a} = \limsup_{k\to\infty} k\mathbf{a}_k > 0$ and define $\{\mathbf{b}_l\colon l \in \mathcal{I}\}$, with $\mathcal{I} \subseteq \mathbb{N}$, as a subsequence of $\{k\mathbf{a}_k\colon k \in \mathbb{N}\}$ such that $\mathbf{b}_l \to \mathbf{a}$. The existence of such a convergent subsequence is guaranteed, for example, by Theorem 3.17 in [33]. The claim then follows directly from the inequality

$$\sum_{k\in\mathbb{N}} \mathbf{a}_k \geq \sum_{l\in\mathcal{I}} \frac{l\mathbf{a}_l}{l} = \sum_{l\in\mathcal{I}} \frac{\mathbf{b}_l}{l} = \infty \tag{A.1}$$

which is due to the subsequence definition and is tight (for partial sums) when $\mathcal{I} = \mathbb{N}$. ■

Lemma A.0.2. *Let $T\colon \mathbb{R}^K \to \mathbb{R}^K$ denote a passive everywhere system operator with a non-empty fixed-point set $\mathcal{F}_T$ and define $f\colon \mathbb{R}\times\mathbb{R}\to\mathbb{R}$ as the scalar valued function*

$$f(\mathbf{l},\mathbf{u}) \triangleq \inf_{\substack{\mathbf{l}\leq\|\mathbf{v}-\mathbf{v}^\star\|^2\leq\mathbf{u}\\ \mathbf{v}^\star\in\mathcal{F}_T}} \|\mathbf{v}-T(\mathbf{v})\|^2 . \tag{A.2}$$

Then, $f(\mathbf{l},\mathbf{u}) > 0$ for every pair of scalars $(\mathbf{l},\mathbf{u})$ satisfying $\mathbf{u} > \mathbf{l} > 0$.

Proof: This result is a direct consequence of the fact that T is continuous system operator and the objective function in (A.2) is defined over a non-empty compact set which does not contain any fixed-points of T. ■

We now prove the main result in Theorem 4.3.6.

Proof: Observe that the sequence of scalars $\mathbb{E}[\|\mathbf{v}^n-\mathbf{v}^\star\|^2]$ for $n\geq 0$ is non-increasing:

$$\begin{aligned}
\mathbb{E}\left[\|\mathbf{v}^n-\mathbf{v}^\star\|^2\right] &= p\mathbb{E}\left[\left\|T_f(\rho,\mathbf{v}^{n-1})-\mathbf{v}^\star\right\|^2\right] + (1-p)\mathbb{E}\left[\left\|\mathbf{v}^{n-1}-\mathbf{v}^\star\right\|^2\right] && \text{(A.3)}\\
&= p\rho\mathbb{E}\left[\left\|T(\mathbf{v}^{n-1})-\mathbf{v}^\star\right\|^2\right] + p(1-\rho)\mathbb{E}\left[\left\|\mathbf{v}^{n-1}-\mathbf{v}^\star\right\|^2\right] && \text{(A.4)}\\
&\quad -p\rho(1-\rho)\mathbb{E}\left[\left\|T(\mathbf{v}^{n-1})-\mathbf{v}^{n-1}\right\|^2\right] + p\mathbb{E}\left[\left\|\mathbf{v}^{n-1}-\mathbf{v}^\star\right\|^2\right]\\
&\leq \mathbb{E}\left[\left\|\mathbf{v}^{n-1}-\mathbf{v}^\star\right\|^2\right] - p\rho(1-\rho)\mathbb{E}\left[\left\|T\left(\mathbf{v}^{n-1}\right)-\mathbf{v}^{n-1}\right\|^2\right] && \text{(A.5)}
\end{aligned}$$

where the second equality is due to (4.26) and the linearity of the expectation operator. Iterating this inequality n times results in the inequality

$$\mathbb{E}\left[\|\mathbf{v}^n-\mathbf{v}^\star\|^2\right] \leq \left\|\mathbf{v}^0-\mathbf{v}^\star\right\|^2 - p\rho(1-\rho)\sum_{m=0}^{n-1}\mathbb{E}\left[\|T(\mathbf{v}^m)-\mathbf{v}^m\|^2\right] \tag{A.6}$$

and so we conclude that the sequence $\mathbb{E}[\|\mathbf{v}^n-\mathbf{v}^\star\|^2]$ for $n\geq 0$ is bounded above by $\|\mathbf{v}^0-\mathbf{v}^\star\|^2$.

Rearranging terms, utilizing the assumption that ρ is restricted to the open unit interval, and loosening the inequality results in

$$\sum_{m=0}^{n-1} \mathbb{E}\left[\|T(\mathbf{v}^m) - \mathbf{v}^m\|^2\right] \quad \leq \quad \frac{1}{p\rho(1-\rho)} \left\|\mathbf{v}^0 - \mathbf{v}^\star\right\|^2 . \tag{A.7}$$

We conclude that the sequence $\mathbf{v}^n - T(\mathbf{v}^n) \to \mathbf{0}$ in mean square, which further implies that $\mathbf{v}^n \to T(\mathbf{v}^n)$ in probability, by taking a limit (note the upper bound is independent of n) and applying the contrapositive to Lemma A.0.1. The rate of this convergence also follows from Lemma A.0.1 and is $o\left(\frac{1}{n}\right)$, i.e. strictly faster than a $\frac{1}{n}$ sequence. We now show by contradiction that $\mathbf{v}^n \to \mathbf{v}^\star$ for some $\mathbf{v}^\star \in \mathcal{F}_T$ in mean square. Suppose that $c = \lim_{n\to\infty} \mathbb{E}[\|\mathbf{v}^n - \mathbf{v}^\star\|^2]$ for some scalar $c > 0$. Note that the limit does indeed exist in part due to the monotone and bounded properties of the sequence $\mathbb{E}[\|\mathbf{v}^n - \mathbf{v}^\star\|^2]$. By application of the law of iterated expectation, we obtain the following inequality:

$$\begin{aligned} \mathbb{E}\left[\|\mathbf{v}^n - \mathbf{v}^\star\|^2\right] \quad &= \quad \mathbb{E}\left[\|\mathbf{v}^n - \mathbf{v}^\star\|^2 \mid \|\mathbf{v}^n - \mathbf{v}^\star\|^2 \geq \frac{c}{2}\right] \mathbb{P}\left(\|\mathbf{v}^n - \mathbf{v}^\star\|^2 \geq \frac{c}{2}\right) && \text{(A.8)} \\ &\qquad + \mathbb{E}\left[\|\mathbf{v}^n - \mathbf{v}^\star\|^2 \mid \|\mathbf{v}^n - \mathbf{v}^\star\|^2 < \frac{c}{2}\right] \mathbb{P}\left(\|\mathbf{v}^n - \mathbf{v}^\star\|^2 < \frac{c}{2}\right) \\ &\leq \quad \left\|\mathbf{v}^0 - \mathbf{v}^\star\right\|^2 \mathbb{P}\left(\|\mathbf{v}^n - \mathbf{v}^\star\|^2 \geq \frac{c}{2}\right) + \frac{c}{2}\left(1 - \mathbb{P}\left(\|\mathbf{v}^n - \mathbf{v}^\star\|^2 \geq \frac{c}{2}\right)\right) && \text{(A.9)} \end{aligned}$$

where the upper bound for the term $\mathbb{E}[\|\mathbf{v}^n - \mathbf{v}^\star\|^2 \mid \|\mathbf{v}^n - \mathbf{v}^\star\|^2 \geq \frac{c}{2}]$ is from (A.6). Taking n large enough and rearranging terms yields

$$0 < \frac{\frac{c}{2}}{\|\mathbf{v}^0 - \mathbf{v}^\star\|^2 - \frac{c}{2}} \leq \mathbb{P}\left(\|\mathbf{v}^n - \mathbf{v}^\star\|^2 \geq \frac{c}{2}\right). \tag{A.10}$$

By application of Lemma A.0.2 we are able to immediately conclude that

$$\mathbb{P}\left(\|\mathbf{v}^n - \mathbf{v}^\star\|^2 \geq \frac{c}{2}\right) \leq \mathbb{P}\left(\|\mathbf{v}^n - T(\mathbf{v}^n)\|^2 \geq f\left(\frac{c}{2}, \|\mathbf{v}^0 - \mathbf{v}^\star\|^2\right)\right) \tag{A.11}$$

since by the definition of the function f in (A.2) the event $\|\mathbf{v}^n - \mathbf{v}^\star\|^2 \geq \frac{c}{2}$ implies $\|\mathbf{v}^n - T(\mathbf{v}^n)\|^2 \geq f(\frac{c}{2}, \|\mathbf{v}^0 - \mathbf{v}^\star\|^2)$. We then extend the chain of inequalities (A.10)-(A.11) by applying Markov's inequality [83] to (A.11) and obtain

$$0 < \frac{\frac{c}{2}}{\|\mathbf{v}^0 - \mathbf{v}^\star\|^2 - \frac{c}{2}} \leq \mathbb{P}\left(\|\mathbf{v}^n - \mathbf{v}^\star\|^2 \geq \frac{c}{2}\right) \leq \frac{\mathbb{E}\left[\|\mathbf{v}^n - T(\mathbf{v}^n)\|^2\right]}{f\left(\frac{c}{2}, \|\mathbf{v}^0 - \mathbf{v}^\star\|^2\right)}. \tag{A.12}$$

Taking a limit produces a contradiction since we have already proven that the term $\mathbb{E}[\|\mathbf{v}^n - T(\mathbf{v}^n)\|^2]$ goes to zero, therefore the scalar c must be zero, i.e.

$$\begin{aligned} \lim_{n\to\infty} \mathbb{E}\left[\|\mathbf{v}^n - \mathbf{v}^\star\|^2\right] &= c && \text{(A.13)} \\ &= 0. && \text{(A.14)} \end{aligned}$$

This concludes the proof that the sequence of states produced using the passive-everywhere filtered system operator T_f converges to an element of $\mathcal{F}_T$ in the mean square sense. ■

Appendix B

List of example constitutive relation modules

In this appendix, we provide a concise list of the modules derived in Chapter 5 for use in directly assembling asynchronous processing systems to solve the transformed stationarity conditions (5.89)-(5.90). The general form used to define an interconnect and constitutive relation module is depicted in Figure B-1. Specific instances of the constitutive relation modules are then depicted in Figures B-2-B-4. Consistent with the general form of a constitutive module, we state the following for each when they exist: the behavior in the canonical coordinate system, the reduced-form primal and dual optimization components, the realization as a map for the coordinate transformations $M^{(i)}$ in (3.90) and $M^{(o)}$ in (3.91), and the α-conicity properties of the realizations.

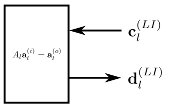
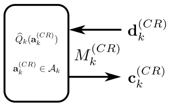
Graph symbol	Canonical behavior	Reduced-form primal components	Reduced-form dual components	Realization		α-connicity
 $A_l\mathbf{a}_l^{(i)} = \mathbf{a}_l^{(o)}$ $\mathbf{c}_l^{(LI)}$ $\mathbf{d}_l^{(LI)}$	$\begin{bmatrix}\mathbf{a}_l^{(i)}\\ \mathbf{a}_l^{(o)}\\ \mathbf{b}_l^{(i)}\\ \mathbf{b}_l^{(o)}\end{bmatrix} \in \text{range}\left(\begin{bmatrix} I_{N_l^{(i)}} & 0\\ A_l & 0\\ 0 & -A_l^T\\ 0 & I_{N_l^{(o)}}\end{bmatrix}\right)$	$A_l\mathbf{a}_l^{(i)} = \mathbf{a}_l^{(o)}$	$\mathbf{b}_l^{(i)} = -A^T\mathbf{b}_l^{(o)}$	$\mathbf{d}_l^{(LI)} = G_l\mathbf{c}_l^{(LI)}$ $G_l = \begin{bmatrix} I & -A_l^T\\ A_l & I\end{bmatrix}\begin{bmatrix} I & A_l^T\\ -A_l & I\end{bmatrix}^{-1}$		passive everywhere
 $\widehat{Q}_k(\mathbf{a}_k^{(CR)})$ $\mathbf{a}_k^{(CR)} \in \mathcal{A}_k$ $M_k^{(CR)}$ $\mathbf{d}_k^{(CR)}$ $\mathbf{c}_k^{(CR)}$	$\left\{\begin{bmatrix} f_k^{(CR)}(\mathbf{x}_k^{(CR)})\\ g_k^{(CR)}(\mathbf{x}_k^{(CR)})\end{bmatrix} : \mathbf{x}_k^{(CR)} \in \mathbb{R}^{N_k^{(CR)}}\right\}$	$\widehat{Q}_k(\mathbf{a}_k^{(CR)})$ $\mathbf{a}_k^{(CR)} \in \mathcal{A}_k$	$\widehat{R}_k(\mathbf{b}_k^{(CR)})$ $\mathbf{b}_k^{(CR)} \in \mathcal{B}_k$	$M^{(i)}$	$\mathbf{c}_k^{(CR)} = m_k^{(i)}(\mathbf{d}_k^{(CR)})$	case by case
				$M^{(o)}$	$\mathbf{c}_k^{(CR)} = m_k^{(o)}(\mathbf{d}_k^{(CR)})$	

Figure B-1: The general form of the interconnect and constitutive relation modules.

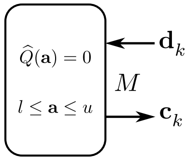
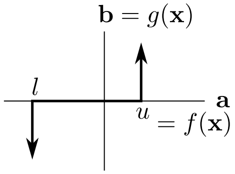
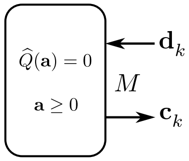
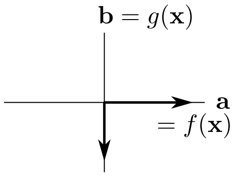
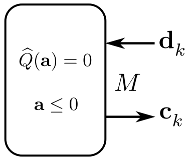
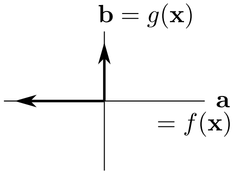
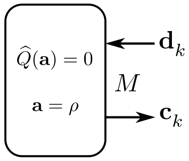
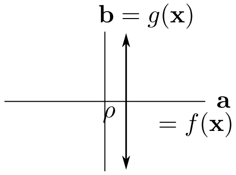
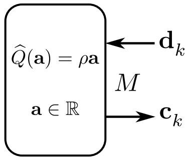
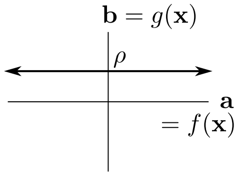
Graph symbol	Canonical behavior	Reduced-form primal components	Reduced-form dual components	Realization		α-connicity
 $\widehat{Q}(\mathbf{a}) = 0$ $l \leq \mathbf{a} \leq u$ $\mathbf{d}_k$ M $\mathbf{c}_k$	 $\mathbf{b} = g(\mathbf{x})$ l u $\mathbf{a}$ $= f(\mathbf{x})$	$\widehat{Q}(\mathbf{a}) = 0$ $l \leq \mathbf{a} \leq u$	$\widehat{R}(\mathbf{b}) = \begin{cases} u\mathbf{b}, & \mathbf{b} \geq 0 \\ l\mathbf{b}, & \mathbf{b} < 0 \end{cases}$ $\mathbf{b} \in \mathbb{R}$	$M^{(i)}$	$\mathbf{c} = \begin{cases} \mathbf{d}, & l \leq \mathbf{d} \leq u \\ 2u - \mathbf{d}, & \mathbf{d} > u \\ 2l - \mathbf{d}, & \mathbf{d} < l \end{cases}$	passive everywhere
				$M^{(o)}$	$\mathbf{c} = \begin{cases} -\mathbf{d}, & l \leq \mathbf{d} \leq u \\ \mathbf{d} - 2u, & \mathbf{d} > u \\ \mathbf{d} - 2l, & \mathbf{d} < l \end{cases}$	
 $\widehat{Q}(\mathbf{a}) = 0$ $\mathbf{a} \geq 0$ $\mathbf{d}_k$ M $\mathbf{c}_k$	 $\mathbf{b} = g(\mathbf{x})$ $\mathbf{a}$ $= f(\mathbf{x})$	$\widehat{Q}(\mathbf{a}) = 0$ $\mathbf{a} \geq 0$	$\widehat{R}(\mathbf{b}) = 0$ $\mathbf{b} \leq 0$	$M^{(i)}$	$\mathbf{c} = \lvert\mathbf{d}\rvert$	passive everywhere
				$M^{(o)}$	$\mathbf{c} = -\lvert\mathbf{d}\rvert$	
 $\widehat{Q}(\mathbf{a}) = 0$ $\mathbf{a} \leq 0$ $\mathbf{d}_k$ M $\mathbf{c}_k$	 $\mathbf{b} = g(\mathbf{x})$ $\mathbf{a}$ $= f(\mathbf{x})$	$\widehat{Q}(\mathbf{a}) = 0$ $\mathbf{a} \leq 0$	$\widehat{R}(\mathbf{b}) = 0$ $\mathbf{b} \geq 0$	$M^{(i)}$	$\mathbf{c} = -\lvert\mathbf{d}\rvert$	passive everywhere
				$M^{(o)}$	$\mathbf{c} = \lvert\mathbf{d}\rvert$	
 $\widehat{Q}(\mathbf{a}) = 0$ $\mathbf{a} = \rho$ $\mathbf{d}_k$ M $\mathbf{c}_k$	 $\mathbf{b} = g(\mathbf{x})$ ρ $\mathbf{a}$ $= f(\mathbf{x})$	$\widehat{Q}(\mathbf{a}) = 0$ $\mathbf{a} = \rho$	$\widehat{R}(\mathbf{b}) = \rho\mathbf{b}$ $\mathbf{b} \in \mathbb{R}$	$M^{(i)}$	$\mathbf{c} = -\mathbf{d} + 2\rho$	passive everywhere
				$M^{(o)}$	$\mathbf{c} = \mathbf{d} - 2\rho$	
 $\widehat{Q}(\mathbf{a}) = \rho\mathbf{a}$ $\mathbf{a} \in \mathbb{R}$ $\mathbf{d}_k$ M $\mathbf{c}_k$	 $\mathbf{b} = g(\mathbf{x})$ ρ $\mathbf{a}$ $= f(\mathbf{x})$	$\widehat{Q}(\mathbf{a}) = \rho\mathbf{a}$ $\mathbf{a} \in \mathbb{R}$	$\widehat{R}(\mathbf{b}) = 0$ $\mathbf{b} = \rho$	$M^{(i)}$	$\mathbf{c} = \mathbf{d} - 2\rho$	passive everywhere
				$M^{(o)}$	$\mathbf{c} = -\mathbf{d} + 2\rho$	

Figure B-2: Example constitutive relation modules derived in this thesis.

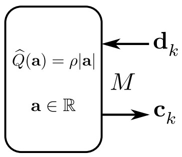
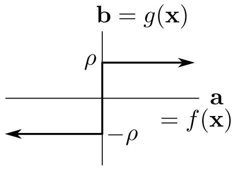
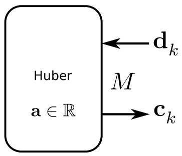
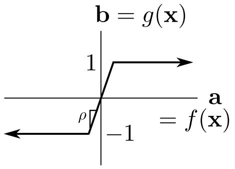
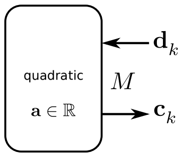
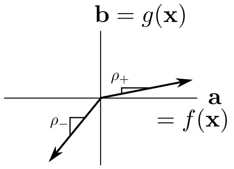
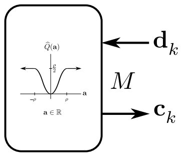
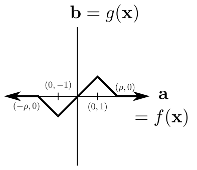
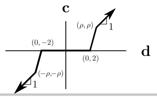
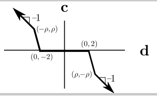
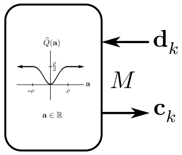
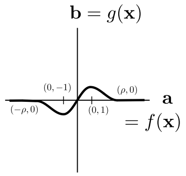
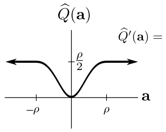
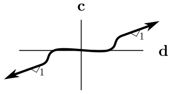
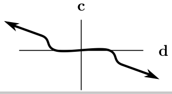
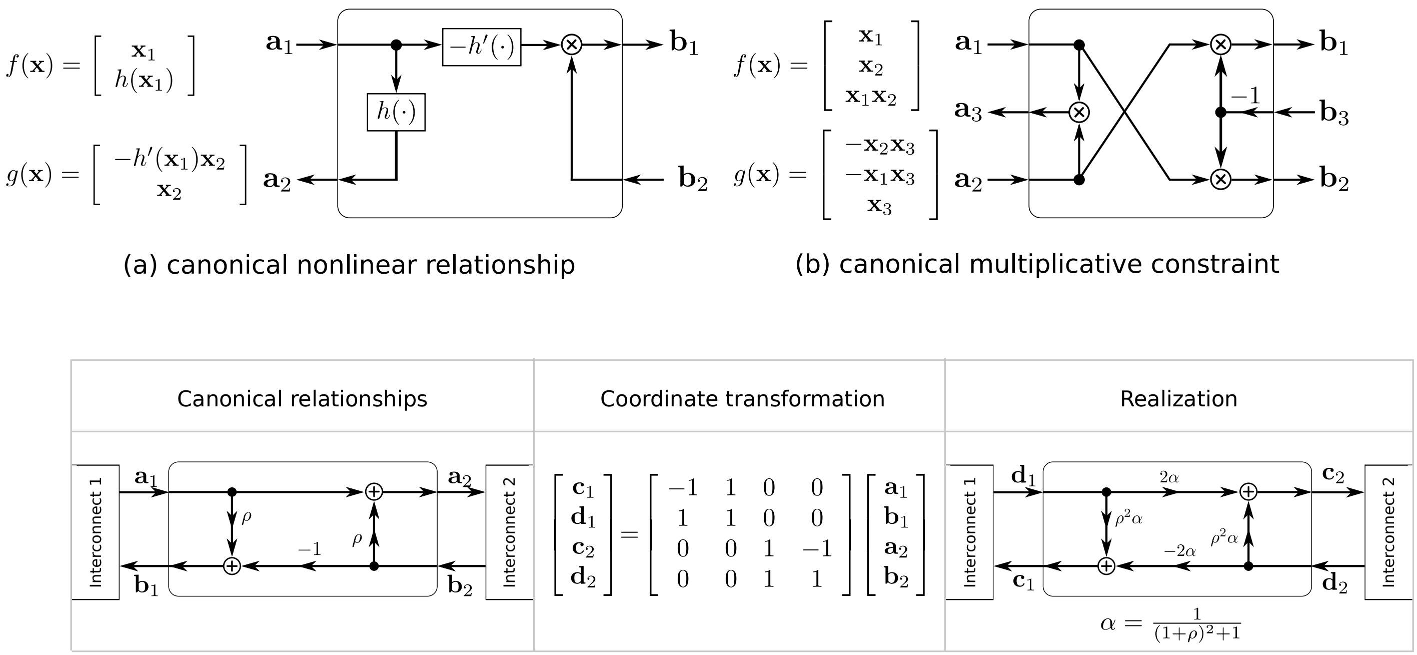
Graph symbol	Canonical behavior	Reduced-form primal components	Reduced-form dual components	Realization		$\alpha-$ connicity
 $\widehat{Q}(\mathbf{a}) = \rho\lvert\mathbf{a}\rvert$ $\mathbf{a} \in \mathbb{R}$ $\mathbf{d}_k$ M $\mathbf{c}_k$		$\widehat{Q}(\mathbf{a}) = \rho\lvert\mathbf{a}\rvert$ $\mathbf{a} \in \mathbb{R}$	$\widehat{R}(\mathbf{b}) = 0$ $-\rho \le \mathbf{b} \le \rho$	$M^{(i)}$	$\mathbf{c} = \begin{cases} -\mathbf{d}, & \lvert\mathbf{d}\rvert \le \rho \\ \mathbf{d} - 2\rho\mathrm{sgn}(\mathbf{d}), & \lvert\mathbf{d}\rvert > \rho \end{cases}$	passive everywhere
				$M^{(o)}$	$\mathbf{c} = \begin{cases} \mathbf{d}, & \lvert\mathbf{d}\rvert \le \rho \\ 2\rho\mathrm{sgn}(\mathbf{d}) - \mathbf{d}\ , & \lvert\mathbf{d}\rvert > \rho \end{cases}$	
 Huber $\mathbf{a} \in \mathbb{R}$ $\mathbf{d}_k$ M $\mathbf{c}_k$		$\widehat{Q}(\mathbf{a}) = \begin{cases} \lvert\mathbf{a}\rvert, & \lvert\mathbf{a}\rvert \ge \frac{1}{\rho} \\ \frac{\rho}{2}\mathbf{a}^2 + \frac{1}{2\rho}, & \lvert\mathbf{a}\rvert < \frac{1}{\rho} \end{cases}$ $\mathbf{a} \in \mathbb{R}$	$\widehat{R}(\mathbf{b}) = \frac{1}{2\rho}(\mathbf{b}^2 - 1)$	$M^{(i)}$	$\mathbf{c} = \begin{cases} \frac{1-\rho}{1+\rho}\mathbf{d} & \lvert\mathbf{d}\rvert \le \frac{1}{\rho}+1 \\ \mathbf{d} - 2\mathrm{sgn}(\mathbf{d}), & \lvert\mathbf{d}\rvert > \frac{1}{\rho}+1 \end{cases}$	passive everywhere dissipative everywhere if $\lvert\mathbf{d}\rvert \le \frac{1}{\rho} + 1$ $0 < \rho < \infty$
				$M^{(o)}$	$\mathbf{c} = \begin{cases} \frac{\rho-1}{\rho+1}\mathbf{d} & \lvert\mathbf{d}\rvert \le \frac{1}{\rho}+1 \\ -\mathbf{d} + 2\mathrm{sgn}(\mathbf{d}), & \lvert\mathbf{d}\rvert > \frac{1}{\rho}+1 \end{cases}$	
 quadratic $\mathbf{a} \in \mathbb{R}$ $\mathbf{d}_k$ M $\mathbf{c}_k$		$\widehat{Q}(\mathbf{a}) = \begin{cases} \frac{\rho_+}{2}\mathbf{a}^2, & \mathbf{a} \ge 0 \\ \frac{\rho_-}{2}\mathbf{a}^2, & \mathbf{a} < 0 \end{cases}$ $\mathbf{a} \in \mathbb{R}$	$\widehat{R}(\mathbf{b}) = \begin{cases} \frac{1}{2\rho_+}\mathbf{b}^2, & \mathbf{b} \ge 0 \\ \frac{1}{2\rho_-}\mathbf{b}^2, & \mathbf{b} < 0 \end{cases}$ $\mathbf{b} \in \mathbb{R}$	$M^{(i)}$	$\mathbf{c} = \begin{cases} \frac{1-\rho_+}{1+\rho_+}\mathbf{d}, & \mathbf{d} \ge 0 \\ \frac{1-\rho_-}{1+\rho_-}\mathbf{d}, & \mathbf{d} < 0 \end{cases}$	dissipative everywhere if $0 < \rho_- < \infty$ $0 < \rho_+ < \infty$
				$M^{(o)}$	$\mathbf{c} = \begin{cases} \frac{\rho_+ - 1}{\rho_+ + 1}\mathbf{d}, & \mathbf{d} \ge 0 \\ \frac{\rho_- - 1}{\rho_- + 1}\mathbf{d}, & \mathbf{d} < 0 \end{cases}$	passive everywhere if $\rho_+ = \rho_- = 0$
 $\widehat{Q}(\mathbf{a})$ $\frac{\rho}{2}$ $-\rho$ ρ $\mathbf{a}$ $\mathbf{a} \in \mathbb{R}$ $\mathbf{d}_k$ M $\mathbf{c}_k$		$\widehat{Q}(\mathbf{a}) = \begin{cases} \frac{\rho}{2}, & \lvert\mathbf{a}\rvert > \rho \\ \frac{1}{2}\mathbf{a}^2, & \lvert\mathbf{a}\rvert < 1 \\ \frac{1}{2(1-\rho)}\mathbf{a}^2 - \mathrm{sgn}(\mathbf{a})\frac{\rho}{1-\rho}\mathbf{a} + \frac{\rho}{2(1-\rho)}, & 1 < \lvert\mathbf{a}\rvert \le \rho \end{cases}.$ $\mathbf{a} \in \mathbb{R}$ no reduced-form dual components		$M^{(i)}$		passive about $\mathbf{d} = 0$
				$M^{(o)}$		
 $\widehat{Q}(\mathbf{a})$ $\frac{\rho}{2}$ $-\rho$ ρ $\mathbf{a}$ $\mathbf{a} \in \mathbb{R}$ $\mathbf{d}_k$ M $\mathbf{c}_k$		 $\widehat{Q}'(\mathbf{a}) = \begin{cases} 0, & \lvert\mathbf{a}\rvert > \rho \\ \sin\left(\frac{\pi}{2}\mathbf{a}\right), & \lvert\mathbf{a}\rvert < 1 \\ \frac{\mathrm{sgn}(\mathbf{a})}{2}\left(\cos\left(\frac{\pi}{\rho-1}(1 - \mathrm{sgn}(\mathbf{a})\mathbf{a}\right) + 1\right), & 1 \le \lvert\mathbf{a}\rvert \le \rho \end{cases}.$ $\mathbf{a} \in \mathbb{R}$ no reduced-form dual components		$M^{(i)}$		passive about $\mathbf{d} = 0$
				$M^{(o)}$		

Figure B-3: A continuation of the example constitutive relation modules derived in this thesis.



$$f(\mathbf{x}) = \begin{bmatrix} \mathbf{x}_1 \\ h(\mathbf{x}_1) \end{bmatrix} \qquad g(\mathbf{x}) = \begin{bmatrix} -h'(\mathbf{x}_1)\mathbf{x}_2 \\ \mathbf{x}_2 \end{bmatrix}$$

(a) canonical nonlinear relationship

$$f(\mathbf{x}) = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_1\mathbf{x}_2 \end{bmatrix} \qquad g(\mathbf{x}) = \begin{bmatrix} -\mathbf{x}_2\mathbf{x}_3 \\ -\mathbf{x}_1\mathbf{x}_3 \\ \mathbf{x}_3 \end{bmatrix}$$

(b) canonical multiplicative constraint

Canonical relationships

Coordinate transformation

$$\begin{bmatrix} \mathbf{c}_1 \\ \mathbf{d}_1 \\ \mathbf{c}_2 \\ \mathbf{d}_2 \end{bmatrix} = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{b}_1 \\ \mathbf{a}_2 \\ \mathbf{b}_2 \end{bmatrix}$$

Realization

$$\alpha = \frac{1}{(1+\rho)^2+1}$$

Figure B-4: Example constitutive relation modules derived in this thesis that are only stated in the canonical coordinate system.

References

[1] G. Zames. On the input-output stability of time-varying nonlinear feedback systems part I: Conditions derived using concepts of loop gain, conicity, and positivity. *IEEE Transactions on Automatic Control*, 11(2):228–238, 1966.

[2] L. N. Trefethen and D. Bau. *Numerical linear algebra.* Society for Industrial and Applied Mathematics, June 1997.

[3] D. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52:1289–1306, 2006.

[4] P. Penfield, R. Spence, and S. Duinker. *Tellegen's theorem and electrical networks.* M.I.T. Press research monographs. Mit Press, 1970.

[5] G. Zames. On the input-output stability of time-varying nonlinear feedback systems part II: Conditions involving circles in the frequency plane and sector nonlinearities. *IEEE Transactions on Automatic Control*, 11(3):465–476, 1966.

[6] R. M. Edelstein and K. S. Govinder. Conservation laws for the Black-Scholes equation. *Nonlinear Analysis: Real World Applications*, 10(6):3372 – 3380, 2009.

[7] J. B. Dennis. Distributed solution of network programming problems. *IEEE Transactions on Communications Systems*, 12(2):176–184, June 1964.

[8] J. B. Dennis. *Mathematical programming and electrical networks.* PhD thesis, Massachusetts Institute of Technology, 1959.

[9] L. O. Chua and G. N. Lin. Nonlinear programming without computation. *IEEE Transactions on Circuits and Systems*, 3(2):182–188, 1984.

[10] L. O. Chua. Stationary principles and potential functions for nonlinear networks. *Journal of the Franklin Institute*, 296(2):91 – 113, 1973.

[11] T. A. Baran. *Conservation in Signal Processing Systems.* PhD thesis, Massachusetts Institute of Technology, 2012.

[12] K. Slavakis, G. B. Giannakis, and G. Mateos. Modeling and optimization for big data analytics: (statistical) learning tools for our era of data deluge. *IEEE Signal Processing Magazine*, 31(5):18–31, Sept 2014.

[13] D. Yu and L. Deng. Deep learning and its applications to signal and information processing. *IEEE Signal Processing Magazine*, 28(1):145–154, Jan 2011.

[14] T. A. Lahlou and T. A. Baran. Asynchronous systems for constraint satisfaction: filtering and stability. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, March 2016.

[15] J. C. Willems. The behavioral approach to open and interconnected systems. *IEEE Control Systems Magazine*, 27(6):46–99, 2007.

[16] J. Polderman and J. C. Willems. *Introduction to Mathematical Systems Theory.* Springer-Verlag, 1998.

[17] T. A. Baran and T. A. Lahlou. Implementation of interconnective systems. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1101–1105, April 2015.

[18] T. A. Baran and A. V. Oppenheim. Inversion of nonlinear and time-varying systems. In *IEEE Digital Signal Processing Workshop and IEEE Signal Processing Education Workshop*, pages 283–288. IEEE, 2011.

[19] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.

[20] L. Lee. *Distributed Signal Processing.* PhD thesis, Massachusetts Institute of Technology, 2000.

[21] R. E. Crochiere and A. V. Oppenheim. Analysis of linear digital networks. *Proceedings of the IEEE*, 63(4):581–595, 1975.

[22] N. Parikh and S. Boyd. Block splitting for distributed optimization. *Mathematical Programming Computation*, 6(1):77–102, 2014.

[23] P. A. Forero, A. Cano, and G. B. Giannakis. Consensus-based distributed support vector machines. *J. Mach. Learn. Res.*, 11:1663–1707, August 2010.

[24] Nancy A. Lynch. *Distributed Algorithms.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1996.

[25] L. F. Shampine. Conservation laws and the numerical solution of ODEs. *Computers and Mathematics with Applications*, 12(5):1287 – 1296, 1986.

[26] A. Härmä. Implementation of recursive filters having delay free loops. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 3, pages 1261–1264, May 1998.

[27] S. Mitra. *Digital Signal Processing: A Computer-Based Approach.* McGraw-Hill Higher Education, 2nd edition, 2000.

[28] F. Fontana. Computation of linear filter networks containing delay-free loops, with an application to the waveguide mesh. *IEEE Transactions on Speech and Audio Processing*, 11(6):774–782, Nov 2003.

[29] F. Fontana and F. Avanzini. Computation of delay-free nonlinear digital filter networks: Application to chaotic circuits and intracellular signal transduction. *IEEE Transactions on Signal Processing*, 56(10):4703–4715, Oct 2008.

[30] S. D'Angelo and V. Valimaki. Generalized moog ladder filter: Part ii: Explicit nonlinear model through a novel delay-free loop implementation method. *IUEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(12):1873–1883, Dec 2014.

[31] G. Borin, G. De Poli, and D. Rocchesso. Elimination of delay-free loops in discrete-time models of nonlinear acoustic systems. *IEEE Transactions on Speech and Audio Processing*, 8(5):597–605, Sep 2000.

[32] Hsing-Tsung Kung and 003 Carnegie-Mellon University.Computer science. Pittsburgh (PA US). Synchronized and asynchronous parallel algorithms for multiprocessors, 1976.

[33] W. Rudin. *Principles of Mathematical Analysis.* International series in pure and applied mathematics. McGraw-Hill, 1976.

[34] S. Axler. *Linear Algebra Done Right.* Linear Algebra Done Right. Springer, 1997.

[35] A. V. Oppenheim and R. W. Schafer. *Discrete-Time Signal Processing.* Prentice Hall Press, Upper Saddle River, NJ, USA, 3rd edition, 2009.

[36] T. A. Baran and B. K. P. Horn. A robust signal-flow architecture for cooperative vehicle density control. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 2790–2794, May 2013.

[37] I. Gohberg, P. Lancaster, and L. Rodman. *Indefinite Linear Algebra and Applications.* Birkhäuser Basel, 2006.

[38] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, New York, NY, USA, 2004.

[39] D. P. Bertsekas. *Nonlinear Programming.* Athena Scientific, Belmont, MA, 1999.

[40] R. T. Rockafellar. *Convex Analysis.* Princeton landmarks in mathematics and physics. Princeton University Press, 1970.

[41] C. Villani. *Optimal transport : old and new.* Grundlehren der mathematischen Wissenschaften. Springer, 2009.

[42] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, 31(9):803–812, Sep 1986.

[43] B. Recht, C. Re, S. Wright, and F. Niu. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in Neural Information Processing Systems 24*, pages 693–701. Curran Associates, Inc., 2011.

[44] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, January 2011.

[45] E. Wei and A. Ozdaglar. On the O(1/k) Convergence of Asynchronous Distributed Alternating Direction Method of Multipliers. *ArXiv e-prints*, July 2013.

[46] N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):127–239, January 2014.

[47] E. K. Ryu and S. Boyd. A primer on monotone operator methods. Technical report, Stanford University, November 2015.

[48] J. W. Brewer. Progress in the bond graph representations of economics and population dynamics. *Journal of the Franklin Institute*, 328:675 – 696, 1991.

[49] A. V. Oppenheim, R. W. Schafer, and T. G. Stockham Jr. Nonlinear filtering of multiplied and convolved signals. *IEEE Transactions on Audio and Electroacoustics*, 16(3):437–466, Sep 1968.

[50] C. Fan and F. Zhang. Homomorphic filtering based illumination normalization method for face recognition. *Pattern Recognition Letters*, 32(10):1468 – 1479, 2011.

[51] R. S. Nikhil and Arvind. *Implicit Parallel Programming in PH.* Morgan Kaufmann Publishers, 2001.

[52] R. E. Crochiere. *Digital network theory and its application to the analysis and design of digital filters.* PhD thesis, Massachusetts Institute of Technology, 1974.

[53] L.-F. Chao and E.H.-M. Sha. Scheduling data-flow graphs via retiming and unfolding. *IEEE Transactions on Parallel and Distributed Systems*, 8(12):1259–1267, Dec 1997.

[54] P. P. Vaidyanathan. *Multirate systems and filter banks.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[55] D. M. Pozar. *Microwave Engineering.* Wiley, 2004.

[56] S. Y. Kung. Inverse systems and an inversion rule. In *IEEE Conference on Decision and Control*, pages 771–776, Dec 1977.

[57] S. J. Mason and H. J. Zimmerman. *Electronic circuits, signals, and systems.* Wiley, 1960.

[58] A. Carini, G. L. Sicuranza, and V. J. Mathews. On the inversion of certain nonlinear systems. *IEEE Signal Processing Letters*, 4(12):334–336, 1997.

[59] P. A Regalia, S .K. Mitra, and P. P. Vaidyanathan. The digital all-pass filter: a versatile signal processing building block. *Proceedings of the IEEE*, 76(1):19–37, Jan 1988.

[60] J. Yen. On nonuniform sampling of bandwidth-limited signals. *IRE Transactions on Circuit Theory*, 3(4):251–257, Dec 1956.

[61] G. W. Hart. Nonintrusive appliance load monitoring. *Proceedings of the IEEE*, 80(12):1870–1891, Dec 1992.

[62] H. T. Kung. Synchronized and asynchronous parallel algorithms for multiprocessors. *Algorithms and Complexity*, pages 153–200, 1976.

[63] Y. Brun, G. Edwards, J. Y. Bang, and N. Medvidovic. Smart redundancy for distributed computation. In *International Conference on Distributed Computing Systems*, pages 665–676, June 2011.

[64] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione. Gossip algorithms for distributed signal processing. *Proceedings of the IEEE*, 98(11):1847–1864, 2010.

[65] A. Sandryhaila and J. M. F. Moura. Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure. *IEEE Signal Processing Magazine*, 31(5):80–90, Sept 2014.

[66] S. Parekh and P. Shah. Nyquist filter design using pocs methods: Including constraints in design. *arXiv*, 2013.

[67] K. C. Haddad, H. Stark, and N. P. Galatsanos. Constrained fir filter design by the method of vector space projections. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 47(8):714–725, Aug 2000.

[68] J. A. Cadzow. Signal enhancement-a composite property mapping algorithm. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 36(1):49–62, 1988.

[69] W. Dai and O. Milenkovic. Subspace pursuit for compressive sensing: Closing the gap between performance and complexity, 2008.

[70] T. Blumensath and M. E. Davies. Iterative hard thresholding for compressed sensing. *Applied and Computational Harmonic Analysis*, 2008.

[71] D. Needell and J. A. Tropp. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Communications of the ACM*, 53(12):93–100, December 2010.

[72] C.T. Kelley. *Iterative Methods for Linear and Nonlinear Equations.* Frontiers in Applied Mathematics. Society for Industrial and Applied Mathematics, 1995.

[73] G. H. Golub and C. F. Van Loan. *Matrix Computations (3rd Ed.).* Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[74] Henk A. Van der Vorst. *Iterative Krylov methods for large linear systems.* Cambridge monographs on applied and computational mathematics. Cambridge University Press, Cambridge, UK, New York, 2003.

[75] Y. S. Song and Y. H. Lee. Design of sparse fir filters based on branch-and-bound algorithm. In *Proceedings of the 40th Midwest Symposium on Circuits and Systems*, Aug 1997.

[76] D. Needell, J. Tropp, and R. Vershynin. Greedy signal recovery review. In *42nd Asilomar Conference on Signals, Systems and Computers*, 2008.

[77] Y. C. Pati R. Rezaiifar and P. S. Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of the 27 th Annual Asilomar Conference on Signals, Systems, and Computers*, 1993.

[78] E. L. Allgower and K. Georg. *Introduction to Numerical Continuation Methods.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2003.

[79] B. Widrow and S. D. Stearns. *Adaptive Signal Processing.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1985.

[80] J. W. Russell. *An elementary treatise on pure geometry: With numerous examples.* Clarendon Press series. Clarendon Press, 1893.

[81] T. A. Lahlou and T. A. Baran. Signal Processing Structures for Solving Conservative Constraint Satisfaction Problems. *ArXiv e-prints*, October 2015.

[82] D. Bharucha-Rei. Fixed point theorems in probabilistic analysis. *BULLETIN OF THE AMERICAN MATHEMATICAL SOCIETY*, 1976.

[83] R. G. Gallager. *Stochastic processes, theory for applications.* Cambridge University Press, 2013.

[84] T. A. Lahlou and A. V. Oppenheim. Unveiling the tree: A convex framework for sparse problems. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3826–3830, April 2015.

[85] D. Avis and K. Fukuda. A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra. *Discrete and Computational Geometry*, 8(1):295–313, 1992.

[86] P. McMullen. Metrical and combinatorial properties of convex polytopes. *Proceedings of International Congress of Mathematicians*, pages 491–495, 1974.

[87] G. Valiente. *Algorithms on trees and graphs.* Springer Science & Business Media, 2013.

[88] T. A. Lahlou and T. A. Baran. Asynchronous algorithms for solving linear programs. *ArXiv e-prints*, February 2015.

[89] T. A. Lahlou and A. V. Oppenheim. Trading accuracy for numerical stability: Orthogonalization, biorthogonalization and regularization. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, March 2016.

[90] D. Wei and A. V. Oppenheim. Sparsity maximization under a quadratic constraint with applications in filter design. In *IEEE International Conference on Acoustics Speech and Signal Processing*, pages 3686–3689, March 2010.

[91] D. Wei. Non-convex optimization for the design of sparse fir filters. In *IEEE/SP 15th Workshop on Statistical Signal Processing*, pages 117–120, Aug 2009.

[92] T. Baran, D. Wei, and A. V. Oppenheim. Linear programming algorithms for sparse filter design. *Signal Processing, IEEE Transactions on*, 58(3):1605–1617, March 2010.

[93] J. Mattingley and S. Boyd. Real-time convex optimization in signal processing. *IEEE Signal Processing Magazine*, 27(3):50–61, May 2010.

[94] T. A. Baran and T. A. Lahlou. Conservative signal processing architectures for asynchronous, distributed optimization part I: General framework. In *IEEE Global Conference on Signal and Information Processing*, pages 35–39, Dec 2014.

[95] W. Millar. Some general theorems for non-linear systems possessing resistance. *Philosophical Magazine Series 7*, 42(333):1150–1160, 1951.

[96] G. Romano. New results in subdifferential calculus with applications to convex optimization. *Applied Mathematics and Optimization*, 32(3):213–234.

[97] J.J. Moreau. ProximitÃľ et dualitÃľ dans un espace hilbertien. *Bulletin de la SociÃľtÃľ MathÃľmatique de France*, 93:273–299, 1965.

[98] D. P. Bertsekas. Projected newton methods for optimization problems with simple constraints. In *IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, pages 762–767, Dec 1981.

[99] P. L. Combettes and J.-C. Pesquet. Proximal Splitting Methods in Signal Processing. *ArXiv e-prints*, December 2009.

[100] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Img. Sci.*, 2(1):183–202, March 2009.

[101] Shao-Po Wu, S. Boyd, and L. Vandenberghe. Fir filter design via semidefinite programming and spectral factorization. In *IEEE Conference on Decision and Control*, volume 1, pages 271–276 vol.1, Dec 1996.

[102] J. J. Fuchs. Linear programming in spectral estimation. application to array processing. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 6, pages 3161–3164 vol. 6, May 1996.

[103] E. J. Candes and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, December 2005.

[104] D. Bertsimas and J. Tsitsiklis. *Introduction to Linear Optimization.* Athena Scientific, 1st edition, 1997.

[105] J. Bisschop. *Aimms Optimization Modeling.* Paragon Decision Technology, 2006.

[106] T. A. Lahlou and T. A. Baran. Web Services for Asynchronous, Distributed Optimization Using Conservative Signal Processing. *ArXiv e-prints*, September 2015.

[107] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Rev.*, 43(1):129–159, January 2001.

[108] S. J. Orfanidis. *Optimum Signal Processing, An Introduction.* Prentice-Hall, second edition, 1996.

[109] P. M. Pardalos and S. A. Vavasis. Quadratic programming with one negative eigenvalue is np-hard. *Journal of Global Optimization*, 1(1):15–22.

[110] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics).* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[111] T. A. Baran and T. A. Lahlou. Conservative signal processing architectures for asynchronous, distributed optimization part II: Example systems. In *IEEE Global Conference on Signal and Information Processing*, pages 40–44, Dec 2014.

[112] T. A. Lahlou and T. A. Baran. Signal processing conservation. [Online] Available: http://optimization.SPConservation.org (visited 15-Sept-2015).

[113] Tessel 2. Technical machine. [Online] Available: https://tessel.io/.

[114] Espruino. Pur3 ltd. [Online] Available: http://espruino.com/.

[115] Firebase. Firebase Inc., San Francisco, California, 2015. Available: https://www.firebase.com.

[116] M. Li, L. Zhou, Z. Yang, A. Li, F. Xia, D. Anderson, and A. Smola. Parameter server for distributed machine learning. *Big Learning NIPS Workshop*, 2013.

[117] MongoDB. MongoDB Inc., New York City, New York, 2015. Available: https://www.mongodb.org/.

[118] S. Sanfilippo. Redis, 2015. Available: http://redis.io/.

[119] K. Jeong and A. B. Kahng. A power-constrained mpu roadmap for the international technology roadmap for semiconductors (itrs).

[120] S. M. Nowick and M. Singh. Asynchronous design part 1: Overview and recent advances. *IEEE Design Test*, 32(3):5–18, June 2015.

[121] R. Kol and R. Ginosar. Adaptive synchronization for multi-synchronous systems. *Comput. Methods Appl. Mech. Engrg*, 117:98–188, 1994.

[122] D. M. Chapiro. *Globally-asynchronous Locally-synchronous Systems (Performance, Reliability, Digital)*. PhD thesis, Stanford, CA, USA, 1985.

Epilogue

One of the extraordinary things about Al Oppenheim is his ability to mentor students by starting with a simple research question aimed at pushing the boundaries of signal processing and ending with fundamental insights in relatively well-developed and mature areas of the field. For example, the story of my master's thesis began with Al posing the question "What can quantum tunneling inspire about signal processing?" and ended with transforms and spectral techniques tailored to handling exponential signals. Examples similar to this are quite common in DSPG, and are in-line with the group's research mantra of "solutions in search of problems." The story of this thesis also continues along the same thread.

In the fall of 2013, upon returning from a summer internship at Texas Instruments where I was working under the guidance of Arthur Redfern to develop automated methods for device sizing, Al made a suggestion during a conversation about identifying interesting thesis directions that would end up drastically shaping the remainder of my graduate experience at MIT. The suggestion was simple: to consider taking Prof. Luca Daniel's course on numerical methods. Two notable things happened as a result of taking Al's advice. First, I began a deep dive into the world of large-scale iterative methods for solving linear and non-linear problems, and this particular area and its unique set of challenges would end up providing the motivating context for the thesis. Second, and most important, I met my future wife.

My time that semester was (disproportionately) split between primarily three things: playing in the research sandbox, taking two numerical methods courses, and being a teaching assistant for the graduate level signal processing course 6.341. The research questions that were on my mind first thing in the morning most days that semester dealt with approximate signal processing concepts and how a framework would look to develop algorithms in the spirit of "good enough is good enough." The teaching assistantship involved a mixture of traditional tasks like leading recitation sections, making problem set and exam questions, holding office hours, etc., as well as less well-defined tasks such as pushing the capabilities of the edX platform in developing online course content and brainstorming ways to leverage online and in-class clicker technology to enhance the learning experience for students. The projects on edX involved working closely with Tom Baran and those interactions helped form a meaningful friendship and collaboration that would later result in him agreeing to co-advise the thesis. For example, brainstorming with Tom at the white board about possible "mistakes" students could make using graphical signal-flow editors and issues of computability in polyphase structures with delay-free loops led to the paper [17] and also sparked my interest in the representational power of behavioral descriptions.

The spring semester of 2014 was spent focused entirely on research, although a variety of interesting distractions wouldn't allow the direction of that research to become too focused. A former DSPG member, Petros Boufounos, was lecturing a course on information acquisition and processing, which inspired me to look into ways of combining sparse signal recovery algorithms with approximate signal processing concepts in the context of the compressed sensing paradigm. Regular research meetings with Al about optimization-free methods for sparse filter design and clever uses of convexity principles led to the development of a class of greedy algorithms and eventually the paper [84]. An interesting seminar on distributed computing models diverted my efforts toward yet another direction, and that investigation resulted in a precursor to the interconnective description in this thesis, although the purpose at that time was to design low complexity approximations to a system to improve distributability. As the semester was winding down, a different direction arose from a group meeting I led on the relationships between orthogonality and biorthogonality principles. That direction quickly turned into the approximate methods for solving linear problems appearing in [89] and later helped shed light on properties of certain scattering systems in the thesis. In full disclosure, a non-research related distraction named Chu also became an important person in the story that semester as well.

Through regular meetings with Tom at the Muddy Charles Pub that academic year, I began to develop an interest in his thesis work. In a particular conversation at the Muddy about conservation principles and conserved quantities in steepest-descent algorithms, Tom and I unearthed the idea of rotating the behaviors of certain features in these algorithms such as integrators, which are difficult to implement using discrete-time systems and whose discrete-time approximations result in numerical instabilities, so that they become well-behaved functional relationships. This idea and the conversations that followed led to the pair of papers [94, 111] and formed the cornerstone of the idea behind the scattering algorithms in this thesis. In fact, before heading off to Texas Instruments for another internship that summer, Tom and I spent a week in the Dominican Republic where many important details in those papers were ironed out while lounging poolside.

The summer of 2014 was spent working with Arthur in the area of high-performance computing with applications to quantitative trading and machine learning. Since it wasn't clear that the stock trading portion of the project would accommodate my retirement by summer's end, I spent the majority of nights and weekends working out the asynchronous and incremental capabilities of the optimization algorithms. By mid-summer, I began to work on web-based JavaScript implementations for some toy-sized example problems that would later, in collaboration with Tom, turn into the paper [106]. The summer progressed with my then-girlfriend Chu visiting Dallas, and ended with my return to Cambridge holding a notebook full of research questions aimed at better understanding the benefits of designing optimization algorithms as signal processing systems with conservation principles.

In the fall semester of 2014, my research progress initially slowed as I ramped into the role of lead instructor for an industry-only beta-version of the edX course 6.341x. Working alongside Al, Tom, and a team from MITx, I learned many valuable lessons related to teaching and online education, and even picked up some video editing skills along the way. This experience also taught me some important time and resource management lessons, especially in the context of working with groups of people scattered around the globe. On the research front, another rendezvous at the Muddy with Tom led to an interesting philosophical discussion of what a coordinate-free model of a signal processing system would look like. This idea persisted to distract me from the courses I was taking for the remainder of the semester. Although the absolute amount of time I could dedicate to research was limited, the progress I made that semester was significant in organizing my thoughts on how a coordinate-free perspective could be useful in designing and realizing fixed-point and optimization algorithms realized as distributed and asynchronous signal processing systems. The key elements to this progress came from studying distributed computing models, and understanding how the strengths of different signal processing platforms can be utilized in both centralized and decentralized settings.

The following spring semester saw a return to research motion (not necessarily progress), and my interest in the subject of variational calculus led to many intriguing research meetings about the sense in which physical systems such as bowling balls rolling down hills are optimal. Toward the end of the semester, a number of personal matters arose that began to affect my daily ability to make progress. During this period, I could not have asked for better or more supportive friends and mentors than Al and Tom, and with their assistance I prepared to spend the summer at home with family. Throughout the summer, research progress formed a healthy distraction from otherwise consuming issues, and this progress manifested itself in the areas of convergence analysis for asynchronous scattering algorithms and the development of the large-scale web-service available in [112].

By the time the fall 2015 semester rolled around, I found myself in a familiar position as a teaching assistant for 6.341. The two key goals this time around were to leverage the online tools and experiences from previous semesters to enhance the residential version of the course in a flipped classroom mode and to get the platform into a stable operating mode since this would be Al's final semester lecturing the course, and the online material developed over the past few years would transfer into Jim Ward's hands moving forward. The two key research tasks I focused my energy on that semester involved identifying how the scattering algorithms fit into the space of existing optimization methods and compiling the initial semblances of a thesis draft. The first task was accelerated by advice from Prof. Pablo Parrilo, which led to the coordinate transforms used to connect the scattering algorithms and their gradient-based and proximal counterparts. This connection also helped to identify additional benefits to the framework since these existing algorithm classes can also be designed in modular and distributable ways with minimal adjustment as well.

The start of the spring 2016 semester brought with it the beginning of an intense writing and revision period. With insightful feedback and commentary from my thesis committee, the level of discourse in the thesis continued to improve. Presenting various aspects of the research at ICASSP in Shanghai led to many intriguing conversations about the interplay between asynchronous architectures and algorithms and the benefits afforded by a signal-flow perspective. In particular, a discussion with Prof. Stephen Boyd helped identify several contexts where properties of scattering algorithms are desirable within the distributed optimization community. Upon returning to MIT and having been exposed to excellent research talks during the two years I served on the student subcommittee for faculty search, I spent the final days before the defense putting together a presentation that would hopefully appeal to a broad audience while simultaneously providing enough detail and rigor for signal processing and optimization experts. In parallel to thesis work, and with no small amount of careful planning and coordination, Chu and I were able to solve the classic two-body problem by securing full-time positions at Texas Instruments for that coming summer. In fact, before the semester got too far underway, on the first of March, Chu, the girl whose project group I was assigned to at the last minute in Luca's numerical methods class several semesters ago, and I got married.

It has become customary in DSPG for students to include a six-word summary of their journey through graduate school somewhere in their thesis. In thinking about my experiences at MIT, there is one common thread to all facets which I'd like to put forward as commentary to future graduate students (who somehow happen to read this):

Distractions, not directions, lead to progress.

As my story above illustrates, both personal and professional distractions can help in understanding which paths we should ultimately take and what parts of life matter the most. Distractions come in many forms, and repeatedly being distracted by the same thing may very well be a sign of what you're passionate about pursuing. Extending this to graduate life, I'd highly recommend discussing your distractions and ideas with collaborators and friends over a beer. You never know what insights and inspirations will arise. And finally, continue taking classes long after meeting the minimum coursework requirements. You never know what distractions you'll find or which distractions you'll meet.