ABSTRACT

Title: Optimization-based Robustness and Stabilization in Decentralized Control

Alborz Alavian, 2017

Dissertation directed by: Professor Michael C. Rotkowitz
Department of Electrical and Computer Engineering

This dissertation pertains to the stabilization, robustness, and optimization of Finite Dimensional Linear Time Invariant (FDLTI) decentralized control systems. We study these concepts for FDLTI systems subject to decentralizations that emerge from imposing sparsity constraints on the controller. While these concepts are well-understood in absence of an information structure, they continue to raise fundamental interesting questions regarding an optimal controller, or on suitable notions of robustness in presence of information structures.

Two notions of stabilizability with respect to decentralized controllers are considered. First, the seminal result of Wang & Davison in 1973 regarding internal stabilizability of perfectly decentralized system and its connection to the decentralized fixed-modes of the plant is revisited. This seminal result would be generalized to any arbitrary sparsity-induced information structure by providing an inductive proof that verifies and shows that those mode of the plant that are fixed with respect to the static controllers would remain fixed with respect to the dynamic ones. A

constructive proof is also provided to show that one can move any non-fixed mode of the plant to any arbitrary location within desired accuracy provided that they remain symmetric in the complex plane. A synthesizing algorithm would then be derived from the inductive proof.

A second stronger notion of stability referred to as "non-overshooting stability" is then addressed. A key property called "feedthrough consistency" is derived, that when satisfied, makes extension of the centralized results to the decentralized case possible.

Synthesis of decentralized controllers to optimize an $\mathcal{H}_\infty$-norm for model-matching problems is considered next. This model-matching problem corresponds to an infinite-dimensional convex optimization problem. We study a finite-dimensional parametrization, and show that once the poles are chosen for this parametrization, the remaining problem of coefficient optimization can be cast as a semidefinite program (SDP). We further demonstrate how to use first-order methods when the SDP is too large or when a first-order method is otherwise desired. This leaves the remaining choice of poles, for which we develop and discuss several methods to better select the most effective poles among many candidates, and to systematically improve their location using convex optimization techniques.

Controllability of LTI systems with decentralized controllers is then studied. Whether an LTI system is controllable (by LTI controllers) with respect to a given information structure can be determined by testing for fixed modes, but this gives a binary answer with no information about robustness. Measures have already been developed to determine how far a system is from having a fixed mode when one

considers complex or real perturbations to the state-space matrices. These measures involve intractable minimizations of a non-convex singular value over a power-set, and hence cannot be computed except for the smallest of the plants. We replace these problem by equivalent optimization problems that involve a binary vector rather than the power-set minimization and prove their equality. Approximate forms are also provided that would upper bound the original metrics, and enable us to utilize MINLP techniques to derive scalable upper bounds. We also show that we can formulate lower bounds for these measures as polynomial optimization problems, and then use sum-of-squares methods to obtain a sequence of SDPs, whose solutions would lower bound these metrics.

Optimization-based Robustness and Stabilization in Decentralized Control

by

Alborz Alavian

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2017

Advisory Committee:
Professor Michael C. Rotkowitz, Chair/Advisor
Professor Nuno Martins
Professor Perinkulam Krishnaprasad
Professor Andre Tits
Professor Nikhil Chopra, Dean's Representative

Acknowledgments

First and foremost, I would like to thank my advisor, Professor Michael Rotkowitz for extraordinary support through every single level of this work. Thanks for broadening my view through various abstractions regarding the research problems discussed herein, and many more that are not yet drafted. Thanks for keeping every kind of support available to me during hard times. Thanks for many discussions through which I could better grasp and develop a mindset of a researcher. Thanks for being fully supportive and communicative during conferences. I can not count the many ways you have guided me in formulating ideas, developing them, and making them better in research. Neither I can count the many times you have helped me regarding every professional activity that was concerning for me. It has been, and will be an honor to have the opportunity of having your supervision, and being your first student. Thanks for making my PhD experience rich in every possible way that I can think of.

Thank you Professor Nuno Martins, Professor P. S. Krishnaprasad, Professor Andre Tits, and Professor Nikhil Chopra for accepting to be on my committee.

Thank you Professor Nobakhti at the Sharif University of Technology for introducing me to the field of control, and creating a long lasting motivation to further study this field. Thanks for your continuous and extensive support during the graduate school applying process, and helping me in a great detail to do so.

Thanks to my colleagues for many nice and informative discussions. Thanks for sharing your own experience, and technical knowledge with me whenever I needed.

Thank you Waseem Malik, James Ferlez, Bhaskar Ramasubramanian, Mai Van Sy, and many others.

I can not put in words how much I am thankful to my family - my mother, father, brother and sister-in-law; Farah Naz, Khalil, Siavash and Neda, who have always shown the greatest level of support through my life. Thank you for making me feel positive on every challenging step that I had encountered through my academic and non-academic life.

Thanks to many friends who created memorable moments out of this time. Thanks for your continuous support. Thanks for being available even before this program started, and through the rest. Thanks for sharing your time, experience, and information with me. Thank you Pouya, Ladan, Ali, Mahshid, Farhad, Alireza, Niloofar, Behzad, Kian, Ali, Sina, Shervin, Ali, Arman, Mohammad Reza, Ashkan, Erfan, and Amir.

I would like to acknowledge financial support from the National Science Foundation (Grant No. 1351674) for the majority of the work herein, and also from the Department of Electrical and Computer Engineering at the University of Maryland.

It is impossible to remember all, and I apologize to those I have inadvertently left out.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Decentralization has long played a crucial role in equipping engineers with a significant paradigm based on which efficient control of dynamical systems could be designed and maintained. Not only such decentralized architectures would reduce the communication burden of large-scale systems and allow a scalable execution, but they are also more robust to maintain in critical situations, when such control systems are most needed.

Fundamental theoretical blocks of this paradigm has been a source of interesting open questions rooting back to the illustrious counter example of Witsenhausen [1]. There has been a re-newed interest in tackling the challenges that arise in this domain by employing optimization theory, mostly due to its thriving numerical capabilities.

This dissertation is concerned with synthesis of decentralized control systems by taking into account desirable steady-state and transient behaviors in the presence of constraints on the structure of the information exchange between subsystems. We will further analyze such decentralized architectures when the systems for which one want to design a decentralized controller is subjected to uncertainties of different sources that could possibly affect their steady-state behaviors.

Stabilization of LTI systems with respect to LTI decentralized controllers is studied in Chapter 2. Two notions of stabilizability are considered in that chapter. The first one is the conventional internal stabilizability . This corresponds to a seminal result in decentralized control regarding the development of fixed modes by Wang and Davison in 1973 – that, given a control structure, plant modes which cannot be moved with a static decentralized controller cannot be moved by a dynamic one either, and that the other modes which can be moved can be shifted to any chosen location with arbitrary precision. These results were developed for perfectly decentralized, or block diagonal, information structure, where each control input may only depend on a single corresponding measurement.

We consider fixed modes for arbitrary information structures. We provide a comprehensive proof that with a given information structure, the modes which cannot be altered by a static controller with the cannot be moved by a dynamic one either, and the modes which can be altered by a static controller can be moved by a dynamic one to any chosen location with arbitrary precision, thus generalizing and solidifying Wang and Davison's results.

This shows that a system can be internally stabilized by an LTI controller with the given information structure if and only if all of the modes which are fixed with respect to that structure are in the left half-plane; an algorithm for synthesizing such a stabilizing decentralized controller is also distilled from the proof.

Another notion of stability is also considered in the same chapter. This second criterion is stronger than internal stabilizability and requires that the energy of the state would always be decreasing, and is thus called non-overshooting stabilizability. We identify a key property which allows extension of the centralized results for this type of stabilizability. This property indeed holds for the most common classes of decentralized control problems. This enables one to determine that non-overshooting stabilizability with respect to static controllers is equivalent to non-overshooting stabilizability with respect to dynamic controllers, and to derive a linear matrix inequality (LMI) which either synthesizes a stabilizing controller or produces a certificate of non-stabilizability. We then compare these results with those for internal stability.

We then turn our attention to the problem of synthesizing decentralized controllers to optimize an $\mathcal{H}_\infty$-norm for model-matching problems. This is the subject of Chapter 3. The model-matching form that we address can arise from the closed-loop decentralized $\mathcal{H}_\infty$ problem when a certain condition on the information structure (namely Quadratic Invariance) is satisfied. There are no known methods to obtain an exact $\mathcal{H}_\infty$-optimal controller for a general information structure, and we develop several methods of obtaining an approximate solution by constructing finite-dimensional parametrizations of the controller. We show that once the poles are chosen for this parametrization, the remaining problem of coefficient optimization can be cast as a semidefinite program (SDP), and also demonstrate how to use first-order methods when the SDP is too large or when a first-order method is otherwise desired.

As for the poles, we consider them as dictionary elements for which an SDP gives their corresponding optimal coefficients. We use the poles from the centralized $\mathcal{H}_\infty$-optimal controller to suggest those for an initial dictionary, and then use sparsity promoting optimization methods to effectively select poles from many candidates. A first-order Taylor approximation is then explored that allows one to formulate another SDP that systematically adjusts the poles and the coefficients to improve the closed-loop performance.

Next, we consider two purely optimization problems as they provide the necessary optimization-based perspective and tools that we need in Chapter 6 to provide a tractable non-binary measure of robustness in the decentralized settings. We explore problems on the minimization of a particular singular value in Chapter 4, and on a subclass of Mixed Integer Non-Linear Programs (MINLP) in Chapter 5.

In Chapter 4 we consider the problem of minimizing a particular singular value of a matrix variable, subject to convex constraints. Convex heuristics for this problem are discussed, including some perhaps counter-intuitive results regarding which heuristic is the best, which provide upper bounds on the optimal value of the problem. The use of polynomial optimization formulations of the problem is considered, to yield lower bounds on the value of the problem. Sum-of-Squares (SOS) techniques are then used to formulate a lower bound on the polynomial optimization problem as an SDP. We show that the problem can also be formulated as an optimization problem with a bilinear matrix inequality (BMI), and discuss the use of this formulation.

In Chapter 5 we consider a class of MINLP problems that are convex except in terms of a vector of discrete variables. We introduce a class of methods, whereby part of the objective is replaced by a new variable that makes it possible to separately update each of the discrete variables. This maintains linear complexity of this update, while incorporating part of the objective minimization into the update. When a certain condition on the separability of the discrete variable in the objective is met, the resulting method shows significant improvements. It is still possible to capture these improvements even when this condition is not met, by means of hybrid methods that approximately decouple the discrete variables while preserving the same linear per-iteration complexity for the discrete variable update. Numerical comparison shows that a certain class of such hybrid algorithms, which only linearizes the effect of the non-dominant part of the coupling matrix, exhibits clear improvements in performance.

In Chapter 6 we consider the determination of non-binary measures of robust controllability with respect to decentralized controllers. Of course, whether an LTI system is controllable (by LTI controllers) with respect to a given information structure can be determined by testing for fixed modes, but this gives a binary answer with no information about robustness. A measure developed by Vaz and Davison in 1988 [2] nicely captures the distance from a plant to the closest one with a decentralized fixed-mode (DFM), and ties it to eigenvalue assignability; that is, how much effort is at most required to move the modes a given amount with the prescribed information structure. This is equivalently referred to as the complex DFM radius, that captures the smallest complex perturbation of the state-space matrices which would result in a fixed mode. The real DFM radius is a more realistic and less conservative measure, and captures the smallest real perturbation of the state-space matrices required to render the system to have a a fixed mode. This was also developed by Lam and Davison [3], more recently.

The main difficulty which have precluded widespread usage of these measures, is that they involve the minimization of a non-convex singular value of a matrix, which must further be minimized over a power set of the subsystems. This also includes an inner non-concave maximization over an additional parameter for the real DFM radius. We thus attained an easily computable, non-binary measure of controllability for LTI systems with decentralized controllers of arbitrary information structure.

We first transform this problem into a form that involves a polynomial over integer variables in the objective, and show that this would indeed result in exactly equal metrics for the complex and real DFM radius. Simpler forms involving affine combinations of the integer variables (rather than monomials) are then derived. We show that these simpler forms would correspond to an upper bound on the complex and real DFM radius, and use them in conjunction with MINLP approaches in Chapter 5 to derive an ADMM-based algorithm that decouples the effects of the integer variables, such that they can be optimized directly with per-iteration computations scaling linearly, rather than exponentially, with the number of subsystems. This method is shown to produce results which closely track the assignability measure across a variety of fixed mode types.

We conclude Chapter 6 with a discussion of upper and lower bounds for these metrics. Finding lower bounds is not only important for providing guarantees on where the true metric lies, but is typically more important since determining whether the metric is bounded away from zero corresponds to whether the system can be controlled at all. We will address these lower bounds by using the machinery developed for obtaining lower bounds on the k-th singular value of a polynomial matrix variable in Chapter 4.

1.1 Preliminaries

We will proceed by stating some preliminary notations, and then define the transfer functions of interest, and review the standard and conventional notion of stability that is mostly used in this dissertation. We then define plant and controller, their types, and review properties regarding their interconnection. We then define sparsity constraints and informations structures and their relation. Finally, Quadratic Invariance (QI) is reviewed, for use in Chapter 3.

Numbers

We proceed with the following preliminary definitions. Let $\mathbb{R}$ denote the set of real numbers, and $\bar{\mathbb{R}}$ be the extended real numbers: $\bar{\mathbb{R}} = \mathbb{R} \cup \{\pm\infty\}$. Denote binary set by $\mathbb{B} = \{0, 1\}$, and any finite subset of $\mathbb{R}$ by $\mathcal{Z}$, i.e.:

$$\mathcal{Z} \triangleq \{\alpha_1, \cdots, \alpha_{|\mathcal{Z}|}\},$$

where $|\mathcal{Z}| < \infty$, and $\alpha_i \in \mathbb{R}$, for $i \in \{1, \cdots, |\mathcal{Z}|\}$. Furthermore, denote the Cartesian product of m possibly different instances of such sets by:

$$\mathcal{Z}^{(m)} \triangleq \mathcal{Z}_1 \times \cdots \times \mathcal{Z}_m.$$

We will denote the projection of a real variable $y \in \mathbb{R}^m$ onto the set $\mathcal{Z}^{(m)}$ by:

$$\Pi_{\mathcal{Z}^{(m)}}\left(y\right) \;=\; \begin{bmatrix}\Pi_{\mathcal{Z}_1}\left(y_1\right) & \cdots & \Pi_{\mathcal{Z}_m}\left(y_m\right)\end{bmatrix}^T,$$

where

$$\Pi_{\mathcal{Z}_i}\left(y_i\right) \;=\; \arg\min_{z \in \mathcal{Z}_i} \;\|y_i - z\|_2,$$

for $i \in \{1, \cdots, m\}$. In its simplest form, all such $\mathcal{Z}_i$ could be taken the binary set, i.e., $\mathcal{Z}_i = \mathbb{B}$ for every $i \in \{1, \cdots, m\}$, which would result in $\mathcal{Z}^{(m)} = \mathbb{B}^m$, for which the projection would be simply picking the closest of either 0 or 1 to each element of the vector y, i.e., $\Pi_{\mathbb{B}}\left(y_i\right) = 1$, if $y_i > 0.5$ and 0 otherwise.

Define $\mathbb{C}$ to be the complex plane, and let $\Re(\cdot)$ and $\Im(\cdot)$ be the real and imaginary part of any complex number or matrix. Let $B(\lambda_0, \epsilon) \triangleq \{\lambda \in \mathbb{C} \mid |\lambda - \lambda_0| < \epsilon\}$ denote the open ϵ-ball around λ_0. Denote the unit imaginary number $\sqrt{-1}$ by $\mathbf{j}$, the imaginary axis by $\mathbf{j}\mathbb{R} \triangleq \left\{z \in \mathbb{C} \mid \Re(z) = 0\right\}$, the open left-half plane (LHP) by $\mathbb{C}^- \triangleq \{\lambda \in \mathbb{C} | \Re(\lambda) < 0\}$, and the closed right half of the complex plane by $\bar{\mathbb{C}}^+ \triangleq \mathbb{C} \setminus \mathbb{C}^-$. Denote the unit disk by $\mathbb{D} = \left\{z \in \mathbb{C} \mid |z| < 1\right\}$ and unit circle by $\partial\mathbb{D}$, and the closed space outside the unit disk by $\bar{\mathbb{D}}^+ = \mathbb{C} \setminus \mathbb{D}$.

Vectors

Let $\mathbf{e}_i$ denote the unit vector of all zeros except for i^{th} element which is 1. Define $\mathbf{1}$ to be the vector of all ones, and I to be the identity matrix. Note that dimension of $\mathbf{e}_k$, $\mathbf{1}$, and I should be clear from the context and thus we suppress their explicit dependence in the notation. Unless otherwise declared, vector norms in this paper are all standard Euclidean norm

$$\|v\|^2 \ = \ \langle v, v\rangle \ = \ v^*v \quad \text{for } v \in \mathbb{C}^n,$$

where superscript $(\cdot)^*$ denotes the Hermitian operator.

Vectorization

For a real matrix R define $\mathrm{vec}\,(R)$ as the vectorization operator that puts columns of $R \in \mathbb{R}^{m\times n}$ on top of each other, and produces a vector in $\mathbb{R}^{mn}$. We will take the vectorization operator for a complex matrix $A \in \mathbb{C}^{m\times n}$ as

$$\mathrm{vec}\,(A) = \begin{bmatrix} \mathrm{vec}\,(\Re(A)) \\ \\ \mathrm{vec}\,(\Im(A)) \end{bmatrix}.$$

Similarly, define $\mathrm{vec}^{-1}\,(v)$ as a reshaping operator which puts elements of $v \in \mathbb{C}^{2mn}$ in the right position in a matrix in $\mathbb{C}^{m\times n}$ such that $\mathrm{vec}^{-1}\,(\mathrm{vec}\,(A)) = A$. We do not explicitly indicate the dimensions required for the reshaping operator $\mathrm{vec}^{-1}\,(\cdot)$, and note that it should be clear from the context wherever used.

Matrices

For a matrix $A \in \mathbb{C}^{m\times n}$, let $\sigma_{\max}(A) \triangleq \sigma_1(A) \geq \cdots \geq \sigma_{\max(m,n)}(A) \triangleq \sigma_{\min}(A)$ denote its singular values. It is obvious that $\sigma_k(A) = 0$ for $\min(m,n) < k \leq \max(m,n)$. If $m = n$ and A is a Hermitian matrix, i.e., $A^* = A$, then all of its eigenvalues would be in $\mathbb{R}$ and one can index them so that $\lambda_{\max}(A) \triangleq \lambda_1(A) \geq \cdots \geq \lambda_n(A) \triangleq \lambda_{\min}(A)$. We refer to the non-negative and negative eigenvalues respectively by $\mathrm{eig}_{+,0}(A) \triangleq \mathrm{eig}\,(A) \cap \bar{\mathbb{C}}^{+}$, and $\mathrm{eig}_{-}(A) \triangleq \mathrm{eig}\,(A) \cap \mathbb{C}^{-}$.

We use the standard inner product on $\mathbb{C}^{n\times n}$ matrices, given by $\langle L, V\rangle = \mathrm{tr}\,(L^{*}V)$, where $\mathrm{tr}\,(\cdot)$ is the trace operator.

Matrix Norms

For a complex matrix $A \in \mathbb{C}^{m\times n}$ denote its Ky Fan k-norm [4, Eq. IV-33, p. 92] by:

$$s_k(A) \;\triangleq\; \sum_{\ell=1}^{k} \sigma_\ell(A). \tag{1.1}$$

We then have that the matrix 2-norm (also known as the spectral norm) of A, denoted by $\|A\|_2$, is $\|A\|_2 = s_1(A) = \sigma_1(A)$, and its Nuclear norm, denoted by $\|\cdot\|_*$ is equal to $\|A\|_* = s_{\max(m,n)}(A) = \sum_k \sigma_k(A)$. We also denote the Frobenius norm by $\|\cdot\|_{\mathrm{F}}$ for which we have:

$$\|A\|_{\mathrm{F}}^2 \;=\; \langle A, A\rangle \;=\; \sum_{ij}|A_{ij}|^2 \;=\; \sum_k \sigma_k^2(A).$$

Also the $\|\cdot\|_\infty$-norm for real matrices is defined as:

$$\|A\|_\infty = \max_i \left(\sum\nolimits_j |A_{ij}|\right).$$

Continuous Time Transfer Functions

We are mostly interested in continuous-time systems. We define transfer functions for continuous-time systems. A rational function $G : \mathbb{C} \to \mathbb{C}$ is called ***real-rational*** if the coefficients of its numerator and denominator polynomials are real.

Similarly, a matrix-valued function $G : \mathbb{C} \to \mathbb{C}^{m\times n}$ is called real-rational if G_{ij} is real-rational for all i, j. A rational polynomial is called ***proper*** if the degree of its denumerator is greater than or equal to the degree of its numerator, and ***strictly proper*** if the degree of its denumerator is strictly greater than the degree of its numerator.

Denote by $\mathcal{R}_{\mathrm{p}}^{m\times n}$ the set of matrix-valued real-rational proper transfer matrices

$$\mathcal{R}_{\mathrm{p}}^{m\times n} = \Big\{ G : \mathbb{C} \to \mathbb{C}^{m\times n} \mid G \text{ proper, real-rational} \Big\},$$

and let

$$\mathcal{R}_{\mathrm{sp}}^{m\times n} \triangleq \Big\{ G \in \mathcal{R}_{\mathrm{p}}^{m\times n} \mid G \text{ strictly proper} \Big\}.$$

Also let $\mathcal{RH}_\infty$ be the set of real-rational proper stable transfer matrices:

$$\mathcal{RH}_\infty^{m\times n} = \Big\{ G \in \mathcal{R}_{\mathrm{p}}^{m\times n} \mid G \text{ has no poles in } \bar{\mathbb{C}}^+ \Big\}.$$

Discrete Time Transfer Functions

We use Discrete Time systems explicitly in Chapter 3. We define transfer functions for discrete-time systems determined on unit circle. A rational function $G : \mathbb{C} \to \mathbb{C}$ is called ***real-rational*** if the coefficients of its numerator and denominator polynomials are real. Similarly, a matrix-valued function $G : \partial\mathbb{D} \to \mathbb{C}^{m\times n}$ is called real-rational if G_{ij} is real-rational for all i, j. Similarly denote by $\mathcal{R}_{\mathrm{p}}^{m\times n}$ the set of matrix-valued real-rational proper transfer matrices

$$\mathcal{R}_{\mathrm{p}}^{m\times n} = \Big\{ G : \partial\mathbb{D} \to \mathbb{C}^{m\times n} \mid G \text{ proper, real-rational} \Big\},$$

and let $\mathcal{R}_{\mathrm{sp}}^{m\times n}$ be

$$\mathcal{R}_{\mathrm{sp}}^{m\times n} = \Big\{ G \in \mathcal{R}_{\mathrm{p}}^{m\times n} \mid G \text{ strictly proper} \Big\}.$$

Also let $\mathcal{RH}_\infty$ be the set of real-rational proper stable transfer matrices

$$\mathcal{RH}_\infty^{m\times n} = \Big\{ G \in \mathcal{R}_{\mathrm{p}}^{m\times n} \mid G \text{ has no poles in } \bar{\mathbb{D}}^{+} \Big\}.$$

It can be shown that functions in $\mathcal{RH}_\infty$ are determined by their values on $\partial\mathbb{D}$, and thus we can regard $\mathcal{RH}_\infty$ as a subspace of $\mathcal{R}_{\mathrm{p}}$.

A discrete-time transfer function matrix G belongs to $\mathcal{H}_\infty$ if and only if

$$\operatorname{ess\,sup}_{\omega\in[0,2\pi)} \sigma_{\max}\left(G(e^{\mathbf{j}\omega})\right) < \infty,$$

where $\sigma_{\max}(\cdot)$ gives the maximum singular value. Similarly the $\mathcal{H}_\infty$-norm of an m-by-n $G \in \mathcal{H}_\infty$ is:

$$\begin{aligned} \|G\|_{\mathcal{H}_\infty} &= \operatorname{ess\,sup}_{\omega\in[0,2\pi)} \sigma_{\max}\left(G(e^{\mathbf{j}\omega})\right) \\ &= \operatorname{ess\,sup} \quad \Re\left(u^{*}G(e^{\mathbf{j}\omega})v\right), \end{aligned}$$

where the last essential supremum is taken over $\omega \in [0, 2\pi), u \in \mathbb{C}^m, v \in \mathbb{C}^n$ with $\|u\| = \|v\| = 1$.

We could now equivalently define $\mathcal{RH}_\infty^{m\times n} = \mathcal{R}_\mathrm{p}^{m\times n} \cap \mathcal{H}_\infty^{m\times n}$. When the dimensions are implied by context, we omit the superscripts of $\mathcal{R}_\mathrm{p}^{m\times n}, \mathcal{R}_\mathrm{sp}^{m\times n}, \mathcal{RH}_\infty^{m\times n}, \mathcal{H}_\infty^{m\times n}$.

Stability Notions

We give the general definitions for stability of a linear system for its equilibrium at zero. If the equilibrium point is not zero, a shift of the state could be applied to the dynamics, to make the equilibrium point zero: We will specially consider the following LTI dynamical system for which the only possible equilibrium point would be zero.

$$\dot{x}(t) = Ax(t), \qquad x(t_0) = x_0, \tag{1.2}$$

where $x \in \mathbb{R}^n$ and $A \in \mathbb{R}^{n\times n}$.

Definition 1 ([5, Definition 2.2])**.** *A dynamical system is said to be:*

1. *Lyapunov stable at time t_0 if and only if for each $\epsilon > 0$, there exists a $\delta(\epsilon) > 0$ such that $\|x(t_0)\| < \delta(\epsilon)$ implies that $\|x(t)\| < \epsilon$ for all $t \geq t_0$.*

2. *Asymptotically stable if and only if it is Lyapunov stable and $\delta(\epsilon)$ in the above part can be selected so that $\lim_{t\to\infty} \|x(t)\| = 0$.*

As a widely known notion of stability for LTI systems, we consider the second part of the above definition and refer to it by the following name:

Remark 2. *A necessary and sufficient condition for asymptotic stability of the* *system of the form* (1.2) *is that* $\Re(\mathrm{eig}(A)) < 0$. *We will also refer to this type of stability as* ***internal stability****.*

Except otherwise noted, all the stability notions through the rest are regarding internal stability.

Plant and Controller

We suppose that we have a Finite Dimensional Linear Time Invariant (FDLTI) causal, generalized plant $P \in \mathcal{R}_{\mathrm{p}}^{(n_z+n_y)\times(n_w+n_u)}$, partitioned as

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & G \end{bmatrix}.$$

Given a controller $K \in \mathcal{R}_{\mathrm{p}}^{n_u \times n_y}$, we define the (lower) ***linear fractional transformation*** (LFT) of P and K

$$f_{\mathrm{LFT}}(P, K) \triangleq P_{11} + P_{12}K(I - GK)^{-1}P_{21}. \tag{1.3}$$

This interconnection is shown in Figure 1.1. This generalized plant P is only used in Chapter 3. Elsewhere, when we talk about about the plant, we mean the map G from u to y.

We suppose that there are n_y sensor measurements and n_u control actions,

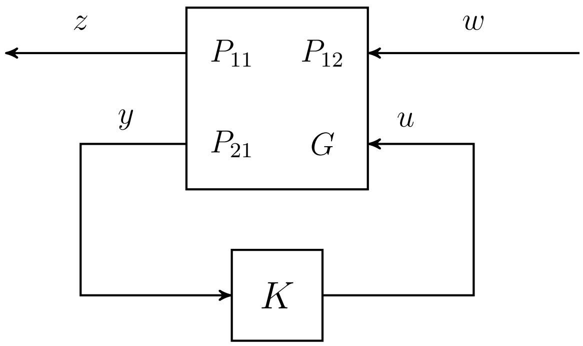


Figure 1.1: Linear fractional interconnection of P and K

and thus partition the sensor measurements and control actions as

$$y = \begin{bmatrix} y_1 \\ \vdots \\ y_{n_y} \end{bmatrix}, \qquad u = \begin{bmatrix} u_1 \\ \vdots \\ u_{n_u} \end{bmatrix},$$

and then further partition K as

$$K = \begin{bmatrix} K_{11} & \dots & K_{1n_y} \\ \vdots & & \vdots \\ K_{n_u1} & \dots & K_{n_un_y} \end{bmatrix}.$$

We also assume that we are provided with a minimal state-space representation of G, denoted by (A, B, C, D). Wherever the results depend on a specific state-space realization, we refer to G as a ***state-space system***. We have that $A \in \mathbb{R}^{n \times n}$,

$B \in \mathbb{R}^{n \times n_u}$, $C \in \mathbb{R}^{n_y \times n}$, and $D \in \mathbb{R}^{n_y \times n_u}$. We will decompose B column-wise as:

$$B = \begin{bmatrix} B_1 & B_2 & \cdots & B_{n_u} \end{bmatrix},$$

and C row-wise as:

$$C = \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_{n_y} \end{bmatrix}.$$

With a slight abuse of notation, when $\text{eig}_-(\cdot)$, and $\text{eig}_{+,0}(\cdot)$ are applied on a general LTI system G, we mean the negative, and non-positive eigenvalues of dynamic matrix of the minimal state-space representation of that system, i.e., $\text{eig}_-(G) \triangleq \text{eig}_-(A)$, and $\text{eig}_{+,0}(G) \triangleq \text{eig}_{+,0}(A)$.

We also denote a state-space representation of K by (A_K, B_K, C_K, D_K), and define types of controllers that will help us to easily refer to whether a controller K is static, proper dynamic, or static for some elements but proper dynamic for others. We will make use of the following controller types:

- $\mathcal{T}^{\text{d}}$: Set of finite order proper dynamic controllers, i.e., A_K, B_K, C_K, D_K each are real matrices of compatible dimension.

- $\mathcal{T}^{\text{s}}$: Set of static controllers, i.e., A_K, B_K, and C_K are all zero and only D_K could be non-zero, i.e., $K_{ij} \in \mathbb{R}$ for all (i, j).

- $\mathcal{T}^{\text{s}+1}_{i,j}$: Set of controllers such that all the elements of controller are static except for $(i,j)^{\text{th}}$ element which could be proper dynamic, i.e., we have $K_{\ell k} \in \mathbb{R}$ for all $(\ell, k) \neq (i,j)$, and K_{ij} is a proper transfer function in σ, where σ could be taken s in the CT, or z in DT. This could be read as "static plus one".

- $\mathcal{T}_{\mathcal{I}}^{\text{s}+k}$: Set of controllers such that all the elements of controller are static except for k indices in the set $\mathcal{I} \triangleq \{(i_1, j_1), \cdots, (i_k, j_k)\}$, i.e., for all $(i,j) \notin \mathcal{I}$, $K_{ij} \in \mathbb{R}$ and for all $(i,j) \in \mathcal{I}$, K_{ij} is a proper transfer function in σ. This could be read as "static plus k".

Given a (not necessarily minimal) state-space representation of G and K, the ***closed-loop*** of G and K (Figure 1.2) has a state-space representation with dynamics matrix denoted by $A_{\text{CL}}(G,K)$, given by:

$$A_{\text{CL}}(G,K) \triangleq \begin{pmatrix} A + BD_KNC & BMC_K \\ B_KNC & A_K + B_KNDC_K \end{pmatrix}, \tag{1.4}$$

where $M \triangleq (I - D_KD)^{-1}$, and $N \triangleq (I - DD_K)^{-1}$. We have $MD_K = D_KN$, and similarly $DM = ND$. Let $\Gamma(G,K)$, as illustrated in Figure 1.2, denote the map from the set-point to the outputs of G (i.e., from r to y), when K is closed around G. A state-space representation for this closed-loop interconnection $\Gamma(G,K)$ is given by:

$$\Gamma(G,K) = \left[\begin{array}{c|c} A_{\text{CL}}(G,K) & \begin{matrix} BM \\ B_KDM \end{matrix} \\ \hline \begin{matrix} NC & DMC_K \end{matrix} & DM \end{array}\right](\sigma)$$

As a property of $\Gamma(\cdot,\cdot)$, we have the following relation for the given state-space

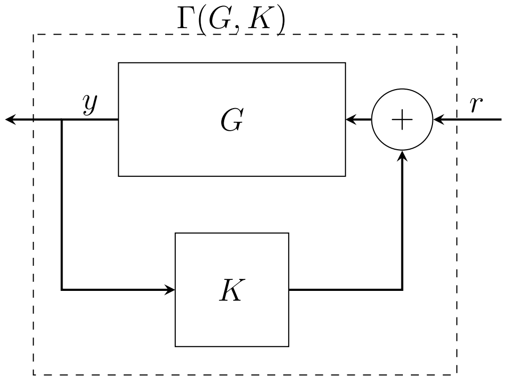


Figure 1.2: The map from set-points to outputs when K is closed around G.

realization of the plant G, and controllers K_1 and K_2:

$$\Gamma(\Gamma(G, K_1), K_2) = \Gamma(G, K_1 + K_2), \qquad \forall\ K_1, K_2. \tag{1.5}$$

which can be verified by writing the state-space representation of both sides.

Sparsity Patterns

Suppose that $K^{\text{bin}} \in \mathbb{B}^{m \times n}$ is a binary matrix. The following is the subspace of $\mathcal{R}_{\text{p}}^{m \times n}$ comprising the transfer function matrices that satisfy the sparsity constraints imposed by K^{bin}:

$$\begin{aligned} \text{Sparse}(K^{\text{bin}}) \triangleq \{K \in \mathcal{R}_{\text{p}}^{m \times n} \mid K_{ij}(\sigma) = 0 \text{ for almost all } \sigma \in \mathbb{C} \\ \text{and for all } i, j \text{ such that } K_{ij}^{\text{bin}} = 0\} \end{aligned}$$

Conversely, given $K \in \mathcal{R}_{\text{p}}^{m \times n}$, we define $\text{Pattern}(K) \triangleq K^{\text{bin}}$, where K^{bin} is the binary matrix given by:

$$K_{ij}^{\text{bin}} = \begin{cases} 0, & \text{if } K_{ij}(\sigma) = 0 \text{ for almost all } \sigma \in \mathbb{C} \\ 1, & \text{otherwise,} \end{cases}$$

for $i \in \{1, \dots, m\},\ j \in \{1, \dots, n\}$.

Information Structures

We want to impose ***information structures*** on the controller which could be described by a subset $\mathcal{S} \subset \mathcal{R}_{\text{p}}^{n_u \times n_y}$, and want to have $K \in \mathcal{S}$. It is only in Section 2.2 that we account for **general information structures**, and everywhere else in this dissertation we consider **sparsity-induced structures** on controllers such that each control input may access certain sensor measurements, but not others.

We represent sparsity constraints on the overall controller via a binary matrix $K^{\text{bin}} \in \mathbb{B}^{n_u \times n_y}$. Its entries can be interpreted as follows:

$$K_{kl}^{\text{bin}} \;=\; \begin{cases} 1, & \text{if control input } k \text{ may access sensor measurement } l, \\ 0, & \text{if not,} \end{cases}$$

for all $k \in \{1, \cdots, n_u\},\ l \in \{1, \dots, n_y\}$.

The subspace of admissible controllers can be expressed as:

$$\mathcal{S} \;=\; \text{Sparse}(K^{\text{bin}}).$$

For a sparsity pattern $\mathcal{S}$, we similarly let $\mathrm{Adm}(\mathcal{S})$ denote the set of admissible indices for which the controller is allowed to be non-zero, i.e., $(i,j) \notin \mathrm{Adm}(\mathcal{S})$ if and only if $K_{ij}^{\mathrm{bin}} = 0$.

Also for simplicity we define the following sparsity patterns:

- $\mathcal{S}_{\mathrm{c}}$: Centralized sparsity patterns, i.e., no sparsity constraints are imposed on the controller. $\mathrm{Adm}(\mathcal{S}) = \{(i,j) \ \ \forall\, i,j\}$.

- $\mathcal{S}_{\mathrm{d}}$: Diagonal sparsity patterns, i.e., $n_u = n_y$ and $K(\sigma)$ must be zero for all off-diagonal term (for almost all σ). $\mathrm{Adm}(\mathcal{S}) = \{(i,i) \ \ \forall\, i\}$.

For any sparsity pattern $\mathcal{S}$, let $a \triangleq |\mathrm{Adm}(\mathcal{S})|$ be the number of admissible non-zero indices in controller, and let the tuple

$$\mathcal{I}(\mathcal{S}) \ \triangleq \ \{(i_1, j_1), \cdots, (i_a, j_a)\} \tag{1.6}$$

be any arbitrary ordering of admissible non-zero indices of controller.

For any $D \in \mathcal{T}^{\mathrm{s}} \cap \mathcal{S}$, we define the sequence of matrices $D|_{(m)} \in \mathbb{R}^{n_u \times n_y}$, $m \in \{0, 1, \cdots, a\}$ with $a = |\mathrm{Adm}(\mathcal{S})|$ as:

$$D|_{(0)} \triangleq 0, \qquad D|_{(m)} \triangleq \sum_{\ell=1}^{m} \mathbf{e}_{i_\ell} D_{i_\ell j_\ell} \mathbf{e}_{j_\ell}^T \quad \text{for } m \in \{1, \cdots, a\} \tag{1.7}$$

where $\mathbf{e}_{i_\ell} \in \mathbb{R}^{n_u}$ and $\mathbf{e}_{j_\ell} \in \mathbb{R}^{n_y}$, for $\ell \in \{1, \cdots, a\}$. We thus have $D|_{(a)} = D$. This $D|_{(m)}$ gives the static controller matrix with only the first m admissible indices.

For a given $i \in \{1, \cdots, n_u\}$, define

$$\mathtt{J}_i \;\triangleq\; \{j \mid j \in \{1, \cdots, n_y\} \text{ and } K^{\text{bin}}_{ij} = 1\}, \tag{1.8}$$

which are the set of sensor measurements y_j that control action u_i is allowed to access. For a subset $\mathtt{I} \subseteq \{1, \cdots, n_u\}$, denote its complement by $\bar{\mathtt{I}} \triangleq \{1, \cdots, n_u\} \setminus \mathtt{I}$. Similarly define $\mathtt{J}_{\mathtt{I}} \triangleq \bigcup_{i \in \mathtt{I}} \mathtt{J}_i$, which are the set of sensor measurements that can be seen from inputs in $\mathtt{I}$. Also, for any subset $\mathtt{I} = \{i_1, \cdots, i_{|I|}\}$, we define

$$B_{\mathtt{I}} \;\triangleq\; \begin{bmatrix} B_{i_1} & \cdots B_{i_{|\mathtt{I}|}} \end{bmatrix}.$$

Likewise, for any subset $\mathtt{J} = \{j_1, \cdots, j_{|\mathtt{J}|}\}$, define

$$C_{\mathtt{J}} \;\triangleq\; \begin{bmatrix} C_{j_1} \\ \vdots \\ C_{j_{|\mathtt{J}|}} \end{bmatrix}.$$

As there is no inherent ordering in the sets $\mathtt{I}$ (and $\mathtt{J}$), the aforementioned $B_{\mathtt{I}}$ (and $C_{\mathtt{J}}$) could defer up to column (and row) permutations, in which case, any column (and row) permutation of $B_{\mathtt{I}}$ (and $C_{\mathtt{J}}$) is a valid choice.

Quadratic Invariance (QI)

We now define quadratic invariance,

Definition 3 ([6, Definition 2]). *Let a causal linear time-invariant plant, rep-*

resented via a transfer function matrix G in $\mathcal{R}_\mathrm{p}^{n_y \times n_u}$, be given. If $\mathcal{S}$ is a subset of $\mathcal{R}_\mathrm{p}^{n_u \times n_y}$ then $\mathcal{S}$ is called ***quadratically invariant*** *under G if the following inclusion holds:*

$$KGK \in \mathcal{S} \qquad \textit{for all } K \in \mathcal{S}.$$

For the case of sparsity constraints, it was shown in [6] that a necessary and sufficient condition for quadratic invariance is

$$K_{ki}^{\mathrm{bin}}\ G_{ij}^{\mathrm{bin}}\ K_{jl}^{\mathrm{bin}}\ (1 - K_{kl}^{\mathrm{bin}}) = 0, \tag{1.9}$$

for all $i, l \in \{1, \ldots, n_y\}$, and all $j, k \in \{\{1, \cdots, n_u\}\}$.

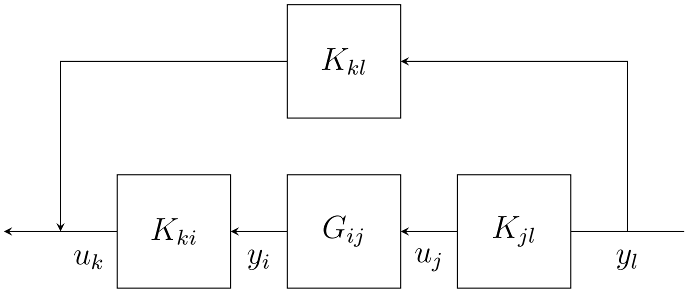


Figure 1.3: QI interpretation for sparse controllers

An interpretation (see Figure 1.3) is that if a sensor measurement (y_l) can indirectly effect a control input (u_k) through the plant, then that controller must be able to directly observe that measurement ($K_{kl}^{\mathrm{bin}} = 1$). This is closely related to the notion of partial nestedness [7, 8], and many problems of interest either fall in this class or can be relaxed or approximated to fall in this class.

Chapter 2: Stabilizability

This chapter is concerned with the stabilization of decentralized control systems, for which certain controller inputs may depend on some measurements but not others. This corresponds to finding a stabilizing controller which satisfies a given sparsity constraint. A special case of this, sometimes referred to as *perfectly decentralized control*, occurs when each control input u_i may depend only on a single associated measurement y_i, which corresponds to finding a stabilizing controller which is (block) diagonal. This special case is sometimes itself referred to as *decentralized control*, particularly in the literature from a few decades ago. This malleability or evolution of the definition has not only caused some confusion, but has also resulted in some important results in the field only being studied for this special case.

We will consider two different notions of stabilization in this chapter. Section 2.1 discusses internal stabilizability for the FDLTI decentralized systems and is built upon a seminal result in decentralized control regarding the development of *fixed modes* by Wang and Davison in 1973 [9]. That paper studies (FDLTI) perfectly decentralized stabilization of FDLTI plants. Its contributions can be broken into three main components - a definition establishing the framework, and two subsequent results. Fixed modes were defined as those modes of the plant which could not be altered by any static perfectly decentralized controller (that is, by any diagonal matrix). The first result was that these fixed modes could also not be altered by any dynamic perfectly decentralized controller: if you can't move it with a static diagonal controller, you can't move it with a dynamic diagonal controller. The second result was that if a mode is not fixed, then it can be moved arbitrarily close to any chosen location in the complex plane (provided that it has a complex conjugate pair if it is not real). These can be taken together to state that a system is stabilizable by a (dynamic) perfectly decentralized controller if and only if all of its (static) fixed modes are in the left half-plane (LHP).

When proving these results, it was shown that allowing one part of the controller to be dynamic does not result in any fewer fixed modes than a static controller, and then claimed that the first result followed; that is, that a dynamic controller would not be able to move any of the fixed modes. Similarly, it was shown that a single non-fixed mode could be moved to any chosen location, and then claimed that the second result followed; that is, that an arbitrary number of non-fixed modes could be simultaneously moved to chosen locations by a single controller. Getting from these initial steps to a rigorous inductive argument, however, is not trivial.

Here in the present chapter, we extend these fundamental concepts for arbitrary information structure, while developing robust notation and rigorous proofs, thus placing the new and existing results on a sound mathematical footing as was considered in [10–12].

We first introduce notation for fixed modes that allows its dependence on information structure, as well as with the allowed type of controllers (linear static, linear dynamic, non-linear, etc.). We then show that, for arbitrary information structure, the fixed modes with respect to dynamic controllers are the same as the fixed modes with respect to static controllers. Moreover, we provide a rigorous proof that the non-fixed modes can then be moved to within an arbitrarily small distance of chosen (conjugate) locations, using a dynamic LTI controller with the given structure, thus extending and solidifying the seminal results of Wang and Davison. The proof is constructive, and we lastly distill an explicit algorithm for the stabilizing decentralized controller synthesis from the proof.

The obvious potential benefits of this are an increased understanding of decentralized stabilizability, and the verification of important existing results. It is also our hope that the notation developed will be useful in further extending our understanding of decentralized stabilizability to richer classes of controllers for which the fixed modes may diminish relative to the original static definition, particularly non-linear and/or time-varying controllers [13–16].

It is known that the centralized fixed modes (that are fixed with respect to a centralized linear static controller) will still be present after applying any causal controller including linear or nonlinear controllers, dynamic or static ones, and finite-dimensional or infinite-dimensional ones [17, Section 6.1, p. 237, Remark 3]. A certain class of the decentralized fixed modes, namely quotient fixed modes, also have the same property and will remain present after applying any causal controller that satisfy the information structure [16], while non-quotient decentralized fixed modes can be eliminated by a periodically time-varying decentralized controller [18, 19].

We further note that demonstrating the results of this section directly for arbitrary structure, as opposed to attempting to diagonalize the problem and then prove the original perfectly decentralized results, would likely be useful when other types of stability are required which are not invariant under such transformations, though we currently focus on internal stability. As an example of the diagonalization approach, readers are referred to [20], where existence of a stabilizing controller under arbitrary sparsity-induced information structures has been demonstrated by transforming the problem into a diagonal one to which [9] could be applied. Furthermore, [20] demonstrates an analytical test for determining structural fixed modes under arbitrary sparsity-induced information structure and shows its equivalence to a graph-theoretical condition.

Also, dealing with the original structure is preferable since stabilizing controllers can be constructed without having to first expand their size. Finally, while the proofs in [9], (as well as [16]), are constructive in nature, they do not clearly lead to an explicit synthesis algorithm. A further advantage of proving this result in earnest was the ability to extract such an algorithm, which then finds a stabilizing LTI decentralized controller whenever one exists.

In contrast with Section 2.1, we study the stabilization of systems with decentralized controllers when the stability criterion of interest is instead "non-overshooting stability" in Section 2.2. This criterion is stronger than those which have typically been studied, particularly for decentralized control, and requires that the energy of the state always be decreasing. We identify a key property which allows centralized results for this type of stability to be extended, which indeed holds for the most common classes of decentralized control problems. Stabilizability with respect to static controllers is equivalent to stabilizability with respect to the dynamic ones for this notion of stabilization. This allows one to derive a linear matrix inequality (LMI) which either synthesizes a stabilizing controller or produces a certificate of non-stabilizability. We then compare these results with those for internal stability, i.e., fixed modes.

2.1 Stabilization for Arbitrary Sparsity-Induced Information Structure

We will define fixed modes, and introduce some auxiliary notation regarding that in Section 2.1.1. We review the Kalman canonical form, and state some preliminary results regarding the relation between that form and fixed-modes in Section 2.1.2. We then provide comprehensive proofs, to verify that those modes of the plant that are fixed with respect to the static controller will remain fixed with respect to a dynamic one (in Section 2.1.3), and that all the non-fixed modes of the plant could be placed arbitrary close to any conjugate location in the complex plane (in Section 2.1.4). Finally, in Section 2.1.5 we derive an explicit stabilization algorithm from the main proof.

2.1.1 Review

We will begin by defining fixed-modes, also known as Decentralized Fixed Modes (DFM):

Definition 4. *The set of fixed modes of a plant G with respect to a sparsity pattern $\mathcal{S}$*

and a type $\mathcal{T}$, is defined to be:

$$\begin{aligned}\Lambda\left(G,\mathcal{S},\mathcal{T}\right) &\triangleq \left\{\lambda\in\mathbb{C} \mid \lambda\in\operatorname{eig}\left(A_{\mathrm{CL}}(G,K)\right)\ \forall\ K\in\mathcal{S}\cap\mathcal{T}\right\} \\ &= \bigcap_{K\in\mathcal{S}\cap\mathcal{T}}\operatorname{eig}\left(A_{\mathrm{CL}}(G,K)\right).\end{aligned}$$

Remark 5. *This reduces to the definition of fixed modes in [9] if $\mathcal{S}=\mathcal{S}_{\mathrm{d}}$ (diagonal structure) and $\mathcal{T}=\mathcal{T}^{\mathrm{s}}$ (static controllers).*

For any FDLTI plant G, denote its open-loop modes by $\zeta(G)=\operatorname{eig}(A)$, and for each mode $\lambda\in\zeta(G)$, let $\mu(\lambda,G)$ denote its algebraic multiplicity. We will partition the open-loop modes as:

$$\zeta(G) \;=\; \Lambda\left(G,\mathcal{S},\mathcal{T}^{\mathrm{s}}\right)\ \cup\ \tilde{\Lambda}(G,\mathcal{S},\mathcal{T}^{\mathrm{s}}) \tag{2.1}$$

where

$$\tilde{\Lambda}(G,\mathcal{S},\mathcal{T}^{\mathrm{s}})=\operatorname{eig}(A)\setminus\Lambda\left(G,\mathcal{S},\mathcal{T}^{\mathrm{s}}\right)$$

gives the non-fixed modes, which we then further partition as:

$$\tilde{\Lambda}(G,\mathcal{S},\mathcal{T}^{\mathrm{s}}) \;=\; \tilde{\Lambda}_{+}(G,\mathcal{S},\mathcal{T}^{\mathrm{s}})\ \cup\ \tilde{\Lambda}_{-}(G,\mathcal{S},\mathcal{T}^{\mathrm{s}}),$$

where

$$\begin{aligned}\tilde{\Lambda}_{+}(G,\mathcal{S},\mathcal{T}^{\mathrm{s}}) &= \{\alpha\in\zeta(G)\mid\Re(\alpha)\geq 0\}\setminus\Lambda\left(G,\mathcal{S},\mathcal{T}^{\mathrm{s}}\right) \\ &= \tilde{\Lambda}(G,\mathcal{S},\mathcal{T}^{\mathrm{s}})\ \cap\ \bar{\mathbb{C}}^{+} \\ &= \{\alpha_1,\cdots,\alpha_{|\tilde{\Lambda}_{+}(G,\mathcal{S},\mathcal{T}^{\mathrm{s}})|}\}\end{aligned}$$

are distinct unstable non-fixed open-loop eigenvalues of A, and

$$\begin{aligned}\tilde{\Lambda}_{-}(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}}) &= \{\beta \in \zeta(G) \mid \Re(\beta) < 0\} \setminus \Lambda\left(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}}\right) \\ &= \tilde{\Lambda}(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}}) \ \cap \ \mathbb{C}^{-} \\ &= \{\beta_1, \cdots, \beta_{|\tilde{\Lambda}_{-}(G,\mathcal{S},\mathcal{T}^{\mathrm{s}})|}\}\end{aligned}$$

are distinct stable non-fixed open-loop eigenvalues of G. We may suppress the dependence of these collections of eigenvalues on some of their arguments when clear from context.

We note that one can adopt the notion of the multiset to discriminate between copies of a mode with multiplicity greater than one. This would have some conceptual advantages, but would unnecessarily complicate some definitions and proofs, and so we maintain the use of standard sets, while tracking the multiplicities of the modes which we will want to move (the unstable non-fixed modes). This is equally acceptable, provided that a fixed and a non-fixed mode do not have the same value, which would require the non-fixed modes to be defined as something other than the complement of those which are fixed, as above (and multiset complementation could handle this aspect nicely). Even that situation could not be problematic if we are considering the complex plane as being split into an acceptable and an unacceptable region, since such an overlap would either represent an acceptable situation, or one which is fatal anyway.

Denote the total (with multiplicities) number of unstable non-fixed modes of a plant G by

$$\nu(G) \triangleq \sum_{\alpha \in \tilde{\Lambda}_+(G,\mathcal{S},\mathcal{T}^{\mathrm{s}})} \mu(\alpha, G).$$

2.1.2 Centralized Results

In this section we review and establish results on controllability, observability, and fixed modes for centralized control of linear time-invariant systems. We begin with Kalman canonical form with the help of the following lemma:

Lemma 6. *For every FDLTI plant G, there exists a similarity transformation matrix T such that*

$$\begin{bmatrix} T & 0 \\ 0 & I \end{bmatrix} \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] \begin{bmatrix} T^{-1} & 0 \\ 0 & I \end{bmatrix} = \left[\begin{array}{cccc|c} \tilde{A}_{11} & 0 & \tilde{A}_{13} & 0 & \tilde{B}_1 \\ \tilde{A}_{21} & \tilde{A}_{22} & \tilde{A}_{23} & \tilde{A}_{24} & \tilde{B}_2 \\ 0 & 0 & \tilde{A}_{33} & 0 & 0 \\ 0 & 0 & \tilde{A}_{43} & \tilde{A}_{44} & 0 \\ \hline \tilde{C}_1 & 0 & \tilde{C}_2 & 0 & D \end{array}\right]. \tag{2.2}$$

In the above equation we have the following correspondence between eigenvalues of $\tilde{A}_{ii}$ and modes of G:

- $\mathrm{eig}\left(\tilde{A}_{11}\right)$: *controllable and observable modes of G,*
- $\mathrm{eig}\left(\tilde{A}_{22}\right)$: *controllable and unobservable modes of G,*
- $\mathrm{eig}\left(\tilde{A}_{33}\right)$: *uncontrollable and observable modes of G,*
- $\mathrm{eig}\left(\tilde{A}_{44}\right)$: *uncontrollable and unobservable modes of G.*

Proof. See, for example, [21]. □

In order to reduce some of the notation, we do not explicitly show the dependence of $\tilde{A}_{ij}, \tilde{B}_i, \tilde{C}_j$ on A, B, C, and T, but it should be kept in mind that wherever we use Lemma 6 on a system, the resulting $\tilde{(\cdot)}$ variables are function of that system's state-space matrices, and Kalman similarity transformation matrix T.

The following lemma is useful in connecting centralized fixed modes with the familiar notion of controllability and observability. It was shown for strictly proper plants in [22]; we establish the following generalization before proceeding.

Lemma 7. *Given a proper controllable and observable plant G_{co}, for almost any $D_K \in \mathcal{S}_{\text{c}} \cap \mathcal{T}^{\text{s}}$, we have that:*

$$\text{eig}\left(A_{\text{CL}}(G_{\text{co}}, D_K)\right) \cap \text{eig}\left(G_{\text{co}}\right) \;=\; \varnothing. \tag{2.3}$$

Proof. For a strictly proper plant refer to [22, Theorem 2]. Given the proper plant G_{co}, consider the strictly proper part of it, namely $G_{\text{co}} - D$. Then, by [22, Theorem 2] the set of static feedback gains $\tilde{D}_K$ for which $\text{eig}\left(A_{\text{CL}}(G_{\text{co}} - D, \tilde{D}_K)\right) \cap \text{eig}\left(G_{\text{co}} - D\right) \neq \varnothing$ constitute a finite union of hyperplanes in the ambient space, and hence almost any $\tilde{D}_K \in \mathcal{S}_{\text{c}} \cap \mathcal{T}^{\text{s}}$ moves the open-loop eigenvalues of $G_{\text{co}} - D$. If $(I + \tilde{D}_K D)$ is invertible, then by the change of variable $D_K = (I + \tilde{D}_K D)^{-1}\tilde{D}_K$, we have:

$$A_{\text{CL}}(G_{\text{co}} - D, \tilde{D}_K) \;=\; A_{\text{CL}}(G_{\text{co}}, D_K).$$

To complete the proof, we show that $(I + \tilde{D}_K D)$ is invertible for almost any $\tilde{D}_K$. This can be seen as $\det(I+\tilde{D}_K D)=0$ is a non-trivial polynomial in $\tilde{D}_K$ (choosing $\tilde{D}_K=0$ would yield non-zero determinant), and hence the set of $\tilde{D}_K$ for which $\det(I+\tilde{D}_K D)=0$ is a set with dimension less than the ambient space and has zero Lebesgue measure. □

Next we state the following result regarding fixed modes with respect to a centralized sparsity pattern $\mathcal{S}_{\mathrm{c}}$, which tells us that the fixed modes of a plant with respect to a centralized information structure are precisely its uncontrollable or unobservable modes.

Lemma 8. *For any FDLTI plant G,*

$$\Lambda\left(G,\mathcal{S}_{\mathrm{c}},\mathcal{T}^{\mathrm{s}}\right)=\bigcup_{i=2,3,4}\mathrm{eig}\left(\tilde{A}_{ii}\right),$$

where $\tilde{A}_{ii}$ are the blocks in the Kalman canonical decomposition of plant G, such that the fixed modes are the union of uncontrollable or unobservable modes of G.

Proof. Denote the controllable and observable part of G by $G_{\mathrm{co}}\triangleq\tilde{C}_1(sI-\tilde{A}_{11})^{-1}\tilde{B}_1+D$. We first establish that for any arbitrary $D_K\in\mathcal{S}_{\mathrm{c}}\cap\mathcal{T}^{\mathrm{s}}$ that is closed around G, we have:

$$\mathrm{eig}\left(A_{\mathrm{CL}}(G,D_K)\right)\ =\ \mathrm{eig}\left(A_{\mathrm{CL}}(G_{\mathrm{co}},D_K)\right)\cup\left(\bigcup_{i=2,3,4}\mathrm{eig}\left(\tilde{A}_{ii}\right)\right). \tag{2.4}$$

To see this, apply the similarity transformation T given in Lemma 6 on $A_{\mathrm{CL}}(G,D_K)$. Then $TA_{\mathrm{CL}}(G,D_K)T^{-1}$ would only differ in blocks $\tilde{A}_{11}$, $\tilde{A}_{21}$, $\tilde{A}_{13}$, and $\tilde{A}_{23}$ compared to the open-loop $\tilde{A}$ in (2.2). This leaves the structure of $\tilde{A}$ unchanged, and renders (2.4).

For any $D_K \in \mathcal{S}_\mathrm{c} \cap \mathcal{T}^\mathrm{s}$, and for $i = 2, 3, 4$, we then have:

$$\mathrm{eig}\left(\tilde{A}_{ii}\right) \subseteq \mathrm{eig}\left(TA_\mathrm{CL}(G, D_K)T^{-1}\right) = \mathrm{eig}\left(A_\mathrm{CL}(G, D_K)\right),$$

and so $\bigcup\limits_{i=2,3,4} \mathrm{eig}\left(\tilde{A}_{ii}\right) \subseteq \Lambda\left(G, \mathcal{S}_\mathrm{c}, \mathcal{T}^\mathrm{s}\right).$

For any remaining modes of G, i.e., $\lambda \in \mathrm{eig}\left(\tilde{A}_{11}\right)$, it follows from (2.4) and Lemma 7 that there exists a static controller $D_K \in \mathcal{S}_\mathrm{c} \cap \mathcal{T}^\mathrm{s}$ such that $\lambda \notin \mathrm{eig}\left(A_\mathrm{CL}(G, D_K)\right)$, and so $\lambda \notin \Lambda\left(G, \mathcal{S}_\mathrm{c}, \mathcal{T}^\mathrm{s}\right).$ □

Remark 9. *In view of Lemma 7 and 8, almost any randomly chosen* $D_K \in \mathcal{S}_\mathrm{c} \cap \mathcal{T}^\mathrm{s}$ *moves all the open-loop modes of* G*, except those of* $\Lambda\left(G, \mathcal{S}_\mathrm{c}, \mathcal{T}^\mathrm{s}\right)$*.*

We use our notation to restate the following result, which tells us that the fixed modes of a plant with centralized information structure are the same with respect to static or dynamic control.

Theorem 10. *Given an FDLTI plant* G*,*

$$\Lambda\left(G, \mathcal{S}_\mathrm{c}, \mathcal{T}^\mathrm{s}\right) = \Lambda\left(G, \mathcal{S}_\mathrm{c}, \mathcal{T}^\mathrm{d}\right).$$

Proof. The $\supseteq$ inclusion follows immediately since $\mathcal{T}^\mathrm{s} \subseteq \mathcal{T}^\mathrm{d}$.

We now need to show that $\Lambda\left(G, \mathcal{S}_\mathrm{c}, \mathcal{T}^\mathrm{s}\right) \subseteq \Lambda\left(G, \mathcal{S}_\mathrm{c}, \mathcal{T}^\mathrm{d}\right)$; using Lemma 8, we can achieve this by showing that $\bigcup\limits_{i=2,3,4} \mathrm{eig}\left(\tilde{A}_{ii}\right) \subseteq \Lambda\left(G, \mathcal{S}_\mathrm{c}, \mathcal{T}^\mathrm{d}\right)$, which can be achieved by showing that $\bigcup\limits_{i=2,3,4} \operatorname{eig}\left(\tilde{A}_{ii}\right) \subseteq \operatorname{eig}\left(A_{\mathrm{CL}}(G,K)\right)$ for arbitrary $K \in \mathcal{S}_{\mathrm{c}} \cap \mathcal{T}^{\mathrm{d}}$.

Given an arbitrary $K \in \mathcal{S}_{\mathrm{c}} \cap \mathcal{T}^{\mathrm{d}}$, and letting T be the similarity transformation matrix from Lemma 6, we can then apply (2.2) to (1.4) to get

$$\begin{aligned}
\begin{pmatrix} T & 0 \\ 0 & I \end{pmatrix} A_{\mathrm{CL}}(G,K) \begin{pmatrix} T^{-1} & 0 \\ 0 & I \end{pmatrix} &= \begin{pmatrix} \tilde{A} + \tilde{B} M D_K \tilde{C} & \tilde{B} M C_K \\ B_K N \tilde{C} & A_K + B_K D M C_K \end{pmatrix} \\
&= \left(\begin{array}{cccc:c} * & 0 & * & 0 & \tilde{B}_1 M C_K \\ * & \tilde{A}_{22} & * & * & \tilde{B}_2 M C_K \\ 0 & 0 & \tilde{A}_{33} & 0 & 0 \\ 0 & 0 & * & \tilde{A}_{44} & 0 \\ \hdashline B_K N \tilde{C}_1 & 0 & B_K N \tilde{C}_2 & 0 & * \end{array}\right)
\end{aligned}$$

where $(\tilde{A}, \tilde{B}, \tilde{C}, D)$ are as in (2.2).

If we apply another similarity transformation which swaps the first/second and third/fifth row and column blocks, the result is an upper block triangular matrix for which the eigenvalues clearly include those of $\tilde{A}_{22}$, $\tilde{A}_{33}$, and $\tilde{A}_{44}$, as desired. □

In the next two subsections, we generalize the result of [9] to arbitrary information structures, and provide a comprehensive proof. Section 2.1.3 establishes the invariance of fixed modes with respect to static and dynamic controllers, thereby demonstrating the necessity of having all of the fixed modes in the LHP for decentralized stabilizability, while Section 2.1.4 gives a constructive proof of existence of a stabilizing controller when all of the fixed modes of G are in the LHP, thereby demonstrating the sufficiency.

2.1.3 Invariance of fixed modes

We will show in this subsection that for any arbitrary sparsity pattern $\mathcal{S}$, the set of fixed modes with respect to static controllers is the same as the set of fixed modes with respect to dynamic controllers.

We first state a lemma which is obvious but will be helpful. This lemma states that if λ is a fixed mode of a system with respect to static controllers and sparsity pattern $\mathcal{S}$, then after closing the loop with an arbitrary matrix $D_K \in \mathcal{S} \cap \mathcal{T}^{\mathrm{s}}$, if we further allow only one of the static admissible elements of the controller to vary, then λ will remain a fixed mode. Given any matrix $D_K \in \mathcal{S} \cap \mathcal{T}^{\mathrm{s}}$ and any $(i,j) \in \mathrm{Adm}(\mathcal{S})$, define $G^{+}(D_K)$, as illustrated in Figure 2.1, as:

$$G^{+}(D_K) \triangleq \mathbf{e}_j^T \Gamma(G, D_K) \mathbf{e}_i = \left[\begin{array}{c|c} A_{G^{+}} & B_{G^{+}} \\ \hline C_{G^{+}} & D_{G^{+}} \end{array}\right],$$

where $A_{G^{+}} \triangleq A_{\mathrm{CL}}(G, D_K) = A + BMD_KC$, $B_{G^{+}} \triangleq BM\mathbf{e}_i$, $C_{G^{+}} \triangleq \mathbf{e}_j^T NC$, and $D_{G^{+}} \triangleq \mathbf{e}_j^T DM\mathbf{e}_i$. We note that this notation suppresses the dependence of G^{+} on the particular choice of the admissible index pair.

Lemma 11. *Given any matrix $D_K \in \mathcal{S} \cap \mathcal{T}^{\mathrm{s}}$, and any $(i,j) \in \mathrm{Adm}(\mathcal{S})$, if $\lambda \in \Lambda\left(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}}\right)$, then $\lambda \in \Lambda\left(G^{+}(D_K), \mathcal{S}_{\mathrm{c}}, \mathcal{T}^{\mathrm{s}}\right)$, i.e., $\Lambda\left(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}}\right) \subseteq \Lambda\left(G^{+}(D_K), \mathcal{S}_{\mathrm{c}}, \mathcal{T}^{\mathrm{s}}\right)$.*

Proof. Suppose that $\lambda \in \Lambda\left(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}}\right)$. For an arbitrary real scalar controller V, we have:

$$\begin{aligned} A_{\mathrm{CL}}(G^{+}(D_K), V) &= A_{\mathrm{CL}}(\mathbf{e}_j^T \Gamma(G, D_K)\mathbf{e}_i, V) \\ &= A_{\mathrm{CL}}(\Gamma(G, D_K), \mathbf{e}_i V \mathbf{e}_j^T) \\ &\overset{(1.5)}{=} A_{\mathrm{CL}}(G, D_K + \mathbf{e}_i V \mathbf{e}_j^T) \\ &= A_{\mathrm{CL}}(G, D_K^V), \end{aligned} \tag{2.5}$$

where we have defined $D_K^V \triangleq D_K + \mathbf{e}_i V \mathbf{e}_j^T$ as the static controller which is now effectively being closed around the plant. Since we clearly have $D_K^V \in \mathcal{S} \cap \mathcal{T}^{\mathrm{s}}$ and since $\lambda \in \Lambda\,(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}})$, it follows that $\lambda \in \operatorname{eig}\left(A_{\mathrm{CL}}(G^{+}(D_K), V)\right)$. Since V was arbitrary, we have $\lambda \in \Lambda\,(G^{+}(D_K), \mathcal{S}_{\mathrm{c}}, \mathcal{T}^{\mathrm{s}})$. □

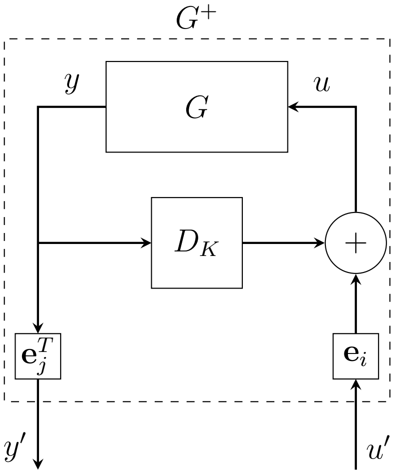


Figure 2.1: G^{+} is the SISO map from u' to y'.

Next, we relate fixed modes with respect to static controllers to those where only one of the admissible elements is allowed to be dynamic; that is, to "static plus one" controllers. The lemma will prove useful because closing such a scalar controller around the plant is equivalent to interconnecting a SISO dynamic controller with G^{+}, and we can then leverage our knowledge of centralized controllers. This result will be the foundation of the induction that we want to use later on. The outline of the proof is similar to that of [9, Proposition 1].

Theorem 12. *For any sparsity pattern $\mathcal{S}$, and any arbitrarily fixed indices $(i,j) \in \mathrm{Adm}(\mathcal{S})$:*

$$\Lambda\left(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}}\right) = \Lambda\left(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}+1}_{i,j}\right).$$

Proof. The $\supseteq$ inclusion follows immediately since $\mathcal{T}^{\mathrm{s}} \subseteq \mathcal{T}^{\mathrm{s}+1}_{i,j}$.

We now need to show that $\Lambda\left(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}}\right) \subseteq \Lambda\left(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}+1}_{i,j}\right)$. We have:

$$\begin{aligned}
\Lambda\left(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}}\right) &\overset{\text{Lem.11}}{\subseteq} \bigcap_{D_K \in \mathcal{S} \cap \mathcal{T}^{\mathrm{s}}} \Lambda\left(G^{+}(D_K), \mathcal{S}_{\mathrm{c}}, \mathcal{T}^{\mathrm{s}}\right) \\
&\overset{\text{Thm.10}}{=} \bigcap_{D_K \in \mathcal{S} \cap \mathcal{T}^{\mathrm{s}}} \Lambda\left(G^{+}(D_K), \mathcal{S}_{\mathrm{c}}, \mathcal{T}^{\mathrm{d}}\right) \\
&= \bigcap_{D_K \in \mathcal{S} \cap \mathcal{T}^{\mathrm{s}}} \ \bigcap_{k^{\mathrm{d}} \in \mathcal{T}^{\mathrm{d}}} \mathrm{eig}\left(A_{\mathrm{CL}}(G^{+}(D_K), k^{\mathrm{d}})\right) \\
&= \bigcap_{D_K \in \mathcal{S} \cap \mathcal{T}^{\mathrm{s}}} \ \bigcap_{k^{\mathrm{d}} \in \mathcal{T}^{\mathrm{d}}} \mathrm{eig}\left(\Gamma(\Gamma(G, D_K), \mathbf{e}_i k^{\mathrm{d}} \mathbf{e}_j^T)\right) \\
&\overset{(1.5)}{=} \bigcap_{D_K \in \mathcal{S} \cap \mathcal{T}^{\mathrm{s}}} \ \bigcap_{k^{\mathrm{d}} \in \mathcal{T}^{\mathrm{d}}} \mathrm{eig}\left(\Gamma(G, D_K + \mathbf{e}_i k^{\mathrm{d}} \mathbf{e}_j^T)\right) \\
&= \bigcap_{K^{\mathrm{s}+1} \in \mathcal{S} \cap \mathcal{T}^{\mathrm{s}+1}_{i,j}} \mathrm{eig}\left(\Gamma(G, K^{\mathrm{s}+1})\right) \\
&= \Lambda\left(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}+1}_{i,j}\right)
\end{aligned}$$

where the penultimate equality follows since $(\mathcal{S} \cap \mathcal{T}^{\mathrm{s}}) + \mathbf{e}_i \mathcal{T}^{\mathrm{d}} \mathbf{e}_j^T = \mathcal{S} \cap \mathcal{T}^{\mathrm{s}+1}_{i,j}$, and this completes the proof. □

We note that it was this result, showing that modes which are fixed with respect to static controllers are still fixed with respect to “static plus one” controllers, that was established for $\mathcal{S} = \mathcal{S}_{\mathrm{d}}$ in [9]. We will now show how to extend this result to show that modes which are fixed with respect to controllers with any given number of dynamic indices; that is, with respect to “static plus k” controllers, are still fixed when an additional index is allowed to become dynamic; that is, with respect to “static plus $k+1$” controllers. The main result of this subsection will indeed follow once that has been established.

We will proceed with the following definitions. Let $K^{(k)}(\sigma)$ be the controller after k steps, with k of its indices allowed to be dynamic, and define $\mathcal{I}^{(k)} \triangleq \{(i_1, j_1), \cdots, (i_k, j_k)\} \subset \mathrm{Adm}(\mathcal{S})$ as the set of such indices where $K^{(k)}(\sigma)$ is allowed to be dynamic, such that $K^{(k)} \in \mathcal{S} \cap \mathcal{T}^{\mathrm{s}+k}_{\mathcal{I}^{(k)}}$. Also let $(A^{(k)}_K, B^{(k)}_K, C^{(k)}_K, D^{(k)}_K)$ be a state-space representation for $K^{(k)}(\sigma)$.

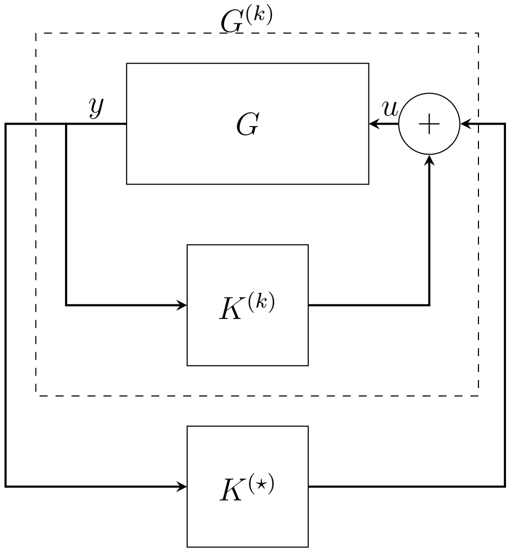


Figure 2.2: Plant $G^{(k)}$ and its respective controller $K^{(\star)}$.

Define $G^{(k)}(\sigma)$, illustrated in Figure 2.2, by closing $K^{(k)}(\sigma)$ around $G(\sigma)$ in such a way that the outputs of $G^{(k)}$ are the same as the outputs of G, and such that the inputs of $G^{(k)}$ are added to the outputs of $K^{(k)}$ and fed into G.

A state-space representation for $G^{(k)}(\sigma)$ is given by

$$G^{(k)} \triangleq \Gamma(G, K^{(k)}), \tag{2.6}$$

i.e., by replacing (A_K, B_K, C_K, D_K, M) with $({A_K}^{(k)}, {B_K}^{(k)}, {C_K}^{(k)}, {D_K}^{(k)}, M^{(k)})$ in (1.1) on page 17, where $M^{(k)} = (I - {D_K}^{(k)} D)^{-1}$.

We prove one remaining lemma before our main inductive step. This lemma relates the modes which are fixed when closing controllers with $k+1$ dynamic elements around the plant, to the modes which are fixed when first closing controllers with k dynamic elements around the plant, and then closing a controller with an additional dynamic element around the resulting plant, as in Figure 2.2. This will allow us to use our result relating static and "static plus one" controllers to make conclusions relating "static plus k" and "static plus $k+1$" controllers.

Remark 13. *We used the fact that given $(i,j) \in \mathrm{Adm}(\mathcal{S})$, we have $(\mathcal{S} \cap \mathcal{T}^{\mathrm{s}}) + \mathbf{e}_i \mathcal{T}^{\mathrm{d}} \mathbf{e}_j^T = \mathcal{S} \cap \mathcal{T}^{\mathrm{s}+1}_{i,j}$; that is, that adding static controllers and a dynamic element is equivalent to taking all of the "static plus one" controllers, at the end of the proof of Theorem 12. If this could be extended to state that*

$$(\mathcal{S} \cap \mathcal{T}^{\mathrm{s}+k}_{\mathcal{I}^{(k)}}) + \mathcal{T}^{\mathrm{s}+1}_{i,j} = \mathcal{S} \cap \mathcal{T}^{\mathrm{s}+k+1}_{\mathcal{I}^{(k)} \cup (i,j)}, \tag{2.7}$$

that is, that adding "static plus k" controllers and "static plus one" controllers *at* $(i,j) \in \mathrm{Adm}(\mathcal{S}) \setminus \mathcal{I}^{(k)}$ *is equivalent to taking all of the "static plus* $k+1$*" controllers, then Theorem 15 would follow similarly and easily, and the upcoming lemma would be trivial and unnecessary. It is not clear, however, that a "static plus* $k+1$*" controller can always be decomposed in that manner. We thus first introduce the following lemma, which states that, regardless of whether those two sets in (2.7) are the same, the modes which remain fixed as the controller varies over them are indeed identical.*

Lemma 14. *Given a set of indices* $\mathcal{I}^{(k)} \subset \mathrm{Adm}(\mathcal{S})$*, an additional index pair* $(i,j) \in \mathrm{Adm}(\mathcal{S}) \setminus \mathcal{I}^{(k)}$*, let* $\mathcal{I}^{(k+1)} \triangleq \mathcal{I}^{(k)} \cup (i,j)$*, and let* $G^{(k)}$ *be as in (2.6), then we have:*

$$\Lambda\left(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}+k+1}_{\mathcal{I}^{(k+1)}}\right) = \bigcap_{K^{(k)} \in \mathcal{S} \cap \mathcal{T}^{\mathrm{s}+k}_{\mathcal{I}^{(k)}}} \Lambda\left(G^{(k)}, \mathcal{S}, \mathcal{T}^{\mathrm{s}+1}_{i,j}\right). \tag{2.8}$$

Proof. For ease of notation, when the controllers are unambiguous such that we can suppress the dependency upon them, define $A^{\mathrm{LHS}}_{\mathrm{CL}} = A_{\mathrm{CL}}(G, K^{(k+1)})$ and $A^{\mathrm{RHS}}_{\mathrm{CL}} = A_{\mathrm{CL}}(G^{(k)}, K^{(\star)})$ to be the closed-loop dynamics matrices arising on each side of the equation for given controllers. Also let $\mathcal{K}_{\mathrm{LHS}} \triangleq \{K^{(k+1)} \mid K^{(k+1)} \in \mathcal{S} \cap \mathcal{T}^{\mathrm{s}+k+1}_{\mathcal{I}^{(k+1)}}\}$, and $\mathcal{K}_{\mathrm{RHS}} \triangleq \{(K^{(k)}, K^{(\star)}) \mid K^{(k)} \in \mathcal{S} \cap \mathcal{T}^{\mathrm{s}+k}_{\mathcal{I}^{(k)}},\ K^{(\star)} \in \mathcal{S} \cap \mathcal{T}^{\mathrm{s}+1}_{i,j}\}$ give the sets of controllers that must be considered on each side, such that the LHS can be abbreviated as $\bigcap_{\mathcal{K}_{\mathrm{LHS}}} \mathrm{eig}\left(A^{\mathrm{LHS}}_{\mathrm{CL}}\right)$, and the RHS can be abbreviated as $\bigcap_{\mathcal{K}_{\mathrm{RHS}}} \mathrm{eig}\left(A^{\mathrm{RHS}}_{\mathrm{CL}}\right)$.

First we prove the $\subseteq$ part by showing that for every admissible $K^{(\star)}$, i.e., $K^{(\star)} \in \mathcal{S} \cap \mathcal{T}^{\mathrm{s}+1}_{i,j}$, and admissible $K^{(k)}$ in RHS, there exist a $K^{(k+1)}$ in LHS such that $A^{\mathrm{RHS}}_{\mathrm{CL}} =$

$A_{\text{CL}}^{\text{LHS}}$. To see this observe that:

$$\Gamma(\Gamma(G, K^{(k)}), K^{(\star)}) \overset{(1.5)}{=} \Gamma(G, K^{(k)} + K^{(\star)}).$$

Thus we choose $K^{(k+1)} = K^{(k)} + K^{(\star)}$. This $K^{(k+1)}$ is admissible because it has only one further dynamic element at position $(i,j) \in \text{Adm}(\mathcal{S})$, and thus is in $\mathcal{T}_{\mathcal{I}^{(k+1)}}^{\text{s}+k+1}$. Hence for every admissible $(K^{(k)}, K^{(\star)})$, there exists an admissible $K^{(k+1)} \in \mathcal{K}_{\text{LHS}}$ constructed as above such that $A_{\text{CL}}^{\text{LHS}} = A_{\text{CL}}^{\text{RHS}}$, and so $\bigcap\limits_{\mathcal{K}_{\text{LHS}}} \text{eig}\left(A_{\text{CL}}^{\text{LHS}}\right) \subseteq \bigcap\limits_{\mathcal{K}_{\text{RHS}}} \text{eig}\left(A_{\text{CL}}^{\text{RHS}}\right)$.

We will prove the $\supseteq$ by contraposition, by showing that if $\lambda \notin \Lambda\left(G, \mathcal{S}, \mathcal{T}_{\mathcal{I}^{(k+1)}}^{\text{s}+k+1}\right)$, then $\lambda \notin \bigcap\limits_{K^{(k)} \in \mathcal{S} \cap \mathcal{T}_{\mathcal{I}^{(k)}}^{\text{s}+k}} \Lambda\left(G^{(k)}, \mathcal{S}, \mathcal{T}_{i,j}^{\text{s}+1}\right)$, for any $\lambda \in \mathbb{C}$. We can equivalently assume that:

$$\exists\ K^{(k+1)} \in \mathcal{K}_{\text{LHS}} \quad \text{s.t.} \quad \lambda \notin \text{eig}\left(A_{\text{CL}}(G, K^{(k+1)})\right), \tag{2.9}$$

and then show that:

$$\exists\ (K^{(k)}, K^{(\star)}) \in \mathcal{K}_{\text{RHS}} \quad \text{s.t.} \quad \lambda \notin \text{eig}\left(A_{\text{CL}}(G^{(k)}, K^{(\star)})\right). \tag{2.10}$$

Starting with $K^{(k+1)}$ from (2.9), we will show that we can then construct a $K^{(k)}$ and $K^{(\star)}$ to satisfy (2.10).

Based on $K^{(k+1)}$ in (2.9), we choose $\tilde{K}^{(\star)} \in \mathcal{S} \cap \mathcal{T}_{i,j}^{\text{s}+1}$ to be the strictly proper dynamic part of the last dynamic index by defining $\tilde{K}^{(\star)} = (\tilde{A_K}^{(\star)}, \tilde{B_K}^{(\star)}, \tilde{C_K}^{(\star)}, \tilde{D_K}^{(\star)})$

as:

$$\begin{aligned}
\tilde{A_K}^{(\star)} &= A_K^{(k+1)}, \\
\tilde{B_K}^{(\star)} &= B_K^{(k+1)}\mathbf{e}_j\mathbf{e}_j^T = \begin{bmatrix} 0 & \cdots & B_{K,j}^{(k+1)} & \cdots & 0 \end{bmatrix}, \\
\tilde{C_K}^{(\star)} &= \mathbf{e}_i\mathbf{e}_i^T C_K^{(k+1)} = \begin{bmatrix} 0 & \cdots & (C_{K,i}^{(k+1)})^T & \cdots & 0 \end{bmatrix}^T, \\
\tilde{D_K}^{(\star)} &= 0,
\end{aligned}$$

i.e., $\tilde{B_K}^{(\star)}$ is of the same dimension as $B_K^{(k+1)}$ with all its columns being zero except the j-th column, and $\tilde{C_K}^{(\star)}$ is of the same dimension as $C_K^{(k+1)}$ with all of its rows being zero except the i-th row. Then define $\tilde{K}^{(k)} \triangleq K^{(k+1)} - \tilde{K}^{(\star)}$, thus a (not necessarily minimal) state-space representation for $\tilde{K}^{(k)}$ is:

$$\begin{aligned}
\tilde{A_K}^{(k)} &= \mathrm{diag}(A_K^{(k+1)}, A_K^{(k+1)}), \\
\tilde{B_K}^{(k)} &= \begin{bmatrix} (B_K^{(k+1)})^T & (\tilde{B_K}^{(\star)})^T \end{bmatrix}^T, \\
\tilde{C_K}^{(k)} &= \begin{bmatrix} C_K^{(k+1)} & -\tilde{C_K}^{(\star)} \end{bmatrix}, \\
\tilde{D_K}^{(k)} &= D_K^{(k+1)}.
\end{aligned}$$

Construct $\tilde{G}^{(k)}$ in the same way as illustrated in Figure 2.2 by closing $\tilde{K}^{(k)}$ around G. Now if we use the following similarity transformation T on $A_{\mathrm{CL}}(\tilde{G}^{(k)}, \tilde{K}^{(\star)})$,

$$T = \begin{bmatrix} 0 & 0 & I & 0 \\ I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & -I & I \end{bmatrix},$$

then $TA_{\mathrm{CL}}(\tilde{G}^{(k)}, \tilde{K}^{(\star)})T^{-1}$ results in an upper block triangular matrix with blocks $A_K^{(k+1)}$, $A_{\mathrm{CL}}(G, K^{(k+1)})$, and $A_K^{(k+1)}$, indicating that:

$$\operatorname{eig}\left(A_{\mathrm{CL}}(\tilde{G}^{(k)}, \tilde{K}^{(\star)})\right) \;=\; \operatorname{eig}\left(A_{\mathrm{CL}}(G, K^{(k+1)})\right) \cup \operatorname{eig}\left(A_K^{(k+1)}\right). \tag{2.11}$$

Thus (2.10) can only be false if:

$$\lambda \in \operatorname{eig}\left(A_K^{(k+1)}\right). \tag{2.12}$$

We have shown that the only way to have an eigenvalue which is not on the LHS (when $K^{(k+1)}$ is closed around the plant) but which is on the RHS (when $\tilde{K}^{(\star)}$ and $\tilde{K}^{(k)}$ are then constructed as above), is if it comes from the dynamics matrix of $K^{(k+1)}$. We will now finish the proof by showing that if this is the case, we can make a small perturbation to $A_K^{(k+1)}$ such that it no longer has this eigenvalue, thus removing it from the RHS, while it is still not a closed-loop eigenvalue on the LHS.

Construct $\hat{K}^{(k+1)}$ by perturbing the A matrix of $K^{(k+1)}$; that is, $\hat{K}^{(k+1)}$ is defined by:

$$\begin{array}{rclrcl} \hat{A_K}^{(k+1)} & = & \tilde{A_K}^{(k+1)} + \epsilon I, & \hat{B_K}^{(k+1)} & = & \tilde{B_K}^{(k+1)}, \\ \hat{C_K}^{(k+1)} & = & \tilde{C_K}^{(k+1)}, & \hat{D_K}^{(k+1)} & = & \tilde{D_K}^{(k+1)}. \end{array}$$

For sufficiently small ϵ this yields

$$\lambda \notin \operatorname{eig}\left(\hat{A_K}^{(k+1)}\right). \tag{2.13}$$

Using the same steps as before to construct $\hat{K}^{(\star)}$ and $\hat{K}^{(k)}$ results in ϵI also being added to $\tilde{A_K}^{(\star)}$ and $\tilde{A_K}^{(k)}$. Then using the same similarity transformation T used to derive (2.11), we have

$$\operatorname{eig}\left(A_{\mathrm{CL}}(\hat{G}^{(k)},\hat{K}^{(\star)})\right) \;=\; \operatorname{eig}\left(A_{\mathrm{CL}}(G,\hat{K}^{(k+1)})\right)\cup\operatorname{eig}\left(\hat{A_K}^{(k+1)}\right), \tag{2.14}$$

where $\hat{G}^{(k)}$ is constructed by closing $\hat{K}^{(k)}$ around G, as illustrated for the unperturbed systems in Figure 2.2.

Since $A_{\mathrm{CL}}(G,K^{(k+1)})$ is continuous in the entries of $K^{(k+1)}$, and since the eigenvalues of a matrix are continuous in its entries (see, for example [23, Theorem 5.2. on p. 89]), it follows that by a sufficiently small perturbation made to $K^{(k+1)}$, along with (2.9), we still have $\lambda \notin \operatorname{eig}\left(A_{\mathrm{CL}}(G,\hat{K}^{(k+1)})\right)$. It then follows from (2.13) and (2.14) that $\lambda \notin \operatorname{eig}\left(A_{\mathrm{CL}}(\hat{G}^{(k)},\hat{K}^{(\star)})\right)$.

Thus we have been able to show that there exists a $(\hat{K}^{(k)},\hat{K}^{(\star)}) \in \mathcal{K}_{\mathrm{RHS}}$ such that $\lambda \notin \operatorname{eig}\left(A_{\mathrm{CL}}(\hat{G}^{(k)},\hat{K}^{(\star)})\right)$, which completes our contraposition argument. □

Now we are ready to prove the main inductive step: that given a certain number of controller indices which are allowed to be dynamic, and the associated set of fixed modes, allowing one additional index to become dynamic does not change the fixed modes.

Theorem 15. *Given an FDLTI plant G, a sparsity pattern $\mathcal{S}$, an admissible set of dynamic elements denoted by $\mathcal{I}^{(k)} \subset \mathrm{Adm}(\mathcal{S})$, an index pair $(i,j) \in \mathrm{Adm}(\mathcal{S}) \setminus \mathcal{I}^{(k)}$, and $\mathcal{I}^{(k+1)} = \mathcal{I}^{(k)} \cup (i,j)$, we have:*

$$\Lambda\left(G,\mathcal{S},\mathcal{T}^{\mathrm{s}+k}_{\mathcal{I}^{(k)}}\right) \;=\; \Lambda\left(G,\mathcal{S},\mathcal{T}^{\mathrm{s}+k+1}_{\mathcal{I}^{(k+1)}}\right).$$

Proof. Beginning with the quantity on the right-hand side, we get:

$$\begin{aligned}\Lambda\left(G,\mathcal{S},\mathcal{T}^{\mathrm{s}+k+1}_{\mathcal{I}^{(k+1)}}\right) &\overset{\text{Lem.14}}{=} \bigcap_{K^{(k)}\in\mathcal{S}\cap\mathcal{T}^{\mathrm{s}+k}_{\mathcal{I}^{(k)}}} \Lambda\left(G^{(k)},\mathcal{S},\mathcal{T}^{\mathrm{s}+1}_{i,j}\right) \\ &\overset{\text{Thm.12}}{=} \bigcap_{K^{(k)}\in\mathcal{S}\cap\mathcal{T}^{\mathrm{s}+k}_{\mathcal{I}^{(k)}}} \Lambda\left(G^{(k)},\mathcal{S},\mathcal{T}^{\mathrm{s}}\right) \\ &= \Lambda\left(G,\mathcal{S},\mathcal{T}^{\mathrm{s}+k}_{\mathcal{I}^{(k)}}\right),\end{aligned}$$

where the final equality follows since clearly $(\mathcal{S}\cap\mathcal{T}^{\mathrm{s}+k}_{\mathcal{I}^{(k)}})+(\mathcal{S}\cap\mathcal{T}^{\mathrm{s}})=\mathcal{S}\cap\mathcal{T}^{\mathrm{s}+k}_{\mathcal{I}^{(k)}}$, and this completes the proof. □

We can now state and easily prove the main result of this subsection. The following shows that for any FDLTI plant G, and any sparsity pattern $\mathcal{S}$, the set of fixed modes with respect to static and dynamic controllers are the same.

Theorem 16. *Given plant G, and sparsity constraint $\mathcal{S}$:*

$$\Lambda\left(G,\mathcal{S},\mathcal{T}^{\mathrm{s}}\right)=\Lambda\left(G,\mathcal{S},\mathcal{T}^{\mathrm{d}}\right). \tag{2.15}$$

Proof. This follows by induction from Theorem 15. □

2.1.4 Stabilization

The results from the previous subsection tell us that having all of the fixed modes of the original system in the LHP is necessary for stabilizability with respect to FDLTI controllers with the given structure. We now address the sufficiency of the condition. With a constructive proof, we will show that we can stabilize a plant

G with arbitrary information structure $\mathcal{S}$, as long as it has no unstable fixed modes. We will achieve this by showing that we can always find a controller which will reduce the number of unstable modes, while leaving all of the fixed modes in the LHP, which can then be applied as many times as required.

We will first state the following lemma from [9], which gives some properties regarding continuity and topology of non-fixed modes with respect to static controllers. It tells us that we can keep the modes within a given distance of the original ones by closing a small enough matrix D around the plant, and that an arbitrarily small D can move all of the non-fixed modes.

Lemma 17. *For any plant G, and any sparsity pattern $\mathcal{S}$, partition the open-loop eigenvalues of G as in (2.1). Then we have:*

1. *For all $\epsilon > 0$, there exist $\gamma > 0$ such that for all $D \in \mathcal{S} \cap \mathcal{T}^{\mathrm{s}}$ with $\|D\|_\infty < \gamma$, there are exactly $\mu(\lambda, G)$ eigenvalues (counting multiplicities) of $A_{\mathrm{CL}}(G, D)$ in $B(\lambda, \epsilon)$, for all $\lambda \in \tilde{\Lambda}(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}})$.*

2. *For all $\gamma > 0$, and for almost any $D \in \mathcal{S} \cap \mathcal{T}^{\mathrm{s}}$ with $\|D\|_\infty < \gamma$, we have that $\lambda \notin \mathrm{eig}\,(A_{\mathrm{CL}}(G, D))$, for all $\lambda \in \tilde{\Lambda}(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}})$.*

Proof. See Lemma 4 in [9]. The proof was developed for strictly proper plants with diagonal information structure. However, it does not use any property specific to only block-diagonal information structure and thus could be replaced by any arbitrary information structure. To generalize it for the proper plants, a similar change of variable technique as in proof of Lemma 7 can be used, which would add an invertibility constraint that would hold for almost any linear static controller. □

Remark 18. *It follows from the proof that the set of D which violate part 2 of Lemma 17, forms a subset with zero Lebesgue measure, and thus a random $D \in \mathcal{S}$ that is sufficiently small satisfies all of the conditions of Lemma 17. Precisely, the space of static controllers that does not move the non-fixed modes is constructed by a finite union of hyper-surfaces in $(\mathcal{S} \cap \mathcal{T}^{\mathrm{s}}) \subset \mathbb{R}^{n_u \times n_y}$. Thus, a D that satisfies all of the conditions of Lemma 17, can be found with probability one by randomly choosing the direction of $D \in \mathcal{S} \cap \mathcal{T}^{\mathrm{s}}$, and then scaling it appropriately such that $\|D\|_\infty < \gamma$.*

We now establish the following theorem, which shows how a given non-fixed mode can be extracted as a controllable and observable mode of a specific SISO system, as illustrated in Figure 2.3.

Theorem 19. *For any plant G with $|\tilde{\Lambda}_+(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}})| \geq 1$, and all fixed modes in the LHP (i.e., $\Lambda(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}}) \subset \mathbb{C}^-$), there exists a $D_K \in \mathcal{S} \cap \mathcal{T}^{\mathrm{s}}$, and an integer $m \in \{1, \cdots, a\}$, such that for the SISO system G_m defined as:*

$$G_m = \mathbf{e}_{j_m}^T \Gamma(G, D_K) \mathbf{e}_{i_m} = \left[\begin{array}{c|c} A_m & B_m \\ \hline C_m & D_m \end{array}\right] \triangleq \left[\begin{array}{c|c} A + BMD_KC & BM\mathbf{e}_{i_m} \\ \hline \mathbf{e}_{j_m}^T NC & \mathbf{e}_{j_m}^T DM\mathbf{e}_{i_m} \end{array}\right], \tag{2.16}$$

where $(i_m, j_m) \in \mathrm{Adm}(\mathcal{S})$ is the m^{th} tuple in (1.6) on page 20, the following hold:

1. *There exists $\alpha \in \tilde{\Lambda}_+(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}})$, such that α is a controllable and observable mode of G_m;*

2. *The total number of unstable modes of G_m is no greater than that of G, i.e., $\nu(G_m) \leq \nu(G)$, where $\nu(G)$ is defined on page 30.*

Proof. The outline of the proof is as follows. To prove the first argument, we find a $D \in \mathcal{S} \cap \mathcal{T}^{\mathrm{s}}$ that when closed around G, moves all of its non-fixed modes, and identify the index $m \in \{1, \cdots, a\}$ for which $D|_{(m)}$ (defined in (1.7) on page 20) is the first in the sequence to alter all of them. This means that only changing the $(i_m, j_m)^{\mathrm{th}}$ element of the static controller will change unstable mode(s) of the closed-loop, and thus those modes must be in the controllable and observable modes of the SISO plant from u_{i_m} to y_{j_m}.

Proof of argument 1: Since $\tilde{\Lambda}_+(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}}) \subseteq \tilde{\Lambda}(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}})$, Lemma 17 guarantees that we can take the static gain $D \in \mathcal{S} \cap \mathcal{T}^{\mathrm{s}}$ such that when closed around G, would move all of its unstable non-fixed modes. It also asserts that by choosing this D small enough, the closed loop $A_{\mathrm{CL}}(G, D)$ would have no more unstable modes than G itself.

Construct a sequence of matrices $D|_{(m)} \in \mathcal{S} \cap \mathcal{T}^{\mathrm{s}}$ as in (1.7), so that $D|_{(a)} = D$ and $D|_{(0)} = 0$, thus:

$$\begin{aligned} \forall\ \alpha \in \tilde{\Lambda}_+(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}}) : &\qquad \alpha \notin \operatorname{eig}\left(A_{\mathrm{CL}}(G, D|_{(a)})\right), \\ \forall\ \alpha \in \tilde{\Lambda}_+(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}}) : &\qquad \alpha \in \operatorname{eig}\left(A_{\mathrm{CL}}(G, D|_{(0)})\right). \end{aligned}$$

By decreasing m from a to 1, there must exist a value of $m \in \{1, \cdots, a\}$, such that:

$$\forall\ \alpha \in \tilde{\Lambda}_+(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}}) : \qquad \alpha \notin \operatorname{eig}\left(A_{\mathrm{CL}}(G, D|_{(m)})\right), \tag{2.17a}$$

$$\exists\ \alpha \in \tilde{\Lambda}_+(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}}) : \qquad \alpha \in \operatorname{eig}\left(A_{\mathrm{CL}}(G, D|_{(m-1)})\right); \tag{2.17b}$$

that is, m is the first index for which all of the unstable non-fixed modes have been moved. If we then set $D_K = D|_{(m-1)}$ and use the definitions from (2.16), as illustrated in Figure 2.3, similar to (2.5) we have:

$$\begin{aligned} A_{\text{CL}}(G, D|_{(m)}) &= A_{\text{CL}}(G, D|_{(m-1)} + \mathbf{e}_{i_m} D_{i_m,j_m} \mathbf{e}_{j_m}^T) \\ &\overset{(1.5)}{=} A_{\text{CL}}(\Gamma(G, D|_{(m-1)}), \mathbf{e}_{i_m} D_{i_m,j_m} \mathbf{e}_{j_m}^T) \\ &= A_{\text{CL}}(\mathbf{e}_{j_m}^T \Gamma(G, D|_{(m-1)}) \mathbf{e}_{i_m}, D_{i_m,j_m}) \\ &= A_{\text{CL}}(G_m, D_{i_m,j_m}). \end{aligned} \tag{2.18}$$

From (2.17b), there exists at least one $\alpha \in \tilde{\Lambda}_+(G, \mathcal{S}, \mathcal{T}^{\text{s}})$ such that:

$$\alpha \;\in\; \text{eig}\left(A_{\text{CL}}(G, D|_{(m-1)})\right) \;=\; \text{eig}\left(A_m\right),$$

but due to (2.17a),

$$\alpha \;\notin\; \text{eig}\left(A_{\text{CL}}(G, D|_{(m)})\right) \;\overset{(2.18)}{=}\; \text{eig}\left(A_{\text{CL}}(G_m, D_{i_m,j_m})\right).$$

For all such α that are thus moved by only closing D_{i_m,j_m} around the SISO system G_m (for which the only information structure is the centralized one, $\mathcal{S}_{\text{c}}$), we have:

$$\begin{aligned} \exists\;\; D_{i_m,j_m} \in \mathbb{R} \text{ s.t. : } \alpha \notin \text{eig}\left(A_{\text{CL}}(G_m, D_{i_m,j_m})\right) \\ \Rightarrow\;\; \alpha \notin \Lambda\left(G_m, \mathcal{S}_{\text{c}}, \mathcal{T}^{\text{s}}\right). \end{aligned}$$

Finally, due to Lemma 8, the fixed modes of any FDLTI plant with centralized information structure are equal to its unobservable or uncontrollable modes, we must have that those α are controllable and observable modes of G_m.

Proof of argument 2: since $A_m = A_{\text{CL}}(G, D|_{(m-1)})$, we need to show that this $D|_{(m-1)}$ satisfies Lemma 17.1 when we take the ϵ-balls in Lemma 17 small enough such that they do not intersect with $\bar{\mathbb{C}}^+$. However this is the case since the given D in part 1 of the proof satisfies Lemma 17, and $D|_{(m)}$ that are constructed from this D, satisfy $\|D|_{(m)}\|_\infty \leq \|D\|_\infty \leq \gamma$ for any $m \in \{0, 1, \cdots, a\}$ based on the definition. $\square$

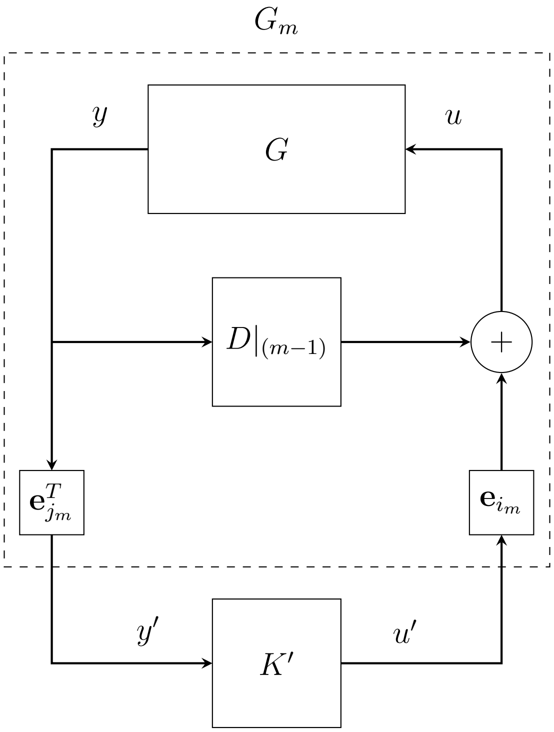


Figure 2.3: G_m is the SISO map from u' to y', and K_m is the map from y to u, giving the total control for the original plant.

In the following proposition, we will use observer-based pole placement for a centralized information structure to show how one can stabilize unstable, non-fixed modes of G_m in (2.16). We will add one further design constraint that the unstable modes of the controller would be different than that of G_m, and will show that this constraint is always achievable by a small perturbation of the gains. This ensures that an induction-based argument can be used later on. This constraint is not mentioned in [9], and it is unclear that without such a constraint how one can guarantee that a rigorous induction could follow, even for a diagonal information structure.

Proposition 20. *All of the controllable and observable unstable modes of the plant G_m can be stabilized by an observer-based controller K' such that:*

$$\text{eig}_{+,0}(K') \cap \text{eig}_{+,0}(\Gamma(G_m, K')) \;=\; \varnothing. \tag{2.19}$$

Proof. Our proof is in a constructive manner, we will first find a K' to only stabilize the controllable and observable modes of G_m without considering (2.19). We will then show that (2.19) is not satisfied only on a set with zero measure, and thus almost any small perturbation in the specific elements of K' will satisfy (2.19).

First find a similarity transformation T that will put G_m in its Kalman canonical form, therefore we would have:

$$\begin{bmatrix} T & 0 \\ 0 & I \end{bmatrix} \left[\begin{array}{c|c} A_m & B_m \\ \hline C_m & D_m \end{array}\right] \begin{bmatrix} T^{-1} & 0 \\ 0 & I \end{bmatrix} = \left[\begin{array}{cccc|c} \tilde{A}^m_{11} & 0 & \tilde{A}^m_{13} & 0 & \tilde{B}^m_1 \\ \tilde{A}^m_{21} & \tilde{A}^m_{22} & \tilde{A}^m_{23} & \tilde{A}^m_{24} & \tilde{B}^m_2 \\ 0 & 0 & \tilde{A}^m_{33} & 0 & 0 \\ 0 & 0 & \tilde{A}^m_{43} & \tilde{A}^m_{44} & 0 \\ \hline \tilde{C}^m_1 & 0 & \tilde{C}^m_2 & 0 & D_m \end{array}\right], \tag{2.20}$$

where as before, all the $\tilde{(\cdot)}$ parameters depend on the transformation matrix T and the state-space representation of G_m. We want to stabilize all the unstable modes in $\tilde{A}_{11}$. Since based on definition $(\tilde{A}_{11}, \tilde{B}_1)$ is a controllable pair and $(\tilde{A}_{11}, \tilde{C}_1)$ is an observable pair, there exists a state feedback gain F and an observer gain L, such that eigenvalues of $\tilde{A}_{11} - \tilde{B}_1 F$ and $\tilde{A}_{11} - L\tilde{C}_1$ can be arbitrary assigned, and hence can be stabilized. We will now show that the following controller will stabilize all the unstable modes of $\tilde{A}_{11}$. Take the controller as:

$$K' = \left[\begin{array}{c|c} A' & B' \\ \hline C' & 0 \end{array}\right] = \left[\begin{array}{c|c} \tilde{A}_{11} - \tilde{B}_1 F - L\tilde{C}_1 + LD_m F & L \\ \hline -F & 0 \end{array}\right];$$

apply T from (2.20) on G_m and close K' around it, then the closed-loop $A_{\mathrm{CL}}(G_m, K')$ would be:

$$\begin{pmatrix} \tilde{A}_{11} & 0 & \tilde{A}_{13} & 0 & -\tilde{B}_1 F \\ \tilde{A}_{21} & \tilde{A}_{22} & \tilde{A}_{23} & \tilde{A}_{24} & -\tilde{B}_2 F \\ 0 & 0 & \tilde{A}_{33} & 0 & 0 \\ 0 & 0 & \tilde{A}_{43} & \tilde{A}_{44} & 0 \\ L\tilde{C}_1 & 0 & L\tilde{C}_2 & 0 & \tilde{A}_{11} - \tilde{B}_1 F - L\tilde{C}_1 \end{pmatrix}$$

Apply another similarity transformation T_1, which keeps the first four rows the same and subtract the first row from the fifth, then we have:

$$\operatorname{eig}\left(A_{\mathrm{CL}}(G_m, K')\right) = \operatorname{eig}\left(T_1 A_{\mathrm{CL}}(G_m, K') T_1^{-1}\right) =$$

$$\operatorname{eig}\left(\begin{array}{cc:ccc} \tilde{A}_{11} - \tilde{B}_1 F & 0 & \tilde{A}_{13} & 0 & -\tilde{B}_1 F \\ \tilde{A}_{21} - \tilde{B}_2 F & \tilde{A}_{22} & \tilde{A}_{23} & \tilde{A}_{24} & -\tilde{B}_2 F \\ \hdashline 0 & 0 & \tilde{A}_{33} & 0 & 0 \\ 0 & 0 & \tilde{A}_{43} & \tilde{A}_{44} & 0 \\ 0 & 0 & L\tilde{C}_2 - \tilde{A}_{13} & 0 & \tilde{A}_{11} - L\tilde{C}_1 \end{array}\right)$$

Thus the eigenvalue of the closed loop would be

$$\operatorname{eig}\left(A_{\mathrm{CL}}(G_m, K')\right) \;=\; \operatorname{eig}\left(\tilde{A}_{11} - \tilde{B}_1 F\right) \cup \operatorname{eig}\left(\tilde{A}_{11} - L\tilde{C}_1\right) \cup \left(\bigcup_{i=2}^{4} \operatorname{eig}\left(\tilde{A}_{ii}\right)\right)$$

Therefore for all observer-based controllers that naturally satisfy $\operatorname{eig}\left(\tilde{A}_{11} - \tilde{B}_1 F\right) \in \mathbb{C}^-$ and $\operatorname{eig}\left(\tilde{A}_{11} - L\tilde{C}_1\right) \in \mathbb{C}^-$; unstable modes of $\Gamma(G_m, K')$ would be independent of F and L, i.e.:

$$\operatorname{eig}_{+,0}((\Gamma(G_m, K')) \;=\; \bigcup_{i=2}^{4} \operatorname{eig}_{+,0}(\tilde{A}_{ii}), \tag{2.21}$$

and all unstable modes in $\tilde{A}_{11}$ can be stabilized by appropriate choice of matrices F and L.

We will now show that (2.19) is not met on a set with zero measure in the ambient space of L. Replacing (2.21) in (2.19) yield that constraint (2.19) is met if and only if:

$$\operatorname{eig}_{+,0}(K') \bigcap \left(\bigcup_{i=2}^{4} \operatorname{eig}_{+,0}(\tilde{A}_{ii})\right) \;=\; \varnothing, \tag{2.22}$$

and if not, we enforce (2.19) by appropriately perturbing the L matrix. Construct the perturbed controller $\hat{K}'$ by replacing L in K' with $\hat{L} = L + L_\epsilon$, i.e.:

$$\hat{K}' \triangleq \left[\begin{array}{c|c} \hat{A}' & \hat{L} \\ \hline -F & 0 \end{array}\right],$$

with $\hat{A}' \triangleq \tilde{A}_{11} - \tilde{B}_1 F - \hat{L}\tilde{C}_1 + \hat{L}D_m F$. We want to show that $\hat{K}'$ satisfies (2.22) for almost any L_ϵ. To see this, first define W as:

$$W \triangleq \left[\begin{array}{c|c} A' & I \\ \hline -\tilde{C}_1 + D_m F & 0 \end{array}\right].$$

It is also straightforward to verify that $A_{\mathrm{CL}}(W, L_\epsilon) = \hat{A}'$. We want to apply Remark 9 on W to show that almost any perturbation L_ϵ moves all the unstable open-loop modes of W (which is equivalent to the unstable modes of K' as $\operatorname{eig}(W) = \operatorname{eig}(K')$). This would be achieved by showing that non of the unstable modes of W would be a fixed one, precisely:

$$\begin{aligned} A_{\mathrm{CL}}(W, -L) &= \tilde{A}_{11} - \tilde{B}_1 F \\ \Rightarrow \Lambda\left(W, \mathcal{S}_{\mathrm{c}}, \mathcal{T}^{\mathrm{s}}\right) &\subseteq \operatorname{eig}\left(\tilde{A}_{11} - \tilde{B}_1 F\right) \subset \mathbb{C}^-, \end{aligned}$$

as F is chosen to stabilize $\tilde{A}_{11}$. Moreover, given that $\operatorname{eig}\left(\tilde{A}_{11} - L\tilde{C}_1\right) \subset \mathbb{C}^-$, if we chose L_ϵ sufficiently small, then due to a continuity argument we have $\operatorname{eig}\left(\tilde{A}_{11} - \hat{L}\tilde{C}_1\right) \subset \mathbb{C}^-$. Thus any sufficiently small perturbation L_ϵ will make $\hat{K}'$ satisfy (2.19) while still keeping $\tilde{A}_{11} - \hat{L}\tilde{C}_1$ stable. □

We now encapsulate the desired properties of the intermediate controller at each step that partially stabilizes the plant in the following corollary, which combines Theorem 19 and Proposition 20.

Corollary 21. *For every plant G that satisfies the assumptions of Theorem 19, there exists an $m \in \{1, \cdots, a\}$ and a controller $K_m \in \mathcal{S} \cap \mathcal{T}^{\mathrm{s}+1}_{i_m, j_m}$ such that:*

$$\begin{aligned} \nu(\Gamma(G, K_m)) \ &\leq \ \nu(G) \ - 1, && (2.23) \\ \mathrm{eig}_{+,0}(K_m) \cap \tilde{\Lambda}_+(\Gamma(G, K_m)) \ &= \ \varnothing, && (2.24) \end{aligned}$$

where $(i_m, j_m) \in \mathrm{Adm}(\mathcal{S})$ is the m^{th} tuple in (1.6) on page 20,

Proof. Use Theorem 19 to find D_K and m, use Proposition 20 to find K', and construct the MIMO controller $K_m \triangleq D|_{(m-1)} + \mathbf{e}_{i_m} K' \mathbf{e}^T_{j_m}$. As illustrated in Figure 2.3, this K_m has the following state-space representation:

$$K_m = \left[\begin{array}{c|c} A^K_m & B^K_m \\ \hline C^K_m & D^K_m \end{array}\right] = \left[\begin{array}{c|c} A' & B'\mathbf{e}^T_{j_m} \\ \hline \mathbf{e}_{i_m} C' & D_K \end{array}\right], \qquad (2.25)$$

and clearly satisfies:

$$A_{\mathrm{CL}}(G_m, K') = A_{\mathrm{CL}}(G, K_m). \qquad (2.26)$$

Due to Theorem 19 and Proposition 20, K' will stabilize at least one unstable mode of G, hence we have $\nu(\Gamma(G_m, K')) \leq \nu(G) - 1$, and thus (2.23) would be an immediate result of this property of K' combined with (2.26). Finally, (2.24) follows from (2.19) as $A^K_m = A'$ and $\tilde{\Lambda}_+(\Gamma(G_m, K')) = \tilde{\Lambda}_+(\Gamma(G, K_m))$, due to (2.25) and (2.26). □

We use induction to prove that if all the fixed modes of G are in LHP, then we can stabilize G by dynamic controller. We will first define the following interconnection that will be useful in the induction. Let $\check{G}^{(0)} \triangleq G$ and at each step k, denote the transfer function from u to y, as illustrated in Figure 2.4, by $\check{G}^{(k+1)}$, i.e., $\check{G}^{(k+1)} = \Gamma(\check{G}^{(k)}, K_m^{(k)})$. Let $(A_{\check{G}}^{(k)}, B_{\check{G}}^{(k)}, C_{\check{G}}^{(k)}, D_{\check{G}}^{(k)})$ be a state-space representation for $\check{G}^{(k)}$, also denote the total number of unstable modes of $\check{G}^{(k)}$ by $\nu^{(k)}$:

$$\nu^{(k)} \triangleq \sum_{\alpha \in \tilde{\Lambda}_+(\check{G}^{(k)})} \mu(\alpha, \check{G}^{(k)}).$$

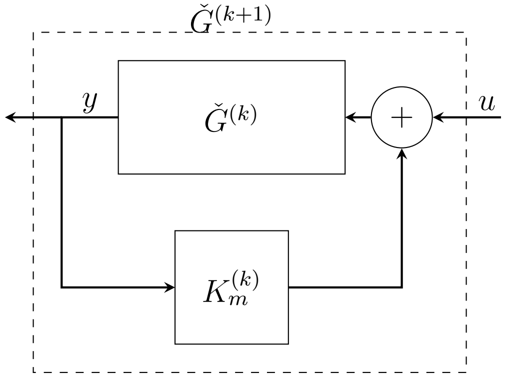


Figure 2.4: Plant $\check{G}^{(k+1)} \triangleq \Gamma(\check{G}^{(k)}, K_m^{(k)})$.

The induction will be in such a way that in each step k, we will find an integer $m^{(k)} \in \{1, \cdots, a\}$, and a $K_m^{(k)} \in \mathcal{S} \cap \mathcal{T}^{s+1}_{i_{m^{(k)}}, j_{m^{(k)}}}$ that when closed around $\check{G}^{(k)}$, will stabilize at least one unstable mode of $\check{G}^{(k)}$, thus $\nu^{(k+1)} \leq \nu^{(k)} - 1$. Then we will treat the corresponding $\check{G}^{(k+1)}$ as the new plant for which we want to stabilize the rest of remaining $\nu^{(k+1)}$ unstable eigenvalues, thus in at most $\nu^{(0)}$ steps, G will be stabilized. A crucial part of induction is that $\check{G}^{(k+1)}$ must have no fixed mode in closed RHP. This is not addressed in [9]. We will formalize this fact with the help of following lemma. It is enough to show that closing K_m around G does not add any unstable fixed modes to $\Gamma(G, K_m)$.

Lemma 22. *Assume that all the fixed modes of G are in LHP, i.e.:*

$$\Lambda\left(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}}\right) \subset \mathbb{C}^{-}, \tag{2.27}$$

and let $K_m \in \mathcal{S} \cap \mathcal{T}^{\mathrm{s}+1}_{i_m, j_m}$ satisfy (2.24). Then we have:

$$\Lambda\left(\Gamma(G, K_m), \mathcal{S}, \mathcal{T}^{\mathrm{s}}\right) \subset \mathbb{C}^{-}.$$

Proof. Proof is done by contradiction, we will first create the following set-up to state the idea. Let (A_K, B_K, C_K, D_K) be a minimal state-space representation for K_m. We have:

$$\Lambda\left(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}}\right) \subseteq \Lambda\left(\Gamma(G, K_m), \mathcal{S}, \mathcal{T}^{\mathrm{s}}\right),$$

since the RHS is the set of fixed modes with respect to controllers in the form $K_m + \mathcal{S} \cap \mathcal{T}^{\mathrm{s}}$, whereas the LHS equals $\Lambda\left(G, \mathcal{S}, \mathcal{T}^{\mathrm{d}}\right)$ (by Theorem 16), that is the set of fixed modes with respect to controllers in $\mathcal{S} \cap \mathcal{T}^{\mathrm{d}}$, which is a bigger set than $K_m \in \mathcal{S} \cap \mathcal{T}^{\mathrm{d}}$. Next, it is trivial to check that if we close $-K_m$ around $\Gamma(G, K_m)$, then by applying a similarity transformation T_2, a state-space realization that does not omit non-minimal modes of $\Gamma(\Gamma(G, K_m), -K_m)$ can be written as:

$$\begin{bmatrix} T_2 & 0 \\ 0 & I \end{bmatrix} \Gamma(\Gamma(G, K_m), -K_m) \begin{bmatrix} T_2^{-1} & 0 \\ 0 & I \end{bmatrix} = \left[\begin{array}{cc:c|c} A & BC_K & 0 & B \\ 0 & A_K & 0 & 0 \\ \hdashline B_K C & B_K DC_K & A_K & B_K D \\ \hline C & DC_K & 0 & D \end{array}\right], \tag{2.28}$$

thus we have

$$\operatorname{eig}\left(\Gamma(\Gamma(G, K_m), -K_m)\right) = \operatorname{eig}\left(A\right) \cup \operatorname{eig}\left(A_K\right).$$

Furthermore, due to (2.27), there exist a $D \in \mathcal{S} \cap \mathcal{T}^{\mathrm{s}}$ that will move all the unstable modes of A. If we apply the same D on (2.28), due to the block-diagonal structure we have $\operatorname{eig}\left(A_{\mathrm{CL}}(\Gamma(\Gamma(G, K_m), -K_m), D)\right) = \operatorname{eig}\left(A_{\mathrm{CL}}(G, D)\right) \cup \operatorname{eig}\left(A_K\right)$, which yields:

$$\Lambda\left(\Gamma(G, K_m), \mathcal{S}, \mathcal{T}^{\mathrm{s}}\right) \subseteq \Lambda\left(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}}\right) \cup \operatorname{eig}\left(A_K\right). \tag{2.29}$$

Now we are ready to do the main contradiction part, assume that there exist an $\alpha \in \Lambda\left(\Gamma(G, K_m), \mathcal{S}, \mathcal{T}^{\mathrm{s}}\right)$, with $\Re(\alpha) \geq 0$, then

$$\begin{aligned} & \alpha \in \Lambda\left(\Gamma(G, K_m), \mathcal{S}, \mathcal{T}^{\mathrm{s}}\right), \qquad \Re(\alpha) \geq 0 \\ & \alpha \overset{(2.29)}{\in} \Lambda\left(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}}\right) \cup \operatorname{eig}\left(A_K\right) \\ \overset{(2.27)}{\Rightarrow} \; & \alpha \in \operatorname{eig}\left(A_K\right) \\ \overset{(2.24)}{\Rightarrow} \; & \alpha \notin \operatorname{eig}\left(\Gamma(G, K_m)\right) \\ \Rightarrow \; & \alpha \notin \Lambda\left(\Gamma(G, K_m), \mathcal{S}, \mathcal{T}^{\mathrm{s}}\right) \end{aligned}$$

thus we have achieved the desired contradiction. □

Constraint (2.24) in Corollary 21 ensures that the unstable modes are non-overlapping, and is one sufficient condition to prove Lemma 22. When this condition is not met for an initial choice of the feedback/observer gain, one way to always make it feasible is by adding the perturbation L_ϵ to the observer gain. This in turn might prevent exact pole placement, but, one can place the poles arbitrarily close to the desired locations by choosing L_ϵ sufficiently small.

Now we are ready to claim that if all the fixed modes of G are in the LHP, then we can stabilize G by a dynamic controller. This stabilizing controller would be a summation of individual controllers $K_m^{(k)}$, each obtained in one step of the induction, where in each step k, $K_m^{(k)}$ would only have one dynamic element (i.e., $K_m^{(k)} \in \mathcal{S} \cap \mathcal{T}^{\mathrm{s}+1}_{i_{m^{(k)}}, j_{m^{(k)}}}$, for some $m^{(k)} \in \{1, \cdots, a\}$).

Theorem 23. *For any FDLTI plant G, and any sparsity pattern $\mathcal{S}$,*

if $\Lambda\left(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}}\right) \subset \mathbb{C}^-$, then there exist a controller $K \in \mathcal{S} \cap \mathcal{T}^{\mathrm{d}}$ that will stabilize G.

Proof. Proof is done by induction. Take $k \leftarrow 0$ and let $\check{G}^{(0)} \triangleq G$. As per assumption of this theorem, $\Lambda\left(\check{G}^{(0)}, \mathcal{S}, \mathcal{T}^{\mathrm{s}}\right) = \Lambda\left(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}}\right) \subset \mathbb{C}^-$. At each induction step k, we would stabilize at least one of the unstable modes of $\check{G}^{(k)}$ by Corollary 21. Specifically, with G replaced by $\check{G}^{(k)}$ in Corollary 21, we can find a $m^{(k)} \in \{1, \cdots, a\}$, and a controller $K_m^{(k)} \in \mathcal{S} \cap \mathcal{T}^{\mathrm{s}+1}_{i_{m^{(0)}}, j_{m^{(0)}}}$, that will stabilize at least one of unstable modes of $\check{G}^{(k)}$. This $K_m^{(k)}$ satisfies (2.24) (with G replaced by $\check{G}^{(k)}$), and thus by Lemma 22, $\check{G}^{(k+1)} = \Gamma(\check{G}^{(k)}, K_m^{(k)})$, would have all of its fixed modes in LHP, i.e., $\Lambda\left(\check{G}^{(k+1)}, \mathcal{S}, \mathcal{T}^{\mathrm{s}}\right) \in \mathbb{C}^-$. This guarantees that we can proceed with the induction by taking $k \leftarrow k+1$, as long as $\check{G}^{(k)}$ has any remaining unstable mode. Since at each step at least one unstable mode is stabilized, G would be stabilized in at most $\nu(G)$ steps. The final $K \in \mathcal{S} \cap \mathcal{T}^{\mathrm{d}}$ that will stabilize G, is equal to the summation of controllers at each step, i.e.:

$$K(s) \overset{(1.5)}{=} \sum\nolimits_k K_m^{(k)}(s).$$

□

We can easily show that stability of all the fixed modes of G, $\Lambda\left(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}}\right) \subset \mathbb{C}^-$, is also a necessary condition for the existence of stabilizing controller:

Theorem 24. *A plant G is stabilizable by a controller $K \in \mathcal{S} \cap \mathcal{T}^{\mathrm{d}}$, if and only if $\Lambda\left(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}}\right) \subset \mathbb{C}^-$.*

Proof. The sufficiency part is done in Theorem 23. For the necessity part note that static fixed modes can not be moved by the dynamic controller either (Theorem 16), i.e.:

$$\begin{aligned} & \Lambda\left(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}}\right) \not\subset \mathbb{C}^- \\ \overset{\text{Thm.16}}{\Rightarrow}\quad & \Lambda\left(G, \mathcal{S}, \mathcal{T}^{\mathrm{d}}\right) \not\subset \mathbb{C}^- \\ \overset{\text{bydef}}{\Rightarrow}\quad & \nexists\ K \in \mathcal{S} \cap \mathcal{T}^{\mathrm{d}} \quad \text{s.t.} \quad A_{\mathrm{CL}}(G, K) \subset \mathbb{C}^-. \end{aligned}$$

□

2.1.5 Synthesis and Numerical Example

In this section we provide an explicit algorithm to stabilize a plant which has no unstable fixed modes, and run it on one numerical example to illustrate its implementation. Algorithm 1 is distilled from the steps taken in the section to prove the sufficiency theorem, and thus can almost certainly be improved upon in several respects.

In Algorithm 1, D is chosen randomly at each outer-step, and as stated in Remark 18, would be a valid choice with probability one. This D must be chosen small enough ($\|D\|_\infty < \gamma^{(k)}$) such that the total number of unstable modes would not increase when each element of the sequence $\{D|_{(m)}\}_{m=1}^{a}$ is closed around $\check{G}^{(k)}$. A prior knowledge of such an upper bound on D, denoted by $\gamma^{(k)}$, is not available and is hard to attain. This leads us to consider the alternative approach of repeatedly making D smaller in a loop until Theorem 19.2 holds. This iterative scaling repeats itself when (2.19) is not met. In this case, as proof suggests, we perturb $L^{(k)}$ by $\hat{L}^{(k)}$. This perturbation must be chosen small enough that it will not make any modes of $\tilde{A}_{11}^{(k)} - (L^{(k)} + \hat{L}^{(k)})\tilde{C}_1^{(k)}$ unstable. The upper bound on the perturbation $\hat{L}^{(k)}$ is unknown, and thus, similar to the case for D, we iterate to make it small enough to meet the constraints.

Remark 25. *The intersection in the* ***if-then*** *section in Algorithm 1 would almost always result in a null set if interpreted with unlimited precision. However, choosing to replace the exact intersection with a proximity condition could possibly avoid very large feedback and observer gains.*

Algorithm 1 Finding a controller $K \in \mathcal{S} \cap \mathcal{T}^{\mathrm{d}}$ to stabilize G

```
Input: Plant $G$, information structure $\mathcal{S}$
Output: Controller $K \in \mathcal{S} \cap \mathcal{T}^{\mathrm{d}}$ that will stabilize $G$
  $k \leftarrow 0$, $\check{G}^{(0)} \leftarrow G$, $K(\sigma) \leftarrow 0$
  /* Repeat the outer loop until the plant is stabilized */
  while $|\tilde{\Lambda}_+(\check{G}^{(k)})| \geq 1$ do
    /* Select a static controller as in Rem. 18 */
    Choose a random $D \in \mathcal{S} \cap \mathcal{T}^{\mathrm{s}}$
    while $\nu(\Gamma(\check{G}^{(k)}, D)) > \nu(\check{G}^{(k)})$ do
      $D \leftarrow D/2$
    end while
    /* Find a controllable index as in Thm. 19 */
    $m^{(k)} \leftarrow a$
    while $\tilde{\Lambda}_+(\Gamma(\check{G}^{(k)}, D|_{(m^{(k)}-1)})) \cap \tilde{\Lambda}_+(\check{G}^{(k)}) = \varnothing$ do
      $m^{(k)} \leftarrow m^{(k)} - 1$
    end while
    /* Form the SISO plant as in Fig. 2.3 */
    $\check{G}^{(k)}_{m^{(k)}} \leftarrow \mathbf{e}^T_{j_{m^{(k)}}} \Gamma(\check{G}^{(k)}, D|_{(m^{(k)}-1)})\mathbf{e}_{i_{m^{(k)}}}$
    /* Stabilize the SISO plant as in Prop. 20 */
    Find a Kalman similarity transformation $T^{(k)}$ for $\check{G}^{(k)}_{m^{(k)}}$
    Name all the corresponding partitions by $\tilde{(\cdot)}^{(k)}$
    Find a $F^{(k)}$ to stabilize $\tilde{A}^{(k)}_{11} - \tilde{B}^{(k)}_1 F^{(k)}$
    Find a $L^{(k)}$ to stabilize $\tilde{A}^{(k)}_{11} - L^{(k)}\tilde{C}^{(k)}_1$
    /* Ensuring that constraint (2.24) holds */
    $M^{(k)} \leftarrow (I - D|_{(m^{(k)}-1)} D^{(k)}_{\check{G}})^{-1}$
    $\check{D}^{(k)} \leftarrow \mathbf{e}^T_{j_{m^{(k)}}} D^{(k)}_{\check{G}} M^{(k)} \mathbf{e}_{i_{m^{(k)}}}$
    if $\mathrm{eig}_{+,0}(\tilde{A}^{(k)}_{11} - \tilde{B}^{(k)}_1 F^{(k)} + L^{(k)}(\check{D}^{(k)}F^{(k)} - \tilde{C}^{(k)}_1)) \cap \left(\bigcup_{i=2}^{4} \mathrm{eig}_{+,0}(\tilde{A}^{(k)}_{ii})\right) \neq \varnothing$ then
      /* Perturb the observer gain if (2.24) does not hold */
      Choose a random $L^{(k)}_\epsilon$
      /* Make the perturbation sufficiently small not to have any new unstable mode */
      while $|\mathrm{eig}_{+,0}(\tilde{A}^{(k)}_{11} - (L^{(k)} + L^{(k)}_\epsilon)\tilde{C}^{(k)}_1)| \geq 1$ do
        $L^{(k)}_\epsilon \leftarrow L^{(k)}_\epsilon/2$
      end while
      $L^{(k)} \leftarrow L^{(k)} + L^{(k)}_\epsilon$
    end if
    /* Construct the MIMO controller as in Cor. 21 */
    $K^{(k)} \leftarrow \left[\begin{array}{c|c} \tilde{A}^{(k)}_{11} - \tilde{B}^{(k)}_1 F^{(k)} + L^{(k)}\left(\check{D}^{(k)}F^{(k)} - \tilde{C}^{(k)}_1\right) & L^{(k)}\mathbf{e}^T_{j_{m^{(k)}}} \\ \hline -\mathbf{e}_{i_{m^{(k)}}} F^{(k)} & D|_{(m^{(k)}-1)} \end{array}\right]$
    $K \leftarrow K + K^{(k)}$
    $\check{G}^{(k+1)} \leftarrow \Gamma(\check{G}^{(k)}, K^{(k)})$
    $k \leftarrow k + 1$
  end while
  return $K$
```

Remark 26. *We can replace $\mathbb{C}^-$ throughout the section with another open set of acceptable closed-loop eigenvalues, letting its complement replace $\bar{\mathbb{C}}^+$ as the closed set of unacceptable closed-loop eigenvalues. The results of Section 2.1.3 hold up to show that the fixed modes must not be in the unacceptable region, the results of Section 2.1.4 hold up to show that if they are not, then all of the modes can be moved to the acceptable region, and Algorithm 1 can be applied to find a controller which achieves that objective. One can further define a smaller open set of desirable closed-loop eigenvalues into which all of the non-fixed modes can be moved by Algorithm 1, taking note of the possibility of fixed and non-fixed modes overlapping in the acceptable-yet-undesirable region, as mentioned in Section 2.1.1.*

The following numerical example will use Algorithm 1 to stabilize the plant G.

Example 27. *Consider the following plant:*

$$A = \mathrm{diag}(2, 3, 5, -1, -1)$$

$$B = \begin{bmatrix} 0 & 0 & 3 & 0 & 2 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 2 & 0 & 5 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \end{bmatrix} \quad C = \begin{bmatrix} 4 & 0 & 8 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 6 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 6 \end{bmatrix}$$

and $D = 0$. Let the sparsity-induced information structure for the controller be given *by the admissible-to-be-nonzero indices:*

$$\mathrm{Adm}(\mathcal{S}) = \{(1,1),(3,1),(4,1),(5,2),(1,3),(3,3),(4,3),(5,4),(5,5)\}.$$

This plant has fixed mode $\Lambda\left(G,\mathcal{S},\mathcal{T}^{\mathrm{s}}\right) = \{-1\}$. *If we follow Algorithm 1 to stabilize* G, *and choose our desired closed-loop modes of* $\Gamma(G,K)$ *to be*

$$\begin{bmatrix} -0.5 & -1 & -1 & -1.5 & -2 & -2.5 & -3 & -3.5 \end{bmatrix}^T,$$

this is achieved by the following resulting controller:

$$A_K = \begin{bmatrix} 14.92 & -460.40 & -4.66 \\ 0.37 & -24.44 & 0.74 \\ 22.92 & -763.84 & -25.42 \end{bmatrix} \quad B_K = \begin{bmatrix} & 317.11 & \\ 0_{3\times1} & 27.44 & 0_{3\times3} \\ & 405.61 & \end{bmatrix}$$

$$C_K = \begin{bmatrix} & 0_{4\times3} & \\ 3.90 & -71.64 & -7.44 \end{bmatrix}^T \quad D_K = \begin{bmatrix} 0.08 & \\ 0 & \\ 0.42 & 0_{5\times4} \\ 0.09 & \\ 0 & \end{bmatrix}.$$

An alternative approach is taken in [24], in which, at each step, a (possibly dynamic) stabilizing controller is applied at the next diagonal element of the controller, and it is shown that by adding stabilizing controllers at each step, the set of (possibly unstable) fixed modes are reduced, until the last step where the remaining fixed modes *must be necessarily stable. Applying the method of [24] on this plant would result in a stabilizing controller of order 7, as compared to 3 here. An explanation could be that in [24], a (possibly dynamic) stabilizing controller is applied at each of the elements, resulting in abundant of controller states, whereas in here, only for each unstable mode, a stabilizing controller (not necessarily of order 1) is needed.*

If we look at each of the nine SISO maps from u_{i_m} to y_{j_m} in G, then the union of controllable and observable modes of all these SISO maps are $\{2, 5\}$, which does not contain the unstable mode 3. This shows that a static gain (the D_{m-1} of Figure 2.3) might be necessary to assign some modes in decentralized settings, which is counter-intuitive compared to the centralized case where a stabilizing observer-based controller would have zero static gain.

2.2 Decentralized Non-overshooting Stabilization

We review existing results regarding centralized non-overshooting stabilizability in Section 2.2.1, and then extend this to the decentralized case in Section 2.2.2. Section 2.2.3 containa numerical examples to further clarify this type of stability, and its comparison to the well known decentralized internal stabilizability that was developed in Section 2.1.

2.2.1 Centralized Case

Materials in this section are mostly adopted from the ones in [5,25], and readers are referred to these papers for detailed discussions and examples of each type of stability and their corresponding properties.

A state-space system of the form (1.2) is said to have ***overshoot***, if for some initial condition x_0, and some $t > t_0$, we have that $\|x(t)\| \geq \|x_0\|$. The following type of stability makes a close connection to overshooting properties.

Definition 28 ([5, Definition 2.3])**.** *A state-space system of the form* (1.2) *on page 13 is called strongly asymptotically stable (in the strict sense), if and only if* $\frac{d\|x(t)\|}{dt} < 0$*, for all* $t \geq t_0$*, and for all* $x(t_0) \neq 0$*.*

Remark 29 ([5, Remark 2.3])**.** *The above definition makes clear that if a state-space system is strongly asymptotically stable in strict sense, each trajectory enters a hyper-sphere* $\|x(t)\| = r \leq r_0 \triangleq \|x_0\|$ *from a non-tangential direction, and thus can not have an overshoot. It is for this reason that we refer to this type of stability as* ***non-overshooting stability****.*

A similar condition to the one in Remark 2 on page 13 is also available for this type of stability and is stated below.

Remark 30 ([5, Theorem 2.1])**.** *A state-space system of the form* (1.2) *is non-overshooting stable if and only if* $A + A^T \prec 0$*.*

Non-overshooting stability is invariant under orthogonal transformations of the form $(U^TAU, U^TB, CU, 0)$, where $U^TU = UU^T = I$, and is not invariant under arbitrary linear transformations, and thus is not a property of transfer matrices.

The following remark connects aforementioned two notion of stability to each other.

Remark 31 ([5, Remark 2.2])**.** *If a state-space system is non-overshooting stable, then it is also internally stable. Algebraically this means that, if $A + A^T \prec 0$, then $\Re\left(\mathrm{eig}\left(A\right)\right) < 0$.*

It holds that the non-overshooting stabilizability with respect to static controllers is a necessary and sufficient condition for non-overshooting stabilizability with respect to dynamic controllers:

Theorem 32 ([25, Proposition 4.2])**.** *A state-space system G is non-overshooting stabilizable by a static controller, if and only if it is non-overshooting stabilizable by a dynamic controller.*

Proof. See, for example Proposition 4.2 in [25]. □

We will follow the above proof and extend it to our constrained version in the next section. Finally, a direct observation in the proof of [25, Proposition 4.2] is that:

Corollary 33. *A state-space system K that makes a strictly proper state-space system G non-overshooting stabilized, is itself non-overshooting stable, and also internally stable (due to Remark 31).*

2.2.2 Decentralized Case

This section extends the non-overshooting stabilization in presence of constraints on the information structure of the controller. We will mainly focus on structures which manifest themselves as a sparsity pattern on the controller, and briefly mention when such an argument could be generalized to other types of structures.

We are interested in information structures that satisfy the following property, and will make it clear in the context when the following property is assumed.

Definition 34. *An information structure $\mathcal{S}$ is said to be **feedthrough consistent** if it is such that:*

$$D_K \in \mathcal{S}, \qquad \textit{for all } K \in \mathcal{S}. \tag{2.30}$$

Remark 35. *Any sparsity-induced information structure is feedthrough consistent.*

Other than sparsity constraints, this property holds for a wide variety of information structures in the control literature, including (but not limited to) delay, and symmetry constraints.

The following theorem ties the non-overshooting stabilizability with respect to static and dynamic controllers, when the information structure of interest is feedthrough consistent.

Theorem 36. *Assume that $\mathcal{S}$ is feedthrough consistent. A state-space system G is non-overshooting stabilizable by a proper dynamic state-space system in $\mathcal{S}$, if and only if it is non-overshooting stabilizable by static output feedback in $\mathcal{S}$.*

Proof. The proof is the extension of the similar one in [25] (in which G was required to be strictly proper, i.e., $D = 0$, while in here we assume G is a proper state-space system). The necessity part follows easily since the static controllers in $\mathcal{S}$ are themselves a subset of the dynamic controllers in $\mathcal{S}$, i.e.:

$$\mathcal{T}^{\mathrm{s}} \subset \mathcal{T}^{\mathrm{d}} \Longrightarrow \mathcal{S} \cap \mathcal{T}^{\mathrm{s}} \subset \mathcal{S} \cap \mathcal{T}^{\mathrm{d}}.$$

For the sufficiency part, assume that there exists a state-space system $K = (A_K, B_K, C_K, D_K) \in \mathcal{S} \cap \mathcal{T}^{\mathrm{d}}$ that makes G non-overshooting stabilized, then by Remark 30:

$$A_{\mathrm{CL}}(G, K) + A_{\mathrm{CL}}(G, K)^T \prec 0. \tag{2.31}$$

Expanding A_{CL} in the above equation will result in:

$$\begin{bmatrix} A + BD_K NC & BMC_K \\ B_K NC & A_K + B_K NDC_K \end{bmatrix} + A_{\mathrm{CL}}(G, K)^T \prec 0. \tag{2.32}$$

Since (2.32) is negative definite, its first block-diagonal element must itself be negative definite, i.e.:

$$A + A^T + BD_K NC + C^T N^T D_K^T B^T \prec 0.$$

Also since (2.30) guarantees that D_K is itself an admissible controller, we have that:

$$\begin{aligned} & A_{\mathrm{CL}}(G, D_K) + A_{\mathrm{CL}}(G, D_K)^T \\ = & A + A^T + BD_K NC + C^T N^T D_K^T B^T \prec 0, \quad D_K \overset{(2.30)}{\in} \mathcal{S}, \end{aligned}$$

thus $D_K \in \mathcal{S} \cap \mathcal{T}^{\mathrm{s}}$ makes G non-overshooting stabilized. □

The following corollary is a direct consequence of Theorem 36, and demonstrates a standard approach for obtaining a static non-overshooting stabilizing controller when G is strictly proper.

Corollary 37. *If a state-space system G is strictly proper ($D = 0$), and information structure $\mathcal{S}$ is such that in addition to being feedthrough consistent, we have that $D_K \in \mathcal{S}$ is a convex criterion, then G is non-overshooting stabilizable if and only if the following convex program is feasible:*

$$\begin{aligned} &\text{find} && D_K \\ &\text{subject to} && A_{\text{CL}}(G, D_K) + A_{\text{CL}}(G, D_K)^T \prec 0, \\ &&& D_K \in \mathcal{S}, \end{aligned} \tag{2.33}$$

with variable $D_K \in \mathbb{R}^{n_u \times n_y}$. Specifically, if we consider sparsity-induced information structures, which all correspond to imposing affine constraints on D_K, we have the following LMI for checking non-overshooting stabilizability:

$$\begin{aligned} &\text{find} && D_K \\ &\text{subject to} && A_{\text{CL}}(G, D_K) + A_{\text{CL}}(G, D_K)^T \prec 0, \\ &&& (D_K)_{ij} = 0 \qquad \text{for all} \quad (i,j) \text{ s.t. } K^{\text{bin}}_{ij} = 0, \end{aligned} \tag{2.34}$$

with variable $D_K \in \mathbb{R}^{n_u \times n_y}$.

However, when $D \neq 0$, the term $D_K(I - DD_K)^{-1}$ in A_{CL} makes (2.33) nonconvex and thus we can not use Corollary 37 directly. We derive an alternative approach for checking non-overshooting stabilizability through a change of variable when G is proper, and when $\mathcal{S}$ is further quadratic invariant under D.

The following theorem will demonstrate when and how a change of variable for the $D \neq 0$ case could make A_{CL} convex in its variable.

Theorem 38. *Assume that we have a proper state-space system G, and the information structure $\mathcal{S}$ is a closed subspace that is QI under D, then:*

$$\{A_{\mathrm{CL}}(G, D_K) \ \mid \ D_K \in \mathcal{S}\} \ = \ \{A_{\mathrm{CL}}(G - D, D_Q) \ \mid \ D_Q \in \mathcal{S}\}. \tag{2.35}$$

Proof. Given D_K, define

$$D_Q \triangleq D_K(I - DD_K)^{-1}. \tag{2.36}$$

Also, by inverting this map, one can verify that:

$$D_K = (I + D_Q D)^{-1} D_Q = D_Q(I + DD_Q)^{-1}. \tag{2.37}$$

Due to [6, Theorem 14], because $\mathcal{S}$ is QI under D, we have the following relation between the set of admissible $D_K \in \mathcal{S}$, and the transformed variable D_Q:

$$D_K \in \mathcal{S} \quad \text{if and only if} \quad D_Q \in \mathcal{S}. \tag{2.38}$$

It is notable that (2.36) is well defined if and only if (2.37) is well defined, i.e., since $(I - DD_K)^{-1} = I + DD_Q$, we have that

$$1 \notin \operatorname{eig}\left(DD_K\right) \Leftrightarrow -1 \notin \operatorname{eig}\left(DD_Q\right) \Leftrightarrow -1 \notin \operatorname{eig}\left(D_Q D\right).$$

Finally, the equivalence of the constraints on the original D_K, and the transformed variable D_Q (2.38), along with

$$\begin{aligned} A_{\mathrm{CL}}(G, D_K) &\overset{(1.4)}{=} A + BD_K(I - DD_K)^{-1}C \\ &\overset{(2.36)}{=} A + BD_QC \\ &\overset{(1.4)}{=} A_{\mathrm{CL}}(G - D, D_Q), \end{aligned}$$

yield that (2.35) holds. □

The following corollary will give an alternative approach to check for non-overshooting stabilizability, when D is not necessarily zero. Precisely, it will demonstrate how to use the result of Corollary 37, and Theorem 38 for the case of $\mathcal{S}$ being QI under D, and G being a proper state-space system.

Corollary 39. *Assume that G is a proper state-space system, and $\mathcal{S}$ is such that further than assumptions of Theorem 38, is feedthrough consistent. Then, a necessary and sufficient condition for non-overshooting stabilizability of G is given by the following convex program:*

$$\begin{aligned} &\text{find} && D_Q \\ &\text{subject to} && A_{\mathrm{CL}}(G - D, D_Q) + A_{\mathrm{CL}}(G - D, D_Q)^T \prec 0, \\ & && D_Q \in \mathcal{S}, \end{aligned} \tag{2.39}$$

with variable $D_Q \in \mathbb{R}^{n_u \times n_y}$. Similarly, if we consider sparsity-induced information *structures, we have:*

$$\begin{aligned} &\text{find} && D_Q \\ &\text{subject to} && A_{\mathrm{CL}}(G-D, D_Q) + A_{\mathrm{CL}}(G-D, D_Q)^T \prec 0, \\ & && (D_Q)_{ij} = 0 \qquad \text{for all} \quad (i,j) \text{ s.t. } K_{ij}^{\mathrm{bin}} = 0, \end{aligned} \tag{2.40}$$

with variable $D_Q \in \mathbb{R}^{n_u \times n_y}$. To recover the original D_K, we can use the inverted map (2.37).

Proof. Due to the Theorem 38, (2.35) holds, thus, we can use Corollary 37 with G, and D_K in Corollary 37 replaced by respectively $G-D = (A, B, C, 0) \in \mathcal{R}_{\mathrm{sp}}$ and D_Q. □

Next, we study a necessary condition for non-overshooting stabilizability with respect to sparsity-induced information structures. Specifically, we take advantage of the necessary and sufficient connection between internal stabilizability of a plant and its fixed-modes locations developed in Section 2.1, and relate it to non-overshooting stabilizability in the following theorem.

Theorem 40. *If a state-space system G is non-overshooting stabilizable with respect to a sparsity-induced information structure $\mathcal{S}$, then it has all of its fixed mode in the open left half plane, i.e.:*

$$\Lambda\left(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}}\right) \subset \mathbb{C}^{-}.$$

Proof. As stated in Remark 31, non-overshooting stabilizability is a stronger condition than internal stabilizability and implies it. Therefore, if G is non-overshooting stabilized, it is also internally stabilized , which in conjunction with Theorem 24 gives the desired result. □

2.2.3 Numerical Examples

We will conclude this section by inspecting the main concepts of this section through two numerical examples. First we use an example to see when non-overshooting stabilizability is feasible, and then inspect its related time response behavior

Example 41. *Let G be given as*

$$A = \operatorname{diag}(-3, -1, 1.5, 2.5),$$

$$B = \begin{bmatrix} 2 & 0 & 0 \\ 3 & 4 & 0 \\ 0 & 5 & 6 \\ 0 & 0 & 7 \end{bmatrix}, \qquad C = \begin{bmatrix} 8 & 0 & 0 & 9 \\ 0 & 10 & 11 & 0 \\ 0 & 0 & 12 & 13 \end{bmatrix},$$

and $D = 0$. Let sparsity be given by

$$K^{\text{bin}} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}.$$

We use cvx *[26] to solve* (2.34) *for D_K, and we have:*

$$D_K = \begin{bmatrix} -0.69 & 0 & 0 \\ 0.12 & -1.05 & 0 \\ 0 & 0.37 & -1.21 \end{bmatrix}.$$

Which gives the following closed-loop modes:

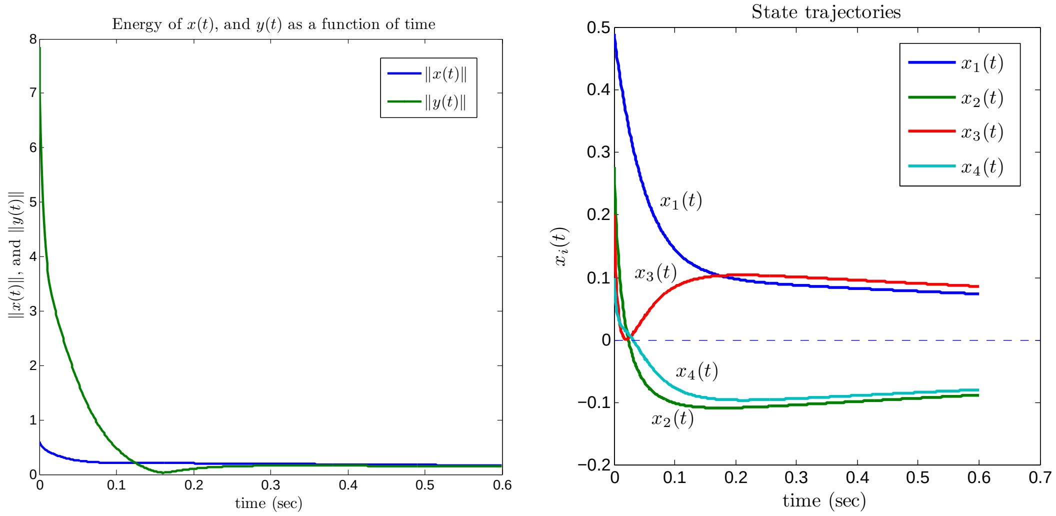
$$\begin{aligned} \operatorname{eig}\left(A_{\text{CL}}(G, D_K)\right) &= \{-194.65, -0.55, -21.66, -66.91\}, \\ \operatorname{eig}\left(A_{\text{CL}}(G, D_K) + A_{\text{CL}}^T(G, D_K)\right) &= \{-396.60, -132.13, -38.71, -0.15\}. \end{aligned}$$

We pick a random initial condition:

$$x_0 = \begin{bmatrix} 0.49 & 0.27 & 0.20 & 0.09 \end{bmatrix}^T,$$

and illustrate $\|x(t)\|$ and $\|y(t)\|$ in Figure 2.5(a), and the corresponding state trajectories in Figure 2.5(b). It is noteworthy that Figure 2.5(a) hints that although the joint energy of states is monotonically decreasing, energy of output variables might not necessarily follow the same pattern, and can have an overshoot. In fact, as illustrated in Figure 2.5(b), it might be the case that increase in some states would be compensated by the other ones such that the norm ($\|x(t)\|$) would be monotonically decreasing. Nevertheless, we have that $\lim_{t\to\infty} \|y(t)\| = \lim_{t\to\infty} \|x(t)\| = 0$.

We established that G having all of its fixed modes in LHP is a necessary condition for non-overshooting stabilizability. However, as the following numerical example shows, having all the fixed modes in the LHP does not guarantee non-



(a) Closed-loop output and state energy

(b) State trajectories

Figure 2.5: State energy, output energy, and state trajectories for a given initial condition x_0 in Example 41

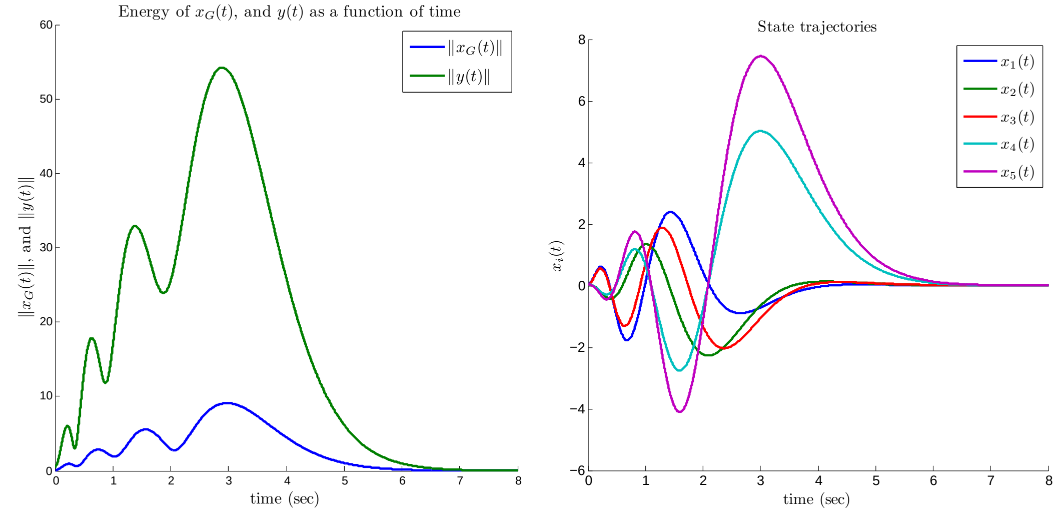
overshooting stabilizability.

Example 42. *Let G be given as:*

$$A = \mathrm{diag}(2, 3, 5, -1, -1),$$

$$B = \begin{bmatrix} -1 & 0 & 2 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \\ 1 & 0 & 0 \\ 0 & 2 & 0 \end{bmatrix}, \qquad C = \begin{bmatrix} 4 & 7 & 8 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 6 \end{bmatrix},$$

and $D = 0$. Also let the binary matrix corresponding to the sparsity-induced infor-



(a) Closed-loop output and state energy

(b) State trajectories

Figure 2.6: State energy, output energy, and state trajectories for a given initial condition x_0 in Example 42

mation structure be given as:

$$K^{\text{bin}} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}.$$

Then we have:

$$\Lambda\left(G, \mathcal{S}, \mathcal{T}^{\text{s}}\right) = \{-1\} \subset \mathbb{C}^{-}.$$

The LMI in (2.34) *is infeasible for D_K while a internally stabilizing controller of order 4 can be found by the algorithm suggested in Section 2.1. State energy, output energy, and plant state trajectories when this internally stabilizing controller is closed around G is illustrated in Figure 2.6. As Figure 2.6(a) shows, even the state energy is no more monotonically decreasing.*

Chapter 3: Optimal $\mathcal{H}_\infty$ Synthesis

Model-matching problems emerge at the core of many estimation and controller design tasks. This problem has been heavily studied for the most significant measures of performance in absence of decentralization constraints. This has resulted in analytical insights, and various synthesizing algorithms along with characterization of their properties for obtaining an exact optimal solution. However, this problem has remained largely intractable in general in presence of decentralization constraints. We focus on the optimal decentralized model-matching problem in $\mathcal{H}_\infty$ sense in this section. The main techniques for centralized $\mathcal{H}_\infty$ control, linear matrix inequalities (LMIs) [27] and Riccati equations [28], do not allow one to optimize directly over the controller, and thus do not present an obvious way to allow one to place constraints on the controller. There has thus been a variety of methods trying to solve for structured $\mathcal{H}_\infty$ controller by approximation, including a homotopy-based method for solving a non-convex bilinear matrix inequality [29], finding local optima with non-smooth optimization techniques [30], and an approach based on dissipative property of systems which result in sub-optimal $\mathcal{H}_\infty$ controller design by LMIs [31]. There are relationships between structured control problems and multi-objective problems [32], and a parametrization based on a finite-dimensional basis has been used to approach the solution for the latter [33].

When the problem one wishes to solves satisfies a certain condition, called quadratic invariance, the optimal decentralized control problem may be cast as a convex model-matching optimization problem, regardless of which closed-loop norm the designer wishes to optimize [6]. For some specific structures or objectives, the problem can be further reduced to one which is solved by standard methods. For example, when both the plant and controller admit lower triangular structure, exact optimal $\mathcal{H}_\infty$ controllers could be found via a finite number of LMIs solved one after another [34]. When the objective of interest is the $\mathcal{H}_2$-norm, the problem can be reduced to a centralized problem [6]. In general, however, the resulting problem is infinite-dimensional and still non-trivial, particularly for certain objectives.

As with centralized control, there are many cases where one must optimize for the worst-case, such as the decentralized control of smart structures to prevent failure during earthquakes [35], for which $\mathcal{H}_\infty$-norm is considered to be a more appropriate objective. While many of the techniques we study could be applied to arbitrary or mixed objectives, we focus on the $\mathcal{H}_\infty$-norm in this chapter.

When the problem of interest is quadratically invariant, one can apply Q-parametrization and transform the closed-loop into a model-matching form, for which we then use a sequence of finite-dimensional (FD) parametrization of the Q-parameter, with the property that the solutions of this sequence approach that of our original problem. We discuss methods for solving the finite-dimensional parametrized version of our problem, which could easily be adapted for any norm of interest. As a keystone of our framework, we show how the main result of [36] can be used to recast any of the FD optimization problems in this sequence as a semidefinite program (SDP) when the norm of interest is indeed the $\mathcal{H}_\infty$-norm.

A natural choice of an FD parametrization in the discrete domain is a Finite Impulse Response (FIR) basis that corresponds to all of the poles being placed at the origin, which is not always the best choice. We will then consider another FD parametrization that places the poles in other locations of the complex plane rather than the origin. This could be equivalently formulated as a basis selection, or a dictionary learning problem for the pole locations. However, there is no clear way to choose the order of the controller and optimize the location of the controller poles from a continuum of choices, both because the $\mathcal{H}_\infty$-optimal decentralized model-matching problem might not always have a rational solution [37], and because the problem is non-convex in pole locations. Hence, we develop and explore several methods for improving pole selection in later parts of this chapter as was first done in [38]. We approach the basis selection problem in three stages as was first done in [39]. We first use the poles from the centralized solution as an initial choice for the pole locations. Second, we adopt the convex program introduced in [40] to automatically choose a sparse combination of the poles among a finite set of stable ones. Third, we use the Taylor linearization to linearize the controller around small perturbations added to its poles, and show that this linearization will result in a convex program that can choose perturbations in a way that improves the objective.

Although this chapter mainly focuses on discrete-time systems, by appropriately adjusting the candidate bases, continuous-time counterparts could be derived similarly. These results can also be extended to include delay constraints in discrete-time as long as they satisfy quadratic invariance.

The organization of this chapter is as follows. We form the problem of finding an $\mathcal{H}_\infty$-optimal decentralized controller in Section 3.1. Quadratic Invariance (QI) and its consequences is discussed in Section 3.2. Section 3.3 demonstrates the finite basis used for approximating an infinite dimensional controller and its corresponding state-space representation. In Section 3.4, we show how, for any norm, the decentralized controller parametrized by a FD basis could be thought of as static output feedback (SOF) with sparsity pattern imposed on the static gain and proceed to solve it in case of $\mathcal{H}_\infty$ norm with help of method proposed by [36]. Section 3.5 will demonstrate improved methods regarding basis selection in three stages, we will illustrate use of poles from centralized solution in Section 3.5.1, adopt a convex program (based on l_1 heuristic) that suits selection of a sparse combination of poles in a finite set of stable ones in Section 3.5.2, and introduce the Taylor approximation for improving pole locations in Section 3.5.3. In order to study practical aspects of proposed methods, numerical examples are introduced throughout the section to both clarify the methods, and to compare them with each other.

3.1 Problem Setup

Given a generalized plant P and a subspace of admissible controllers $\mathcal{S}$, the following is the optimal decentralized control problem we seek to address:

$$\begin{aligned} \underset{K \in \mathcal{R}_{\mathrm{p}}}{\text{minimize}} \quad & \|f_{\mathrm{LFT}}(P, K)\|_{\mathcal{H}_\infty} \\ \text{subject to} \quad & K \text{ stabilizes } P \\ & K \in \mathcal{S}, \end{aligned} \tag{3.1}$$

where the lower linear fractional transformation (LFT) function $f_{\mathrm{LFT}}(\cdot, \cdot)$ is defined as in (1.3). Which subsystems can affect others is embedded in the sparsity pattern of P, and which subsystem controllers can access the sensor information from which others' is embedded in $\mathcal{S}$. We call the subspace $\mathcal{S}$ the ***information structure***.

Many decentralized control problems may be expressed in the form of problem (3.1), including all of those addressed in [41, 42]. The problem is intractable in general, but for some P and $\mathcal{S}$, has been shown to be equivalent to a convex optimization problem. This is the subject of the next section, and we will then focus on methods to solve those problems for the $\mathcal{H}_\infty$-norm.

3.2 Quadratic Invariance

We defined quadratic invariance in Definition 3. Here, we give a brief overview of related results, in particular, that if it holds then convex synthesis of optimal decentralized controllers is possible.

It was shown in [6] that if $\mathcal{S}$ is a closed subspace and $\mathcal{S}$ is quadratically invariant under G, then with a change of variables, problem (3.1) is equivalent to the following optimization problem:

$$\begin{aligned} &\underset{Q \in \mathcal{RH}_\infty}{\text{minimize}} \quad \|T_1 - T_2 Q T_3\|_{\mathcal{H}_\infty} \\ &\text{subject to} \quad Q \in \mathcal{S}. \end{aligned} \tag{3.2}$$

where $T_1, T_2, T_3 \in \mathcal{RH}_\infty$. See Theorem 17 in [6] for finding T_1,T_2, T_3 and recovering K from Q. Through the rest of this chapter we will focus on this equivalent form instead of (3.1).

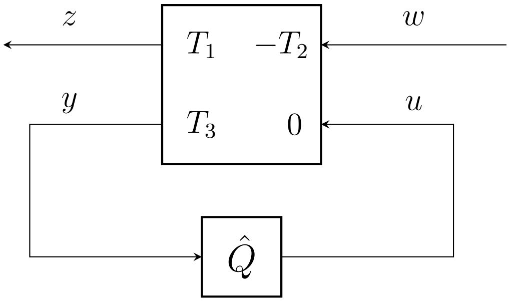


Figure 3.1: Model-matching problem from Youla parametrization

This states that if our problem is quadratically invariant (QI), we may use a particular Youla parametrization [43] to reduce the problem to the model-matching problem shown in Figure 3.1, as one can for centralized problems, and the constraint on the controller is passed on to the Youla parameter. The optimization problem in (3.2) is then convex. We may solve it to find the optimal Q, and then recover the optimal K for our original problem (3.1). Similar results have been achieved [44] for other function spaces as well, also showing that quadratic invariance allows optimal linear decentralized control problems to be recast as convex optimization problems.

While the problem is convex, the domain is infinite-dimensional, and solving it is certainly not straightforward. This equivalence holds for arbitrary closed-loop norm in the objective, and when the norm of interest is instead the $\mathcal{H}_2$-norm, it was shown in [6] that the problem can be further reduced to an unconstrained optimal control problem and then solved with standard software. Some recent progress has also been made to directly compute the optimal state-space controller parameters for some specific information structures in the $\mathcal{H}_2$ case [45].

3.3 Finite-Dimensional Parametrizations of Q

In this section, we discuss a method for addressing the convex infinite dimensional model-matching problem (3.2), by a FD parametrization of the Q-parameter. This has long been used for the centralized problem (without the constraint) for objectives where more elegant solutions are not, or were not available (including multiple-objective problems [33]), and has been suggested as a possible method for (3.2) since the QI results were first available. The idea is to use a finite-dimensional basis to parametrize the domain $\mathcal{RH}_\infty$, where the limit of the span will be dense in the original domain. We will first illustrate this FD parametrization for the usual choice of basis in discrete time, which corresponds to a Finite Impulse Response (FIR) of different delays in different parts of the controller, and then generalize it to other bases.

Suppose we choose a maximum order of N for the map between each input and output of the controller element. Then for each $i \in \{1, \cdots, n_u\}$ and $j \in \{1, \ldots, n_y\}$, we have an approximate FD parametrization:

$$\hat{Q}_{ij}(z) = \sum_{k=0}^{N} \frac{\alpha_{ijk}}{z^k},$$

and there are $n_u \cdot n_y \cdot (N+1)$ variables to find.

We can then state the following FD parametrized approximation to our convex decentralized model-matching problem (3.2):

$$\begin{aligned} \text{minimize} \quad & \|T_1 - T_2\hat{Q}T_3\|_{\mathcal{H}_\infty} \\ \text{subject to} \quad & \hat{Q}_{ij}(z) = \sum_{k=0}^{N} \frac{\alpha_{ijk}}{z^k} \\ & \hat{Q} \in \mathcal{S}, \end{aligned} \tag{3.3}$$

with variables $\hat{Q} \in \mathcal{RH}_\infty^{n_u \times n_y}$, $\alpha \in \mathbb{R}^{n_u \cdot n_y \cdot (N+1)}$, and assuming that we substitute the first constraint into the objective, we have a finite-dimensional convex optimization problem in the vector α.

We can find a state-space representation of $\hat{Q}$ as described below. For each $j \in \{1, \ldots, n_y\}$, let $A_j^Q \in \mathbb{R}^{N \times N}$ and $B_j^Q \in \mathbb{R}^N$ be given as:

$$A_j^Q = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & 0 & 1 \\ 0 & \cdots & \cdots & 0 \end{bmatrix}, \quad B_j^Q = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \tag{3.4}$$

and for each $i \in \{1, \cdots, n_u\}$ and $j \in \{1, \ldots, n_y\}$, let

$$C^Q_{ij} = \begin{bmatrix} \alpha_{ijN} & \cdots & \alpha_{ij1} \end{bmatrix}, \quad D^Q_{ij} = [\alpha_{ij0}].$$

Then define

$$A_Q = \text{diag}(A^Q_1, ..., A^Q_{n_y}), \qquad B_Q = \text{diag}(B^Q_1, ..., B^Q_{n_y}), \tag{3.5}$$

$$C_Q = \begin{bmatrix} C^Q_{11} & \cdots & C^Q_{1n_y} \\ \vdots & \ddots & \vdots \\ C^Q_{n_u1} & \cdots & C^Q_{n_un_y} \end{bmatrix}, D_Q = \begin{bmatrix} D^Q_{11} & \cdots & D^Q_{1n_y} \\ \vdots & \ddots & \vdots \\ D^Q_{n_u1} & \cdots & D^Q_{n_un_y} \end{bmatrix} \tag{3.6}$$

and we have

$$\hat{Q}(z) \;=\; \left[\begin{array}{c|c} A_Q & B_Q \\ \hline C_Q & D_Q \end{array}\right](z). \tag{3.7}$$

Remark 43. *With this representation, all of the parameters α_{ijk} have been gathered in only C_Q and D_Q. This will allow the problem to be cast as one of finding an optimal static output feedback controller.*

Remark 44 (Alternative parametrization)**.** *It is similarly possible to gather all the variable parameters in B_Q and D_Q and have the fixed parts in A_Q and C_Q. Our methods could be very similarly adjusted to account for this case too.*

Remark 45 (Different poles per column)**.** *We can generalize this parametrization to allow for different poles for each column of $\hat{Q}$, as long as the controller remains stable. To this end, suppose each column $j \in \{1, \cdots, n_y\}$ of $\hat{Q}$ has n_j different poles* *and denote them by $p_1^j, \cdots, p_{n_j}^j$ (all in $\mathbb{D}$). Then, instead of (3.4), we have that:*

$$A_j^Q = \operatorname{diag}\left(p_1^j, \cdots, p_{n_j}^j\right), \qquad B_j^Q = \boldsymbol{1}_{n_j},$$

where $A_j^Q \in \mathbb{R}^{n_j \times n_j}$ are possibly different diagonal matrices representing stable poles of the controller for columns $j = 1, \cdots, n_y$. The order of $\hat{Q}$ would then be $n_{\hat{Q}} \triangleq \sum_{j=1}^{n_y} n_j$ and A_Q and B_Q can be constructed as (3.5). We then have $C_{ij}^Q \in \mathbb{R}^{1 \times n_j}$ and thus $C_Q \in \mathbb{R}^{n_u \times n_{\hat{Q}}}$.

With the modification indicated in the following remark, we can further allow for complex poles.

Remark 46 (Complex poles)**.** *Since we are interested in $\hat{Q} \in \mathcal{RH}_\infty$, complex eigenvalues of A_j^Q must appear in conjugate pairs $\alpha \pm \boldsymbol{j}\beta$. In order to keep both A_Q and C_Q real matrices (thus eliminating the need for specifying which coefficients in C_Q must be conjugate of each other) whenever a complex conjugate pair is in the spectrum of A_j^Q, instead of $\begin{bmatrix} \alpha + \boldsymbol{j}\beta & 0 \\ 0 & \alpha - \boldsymbol{j}\beta \end{bmatrix}$, we will represent it as $\begin{bmatrix} \alpha & \beta \\ -\beta & \alpha \end{bmatrix}$ by applying a similarity transformation. Similarly, for the corresponding parts in B_j^Q, we will use $\begin{bmatrix} 2 & 0 \end{bmatrix}^T$ instead of $\begin{bmatrix} 1 & 1 \end{bmatrix}^T$, which is obtained by applying the same similarity transformation.*

Remark 47. *With this representation, we get $\hat{Q}_{ij} = C_{ij}^Q (zI - A_j^Q)^{-1} B_j^Q + D_{ij}^Q$.*

With the parameters all gathered in C_Q and D_Q, we now state a lemma showing how to impose the sparsity-induced information structures on these variables.

Lemma 48. *If* $\hat{Q} = \left[\begin{array}{c|c} A_Q & B_Q \\ \hline C_Q & D_Q \end{array}\right]$*, with* A_Q, B_Q, C_Q, D_Q *given as above, then* $\hat{Q} \in \mathcal{S}$ *if and only if*

$$\begin{aligned} C^Q_{ij} &= 0 \textit{ for all } (i,j) \textit{ s.t. } K^{\text{bin}}_{ij} = 0 \\ D^Q_{ij} &= 0 \textit{ for all } (i,j) \textit{ s.t. } K^{\text{bin}}_{ij} = 0. \end{aligned} \tag{3.8}$$

Proof. It is straightforward to verify that the state-space representation gives

$$\hat{Q}_{ij}(z) = \sum\nolimits_{k=1}^{n_j} \frac{\alpha_{ijk}}{z - p^j_k} + \alpha_{ij0}$$

for the general case of different poles, or similarly

$$\hat{Q}_{ij}(z) = \sum\nolimits_{k=1}^{N} \frac{\alpha_{ijk}}{z^k} + \alpha_{ij0}$$

for the case of FIR parametrization. It then follows that $\hat{Q} \in \mathcal{S}$ if and only if $\alpha_{ijk} = 0$ for all $k \in \{0, 1, \ldots, n_j\}$, and for all (i,j) such that $K^{\text{bin}}_{ij} = 0$, which can be equally expressed as (3.8). □

We can then form an equivalent optimization problem using this lemma and the FD parametrized version of problem (3.2) by replacing Q with $\hat{Q}$ and optimizing over $C_Q \in \mathbb{R}^{n_u \times n_{\hat{Q}}}$ and $D_Q \in \mathbb{R}^{n_u \times n_y}$, thus leaving the following finite-dimensional convex optimization problem

$$\begin{aligned}
\text{minimize} \quad & \|T_1 - T_2 \hat{Q} T_3\|_{\mathcal{H}_\infty} \\
\text{subject to} \quad & \hat{Q} = \left[\begin{array}{c|c} A_Q & B_Q \\ \hline C_Q & D_Q \end{array}\right] \\
& C_{ij}^Q = 0 \text{ for all } (i,j) \text{ s.t. } K_{ij}^{\text{bin}} = 0 \\
& D_{ij}^Q = 0 \text{ for all } (i,j) \text{ s.t. } K_{ij}^{\text{bin}} = 0,
\end{aligned} \tag{3.9}$$

with variables $\hat{Q} \in \mathcal{RH}_\infty^{n_u \times n_y}, C_Q \in \mathbb{R}^{n_u \times n_{\hat{Q}}}, D_Q \in \mathbb{R}^{n_u \times n_y}$, and assuming that we substitute the first constraint into the objective, we have a convex finite-dimensional problem in the matrices C_Q and D_Q.

3.3.1 Subgradient

We first address problem (3.3) directly for the FIR parametrization given in (3.4), (3.5), and (3.6). Generalization of this subgradient for other arbitrary basis could also be derived similarly. We will show that we can compute the objective and its subgradient for a given value of variable α. We can thus solve the optimal decentralized model-matching problem with the considered FD Q-parameter using various methods. We demonstrate it here using the ellipsoid method. If evidence arises that this is remotely competitive with the performance of our main result, then more sophisticated algorithms will be explored for this direct approach.

The main ideas are from [46] and are adapted for the decentralized case. The advantage of this basic approach in here is that it can easily be modified to be adapted for other objectives of interest, such as multi-objectives or other norms.

Given a convex functional $f : \mathcal{X} \mapsto \mathbb{R}$, a subgradient of f at x_0 evaluated on y, denoted by $f^{\mathrm{sg}}(x_0, y) : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$, is a linear functional in its second variable, such that:

$$f(y) \geq f(x_0) + f^{\mathrm{sg}}(x_0, y) - f^{\mathrm{sg}}(x_0, x_0), \quad \forall\, y \in \mathcal{X}. \tag{3.10}$$

This definition suits the convex functionals considered in this chapter, and is a more simplified and standard definition compared to those in Section 4.2 that are generalized to account for the local behavior of non-convex functions.

We want to obtain a subgradient of $\mathcal{H}_\infty$-norm of closed-loop map $T_1 - T_2 Q T_3$ when the controller Q is approximated by the FIR parametrization

$$\hat{Q}(z) \;=\; \sum_{i,j} \sum\nolimits_{k=0}^{N} \frac{\alpha_{ijk}}{z^k} \mathbf{E}_{ij},$$

where $\mathbf{E}_{ij} \triangleq \mathbf{e}_i \mathbf{e}_j^T$. By substituting $\hat{Q}$ for Q, closed-loop map can be written as a function of parameters $\alpha = \{\alpha_{ijk}\}$ as:

$$\begin{aligned} H(\cdot) \;&:\; \mathbb{R}^{n_u n_y (N+1)} \mapsto \mathcal{RH}_\infty^{n_u \times n_y} \\ H(\alpha) \;&\triangleq\; T_1 - T_2 \left(\sum_{i,j} \sum\nolimits_{k=0}^{N} \frac{\alpha_{ijk}}{z^k} \mathbf{E}_{ij} \right) T_3. \end{aligned}$$

We will derive the subgradient vector of $f_\infty(\alpha) \triangleq \|H(\alpha)\|_{\mathcal{H}_\infty}$ at α^0 by first describing it when it is computed on another point α^1, i.e., $f_\infty^{\mathrm{sg}}(\alpha^0, \alpha^1)$, and then will derive the subgradient vector explicitly. Following will achieve the first step:

Theorem 49. *Given T_1, T_2, T_3, a subgradient of $f_\infty(\alpha) = \|H(\alpha)\|_{\mathcal{H}_\infty}$, at α^0 evaluated on α^1 is given by:*

$$f_\infty^{\text{sg}}(\alpha^0, \alpha^1) \;=\; -\Re\left(u_0^* T_2^{\omega_0}\left(\sum_{i,j}\sum\nolimits_{k=0}^{N} \alpha_{ijk}^1 e^{-\boldsymbol{j}k\omega_0}\boldsymbol{E}_{ij}\right) T_3^{\omega_0} v_0\right). \tag{3.11}$$

The equation is illustrated as follows. We first compute the frequency ω_0 at which the $\mathcal{H}_\infty$ norm of $H(\alpha^0)$ is achieved (i.e. $\sigma_{max}\left(H(\alpha^0)(e^{\boldsymbol{j}\omega_0})\right) = \|H(\alpha^0)(e^{\boldsymbol{j}\omega})\|_{\mathcal{H}_\infty}$), then a singular value decomposition of $H(\alpha^0)(e^{\boldsymbol{j}\omega_0})$ would be computed to have $H(\alpha^0)(e^{\boldsymbol{j}\omega_0}) = U_0\Sigma_0 V_0^*$. The first columns of U_0 and V_0 are then extracted and named as u_0 and v_0. Now we can form (3.11), where $T_i^{\omega_0} = T_i(e^{\boldsymbol{j}\omega_0})$, for $i = 1, 2, 3$.

Proof. To prove that (3.11) is a subgradient of f_∞, we must show that it satisfies the subgradient inequality (3.10). We first state the following basic equality that can be derived by direct substitutions in the definitions:

$$\begin{aligned} f_\infty(\alpha^0) \;&=\; \|H(\alpha^0)\|_{\mathcal{H}_\infty} \;=\; \Re\left(u_0^*\left(H(\alpha^0)(e^{\mathbf{j}\omega_0})\right)v_0\right) \\ &\;=\; \Re(u_0^* T_1^{\omega_0} v_0) + f_\infty^{\text{sg}}(\alpha^0, \alpha^0). \end{aligned} \tag{3.12}$$

Next we can see that for all $\alpha^1 \in \mathbb{R}^{n_u n_y (N+1)}$:

$$\begin{aligned} f_\infty(\alpha^1) \;&=\; \operatorname*{ess\ sup}_{\omega, \|u\|=\|v\|=1} \Re\left(u^*\left(H(\alpha^1)(e^{\mathbf{j}\omega})\right)v\right) \\ &\geq\; \Re\left(u_0^*\left(H(\alpha^1)(e^{\mathbf{j}\omega_0})\right)v_0\right) \\ &=\; \Re\left(u_0^* T_1^{\omega_0} v_0\right) + f_\infty^{\text{sg}}(\alpha^0, \alpha^1) \\ &\overset{(3.12)}{=}\; f_\infty(\alpha^0) - f_\infty^{\text{sg}}(\alpha^0, \alpha^0) + f_\infty^{\text{sg}}(\alpha^0, \alpha^1) \\ \Rightarrow f_\infty(\alpha^1) \;&\geq\; f_\infty(\alpha^0) + f_\infty^{\text{sg}}(\alpha^0, \alpha^1) - f_\infty^{\text{sg}}(\alpha^0, \alpha^0), \quad \forall\ \alpha^1. \end{aligned}$$

□

We will now proceed by computing the subgradient vector explicitly, i.e., we will compute the subgradient vector $\phi \in \mathbb{R}^{n_u n_y (N+1)}$ such that $\langle \phi, \alpha^1 \rangle = f_\infty^{\mathrm{sg}}(\alpha^0, \alpha^1)$.

Theorem 50. *Given T_1, T_2, T_3, and α^0, elements of a subgradient vector ϕ for $f_\infty(\alpha^0)$ are given by:*

$$\phi_{ijk} \quad = \quad - \Re \left(u_0^* T_2^{\omega_0} \boldsymbol{E}_{ij} T_3^{\omega_0} v_0 e^{-\mathbf{j} k \omega_0} \right),$$

where $i \in \{1, \cdots, n_u\}$, $j \in \{1, \ldots, n_y\}$, and $k \in \{0, \cdots, N\}$.

Proof. We can write the RHS of (3.11) as:

$$\begin{aligned} (3.11) \; &= \; \langle \phi, \alpha^1 \rangle \\ &= \; - \sum_{i,j} \sum_{k=0}^{N} \alpha_{ijk}^1 \Re \left(u_0^* T_2^{\omega_0} \mathbf{E}_{ij} T_3^{\omega_0} v_0 e^{-\mathbf{j} k \omega_0} \right) \\ \Rightarrow \phi_{ijk} \; &= \; - \Re \left([u_0^* T_2^{\omega_0}]_i [T_3^{\omega_0} v_0]_j e^{-\mathbf{j} k \omega_0} \right), \end{aligned}$$

where the first equality is due to the linearity of the subgradient, the second equality follows as α^1 is a real vector, and the third as (3.11) is valid for all α^1. □

This subgradient vector ϕ is used in implementations of the ellipsoid method in later sections.

3.4 LMI for $\mathcal{H}_\infty$-norm with the Fixed FD Basis

We first review a key result in Section 3.4.1 establishing that finding the $\mathcal{H}_\infty$-optimal static controller for certain plants, including those that admit to a model-matching form through a Q-parametrization, can be cast as a semi-definite program.

We then show in Section 3.4.2 how this result can be used to cast the QI $\mathcal{H}_\infty$-optimal decentralized control problem with a FD basis for the Q-parameter as an SDP.

3.4.1 Static Output Feedback

We review the main result of [36], which will be crucial to have the static parameters of the controller explicitly present in the SDP whose objective will the closed-loop $\mathcal{H}_\infty$-norm and its solution will be the optimal value of these (static) parameters.

Theorem 51. *Consider a generalized discrete-time plant with state-space realization that can be partitioned as follows:*

$$\left[\begin{array}{c:c|c:c} \breve{A}_1 & \breve{A} & \breve{B}_{11} & \breve{B} \\ \hdashline 0 & \breve{A}_2 & \breve{B}_{21} & 0 \\ \hline \breve{C}_{11} & \breve{C}_{12} & \breve{D}_{11} & \breve{D}_{12} \\ \hdashline 0 & \breve{C} & \breve{D}_{21} & 0 \end{array}\right]; \tag{3.13}$$

then, the optimal static output feedback controller K^{static} along with the optimal $\mathcal{H}_\infty$-

norm, can be found by solving the following SDP:

$$\begin{array}{ll} \textit{minimize} & \gamma \\ \textit{subject to} & \begin{bmatrix} \grave{X} & 0 & \grave{A}^T & \grave{C}^T \\ 0 & \gamma I & \grave{B}^T & \grave{D}^T \\ \grave{A} & \grave{B} & \grave{X} & 0 \\ \grave{C} & \grave{D} & 0 & \gamma I \end{bmatrix} \succ 0 \end{array} \tag{3.14}$$

with variables $\gamma \in \mathbb{R}$, and real matrices of appropriate dimension K^{static}, $E = E^T$, S, and $R = R^T$, as well as $\grave{A}$, $\grave{X}$, $\grave{B}$, $\grave{C}$, and $\grave{D}$, which are given by the additional constraints:

$$\begin{aligned} \grave{A} &= \begin{bmatrix} \breve{A}_1 E & \breve{A}_1 S + \breve{A} + \breve{B} K^{\text{static}} \breve{C} - S \breve{A}_2 \\ 0 & R \breve{A}_2 \end{bmatrix}, \\ \grave{B} &= \begin{bmatrix} \breve{B}_{11} + \breve{B} K^{\text{static}} \breve{D}_{21} - S \breve{B}_{21} \\ R \breve{B}_{21} \end{bmatrix}, \\ \grave{C} &= \begin{bmatrix} \breve{C}_{11} E & \breve{C}_{12} + \breve{D}_{12} K^{\text{static}} \breve{C} + \breve{C}_{11} S \end{bmatrix}, \\ \grave{D} &= \breve{D}_{11} + \breve{D}_{12} K^{\text{static}} \breve{D}_{21}, \\ \grave{X} &= \textit{diag}(E, R), \end{aligned} \tag{3.15}$$

each of which is affine in all of the variables.

Proof. See [36]. □

The paper also notes that plants without a 22-block (where the controller inputs to the plant do not affect the measurements which the controller may act on) can be partitioned as in (3.13), and are thus amenable to optimal static feedback with this SDP.

3.4.2 LMI Formulation with Quadratic Invariance

In this subsection we show how the problem of finding the $\mathcal{H}_\infty$-optimal decentralized controller for a QI problem, or the $\mathcal{H}_\infty$-optimal decentralized model-matching problem with a FD basis for the Q-parameter, can be formulated as an SDP.

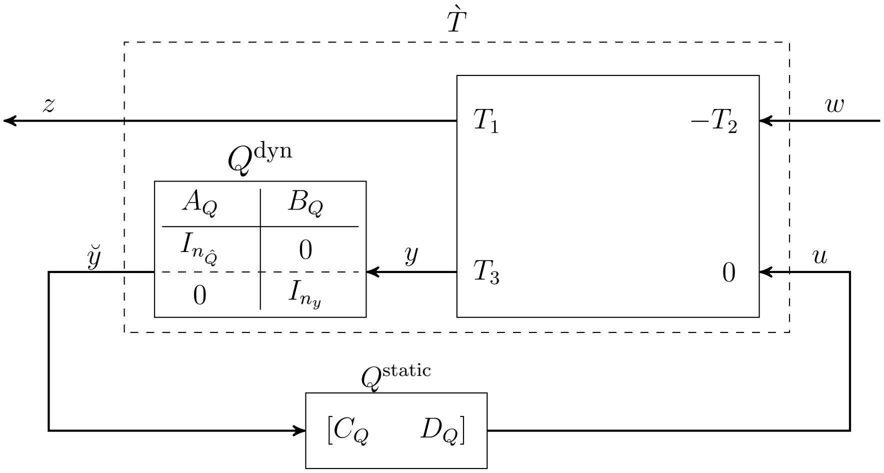


Figure 3.2: $\grave{T}$ defined by augmenting plant with the fixed part of Q

The model-matching problem has (by definition) no 22-block, and can thus be represented as a generalized plant as in (3.13). The FD parametrized Q that we are trying to design for it, was shown to be separable into a fixed dynamic part, which can be represented as:

$$Q^{\text{dyn}} \triangleq \left[\begin{array}{c|c} A_Q & B_Q \\ \hline I_{n_{\hat{Q}}} & 0 \\ 0 & I_{n_y} \end{array}\right],$$

and a variable static part: $Q^{\text{static}} \triangleq [C_Q \;\; D_Q]$.

The fixed dynamic part (Q^{dyn}) can then be considered part of an augmented plant, as illustrated in Figure 3.2. This leaves us to optimize over static controllers (matrices) Q^{static} for the augmented plant $\grave{T}$.

A state-space realization for $\grave{T}$ is given by:

$$\left[\begin{array}{cc:cc|c:c}
A_1 & 0 & 0 & 0 & B_1 & 0 \\
0 & A_2 & 0 & 0 & 0 & B_2 \\
\hdashline
0 & 0 & A_3 & 0 & B_3 & 0 \\
0 & 0 & B_Q C_3 & A_Q & B_Q D_3 & 0 \\
\hline
C_1 & -C_2 & 0 & 0 & D_1 & -D_2 \\
\hdashline
0 & 0 & 0 & I_{n_y N} & 0 & 0 \\
0 & 0 & C_3 & 0 & D_3 & 0
\end{array}\right], \tag{3.16}$$

where (A_i, B_i, C_i, D_i) is a state-space realization of T_i for $i = 1, 2, 3$. This partition still comports with (3.13). We can then apply the results of the previous subsection, and combine this with Lemma 48, to arrive at the key outcome of this section.

Key SDP. The parametrized version of main problem (3.9) using a FD basis for the Q-parameter is solvable by the following SDP:

$$
\begin{aligned}
\text{minimize} \quad & \gamma \\
\text{subject to} \quad & \begin{bmatrix} \grave{X} & 0 & \grave{A}^T & \grave{C}^T \\ 0 & \gamma I & \grave{B}^T & \grave{D}^T \\ \grave{A} & \grave{B} & \grave{X} & 0 \\ \grave{C} & \grave{D} & 0 & \gamma I \end{bmatrix} \succ 0 \\
& C^Q_{ij} = 0 \quad \text{for} \quad (i,j) \text{ s.t. } K^{\text{bin}}_{ij} = 0 \\
& D^Q_{ij} = 0 \quad \text{for} \quad (i,j) \text{ s.t. } K^{\text{bin}}_{ij} = 0,
\end{aligned}
\tag{3.17}
$$

with variables $\gamma \in \mathbb{R}$, and real matrices of appropriate dimension $Q^{\text{static}} = \begin{bmatrix} C_Q & D_Q \end{bmatrix}$, $E = E^T$, S, and $R = R^T$, as well as $\grave{A}$, $\grave{X}$, $\grave{B}$, $\grave{C}$, and $\grave{D}$ which are given by the additional affine constraints (3.15), where in each constraint, the $\breve{(\cdot)}$ constants would be obtained by matching (3.13) to (3.16), and the variable K^{static} is replaced by Q^{static}.

Proof. Finding a Q with fixed FD basis for problem (3.2) is equivalent to finding a static output feedback controller for $\grave{T}$ (see Figure 3.2), for which matrices for the generalized plant (3.13) are given by the specified partitioning in (3.16). The $\mathcal{H}_\infty$-optimal static output feedback controller for this is then given by SDP (3.14), and the sparsity-induced information structure $\hat{Q} \in \mathcal{S}$ could then be enforced as (3.8) due to Lemma 48, which results in (3.17). □

After obtaining the optimal $Q^{\text{static}} = \begin{bmatrix} C_Q & D_Q \end{bmatrix}$, we can recover the optimal $\hat{Q}$ as in (3.7). Then if we take $Q = \hat{Q}$, we can recover the controller K by [6, Theorem 17].

Remark 52. *Through the rest of this chapter, γ denotes the optimal closed-loop $\mathcal{H}_\infty$-norm obtained from the SDP* (3.17)*.*

Remark 53. *This SDP allows for having different poles for each column of $\hat{Q}(z)$ by properly adjusting the matrices A_Q and B_Q, as long as they are constructed as in Remark 45 and Remark 46.*

Remark 54. *We can view the decentralized $\mathcal{H}_\infty$-optimal control design subject to a QI sparsity pattern (or equivalently the decentralized model-matching problem) parametrized by a FD basis as a two-phase problem. The first phase is choosing A_Q and B_Q as in* (3.4) *and* (3.5)*, named as **dictionary selection**, and the second phase is solving the SDP* (3.17) *for the **optimal coefficients** C_Q and D_Q. An admissible B_Q serves as a normalization factor and is irrelevant to the achievable optimality level γ (the same $\hat{Q}_{ij}$ can be obtained by scaling B_j^Q and C_{ij}^Q appropriately. This means that the dictionary selection reduces to choosing an appropriate A_Q.*

We now provide an example to illustrate the FIR Q-parameter. We use a discretized version of the same plant which was used in [6], along with the same sequence of sparsity-induced information structures.

Example 55. *Consider an unstable lower triangular plant*

$$G(z) = \begin{bmatrix} s(z) & 0 & 0 & 0 & 0 \\ s(z) & u(z) & 0 & 0 & 0 \\ s(z) & u(z) & s(z) & 0 & 0 \\ s(z) & u(z) & s(z) & s(z) & 0 \\ s(z) & u(z) & s(z) & s(z) & u(z) \end{bmatrix}$$

with $s(z) = \frac{0.1}{z-0.5}$, $u(z) = \frac{1}{z-2}$, *and* P *given by:*

$$P_{11} = \begin{bmatrix} G & 0 \\ 0 & 0 \end{bmatrix} \qquad P_{12} = \begin{bmatrix} G \\ I \end{bmatrix} \qquad P_{21} = \begin{bmatrix} G & I \end{bmatrix},$$

and a sequence of sparsity constraints $K_1^{\text{bin}}, \ldots, K_6^{\text{bin}}$*:*

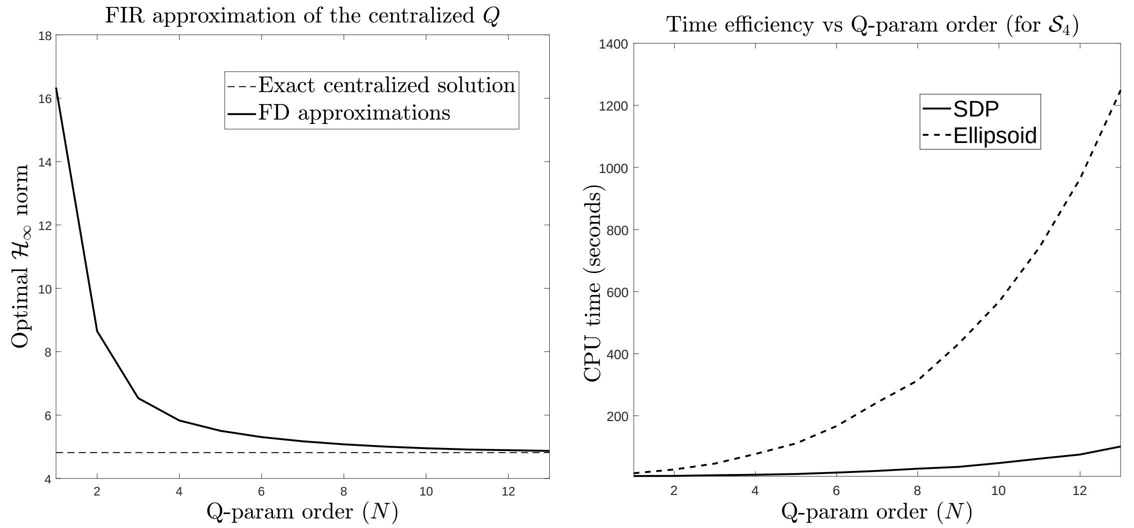
$$K_1^{\text{bin}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix} \qquad K_2^{\text{bin}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

$$K_3^{\text{bin}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix} \qquad K_4^{\text{bin}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

$$K_5^{\text{bin}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 \end{bmatrix} \qquad K_6^{\text{bin}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix},$$

defining a sequence of sparsity-induced information structures $\mathcal{S}_i = \text{Sparse}(K_i^{\text{bin}})$ *such that each subsequent constraint is less restrictive, and such that each is quadratically invariant under* G*. We also use* $\mathcal{S}_7$ *as the set of controllers with no sparsity constraints; i.e., the centralized case.*

First, we apply SDP (3.17) *to the centralized problem, where we can compute the optimal solution with existing methods. This serves as a sanity check to ensure that we get convergence to the optimum, and to explore how the parametrized solutions converge as the order grows.*



(a) Optimal norm versus N for centralized controller

(b) Time efficiency of different methods for decentralized case $\mathcal{S}_4$

Figure 3.3: FIR approximation of the Q-parameter

Figure 3.3(a) plots the optimal $\mathcal{H}_\infty$*-norm obtained by solving our SDP* (3.17) *for a centralized sparsity pattern (i.e., no sparsity constraints), as the order of the FD parametrization N increases, and is solved using the* `cvx` *toolbox [26] for* `MATLAB`*. This is meant to serve primarily as a sanity check, and to provide some initial indication of the satisfactory order of the FD parametrization, before moving on to the decentralized problems of interest. It shows that, as expected, as N increases from 1 to 13, the optimal $\mathcal{H}_\infty$-norm decreases and converges toward the actual solution, indicated by the dashed line, which was obtained using* `MATLAB`*'s internal function* `hinfsyn`*. The plant, and thus the actual optimal controller, are of order 5, and we observe close convergence after increasing the order slightly beyond that.*

We then apply our results to the decentralized problems. Figure 3.3(b) shows how the SDP method and the ellipsoid method that uses the computed subgradient compare for the information constraint $\mathcal{S}_4$, as the order N varies again from 1 to 13. The SDP (3.17) is solved first, and the time it takes is shown with the solid line. The ellipsoid method is then used, with its stopping criterion chosen based on the optimal value of the SDP f^*_{SDP}, setting $\frac{f^{best}_{ellip}-f^*_{SDP}}{f^*_{SDP}} < 0.1$, such that we stop when we have an optimal point that is at least within 10% of the SDP solution. CPU time is reported from a machine with 2.3GHz CPU, and 8GB of RAM. We see that the ellipsoid method takes much more time, and this pattern was consistent over all of the information structures, though it diverged more slowly for some others.

We then turn our attention to computing and comparing the $\mathcal{H}_\infty$-optimal solutions for the sequence of sparsity constraints. The results are presented in Figure 3.4 and are computed by solving the SDP (3.17) for K_i^{bin}, for $i = 1, \cdots, 6$, and with $i = 7$ representing a centralized controller. In each of these cases, N is fixed at 13.

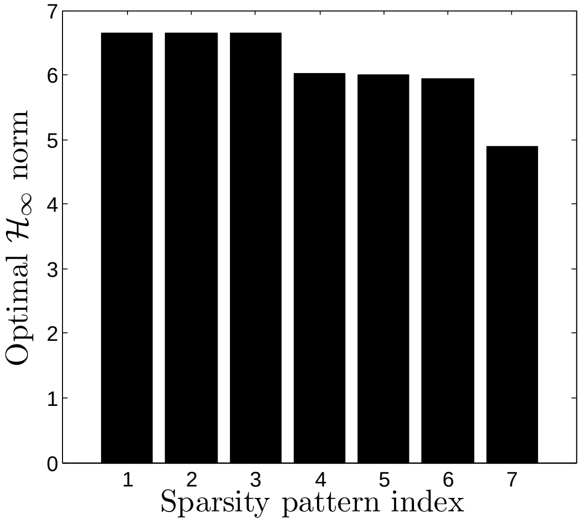


Figure 3.4: Optimal norm for different sparsity patterns

This shows that (as expected) as we relax the information constraint, the optimal norm would also be non-increasing, since $\mathcal{S}_i \subset \mathcal{S}_j$ for $i < j$.

Comparing these results with those for which the $\mathcal{H}_2$-norm was instead optimized for the same plant and sparsity patterns in [6], we similarly see that the first significant drop occurs with the relaxation from $\mathcal{S}_3$ to $\mathcal{S}_4$; i.e., by allowing the fifth controller to access the third measurement. We dissimilarly see that the relaxations from $\mathcal{S}_5$ to $\mathcal{S}_6$ no longer produce a noticeable change in performance.

3.5 Pole Selection Methods

We will now address the dictionary selection phase, and will discuss and develop methods for choosing better poles for the columns of $\hat{Q}$, rather than simply using the FIR Q-parameter. In each method, we will construct $(A_Q,\ B_Q)$ first, and then with those fixed, we will find coefficients $(C_Q,\ D_Q)$ by solving the

SDP (3.17). We will study various schemes for pole selection and compare them through numerical experiments. We first inspect pole selection based on the centralized solution of (3.2) in Section 3.5.1, either by using the full centralized solution in Method 1 (Cent-All), or a reduced set based on the centralized solution in Method 2 (Cent-Red). In Section 3.5.2, we populate the dictionary and then use a variant of the l_1 heuristic to choose a small number of effective poles for each column from a large set, leading to Method 3 (Sparse). Finally in Section 3.5.3, we will derive Method 4 (Taylor) using Taylor approximation to linearize the performance about the current pole locations, enabling us to systematically adjust the pole locations along with their coefficients.

3.5.1 Dictionary Selection Based on Centralized Solution

The optimal centralized solution (K^{cent}) to (3.1) (i.e., without the $K \in \mathcal{S}$ constraint) can be found by several standard methods [27, 28]. One naïve method for obtaining a decentralized controller is to then force the non-admissible elements of K^{cent} to be zero. However, this solution might not even be stabilizing. Whenever QI holds, and thus the equivalent formulation (3.2) is possible, we can first obtain the centralized solution to (3.2), denoted by Q^{cent}, and then force its non-admissible elements to be zero. This approach is at least guaranteed to result in a feasible solution (a stabilizing controller that satisfies the information constraint), and could also be further improved in performance as discussed below.

We will first obtain the $\mathcal{H}_\infty$-optimal centralized model-matching solution Q^{cent},

	P^1	P^2
n_z and n_w	10	8
n_u and n_y	5	4
G	unstable	stable
Order of G	5	16
Order of T	21	48
Order of Q^{cent}	21	48
$\gamma^{\text{cent}} = \|T_1 - T_2 Q^{\text{cent}} T_3\|_{\mathcal{H}_\infty}$	4.816	4.816

Table 3.1: Summaries of the two sample plants

and then use its poles to construct a parametrization for the optimal decentralized parameter which we seek. Denoting the order of Q^{cent} by n_{cent}, this method is then outlined as below:

Method 1 (Cent-All)**.**

1. *Use poles of Q^{cent} to construct the same block-diagonal $\{A_j^Q\}_{j=1}^{n_y}$, with $n_j = n_{\text{cent}}$.*
2. *Construct B_j^Q, A_Q, and B_Q as in Remark 45 for the real poles, or as in Remark 46 for the complex poles.*
3. *After this dictionary selection stage, use the SDP (3.17) to solve for the optimal coefficients.*

Example 56. *We test our methods and compare them based on two sample plants. The first one, denoted as P^1, is the unstable plant from Example 55 along with the choice of $\mathcal{S} = \text{Sparse}(K_4^{\text{bin}})$. The other plant, denoted as P^2, is a stable randomly generated plant. We first fix $n_y = n_u = 4$, and then randomly generate a stable $A_G^0 \in \mathbb{R}^{4\times 4}$, generate B_G^0, C_G^0, and fix $D_G^0 = 0$. Then, the resulting G^0 is projected onto* *a lower triangular information structure that gives G. The information structure of the controller for this plant corresponds to the following binary matrix:*

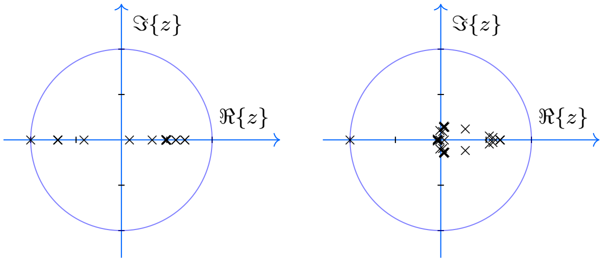
$$K^{\text{bin}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix},$$

and we choose T as:

$$T_1 = \alpha \begin{bmatrix} G & 0 \\ 0 & 0 \end{bmatrix}, \quad T_2 = \alpha \begin{bmatrix} G \\ I \end{bmatrix}, \quad T_3 = \alpha \begin{bmatrix} G & I \end{bmatrix},$$

where α is a scalar chosen to make the optimal centralized closed-loop $\mathcal{H}_\infty$-norm equal for the two plants. Summaries of these two plants are given in Table 3.1.

Following Method 1 (Cent-All), we will choose A_j^Q for each of the plants P^1 and P^2 to match the poles of their corresponding optimal centralized solutions Q^{cent}, which are depicted in Figure 3.5. All of the poles are stable with $\max|\lambda| = 0.99$.



(a) Poles of P^1 (b) Poles of P^2

Figure 3.5: Location of Q^{cent} poles

By applying the Cent-All method to P^1, the SDP solver was able to reach $\gamma =$

6.152 [1], *whereas direct projection of* Q^{cent} *onto* $\mathcal{S}$ *(denoted by* $Q_{\mathcal{S}}$*) would result in* $\|T_1 - T_2 Q_{\mathcal{S}} T_3\|_{\mathcal{H}_\infty} = 278.539$. *The corresponding SDP has 45,130 variables, and it takes approximately 160 minutes to reach this level of optimality on the prescribed PC. For* P^2*, the SDP becomes extensively large (152,177 variables), and thus impractical to solve. Direct projection for* P^2 *will result in* $\|T_1 - T_2 Q_{\mathcal{S}} T_3\|_{\mathcal{H}_\infty} = 12.654$.

The resulting controller $\hat{Q}$ would then have order $n_{\hat{Q}} = n_y n_{\text{cent}}$, which can grow large easily. The centralized $\mathcal{H}_\infty$-controller results in a solution that has the same order as the generalized plant for which it is designed [27, p. 15], i.e., $n_{\text{cent}} = n_T$. This term can be much larger than the order of the original plant, depending on how the model-matching problem (3.2) is derived from the closed-loop problem (3.1).

This motivates us to reduce the dictionary size by considering the most significant poles in each column of the projected Q^{cent}. This method is outlined below:

Method 2 (Cent-Red)**.**

1. *Project* Q^{cent} *onto* $\mathcal{S}$ *by making the non-admissible elements zero, and name it as* $Q_{\mathcal{S}}$.

2. *Use a typical order reduction method on each column of* $Q_{\mathcal{S}}$ *to obtain the* N *most significant poles in each column, and denote the resulting set of poles for each column by* V_j *for* $j = 1, \cdots, n_y$.

3. *Assemble possibly different* $A_j^Q = \operatorname{diag}(V_j)$*, and utilize Remark 45 to construct*

[1] We use the `cvx` package to solve these optimization problems. However, if the problem size becomes large, SDP solvers used in `cvx` (`SDPT3 4.0`, and `Mosek`) fail to reach the solution. We report values from the last step when the solver terminates, and thus it is possible that reported data will be near the solution (depending on performance of solver), but not within pre-specified precision. To keep timing data comparable, we will use `SDPT3 4.0` for all optimizations.

B_j^Q, A_Q, and B_Q. Use Remark 46 for complex poles.

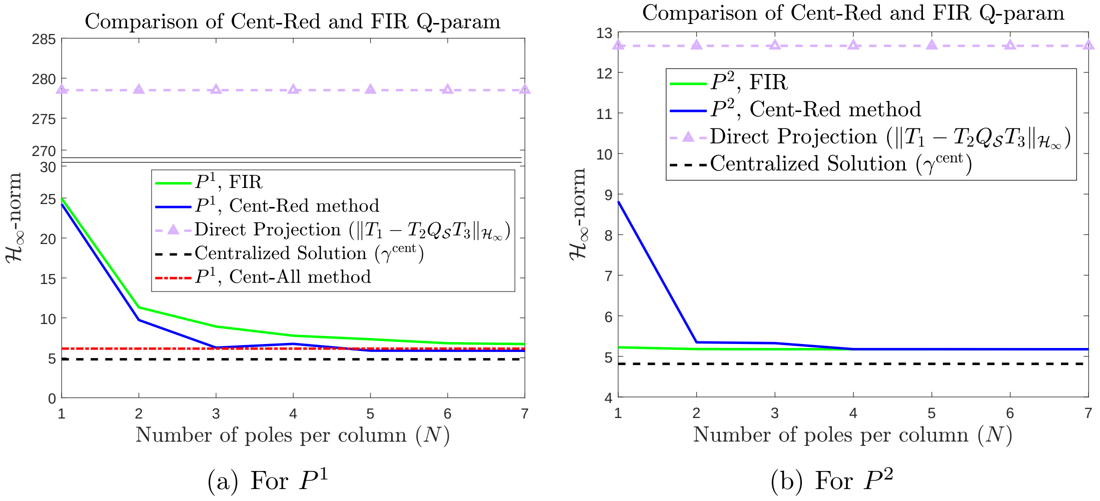
4. *After this dictionary selection stage, use the SDP (3.17) to solve for the optimal coefficients.*

Numerical results for order reduction of different sizes are reflected below.

Example 57. *We continue with Example 56, and apply Method 2 (Cent-Red) on P^1 and P^2. Order reduction on each column is done by Balanced Stochastic model Truncation (BST) via the Schur method (`bstmr` in `MATLAB`). The resulting closed-loop $\mathcal{H}_\infty$-norm as we increase the number of poles in each column (N) is illustrated in Figure 3.6. The closed-loop $\mathcal{H}_\infty$-norm associated with the direct projection for P^1 was about 278.5 and thus we have shown it in Figure 3.6(a) with a jump in the y-axis. For comparison purposes, related data for FIR approximations with increasing N (as in Section 3.3 and Section 3.4.2) is also provided in Figure 3.6 and shown in green.*

We see that for P^1, Cent-Red (shown in blue) slightly but consistently outperformed the FIR approximation over different numbers of poles, while taking more time. We see a slight increase at $N = 4$, which is possible since the selections made when smaller numbers of poles are allowed can end up being better when used in the decentralized closed-loop. It is similarly possible for the Cent-Red method to outperform Cent-All (dashed red), as it does slightly for $N \geq 5$ for P^1, since the model reduction technique can choose poles which were not present in the larger controller. For P^2, we see that a low-order FIR approximation is hardly improved upon, and Cent-Red catches up once it has 2 to 4 poles per column.

This order reduction on $Q_{\mathcal{S}}$ is a fairly fast method for choosing the dictionary,



(a) For P^1

(b) For P^2

Figure 3.6: Applying Method 2 (Cent-Red) in Example 57: Closed-loop $\mathcal{H}_\infty$-norm versus number of poles per column, N

however, it is selected solely based on the contribution of the poles in columns of $Q_\mathcal{S}$. This makes the considered order reduction technique an open-loop approach that cannot directly account for the relative importance of the poles when $\hat{Q}$ is closed around T. This motivates us to consider a framework which selects the poles based on their effect in the closed-loop $\mathcal{H}_\infty$-norm, which will be the topic of the next section.

3.5.2 Sparsity Promoting Framework

In this section we will implement a variant of the l_1 regularization to help us choose a small dictionary while keeping the closed-loop $\mathcal{H}_\infty$-norm small. The sparsity inducing heuristic here would be a convex penalty (on C_Q) added to the objective of the SDP (3.17). This heuristic is adopted from [40] (also known as the *relaxed simultaneous sparse approximation*), and provides a convex relaxation for deciding which poles could be eliminated from an initial choice of poles without significantly increasing the closed-loop $\mathcal{H}_\infty$-norm.

We first outline the aforementioned approach below, and describe each step in more detail afterwards. We will then inspect its performance through the numerical examples from the previous sections.

Method 3 (Sparse)**.**

1. *Add initial choice of poles for the controller to the dictionary, i.e., choose* A_j^Q*, for* $j = 1, \cdots, n_y$*, to be block-diagonal matrices with their eigenvalues inside the open unit disk* $\mathbb{D}$*.*

2. *Construct normalized* B_j^Q *(as described below).*

3. *Construct block-diagonal* A_Q *and* B_Q *as in* (3.5)*.*

4. *Solve SDP* (3.17) *with the following convex penalty on the sparsity of* C_Q*:*

$$\begin{aligned} &\mathit{min} \quad \gamma + \lambda \|C_Q\|_{\mathrm{r\tilde{x}}} \\ &\mathit{s.t.} \quad \textit{remaining conditions of } (3.17), \end{aligned} \tag{3.18}$$

where $\|\cdot\|_{\mathrm{r\tilde{x}}}$ *is defined below.*

5. *For each* $j = 1, \cdots, n_y$*, eliminate those poles of* A_j^Q *that have zero corresponding coefficients in* C_Q*.*

6. *Construct the new* A_j^Q*,* B_j^Q*,* A_Q*, and* B_Q*, and solve the SDP* (3.17) *again for the refined solution.*

The initial choice of poles in step 1 can be selected in many ways. It could be arbitrary finite sets in $\mathbb{D}$ with each set representing the initial choice for each column of $\hat{Q}$. For example, it could be a union of the most effective poles in each column of $Q_{\mathcal{S}}$ and some other poles in $\mathbb{D}$. These poles could be either real with degree one (resulting in a single 1×1 element in the diagonal of A_j^Q), or in complex conjugate pairs (resulting in 2×2 blocks as in Remark 46).

Remark 58. *Each real pole with degree one has all of its corresponding coefficients in a single column of C_Q, and poles in complex conjugate pairs in two consecutive columns. This holds due to A_Q and $\{A_j^Q\}_{j=1}^{n_y}$ all being block-diagonal.*

Elements of the dictionary must be normalized to have the same weight so that the sparsity-promoting regularizer functions properly choose them based on their contribution. By choosing $B_j^Q = \mathbf{1}_{n_j}$, each pole would appear as $\frac{1}{z-p}$. We adopt the normalization suggested in [47], which results in the poles appearing as $\frac{1-|p|^2}{z-p}$. To achieve this normalization, if an element (k,l) of B_Q is not zero by construction (i.e, to enforce block-diagonal structure), it would be normalized by multiplying by $1 - \sum_j |A_Q(k,j)|^2$.

The $\tilde{\text{rx}}$-norm in step 4 is a modified version of the simultaneous sparsity inducing norm (rx-norm) in row format in [40]. This norm could be stated in column format to promote sparsity for each column for a matrix $M \in \mathbb{R}^{m \times n}$ as:

$$\|M\|_{\text{rx}} \triangleq \sum_{j=1}^{n} \max_{i=1,\cdots,m} |M_{ij}|.$$

We want to apply this penalty on C_Q, and the intuition as mentioned in [40] is that if we are going to keep a pole in the dictionary, we want that pole to contribute to as many admissible elements in $\hat{Q}$ as possible. In other words, most of columns of C_Q should be zero, but the non-zero columns should have as many admissible non-zero elements as possible. To do so, we apply the l_∞-norm on each column to promote non-sparsity among admissible elements, and then apply the l_1-norm to promote sparsity on the resulting vector.

The $\|\cdot\|_{\mathrm{rx}}$ does not account for poles in complex conjugate pairs properly. We can not eliminate one of the complex poles but not its conjugate, and thus these poles could be eliminated, if the two corresponding consecutive columns in C_Q would be simultaneously zero. To properly adopt this norm for complex poles, we will apply the l_∞-norm on the coefficients corresponding to conjugate pairs simultaneously. To illustrate this, divide indices of columns of C_Q into two sets. Denote the one that corresponds to real poles as $J_\Re$, and the other one that corresponds to beginning index of consecutive complex pairs as $J_\Im$. This separation of indices is illustrated in the following example.

Example 59. *Let $\mathcal{S}$ be such that $K^{\mathrm{bin}} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$. If we choose the dictionary as $A_1^Q = \mathrm{diag}\left(0.1, 0.2, \begin{bmatrix} 0.3 & 0.4 \\ -0.4 & 0.3 \end{bmatrix}\right)$, and $A_2^Q = \mathrm{diag}\left(-0.6, \begin{bmatrix} 0 & 0.5 \\ -0.5 & 0 \end{bmatrix}\right)$, then the normalized B_j^Q are given by $B_1^Q = \begin{bmatrix} 0.99 & 0.96 & 1.5 & 0 \end{bmatrix}^T$, and $B_2^Q = \begin{bmatrix} 0.64 & 1.5 & 0 \end{bmatrix}^T$; moreover, $n_1 = 4$, $n_2 = 3$, $n_{\hat{Q}} = n_1 + n_2 = 7$, and $C_Q \in \mathbb{R}^{2\times 7}$. We also have $J_\Re = \{1, 2, 5\}$ and $J_\Im = \{3, 6\}$.*

Define the modified $\|\cdot\|_{\tilde{\mathrm{rx}}}$-norm as:

$$\|C_Q\|_{\tilde{\mathrm{rx}}} \triangleq \sum_{j\in J_{\Re}} \max_{i=1,\cdots,n_u} |(C_Q)_{ij}| \\ + \sum_{j\in J_{\Im}} \max_{l\in\{j,j+1\}} \max_{i\in\{1,\cdots,n_u\}} |(C_Q)_{il}|.$$

The aforementioned modification will promote sparsity on the two consecutive columns that correspond to a complex conjugate pair simultaneously, thus keeping them zero, or non-zero at the same time. We can then solve step 4 with λ being a regularization factor that makes γ comparable to the $\tilde{\mathrm{rx}}$-norm. After identifying the appropriate set of poles for a smaller dictionary in step 4, we drop the unnecessary ones in step 5, and then solve the smaller problem for the refined solution in step 6.

We now demonstrate pole selection by the Sparse method (Method 3) in the following example.

Example 60. *Continuing with Example 57, we apply the Method 3 (Sparse) on P^1 and P^2. Matrices A_j^Q are constructed by combination of the 4 most significant poles in each column of $Q_{\mathcal{S}}$ (obtained via BST as in Example 57) and 4 poles located at $\{\pm 0.25, \pm 0.75\}$, thus giving 8 poles per column for both P^1 and P^2. We vary λ and trace its behavior on the optimality level γ, McMillan degree of $\hat{Q}$ (denoted as $n_{\hat{Q}}$), and the time required for computation of this method in Table 3.2.*

One possible reason behind the subtle counter-intuitive decrease in the second row of Table 3.2 that arises from the solver ability is due to the fine-tunning Step 5 in the Sparse method that would eliminate some poles, and hence results in a much smaller SDP that is more numerically well-behaved in the final step of that method.

P^1				P^2			
λ	γ	$n_{\hat{Q}}$	t [sec]	λ	γ	$n_{\hat{Q}}$	t [sec]
0.000	5.932	32	126	0.00	5.178	32	217
0.005	5.919	28	183	0.01	5.176	23	255
0.050	5.925	23	158	0.10	5.176	22	360
0.500	5.932	16	255	1.00	5.176	17	1155

Table 3.2: Applying Method 3 (Sparse) in Example 60

Nevertheless, we see that as we increase λ, we have almost the same closed-loop $\mathcal{H}_\infty$-norm, but for a controller with fewer states. This is one powerful aspect of this method which chooses poles based on their importance in the objective.

The sparsity promoting framework in this section would recover a low-order controller. Determining which poles the sparsity regularizer would choose to be the most effective ones among a large-set of them would however demand solving a large optimization problem thoroughly. We study an alternative approach in the next section that enables us to improve locations of the poles sequentially, where each step entail a small optimization problem, and we can stop at any step upon achieving a satisfactory objective value.

3.5.3 Dictionary Learning Based on Taylor Approximation

We want to improve the controller pole locations for our model-matching problem in a systematic way in this section. We will use a sequential optimization framework that allows us to start from any initial choice for the pole location and adjust them using small optimization problems in each step until we achieve a satisfactory objective value. In each step, we will perturb the controller poles by a small amount and linearize the resulting transfer function of the controller around these perturbations. This approximation makes the convex synthesis of the optimal perturbations feasible. The suggested approach is based on the linearization of the factorized matrices in dictionary learning frameworks [48], and we will adopt it to our problem.

We will first sketch this approximation for a SISO controller with a single pole, and then state the general algorithm. Assume that the controller transfer function is given as $\frac{c}{z-p}$, where the optimal c has been found using SDP (3.17). We will perturb the pole location by δ_p, which would also result in the change δ_c in its respective optimal coefficient. The resulting transfer function would then be $\frac{c+\delta_c}{z-(p+\delta_p)}$ with variables δ_p and δ_c. Solving for these two perturbations when ones uses this controller in our model-matching problem would result in a non-convex problem. Thus, we linearize the controller transfer function around these perturbations and use the following first order approximation:

$$\begin{aligned} \frac{c+\delta_c}{z-(p+\delta_p)} &\approxeq \frac{c}{z-p} + \frac{1}{z-p}\delta_c + \frac{c}{(z-p)^2}\delta_p \\ &= \frac{c+\delta_c}{z-p} + \frac{c}{(z-p)^2}\delta_p, \end{aligned} \tag{3.19}$$

with variables δ_p and δ_c. We will embed this perturbed controller in the closed-loop, which can then be re-written in a way that (similar to our initial FD parametrization of the Q-param, Figure 3.2) would have all of its variables (δ_c and δ_p) in its static part. This would allow us to use an SDP similar to (3.17) to solve for the perturbations. To see this, observe that if we augment A and C to be

$$A = \begin{bmatrix} p & 0 \\ 1 & p \end{bmatrix}, \qquad C = \begin{bmatrix} c + \delta_c & c\delta_p \end{bmatrix},$$

we would have:

$$C(zI - A)^{-1} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{c + \delta_c}{z - p} + \frac{c}{(z-p)^2}\delta_p,$$

which means that all the variables would be in the augmented C, and thus we can accordingly use SDP (3.17) with proper modifications to solve for them. The linearization of (3.19) is valid for small enough δ_p, and thus a bound should be placed on $|\delta_p|$. We will then update the location of the pole p as $p \leftarrow p + \delta_p$, and solve (3.17) for the new coefficients (C_Q, D_Q).

The resulting method in the general case for a modal parametrization is outlined below, and a detailed description will follow after.

Method 4 (Taylor)**.**

1. *Let step number be denoted by $k \leftarrow 0$, choose the initial set of poles per column of $\hat{Q}$, and construct corresponding block-diagonal $\{(A_j^Q)^{(0)}\}_{j=1}^{n_y}$, $\{(B_j^Q)^{(0)}\}_{j=1}^{n_y}$, ${A_Q}^{(0)}$, and ${B_Q}^{(0)}$.*

2. *Solve the SDP (3.17) with $A_Q \leftarrow {A_Q}^{(k)}$, $B_Q \leftarrow {B_Q}^{(k)}$, and name the obtained solutions as $\gamma^{(k)}$, ${C_Q}^{(k)}$, ${D_Q}^{(k)}$.*

3. *If $\gamma^{(k)}$ is satisfactory, or a local minima has been achieved, then terminate; else continue.*

4. *Solve the following optimization problem:*

$$
\begin{aligned}
\mathit{min} \quad & \gamma && (3.20)\\
\mathit{s.t.} \quad & A_Q \leftarrow \begin{bmatrix} {A_Q}^{(k)} & 0 \\ I & {A_Q}^{(k)} \end{bmatrix}, \quad B_Q \leftarrow \begin{bmatrix} {B_Q}^{(k)} \\ 0 \end{bmatrix}, \\
& Q^{\text{static}} \quad = \begin{bmatrix} C_Q & {C_Q}^{(k)}\Delta & D_Q \end{bmatrix}, \\
& \|\Delta\| < \epsilon, \\
& \Delta_{ij} \quad = 0, \quad \text{if } i \in J_{\Re} \ \text{ and } j \neq i, \\
& \Delta_{ij} \quad = 0, \quad \text{if } i \in J_{\Im} \ \text{ and } j \notin \{i, i+1\}, \\
& \Delta_{i+1,j} \quad = 0, \quad \text{if } i \in J_{\Im} \ \text{ and } j \notin \{i, i+1\}, \\
& \Delta_{ii} \quad = \quad \Delta_{i+1,i+1} \quad \text{for all } \ i \in J_{\Im} \\
& \Delta_{i,i+1} \quad = -\Delta_{i+1,i} \quad \text{for all } \ i \in J_{\Im} \\
& \textit{remaining conditions of } (3.17),
\end{aligned}
$$

with all the variables of (3.17)*, and an additional variable* $\Delta \in \mathbb{R}^{n_{\hat{Q}} \times n_{\hat{Q}}}$ *which is described in detail below.*

5. *Update the poles as* ${A_Q}^{(k+1)} \leftarrow {A_Q}^{(k)} + \Delta$*.*

6. *Let* $k \leftarrow k+1$*, and go to step* 2*.*

In step 1, complex conjugate pairs in A_j^Q, and their respective parts in B_j^Q must be handled as in Remark 46, which is illustrated in more detail later in this section.

Remark 61. *Our framework easily allows us to consider transition of complex and real poles into one another, i.e., if a complex conjugate pair becomes very close to the real axis in the current iteration, we can consider them to be real poles in the next iteration by properly updating the sets $J_{\Re}$ and $J_{\Im}$. Likewise, we can allow two real poles to become complex conjugate in the next iteration if they become very close to each other in the current iteration.*

It is noteworthy to mention that the dimension of A_Q, B_Q, Q^{static}, and the auxiliary variables changes from step 2 to 4 (and vice-versa). Also, the Q^{static} in (3.20) replaces the one given in SDP (3.17).

Variables C_Q and Δ in step 4 correspond to $c + \delta_c$ and δ_p respectively in the SISO case (3.19). Variable Δ would then gather the corresponding perturbation to the poles of the multi-modal MIMO parametrization, and is a block-diagonal matrix whose sparsity pattern is same as A_Q. We will illustrate this by the following example:

Example 62. *For the choice of matrices in Example 59, $\Delta \in \mathbb{R}^{7\times 7}$ is given as:*

$$\Delta = \operatorname{diag}\left(\delta_1, \delta_2, \begin{bmatrix} \delta_3 & \delta_4 \\ -\delta_4 & \delta_3 \end{bmatrix}, \delta_5, \begin{bmatrix} \delta_6 & \delta_7 \\ -\delta_7 & \delta_6 \end{bmatrix}\right).$$

The last 5 equality constraints on Δ in (3.20) precisely specify such structure for a general case. This structure indicates that for a real pole p, perturbation $\delta \in \mathbb{R}$ represent the corresponding adjustment on the real line, and appears as a 1×1 variable in the diagonal of Δ in the same position as p. Similarly, for a complex conjugate

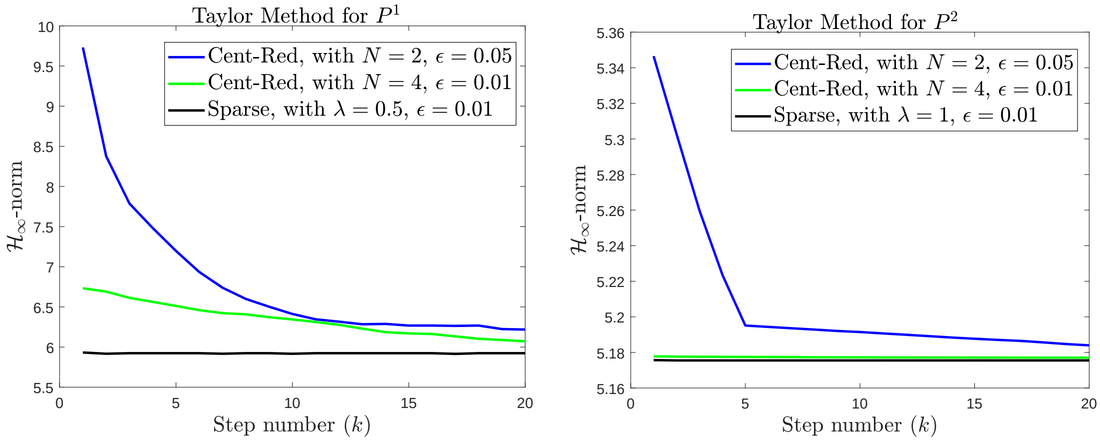
	${A_Q}^{(0)}$	ϵ	$\gamma^{(0)}$	$\gamma^{(\text{best})}$
P^1	Ex. 57, $N = 2$	ϵ_2	9.731	6.218
	Ex. 57, $N = 4$	ϵ_1	6.732	6.072
	Ex. 60, $\lambda\ = 0.5$	ϵ_1	5.932	5.915
P^2	Ex. 57, $N = 2$	ϵ_2	5.347	5.184
	Ex. 57, $N = 4$	ϵ_1	5.178	5.177
	Ex. 60, $\lambda\ = 1$	ϵ_1	5.176	5.176

Table 3.3: The source of initial set of poles for Taylor approximation in Example 63

pair $p_{\Re} \pm \mathbf{j}p_{\Im}$, two conjugate directions $\delta_{\Re} \pm \mathbf{j}\delta_{\Im}$ would specify their adjustment in the complex plane, and would appear in the block diagonal of Δ as $\begin{bmatrix} \delta_{\Re} & \delta_{\Im} \\ -\delta_{\Im} & \delta_{\Re} \end{bmatrix}$, in the same position as $\begin{bmatrix} p_{\Re} & p_{\Im} \\ -p_{\Im} & p_{\Re} \end{bmatrix}$. Thus, $\Delta \in \mathbb{R}^{n_{\hat{Q}} \times n_{\hat{Q}}}$ is a block-diagonal matrix that matches the sparsity and symmetry pattern of A_Q.

We will now inspect the performance of the Taylor method (Method 4) with the following example.

Example 63. *We continue with Examples 57 and 60, and apply the Taylor method with the initial choice of poles picked from particular instances of them. We will bound Δ by $|\Delta_{ij}| < \epsilon$, and will use either $\epsilon_1 = 0.01$, or $\epsilon_2 = 0.05$. Table 3.3 reflects the related data, in which ${A_Q}^{(0)}$ is the initial choice of poles taken from the mentioned examples, $\gamma^{(0)}$ is its corresponding closed-loop $\mathcal{H}_\infty$-norm, and $\gamma^{(\text{best})}$ is the best result in the first 20 iterations. It is notable that $\gamma^{(\text{best})}$ does not necessarily come from the last iteration since for some iterations the linearization might not result in an accurate approximation. Figure 3.7 illustrates the first 20 steps of the Taylor method for aforementioned settings in Table 3.3. It is seen that by adjusting*



(a) Taylor method for P^1 (b) Taylor method for P^2

Figure 3.7: Applying Taylor approx. when the poles are initialized as in Table 3.3

the pole locations properly using the Taylor method, a low-order controller with 2 poles per column has achieved almost the same performance as one with 4 poles per column for both P^1 and P^2.

Remark 64. *One important merit of this dictionary learning approach is that after each step a stabilizing controller $\hat{Q}^{(k)}$ is achieved that has improved the closed-loop $\mathcal{H}_\infty$-norm, whereas in Section 3.5.2, the same level of γ can only be achieved after solving a very big SDP. Moreover, this convex formulation of perturbation directions allows for embedding it in more complex objectives, as for those in multi-objective problems.*

Chapter 4: Minimization of a Particular Singular Value

Here we consider the problem of minimization of the k-singular value of a matrix variable in this chapter. This problems becomes convex only when minimizing the largest singular value ($k = 1$), while for all the other cases ($k \geq 2$) the problem is neither convex nor concave, and could be NP-hard in general.

When one wishes to obtain low-rank solutions, the convex heuristic of nuclear norm has been shown to be effective and even guaranteed to recover a low rank solution in some cases [49–51]. However, when one tries to minimize a specific singular value, the most common approach is to apply a non-smooth non-convex technique by using the subgradient of that singular value [52, 53]. It has also been suggested that due to structural relation of the singular values, one can minimize the partial tail sum of singular values [54], which would also be a non-convex non-smooth problem.

The problem of minimizing the k-th singular value of a matrix is closely related to the problem of minimizing the k-th largest element of a vector, and we note that the approach we derive in this chapter could be very similarly formulated for that case as well. Also, the same behavior regarding our convex heuristic has been seen when we apply it on the k-th largest element of a vector.

One application of minimizing such a singular value arises in decentralized control theory, where one would like to know how far a FDLTI state-space system is from losing decentralized controllability or observability. This is further discussed in Chapter 6.

We will formulate the problem in Section 4.1, and then consider convex heuristics for obtaining upper bounds for this problem in Section 4.3 as was first done in [55]. We will then review a local notion of subgradient of the k-th singular value in Section 4.2. When finding upper bounds, we assume that there are convex constraints that prohibit trivial solutions, and analyze a class of convex heuristics by taking a non-integer partial sum of the singular values from the greatest one up to a non-integer portion of $(k+1)$-th singular value. We inspect this heuristic numerically and compare it against the conventional ones, both in presence and absence of low-rank solutions. It was widely observed that our counter intuitive convex heuristic for minimizing the k-th singular value would perform better in the absence of a low-rank solution. However, we prove that if our convex heuristic recovers a low-rank solution, then the nuclear norm would also recover that solution, suggesting that if one is only concerned with the rank minimization, and not minimizing the k-th singular value even if it would ultimately be greater than zero, nuclear norm would be at least as good as our convex heuristic. We also discuss using subgradient based methods to further improve the solution obtained from our convex heuristic in the same section.

We consider lower bounds for this problem in Section 4.4. We provide a polynomial optimization problem that will be exactly equal to the singular value of consideration using factorization of the semidefinite matrices in Section 4.4.2. However as it involves a large number of variables and constraints, we give an alternative approximate form via characterization of positive definite matrices using leading principle minors in Section 4.4.3. This alternative form would involve fewer variables and constraints, but of higher degrees. We provide another form by sampling from the non-convex constraint in the Courant-Fischer variational formulation of the singular values in Section 4.4.4, an then review and utilize Sum-of-Square (SOS) techniques to derive lower bounds on these polynomial programs, which would in turn result in lower bounds for the minimization problem that we are interested in. These different formulations would be comparable against each other in terms of tightness of the lower bound and the size of the corresponding optimization problem.

We discuss how we can alternatively formulate the problem of minimizing a particular singular value subject to convex constraints by a Bilinear Matrix Inequality (BMI) that will be further subjected to an orthonormality condition on one of its variables in Section 4.5.

4.1 Problem Formulation

Given a matrix $X \in \mathbb{C}^{m\times n}$, convex functions $f_1(X), \cdots, f_{\bar{i}}(X)$, and affine functions $h_1(X), \cdots, h_{\bar{j}}(X)$ all from $\mathbb{C}^{m\times n}$ to $\mathbb{R}$, we are interested in the following optimization problem:

$$\begin{aligned} \text{minimize} \quad & \sigma_k(X) \\ \text{subject to} \quad & f_i(X) \leq 0 \quad i = 1, \cdots, \bar{i} \\ & h_j(X) = 0 \quad j = 1, \cdots, \bar{j}, \end{aligned} \tag{4.1}$$

with variable $X \in \mathbb{C}^{m \times n}$, and where $\sigma_k(X)$ denotes the k-th largest singular value of the matrix X. Without loss of generality we assume that $m \geq n$ and thus $\sigma_{n+1}(X) = \cdots = \sigma_m(X) = 0$. We will hence focus on the non-trivial cases $1 \leq k \leq n$, for which $\sigma_1(X) \geq \cdots \geq \sigma_n(X)$. This problem is convex if and only if $k = 1$ and we are interested in cases where $k > 1$.

4.2 Subgradient of the k-th Singular Value

We review concepts related to definition of the subgradient constrained to the local behavior of the non-smooth non-convex functions considered in this section. Definitions presented here are simplified versions of the ones in [52, Section 2].

Definition 65 (Regular Subgradient)**.** *Given a function $f : \mathbb{R}^p \to [-\infty, +\infty]$, we say $y \in \mathbb{R}^p$ is a* regular subgradient *of f at x, if $f(x) < \infty$, and in a neighborhood of x, we have:*

$$f(x+z) \geq f(x) + \langle y, z \rangle + o(z) \qquad \text{as} \quad z \to 0,$$

where $o(z)$ denotes a real-valued function defined in a neighborhood of the origin which satisfies $\lim_{z \to 0} \|z\|^{-1} o(z) = 0$.

The set of all regular subgradients of f at x is called the *regular subdifferential*, and is denoted by $\hat{\partial} f(x)$. See Figure 4.1(a) as an example of a regular subgradient when f is not convex. The regular subdifferential could be empty at some points, yet a descent direction could still exist in some cases. As an example see Figure 4.1(b) where there does not exist any plane that passes through the edge of the pyramid, and would be below the function at any open neighborhood containing a point on the edge, while any plane that contains the edge of intersection of the blue and red planes also contains descent directions. The following definitions consider this aspect in a more general form.

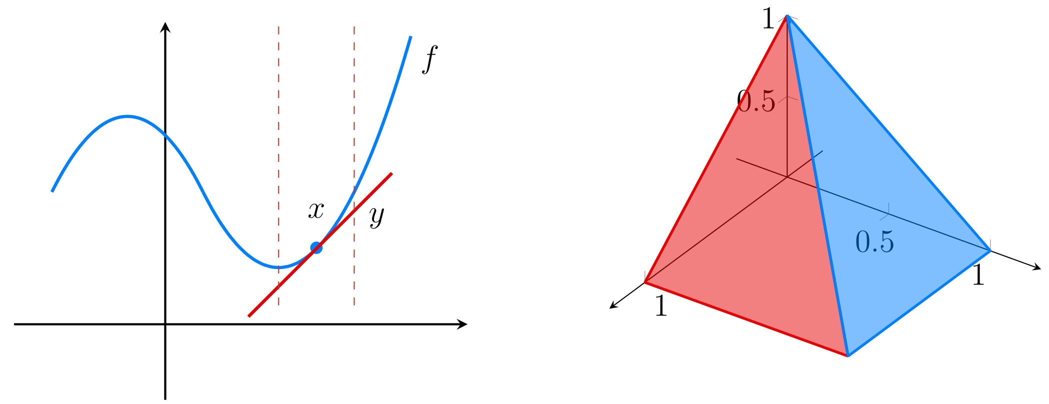
Definition 66 (Limiting Subgradient)**.** *Given a function $f : \mathbb{R}^p \to [-\infty, +\infty]$, we say $y \in \mathbb{R}^p$ is a* limiting subgradient *of f at x, if $f(x) < \infty$, and there exists a sequence of points x^{r} approaching x with values $f(x^{\mathrm{r}})$ approaching $f(x)$, and a sequence of regular subgradients y^{r} in $\hat{\partial} f(x^{\mathrm{r}})$ approaching y.*

The set of all limiting subgradients of f at x is called the *limiting subdifferential.*

Definition 67 (Clarke Subgradient)**.** *Given a function $f : \mathbb{R}^p \to [-\infty, +\infty]$, if f is locally Lipschitz in a neighborhood of x, a convex combination of subgradients at x (regular or limiting subgradients) is called a* Clarke subgradient *at x.*

The set of all Clarke subgradients of f at x is called the *Clarke subdifferential*, and is denoted by $\partial^{\mathrm{C}} f(x)$.

We are now ready to describe the subgradient of the k-th singular value with respect to a matrix variable in the following theorem.



(a) An illustration of a local regular subgradient for a non-convex function

(b) Second singular value of a 2-dimensional matrix as a function of a parameter in $[0,1]^2$

Figure 4.1: Illustrating cases on a local concept of a subgradient for non-convex functions

Theorem 68. *Given a matrix $X \in \mathbb{C}^{m\times n}$ with singular value decomposition $X = U_X \Sigma_X V_X^T$, a Clarke subgradient of $\sigma_k(\cdot)$ at X, for $1 \leq k \leq \min(m,n)$, denoted by $\sigma_{\mathrm{sg},k}(X)$ is also a (rank one) matrix in $\mathbb{C}^{m\times n}$ and is given by:*

$$\sigma_{\mathrm{sg},k}(X) = U_X \mathbf{e}_k \mathbf{e}_k^T V_X^T, \tag{4.2}$$

Proof. See, for example, [53, Corollary 6.4]. □

Remark 69. *A regular subgradient of $\sigma_k(\cdot)$ at X could also be obtained in the same way as (4.2), unless when $\sigma_k(X) = \sigma_{k-1}(X)$, in which case the regular subdifferential would be the empty set. This is the reason definition 67 become advantages to have the right level of technicality for developing a feasible descent direction for our problem.*

4.3 A Convex Heuristic

We analyze a class of convex heuristics for problem (4.1), and inspect their performance via numerical simulations later in this section. To this end, we generalize the Ky Fan k-norm in (1.1) on page 10 and define the generalized non-integer Ky Fan ℓ-norm of X as:

$$s_\ell(X) \triangleq \sum_{i=1}^{\lfloor \ell \rfloor} \sigma_i(X) + (\ell - \lfloor \ell \rfloor)\sigma_{\lfloor \ell \rfloor + 1}(X), \tag{4.3}$$

where $\ell \in [1, n]$ is a real variable, and where $\lfloor \ell \rfloor$ denotes floor of ℓ.

Example 70. *If we take $\ell = 2.7$ then $s_{2.7}(X) = \sigma_1(X) + \sigma_2(X) + 0.7\,\sigma_3(X)$.*

Corollary 71. *$s_\ell(X)$ is convex in X for all $\ell \in [1, n]$.*

Proof. Write the non-integer Ky Fan ℓ-norm as convex combinations of integer Ky Fan k-norms, where each would be convex in X [56, Argument 19, p. 147]. □

Remark 72. *The nuclear norm of X is by definition equal to $s_n(X)$.*

Problem (4.1) is non-convex in X for all $k > 1$ and we would like to replace the objective with the convex heuristic $s_\ell(X)$ and inspect the best ℓ, i.e., we are interested in solving the following problem:

$$\begin{aligned} X_\ell^* \in \quad & \arg \min_{X \in \mathbb{R}^{m \times n}} && s_\ell(X) \\ & \text{subject to} && f_i(X) \le 0 \quad i = 1, \cdots, \bar{i} \\ & && h_j(X) = 0 \quad j = 1, \cdots, \bar{j}, \end{aligned} \tag{4.4}$$

and then reporting $\sigma_k(X_\ell^*)$ as the output of our convex heuristic. Also, denote the best ℓ value for a specific singular value k by ℓ_k^*, i.e.:

$$\ell_k^* \triangleq \arg \min_{\ell \in [1,n]} \sigma_k(X_\ell^*).$$

Remark 73. *Perhaps a first guess could be taking $\ell = k$, however our understanding from a wide variety of numerical examples shows that generally we have $\ell_k^* > k$.*

Example 74. *As an example of (4.4) assume that we want to minimize the $\sigma_2(X)$, and let X_1 and X_2 satisfy the feasibility constraints. Furthermore assume that the singular values of these two matrices are given as in Table 4.1. If we only choose $\ell =$ 2 for our convex heuristic, i.e., minimizing $\sigma_1(X) + \sigma_2(X)$, we would be worse off than taking the sum of the first three singular values. This could happen as the singular values are implicitly tied together via structural constraints $\sigma_1(X) \geq \cdots \geq \sigma_n(X)$. This would be further inspected via a variety of numerical examples in the rest of this chapter.*

	σ_1	σ_2	σ_3	$\sigma_1 + \sigma_2$	$\sigma_1 + \sigma_2 + \sigma_3$
X_1	10	9.9	9	19.9	28.9
X_2	12	8	7.9	20	27.9

Table 4.1: Singular values for Example 74

For numerical examples through the rest of this section we will focus on the following optimization problem.

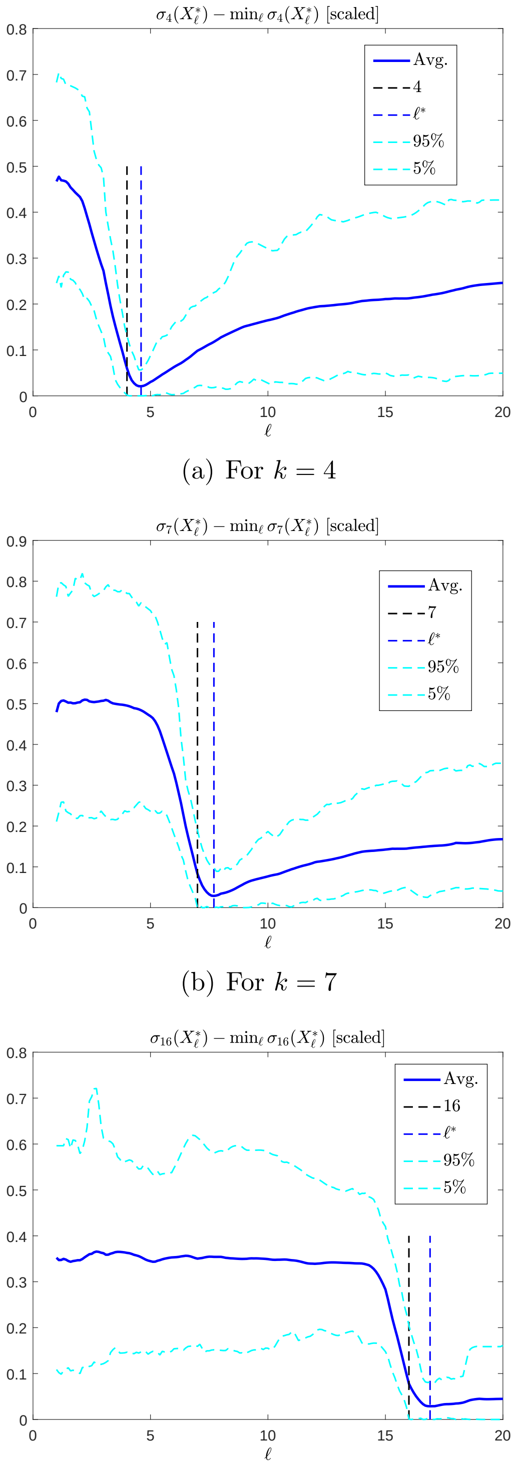
$$\begin{array}{lll} \text{minimize} & \sigma_k(X) & \\ \text{subject to} & X_{ij} = B_{ij} & \text{for} \quad (i,j) \in \mathcal{I} \\ & \underline{X}_{ij} \leq X_{ij} \leq \bar{X}_{ij} & \text{for} \quad (i,j) \in \bar{\mathcal{I}}, \end{array} \tag{4.5}$$

with variable $X \in \mathbb{R}^{m \times n}$, and fixed $\mathcal{I} \subset \{1, \cdots, m\} \times \{1, \cdots, n\}$, $B \in \mathbb{R}^{m \times n}$, $\underline{X} \in \mathbb{R}^{m \times n}$, $\bar{X} \in \mathbb{R}^{m \times n}$, and where $\bar{\mathcal{I}}$ denotes the complement of the set $\mathcal{I}$. We want to inspect our convex heuristic for this problem by replacing the objective with $s_\ell(X)$, i.e., we will solve:

$$\begin{array}{llll} X_\ell^* \in & \arg\min & s_\ell(X) & \\ & \text{subject to} & X_{ij} = B_{ij} & \text{for} \quad (i,j) \in \mathcal{I} \\ & & \underline{X}_{ij} \leq X_{ij} \leq \bar{X}_{ij} & \text{for} \quad (i,j) \in \bar{\mathcal{I}}. \end{array} \tag{4.6}$$

Example 75. *In this example we will fix $m = n = 20$, and consider 200 instances of (4.6) by random uniform generation of $\mathcal{I}$, B, $\underline{X}$, and $\bar{X}$. We will vary ℓ and look at the k^{th} singular value of the solution of (4.6). More precisely we will plot the difference of $\sigma_k(X_\ell^*)$ to its minimum when we vary ℓ, i.e., plotting $\sigma_k(X_\ell^*) - \min_\ell \sigma_k(X_\ell^*)$ versus ℓ. Results for $k = 4$, 7, and 16 are provided in Figures 4.2(a), 4.2(b), and 4.2(c) respectively.*

The vertical black dashed line indicates where $\ell = k$, the blue is the average of $\sigma_k(X_\ell^*) - \min_\ell \sigma_k(X_\ell^*)$ across 200 samples, the dashed cyan lines show the 5% and 95% quantiles for this metric, and the dashed blue shows where the average hits

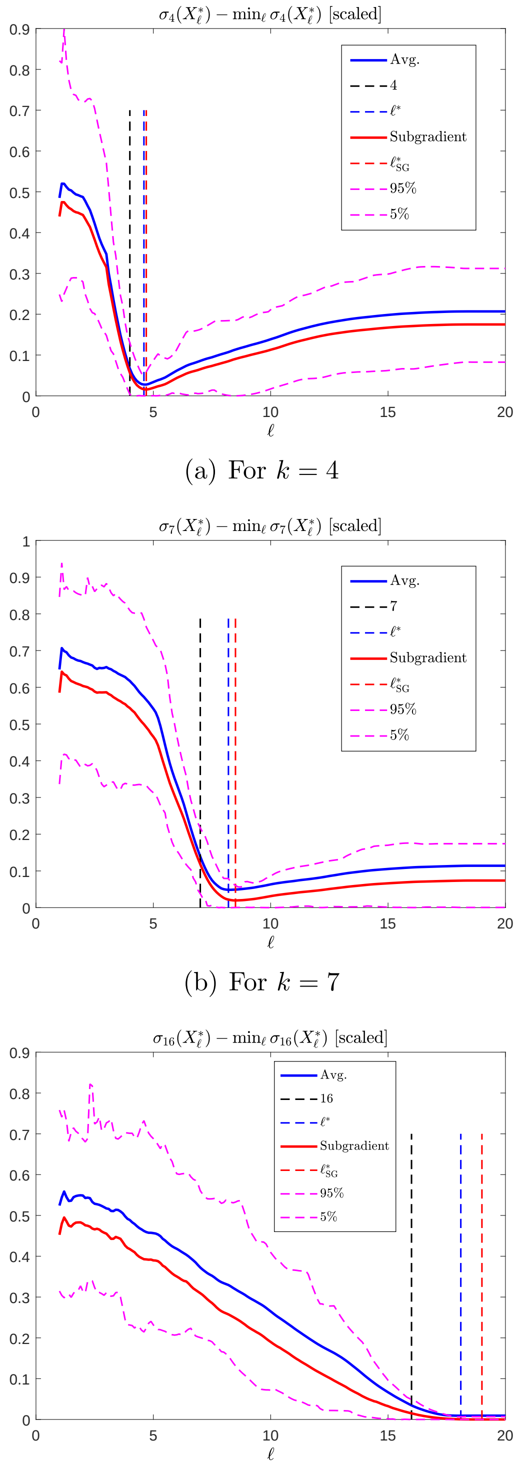


(a) For $k = 4$

(b) For $k = 7$

(c) For $k = 16$

Figure 4.2: Non-integer Ky Fan ℓ-norm as the convex heuristic for minimizing σ_k



(a) For $k = 4$

(b) For $k = 7$

(c) For $k = 16$

Figure 4.3: Non-integer Ky Fan ℓ-norm as the convex heuristic for minimizing σ_k with low-rank solutions

its minimum. It was observed that for almost all k (even for the k that are not presented in this figure), the average hits its minimum is some point after $l > k$, suggesting that perhaps the best convex heuristic for minimizing σ_k in this class (by s_ℓ) is achieved at a $\ell > k$.

We have tested our heuristic on random matrices where the solutions would almost never be rank deficient. However it is interesting to see how this heuristic compares to nuclear norm for rank minimization. We will first show that how our heuristic and the nuclear norm are related to each other in the following theorem, and then discuss some experimental lessons when we apply them on a similar class of problems as in Example 75.

Theorem 76. *If $X^*_{\ell_0}$ is a minimizer of (4.4) with rank $\leq \lceil \ell_0 \rceil - 1$, for some $\ell_0 \in [1, n]$, then it also minimizes (4.4) for any $\ell \geq \ell_0$, i.e.: if $\sigma_{\lceil \ell_0 \rceil}(X^*_{\ell_0}) = 0$ then $s_\ell(X^*_\ell) = s_\ell(X^*_{\ell_0})$.*

Proof. Proof is done by contradiction. Since X^*_ℓ is a minimizer of $s_\ell(\cdot)$, the conclusion can be false only if $s_\ell(X^*_\ell) < s_\ell(X^*_{\ell_0})$, for which we show that some singular value must be negative, which achieves the desired contradiction. This condition can be equivalently written as:

$$s_{\ell_0}(X^*_\ell) + s_\ell(X^*_\ell) - s_{\ell_0}(X^*_\ell) < s_\ell(X^*_{\ell_0}) = s_{\ell_0}(X^*_{\ell_0}) \tag{4.7}$$

where the equality follows as $\sigma_{\lceil \ell_0 \rceil}(X^*_{\ell_0}) = 0$ and thus $\sigma_k(X^*_{\ell_0}) = 0$ for all $k \geq \lceil \ell_0 \rceil$, which by definition results in $s_\ell(X^*_{\ell_0}) = s_{\ell_0}(X^*_{\ell_0})$ for all $\ell \geq \ell_0$. Also, since $X^*_{\ell_0}$ is a minimizer of $s_{\ell_0}(\cdot)$, we have that $s_{\ell_0}(X_\ell^*) \geq s_{\ell_0}(X_{\ell_0}^*)$ for all ℓ, and thus (4.7) becomes true only if $s_\ell(X_\ell^*) - s_{\ell_0}(X_\ell^*) < 0$, meaning that some σ_k must be negative in the accumulative sum. This achieves the contradiction. □

Remark 77. *Theorem 76 suggests that there is no loss in considering the convex heuristic of nuclear norm ($\ell = n$) compared to the cases where $\ell < n$ when we have a low rank solution (of rank $\leq \lceil \ell_0 \rceil - 1$ for some $\ell_0 < n$). However this is not the case in Example 75, where the solutions were almost always full rank.*

In the next example we consider cases where the solution could be low rank and inspect our heuristic on this class also.

Example 78. *We again fix $m = n = 20$, and consider 50 instances of* (4.6)*. In each of these instances a random X_0 with rank 10 is generated first, then $\mathcal{I}$ is randomly selected and we set $B_{ij} = (X_0)_{ij}$ for all $(i,\ j) \in \mathcal{I}$, then $\underline{X}$ and $\bar{X}$ are uniformly generated in a way that X_0 (the rank 10 matrix) remains a feasible point. We again vary ℓ and look at the k^{th} singular value of the solution of* (4.6)*. We then further decrease $\sigma_k(X_\ell^*)$ with a subgradient based method. We take X_ℓ^* as our starting point, and do descent step along the subgradient of σ_k for each ℓ. The subgradient is straightforward to derive when all the singular values are distinct, however more technical considerations would be needed when this would not be the case. See [52, 53] for a detailed derivation of the subgradient of the k-th singular value. Results for $k = 4,\ 7$, and 16 are provided in Figures 4.3(a), 4.3(b),* and *4.3(c).*

The vertical dashed black line indicates where $\ell = k$, the blue is the average of $\sigma_k(X_\ell^*) - \min_\ell \sigma_k(X_\ell^*)$ across 50 samples, the red line shows the average further *enhancement resulted from applying the subgradient based method on each of the samples, the dashed magenta lines show the 5% and 95% quantiles for the red line, the dashed blue shows where the blue hits its minimum, and the dashed red shows where the red hits its minimum. The observation that both the minimum of the convex heuristic and its enhanced version (by subgradient method) would happen for some $\ell > k$ was widespread. It can also be seen that when $k = 16$ the solution, up to numerical errors, would remain the same after some point (see Remark 77), meaning that for rank-deficient solutions nuclear norm would still recover as good as the suggested heuristic in this chapter. The solution obtained from the nuclear norm is displayed in the figures where $\ell = n = 20$.*

Our heuristic for minimizing the k-th singular value results in an upper bound for the global minimizer, and we want to have optimality certificates on how good these upper bounds will be. In the following section we will discuss methods for obtaining such lower bounds.

4.4 Lower Bounds

We will derive lower bounds for the global minimizer of the k-th singular value in this section. In Section 4.4.1, we review Sum of Squares techniques for obtaining lower bounds on problem that could be formulated as polynomial optimization. In each of the subsequent subsections we formulate the k-th singular value as a different polynomial optimization problem which can then be used in conjunction with SOS techniques in Section 4.4.1 to obtain lower bounds. We use a factorization of positive semidefinite matrices to derive a polynomial optimization problem in Section 4.4.2, then we lay out an alternative form with fewer constraints of higher degrees in Section 4.4.3. Finally we utilize the Courant-Fischer variational formulation of singular values to obtain another form which would require less resources for implementation. Properties related to each of these specific formulations are discussed where appropriate.

In the remainder of this section we directly address problem (4.1) with the assumption that $f_i(X)$ and $h_j(X)$ would all be polynomials in X, however we no longer require that they would be convex or affine.

4.4.1 Sum-of-Squares

We will first give a brief overview of polynomial optimization problems, and then review some related results that we will use later in this section. The materials in this subsection are mostly adopted from [57].

Definition 79 (Polynomial Optimization Problem). *Given real-valued polynomials $p(x)$, $g_1(x), \cdots, g_r(x)$ all from $\mathbb{R}^n$ to $\mathbb{R}$, the following optimization problem is called a Polynomial Optimization (P.O.) problem:*

$$p_K^* \triangleq \min_{x \in K} p(x), \tag{4.8}$$

with variable $x \in \mathbb{R}^n$, and where the set $K \subseteq \mathbb{R}^n$ is defined by polynomial inequalities as $K \triangleq \{x \in \mathbb{R}^n \mid g_i(x) \geq 0$, for $i = 1, \cdots, r\}$.

Remark 80. *The equality constraint $h(x) = 0$ can be expressed by two inequality constraints $h(x) \geq 0$, and $(-h(x)) \geq 0$.*

Furthermore, define the sum of squares polynomials as:

Definition 81 (Sum-of-Squares)**.** *A real-valued polynomial $p(x) : \mathbb{R}^n \to \mathbb{R}$ is called* ***Sum-of-Squares*** *(SOS), if it can be written as:*

$$p(x) \;=\; \sum_{i=1}^{\tilde{i}} \left(p_i(x)\right)^2 ,$$

for some $\tilde{i} \in \mathbb{N}$, and where $p_i(x) : \mathbb{R}^n \to \mathbb{R}$ are all polynomials in x for $i = 1, \cdots, \tilde{i}$.

We would like to derive lower bounds on one particular instance of such P.O. problems. This could be achieved if the set K satisfies the following assumption:

Assumption 82 ([57, Assumption 4.1])**.** *The set K is compact and there exists a real-valued polynomial $u(x) : \mathbb{R}^n \to \mathbb{R}$ such that the set $\{x \in \mathbb{R}^n \mid u(x) \geq 0\}$ is compact, and:*

$$u(x) \;=\; u_0(x) + \sum_{k=1}^{r} g_i(x) u_i(x), \qquad \text{for all} \quad x \in \mathbb{R}^n,$$

where $u_i(x)$ are all SOS polynomials for $i = 0, \cdots, r$.

Remark 83. *One way to ensure that this assumption holds is that the variable x would be bounded, i.e., it would be known that the solution of (4.8) would lie in some bounded region $\|x\|_2^2 \leq a$. In this case, one can add an inequality constraint $g_{r+1}(x) \geq 0$, with $g_{r+1}(x) = a - \|x\|_2^2$, to the set K, and take $u_i(x) = 1$, if $i = r + 1$, and 0 otherwise.*

It can then be proved that with this assumption one could obtain a sequence of finite dimensional Semidefinite Programs (SDP) that would converge to the optimum value p_K^* from below. This is stated in the following theorem:

Theorem 84 ([57, Theorem 4.2])**.** *Let $p(x)$, p_K^*, and the set K be given as in Definition 79. Assume that K is a compact set that satisfies Assumption 82, then there exists a sequence of finite dimensional SDP indexed by their order N, denoted by $\mathbb{Q}_K^N$, that converges to the optimal value p_K^* from below, i.e.:*

$$\inf\ \mathbb{Q}_K^N \uparrow p_K^*, \qquad \text{as} \quad N \to \infty.$$

4.4.2 Lower Bound via Factorization

In this section we will utilize the factorization of the semidefinite matrices to transfer problem (4.1) into a polynomial optimization problem.

We will use the following lemma:

Lemma 85. *Given a matrix $X \in \mathbb{R}^{m \times n}$ with nonzero singular values $\sigma_1(X) \geq \cdots \geq \sigma_n(X)$, for any $k \in \{1, \cdots, n\}$ we have that:*

$$\sigma_k(X) = \min_{\substack{R \in \mathbb{R}^{m \times n} \\ \text{rank}(R) = k-1}} \|X - R\|_2.$$

Proof. See, for example [58, Eq. (5.12.10), p. 417]. □

We can further replace $\text{rank}(R) = k - 1$ constraint with $\text{rank}(R) \leq k - 1$ and at the same time extend the result to allow for zero singular values as in the following corollary:

Corollary 86. *Given a matrix $X \in \mathbb{C}^{m\times n}$ we have that:*

$$\sigma_k(X) = \min_{\substack{R\in\mathbb{C}^{m\times n}\\ \mathrm{rank}(R)\leq k-1}} \|X - R\|_2.$$

Proof. Let $U_X \Sigma_X V_X^*$ denote a SVD of X, then a solution of the optimization problem in Lemma 85 would be achieved by taking

$$R = U_X \operatorname{diag}(\sigma_1(X), \cdots, \sigma_{k-1}(X), 0, \cdots, 0) V_X^*,$$

even if $\mathrm{rank}(X) < n$. Any solution with the strict constraint $\mathrm{rank}(R) < k-1$ would imply $\sigma_{k-1}(R^*) = 0$. This would render the minimum value to be equal to $\sigma_{k-1}(X) \geq \sigma_k(X)$, meaning that the aforementioned R would still be an optimal solution. □

By combining the above corollary with the SDP representation of the matrix 2-norm, we would have:

Theorem 87. *The optimization problem* (4.1) *is equivalent to the following:*

$$\begin{aligned} &\text{minimize} && \tau \\ &\text{subject to} && f_i(X) \le 0 && i = 1, \cdots, \bar{i} \\ &&& h_j(X) = 0 && j = 1, \cdots, \bar{j} \\ &&& R = UV \\ &&& \begin{bmatrix} \tau I & X - R \\ (X-R)^* & \tau I \end{bmatrix} \succeq 0, \end{aligned}$$

with variables $\tau \in \mathbb{R}, X \in \mathbb{C}^{m\times n}, R \in \mathbb{C}^{m\times n}, U \in \mathbb{C}^{m\times(k-1)}, V \in \mathbb{C}^{(k-1)\times n}$.

Proof. Constraint $\operatorname{rank}(R) \le k-1$ is equivalent to $R = UV$ where U has $k-1$ columns and V has $k-1$ rows. Then the result follows from SDP representation of the 2-norm. □

We can insert $R = UV$ into the optimization problem and factorize the semidefinite constraint as what follows to derive a polynomial optimization problem:

Corollary 88. *The optimization problem* (4.1) *is equivalent to the following:*

$$\begin{aligned} &\text{minimize} && \tau \\ &\text{subject to} && f_i(X) \le 0 && i = 1, \cdots, \bar{i} \\ &&& h_j(X) = 0 && j = 1, \cdots, \bar{j} \\ &&& \begin{bmatrix} \tau I & X - UV \\ (X-UV)^T & \tau I \end{bmatrix} = G^T G, \end{aligned}$$

with variables $\tau \in \mathbb{R}, X \in \mathbb{C}^{m\times n}, U \in \mathbb{C}^{m\times(k-1)}, V \in \mathbb{C}^{(k-1)\times n}, G \in \mathbb{C}^{(m+n)\times(m+n)}$.

Remark 89. *The minimization problem in Corollary 88 is a polynomial optimization problem with* $(m+n)^2+(k-1)(m+n)+mn+1$ *variables and* $(m+n)^2+\bar{i}+\bar{j}$ *constraints, or equivalently* $2\left((m+n)^2+(k-1)(m+n)+mn\right)+1$ *real variables and* $2(m+n)^2+\bar{i}+\bar{j}$ *real constraints. Furthermore each of the constraints (aside from* $f_i(X)\leq 0$ *and* $h_j(X)=0$*) is of degree two.*

The SOS technique in Section 4.4.1 can now be applied to derive a lower bound for this polynomial optimization problem, however due to the large number of variables and constraints, we will consider an alternative form which give us fewer variables and constraints but of higher degrees in the following section.

4.4.3 Lower Bound via Leading Principle Minors

We will form an alternative formulation with fewer variables and constraints in this section. This form will only replace the semidefinite factorization in Corollary 88 with a constraint on the leading principle minors, which we define below.

Definition 90 (Leading Principle Minors)**.** *Given a square matrix* $A\in\mathbb{R}^{n\times n}$*, the leading principle minors of* A *are the determinants of the* $d\times d$ *sub-matrices obtained from only considering the first* d *rows and columns of* A*, where* $d\in\{1,\cdots,n\}$*. We denote them by* $g_d(A)$*:*

$$g_d(A) \triangleq \det\left(\begin{bmatrix} I_d & 0\end{bmatrix} A \begin{bmatrix} I_d \\ 0\end{bmatrix}\right),$$

for $d\in\{1,\cdots,n\}$*.*

Positive definiteness of a matrix can be equivalently stated in terms of its leading principle minors:

Lemma 91. *Given a symmetric matrix $A \in \mathbb{R}^{n\times n}$, we have that $A \succ 0$ if and only if each of the n leading principle minors of A are strictly positive.*

Proof. See, for example [59, Theorem 3, p. 306], or [60, Sylvester's Criterion] □

Remark 92. *It is noteworthy that although checking for positive definiteness of a matrix is equivalent to n leading principle minors being positive, checking if a matrix is positive semidefinite requires that all the principle minors would be non-negative [59, Theorem 4, p. 307]. There are $2^n - 2$ principle minors of A.*

Although it is possible to derive an exact equivalent to the optimization problem in Theorem 87 by principle minors, due to excessive exponential number of resulting constraints we will derive an approximate one based on leading principle minors by first tightening the positive semidefinite constraint in Theorem 87 to the positive definiteness, and then using Lemma 91 to derive an approximate lower bound with fewer constraints and variables in the following corollary.

Corollary 93. *The optimization problem in Theorem 87 when replacing the positive semidefinite constraint with positive definite constraint, and when X is real is* *equivalent to:*

$$
\begin{aligned}
&\text{minimize} && \tau \\
&\text{subject to} && f_i(X) \le 0 \qquad i = 1, \cdots, \bar{i} \\
& && h_j(X) = 0 \qquad j = 1, \cdots, \bar{j} \\
& && g_d\left(\begin{bmatrix} \tau I & X - UV \\ (X-UV)^T & \tau I \end{bmatrix}\right) > 0 \qquad \text{for} \qquad d = 1, \cdots, m+n,
\end{aligned}
$$

with variables $\tau \in \mathbb{R}, X \in \mathbb{R}^{m \times n}, U \in \mathbb{R}^{m \times (k-1)}, V \in \mathbb{R}^{(k-1) \times n}$.

Proof. The corollary is a direct consequence of Lemma 91 □

Remark 94. *The only approximation in Corollary 93 compared to the exact formulation in Corollary 88 is due to tightening the semidefinite constraint, which is possible as long the feasible set of Corollary 93 is not empty. The minimization problem in Corollary 93 is a polynomial optimization problem with* $(k-1)(m+n)+mn+1$ *variables and* $m + n + \bar{i} + \bar{j}$ *constraints. These constraints (aside from* $f_i(X) \le 0$ *and* $h_j(X) = 0$*) are of degree* $m + n$ *at most.*

4.4.4 Lower Bound by Sampling from Courant-Fischer

We will first review the Courant-Fischer variational formulation of the singular values, and will transform this formulation into a polynomial program by rewriting some of the constraints. To this end, given a Hermitian matrix M, and a non-zero vector v with compatible dimension, define the ***Rayleigh quotient***, denoted by $R(M, v)$, as:

$$R(M, v) \triangleq \frac{v^*Mv}{v^*v}.$$

The next theorem reviews the Courant-Fischer formulation of the singular values, and is adopted to the notation used in this chapter.

Theorem 95 (Courant-Fischer)**.** *Given a matrix $X \in \mathbb{C}^{m\times n}$ the following relations hold:*

$$\sigma_k^2(X) = \min_{v_1, v_2, \cdots, v_{k-1} \in \mathbb{C}^n} \max_{\substack{v \neq 0, v \in \mathbb{C}^n, \\ v \perp v_1, \cdots, v_{k-1}}} R(X^*X, v) \tag{4.9}$$

$$\sigma_k^2(X) = \max_{v_1, v_2, \cdots, v_{n-k} \in \mathbb{C}^n} \min_{\substack{v \neq 0, v \in \mathbb{C}^n, \\ v \perp v_1, \cdots, v_{n-k}}} R(X^*X, v). \tag{4.10}$$

Proof. The proof is a direct consequence of [61, Theorem, 4.2.11, p. 179] when considering that X^*X is an $n \times n$ Hermitian matrix (and thus with real eigenvalues), for which we have that $\lambda_k(X^*X) = \sigma_k^2(X)$. □

Remark 96 (Rayleigh-Ritz)**.** *This theorem extends Rayleigh-Ritz theorem on the largest and smallest eigenvalues of a Hermitian matrix [61, Theorem 4.2.2, p. 176], which states that for a Hermitian matrix $M \in \mathbb{C}^{n\times n}$, with eigenvalues $\lambda_1(M) \geq \cdots \geq \lambda_n(M)$, we have that:*

$$\lambda_{\max}(M) = \lambda_1(M) = \max_{v \in \mathbb{C}^n} R(M, v),$$

$$\lambda_{\min}(M) = \lambda_n(M) = \min_{v \in \mathbb{C}^n} R(M, v).$$

We can equivalently write (4.9) and (4.10) using rank constraints on the considered subspaces:

Corollary 97. *Given a matrix $X \in \mathbb{C}^{m\times n}$ we have:*

$$\sigma_k^2(X) = \min_{\substack{V\in\mathbb{C}^{(k-1)\times n},\\ \mathrm{rank}(V)=k-1}} \max_{\substack{v\in\mathbb{C}^n,\\ Vv=0\\ v^*v=1}} v^*X^*Xv \tag{4.11}$$

$$\sigma_k^2(X) = \max_{\substack{V\in\mathbb{C}^{(n-k)\times n},\\ \mathrm{rank}(V)=n-k}} \min_{\substack{v\in\mathbb{C}^n,\\ Vv=0\\ v^*v=1}} v^*X^*Xv. \tag{4.12}$$

Proof. It is straightforward to replace the Rayleigh quotient with only its numerator, while enforcing the denominator to be equal to one, as any non-zero v can be scaled to have unit norm. For the min-max formulation of (4.9), we can gather $v_1, \cdots, v_{k-1}$ in a $(k-1)\times n$ matrix as $V = \begin{bmatrix} v_1 & \ldots & v_{k-1}\end{bmatrix}^T$. Then, the minimum would be achieved when all the $v_1, \cdots, v_{k-1}$ are independent of one another so as to make the feasible set for the v in the max part as small as possible, which is equivalent to the constraint $\mathrm{rank}(V) = k-1$. Similar reasoning applies to the max-min formulation.

□

We would further relax (4.11) and (4.12) by considering finite samples of the rank-constrained subspaces in those equations, and show that this would lead to upper and lower bounds on the singular values. This is illustrated in the following corollary:

Corollary 98. *Suppose that a matrix $X \in \mathbb{C}^{m\times n}$ is given, Let $q \in \mathbb{N}$ be the desired number of the samples from the rank constrained subspaces, i.e., let $\bar{V}_1, \cdots, \bar{V}_q$ be matrices that are sampled from the subspace specified by the rank constraint in (4.11), i.e., they are all in $\mathbb{C}^{(k-1)\times n}$ and have rank $k-1$. Then we have the following upper* *bound:*

$$\begin{array}{rlll} \sigma_k^2(X) \leq & \max & \kappa & \\ & \text{s.t.} & \kappa \leq \bar{v}_i^* X^* X \bar{v}_i & \text{for} \quad i = 1, \cdots, q \\ & & \bar{V}_i \bar{v}_i = 0 & \text{for} \quad i = 1, \cdots, q \\ & & \bar{v}_i^* \bar{v}_i = 1 & \text{for} \quad i = 1, \cdots, q, \end{array} \tag{4.13}$$

with variables $\kappa \in \mathbb{R}$, and $\bar{v}_1, \cdots, \bar{v}_q \in \mathbb{C}^n$. Similarly let $\underline{V}_1, \cdots, \underline{V}_q$ be matrices that are sampled from the rank constraint in (4.12), that are all in $\mathbb{C}^{(n-k)\times n}$ and have rank $n-k$. Then we have the following lower bound:

$$\begin{array}{rlll} \sigma_k^2(X) \geq & \min & \kappa & \\ & \text{s.t.} & \kappa \geq \underline{v}_i^* X^* X \underline{v}_i & \text{for} \quad i = 1, \cdots, q \\ & & \underline{V}_i \underline{v}_i = 0 & \text{for} \quad i = 1, \cdots, q \\ & & \underline{v}_i^* \underline{v}_i = 1 & \text{for} \quad i = 1, \cdots, q, \end{array} \tag{4.14}$$

with variables $\kappa \in \mathbb{R}$ and $\underline{v}_1, \cdots, \underline{v}_q \in \mathbb{C}^n$.

Proof. The upper-bound in (4.13) is achieved due to the fact that by finite sampling we are minimizing over a smaller set rather than the rank constrained subspace in (4.11), and hence the minimum value would increase. Similarly, in (4.14) we are only considering finite numbers of $\underline{V}_1, \cdots, \underline{V}_q$, rather than the original subspace specified by the rank constraint in (4.12), and hence the maximum value would decrease and we would have a lower bound for $\sigma_k(X)$. □

We will next utilize this lower bound for the optimization problem (4.1).

Corollary 99. *Given a finite number $q \in \mathbb{N}$, let $\underline{V}_1, \cdots, \underline{V}_q$ be q rank $n-k$ samples* *from $\mathbb{C}^{(n-k)\times n}$, then a lower bound for the optimization problem (4.1) can be obtained by taking $\sqrt{\kappa}$ from the following polynomial optimization problem:*

$$\begin{array}{lll} \text{minimize} & \kappa & \\ \text{subject to} & f_i(X) \leq 0 & \text{for} \quad i = 1, \cdots, \bar{i} \\ & h_j(X) = 0 & \text{for} \quad j = 1, \cdots, \bar{j} \\ & \kappa \geq \underline{v}_i^* X^* X \underline{v}_i & \text{for} \quad i = 1, \cdots, q \\ & \underline{V}_i \underline{v}_i = 0 & \text{for} \quad i = 1, \cdots, q \\ & \underline{v}_i^* \underline{v}_i = 1 & \text{for} \quad i = 1, \cdots, q, \end{array}$$

with variables $\kappa \in \mathbb{R},\ X \in \mathbb{C}^{m\times n}$, and $\underline{v}_1, \cdots, \underline{v}_q \in \mathbb{C}^n$.

Remark 100. *The tightness of this lower bound would be dependent on the count (q) and choices of the samples. However, the advantage of this approach compared to the ones in the previous sections is that the required resources (memory and computation time) can be implicitly controlled by choosing a moderate q.*

4.5 An Equivalent Bilinear Matrix Inequality

We will derive an equivalent BMI to (4.1) in this section, which would be based on the min-max form in the Courant-Fischer formulation. This BMI is further subjected to have an orthonormal constraint on one of its variable. To this end we will equivalently write (4.9) as:

$$\sigma_k^2(X) \;=\; \min_{\substack{\mathcal{V}_{n-k+1}\subseteq\mathbb{C}^n \\ \dim(\mathcal{V}_{n-k+1})=n-k+1}} \;\max_{\substack{v\in\mathcal{V}_{n-k+1} \\ v^*v=1}} v^*X^*Xv \tag{4.15}$$

Which is further equivalent to the form stated in the following corollary:

Corollary 101. *Given a matrix $X \in \mathbb{C}^{m\times n}$ we have:*

$$\sigma_k^2(X) = \min_{\substack{R\in\mathbb{C}^{n\times(n-k+1)} \\ R^*R=I}} \max_{\substack{v\in\mathbb{C}^n, x\in\mathbb{C}^{n-k+1} \\ v^*v\leq 1 \\ v=Rx}} v^*X^*Xv \tag{4.16}$$

Proof. We can represent the $n-k+1$ dimensional subspaces $\mathcal{V}_{n-k+1}$ in $\mathbb{C}^n$ via range-spaces of full column rank matrices $R \in \mathbb{C}^{n\times(n-k+1)}$. Since we are only interested in the directions specified by this full column rank R, without loss of generality it can also be assumed that R is orthonormal. It is also straightforward to check that given any direction $v \in \mathbb{C}^n$, the inner maximum would occur at the boundary of the unit circle, and thus we can replace the $v^*v = 1$ constraint with $v^*v \leq 1$. □

We will further insert $v = Rx$ for v in (4.16) to have:

$$\sigma_k^2(X) = \min_{\substack{R\in\mathbb{C}^{n\times(n-k+1)} \\ R^*R=I}} \max_{\substack{x\in\mathbb{C}^{n-k+1} \\ x^*x\leq 1}} x^*R^*X^*XRx. \tag{4.17}$$

The inner maximization has a quadratic objective subject to only a single quadratic constraint, and thus by strong duality [62, Section B.1] we have:

$$\max_{\substack{x\in\mathbb{C}^{n-k+1} \\ x^*x\leq 1}} x^*R^*X^*XRx = \min_{\substack{\tau\in\mathbb{R},\ \tau\geq 0 \\ R^*X^*XR\preceq\tau^2 I}} \tau^2 \tag{4.18}$$

By combining (4.18), (4.17) and (4.1) we would have the following theorem:

Theorem 102. *Problem (4.1) can be equivalently written as:*

$$\begin{aligned} \text{minimize} \quad & \tau \\ \text{subject to} \quad & f_i(X) \leq 0 \qquad\qquad \text{for} \quad i = 1, \cdots, \bar{i} \\ & h_j(X) = 0 \qquad\qquad \text{for} \quad j = 1, \cdots, \bar{j} \\ & R^* R = I \\ & \begin{bmatrix} \tau I & XR \\ (XR)^* & \tau I \end{bmatrix} \succeq 0, \end{aligned} \tag{4.19}$$

with variables $\tau \in \mathbb{R}$, $X \in \mathbb{C}^{m \times n}$, and $R \in \mathbb{C}^{n \times (n-k+1)}$.

Proof. Insert (4.18) into (4.17). We are interested in $\sigma_k(X)$ and not $\sigma_k^2(X)$, hence noting that τ^2 is a monotonic function of τ, we can replace the objective with τ. We will further insert this for $\sigma_k(X)$ in (4.1) which would give us (4.19). □

Chapter 5: Enhanced ADMM-based Heuristics for Mixed Integer Non-Linear Programs

We consider problems that are convex except for a vector of discrete variables in this chapter. One main motivation behind the suggested methods in this chapter is in regard to the MINLP part involved in the approximation of the decentralized assignability measure in Chapter 6.

MINLPs are hard problems in general with much interest in finding bounds or approximate solutions for them. These include linear program (LP) and Semidefinite relaxations (SDP). The LP methods consider a linear relaxation of the integer variable to obtain a lower bound, and its projection to the discrete space for an upper-bound, whereas SDP relaxations consider the trace of a rank 1 matrix instead of the terms that involve products of the integer variables [63, 64]. Tighter relaxations can be obtained by lift-and-project methods, that introduce new auxiliary variables to transform the nonlinear integer constraints into a form with linear constraints in a higher dimension, and then solve the convex problem in the higher dimension to obtain a lower-bound [65,66]. When there would also be higher degree non-convex objectives functions or constraints other than the integer constraint, one can also consider using polynomial optimization methods to obtain such lower-bounds [57,67,68]. We are interested in upper-bounds for MINLP problems in this chapter and refer the reader to the above works for certificates of the optimality.

The so-called relax-and-round algorithm replaces the discrete variable with its continuous counterpart and solve the obtained convex program to obtain a lower-bound on the exact optimal value. Projection of this optimal solution onto the discrete set will then give an upper-bound when the projection satisfies the feasibility constraints. This projection would simply be a rounding step toward the closest discrete value in each dimension. It has been suggested in [69] that one can use the information from the dual problem through Alternating Direction Method of Multipliers (ADMM) to do several passes of such steps, which will result in a better upper-bound, which will re-visit in Section 5.2. Although ADMM has been originally developed for convex problems [70], there has been much recent interest in applying it to the non-convex problems, with some analysis of convergence available in some cases [71]. This has led to a broad class of heuristics with great flexibility for problems that were originally very hard to solve. In particular, the binary quadratic problems (BQP) has attracted much attention and authors of [69] have demonstrated their algorithm for this class of problems, which can simply be extended to a more generalized setting as in [72].

We derive a new class of methods in Section 5.3 as was first done in [73], whereby the introduction of an auxiliary equality constraint that captures the inner part of the objective would be the base for the suggested ADMM-based algorithm. The algorithm finds the best discrete variable in each step by checking the captured part values at the discrete values (rather than rounding). When the effect of the discrete variable can be decoupled in the inner function, the best discrete value at each step can be obtained through a linear number of function evaluations in the dimension of the discrete variable, versus an exponential number that corresponds to the exhaustive search. This has shown significant improvements when the objective is not necessarily symmetrical around its optimal point, and when a certain separability condition is met. We will discuss when and how this could capture the effect of the discrete variables better in Section 5.3.1.

It is still possible to capture these improvements even when this separability condition is not met and the discrete variable enters through a coupling matrix. We will explore various hybrid methods that decouple the discrete variables approximately while preserving the same linear per-iteration complexity for the discrete variable update in Section 5.4. Numerical comparisons have indicated that one class of such hybrid algorithms that only linearizes the effect of the non-dominant part of the coupling matrix exhibits clear improvements in performance.

5.1 Problem Formulation

We formulate the problem of interest of this chapter in this section. We consider objectives that have mixed discrete and continuous parts and then discuss the generalizations and constraints wherever applicable.

Consider the following optimization problem:

$$\text{minimize} \quad f(g(x, z)), \tag{5.1}$$

with variables $x \in \mathbb{R}^n$ and $z \in \mathcal{Z}^{(m)}$. Throughout the rest of this chapter, the inner function $g(\cdot,\cdot)$ is from $\mathbb{R}^n \times \mathcal{Z}^{(m)}$ to $\mathbb{R}^p$. The extended function $f(\cdot)$ is from $\mathbb{R}^p$ to $\bar{\mathbb{R}}$ and is assumed to be convex in its variable.

Even without any further constraints, and even if $f \circ g$ is convex, this would typically be a hard problem due to the presence of the discrete variable z.

Assumption 103. *We will assume that $g(\cdot,\cdot)$ is affine in its variables, i.e.,:*

$$g(x,z) \;=\; Az + Bx + b. \tag{5.2}$$

Remark 104 (affine g)**.** *It is important to mentioned that this assumption does not have to be this restrictives, and could be extended to include a more generalized class of functions.*

Remark 105 (constraints)**.** *All the methods that we will develop, with exception of the Direct Evaluation method (Method 6), allow incorporating equality and inequality constraints in a modestly straightforward fashion. However, in order to avoid illustrating the main concepts with heavy notations, we will not include them in this chapter.*

5.2 A Round-off Based Algorithm

We will reformulate a class of relax-and-round heuristics for mixed integer nonlinear programs in this section. We will then discuss how introduction a new axillary variable could result in a set of different ADMM-based algorithms that could show significant improvements later.

We can rewrite the optimization problem (5.1) with an extra constraint such that instead of having the discrete variable in the objective, we will have it in the constraints:

$$\begin{aligned} &\text{minimize} && f(g(x,y)) \\ &\text{subject to} && y = z, \end{aligned} \tag{5.3}$$

with variables $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$ and $z \in \mathcal{Z}^{(m)}$.

The augmented Lagrangian for this problem, for any parameter $\rho > 0$, can be written as:

$$L_\rho^{\mathrm{R}}(x, y, z, \nu) = f(g(x, y)) + \nu^T(y - z) + \frac{\rho}{2}\|y - z\|_2^2,$$

which with some basic rearrangement of the term and a change of variable $\mu = (1/\rho)\nu$ can be equivalently written as:

$$L_\rho^{\mathrm{R}}(x, y, z, \mu) = f(g(x, y)) + \frac{\rho}{2}\|y - z + \mu\|_2^2 - \frac{\rho}{2}\|\mu\|_2^2.$$

The resulting ADMM algorithm based on this augmented Lagrangian would then consist of joint optimization over the variables (x, y), projecting onto the discrete set $\mathcal{Z}^{(m)}$, and the dual update:

Method 5 (Relax-and-Round)**.**

$$\begin{aligned} (x^{(k+1)}, y^{(k+1)}) &= \arg \min_{\substack{x\in\mathbb{R}^n \\ y\in\mathbb{R}^m}} \; L^{\mathrm{R}}_{\rho}(x, y, z^{(k)}, \mu^{(k)}) \\ z^{(k+1)} &= \Pi_{\mathcal{Z}^{(m)}} \left(y^{(k+1)} + \mu^{(k)} \right) \\ \mu^{(k+1)} &= \mu^{(k)} + y^{(k+1)} - z^{(k+1)}. \end{aligned}$$

The second and third steps of the ADMM are straightforward, and the first step requires convexity of $f(g(x, y))$ in (x, y), which in turn could be guaranteed by the Assumption 103.

This heuristic was considered for the binary quadratic problems in [69], and generalized to allow for some other kinds of mixed integer non-linear programs in [72]. The objective of [69] has the form:

$$f(v) \;=\; 1/2 \; v^T P v + q^T v, \tag{5.4}$$

with positive semidefinite P. This can be matched to (5.3) by choosing:

$$g(x, z) \;=\; z, \tag{5.5}$$

which indicates that there is no continuous variable. Also it is noteworthy that considering another affine function for $g(x, z)$ other than the one mentioned above, such as $g(x, z) = Az + b$, would be in effect the same as a new quadratic function with P being changed to $A^T PA$, and q to $A^T q + A^T Pb$, while still having $g(x, z) = z$.

Although this suggests that considering the composition of functions in (5.3) might not be fundamentally different in the quadratic case as it would yield another similar quadratic function, this is not what one would generally observe in non-quadratic cases. We will describe this aspect in the next section.

5.3 Separable Binary Variables

We will describe the auxiliary variable introduction in this section. We will first lay out the modification to the ADMM algorithm when the A matrix has only one non-zero element in each of its rows and describe its advantages for this special case in Section 5.3.1, and then consider different generalizations and compare them in later sections.

We will begin by putting a different assumption on the function $g(\cdot,\cdot)$. This assumption is more restrictive than Assumption 103 in the sense that it requires partial separability over the discrete variable, Particularly we will require that each element of g would depend on at most one discrete variable:

Assumption 106. *Throughout the rest of this section (and only this section) we will assume that g satisfies Assumption 103 and is such that each g_i depend only on a single discrete variable, i.e., for all $i \in \{1, \cdots, p\}$ there exists a single $\ell_i \in \{1, \cdots, m\}$ such that $g_i(x,z) = \tilde{g}_i(x, z_{\ell_i})$, for some $\tilde{g}_i : \mathbb{R}^n \times \mathcal{Z}_{\ell_i} \mapsto \mathbb{R}$, and for all $x \in \mathbb{R}^n$ and $z \in \mathcal{Z}^{(m)}$.*

Remark 107. *In matrix form, Assumption 106 can be equivalently written as*

$g(x,z) = Dz + Bx + b$, where $D \in \mathbb{R}^{p \times m}$ is such that:

$$D_{ij} \neq 0 \Longrightarrow j = \ell_i. \tag{5.6}$$

With this assumption in place, we will consider the ADMM algorithm that will be based on the following equivalent form of (5.1):

$$\begin{aligned} &\text{minimize} && f(v) \\ &\text{subject to} && v = g(x,z), \end{aligned} \tag{5.7}$$

with variables $x \in \mathbb{R}^n$, $v \in \mathbb{R}^p$ and $z \in \mathcal{Z}^{(m)}$.

The augmented Lagrangian for this problem can be written as:

$$L_\rho^{\mathrm{D}}(x,v,z,\mu) = f(v) + \frac{\rho}{2}\|v - g(x,z) + \mu\|_2^2 - \frac{\rho}{2}\|\mu\|_2^2,$$

for which the ADMM algorithm would be:

Method 6 (Direct Evaluation)**.**

$$\begin{aligned} (x^{(k+1)}, v^{(k+1)}) &= \arg \min_{\substack{x \in \mathbb{R}^n \\ v \in \mathbb{R}^p}} \ L_\rho^{\mathrm{D}}(x, v, z^{(k)}, \mu^{(k)}) \\ z^{(k+1)} &= \arg \min_{z \in \mathcal{Z}^{(m)}} \ \|v^{(k+1)} - g(x^{(k+1)}, z) + \mu^{(k)}\|_2^2 \\ \mu^{(k+1)} &= \mu^{(k)} + v^{(k+1)} - g(x^{(k+1)}, z^{(k+1)}). \end{aligned}$$

The first step of this algorithm would be a convex minimization step due to the convexity of $f(\cdot)$ and Assumption 103. The second step is the origin of the difference from the Relax-and-Round Method 5. In particular, this step would not correspond to a similar projection as in the Relax-and-Round method, yet it would remain computationally tractable due to the dependency of g on a single discrete variable (Assumption 106). This is explicitly stated in the following theorem:

Theorem 108. *Given a function g that satisfies Assumption 106, the optimal z in Method 6 (Direct Evaluation) can be equivalently obtained by independently solving for each of the discrete variables, i.e., for all $j \in \{1, \cdots, m\}$ we have that:*

$$z_j{}^{(k+1)} = \arg\min_{z_j \in \mathcal{Z}_j} \sum_{\{i \,|\, \ell_i = j\}} \left(v_i{}^{(k+1)} - \tilde{g}_i(x^{(k+1)}, z_j) + \mu_i{}^{(k)} \right)^2. \tag{5.8}$$

Proof. We have that:

$$\begin{aligned} z^{(k+1)} &= \arg\min_{z \in \mathcal{Z}^{(m)}} \; L_\rho^{\mathrm{D}}(x^{(k+1)}, v^{(k+1)}, z, \mu^{(k)}) \\ &= \arg\min_{z \in \mathcal{Z}^{(m)}} \; \|v^{(k+1)} - g(x^{(k+1)}, z) + \mu^{(k)}\|_2^2 \\ &= \arg\min_{z \in \mathcal{Z}^{(m)}} \; \sum_{i=1}^{p} \left(v_i{}^{(k+1)} - g_i(x^{(k+1)}, z) + \mu_i{}^{(k)}\right)^2 \\ &= \arg\min_{z \in \mathcal{Z}^{(m)}} \; \sum_{i=1}^{p} \left(v_i{}^{(k+1)} - \tilde{g}_i(x^{(k+1)}, z_{\ell_i}) + \mu_i{}^{(k)}\right)^2 \\ &= \arg\min_{z \in \mathcal{Z}^{(m)}} \; \sum_{j=1}^{m} \sum_{\{i \,|\, \ell_i = j\}} \left(v_i{}^{(k+1)} - \tilde{g}_i(x^{(k+1)}, z_{\ell_i}) + \mu_i{}^{(k)}\right)^2 \end{aligned}$$

$$\Longrightarrow$$

$$z_j{}^{(k+1)} = \arg\min_{z_j \in \mathcal{Z}_j} \sum_{\{i \,|\, \ell_i = j\}} \left(v_i{}^{(k+1)} - \tilde{g}_i(x^{(k+1)}, z_j) + \mu_i{}^{(k)} \right)^2,$$

where the second equality follows because that is the only term involving the discrete variable z, the third is due to the fact that $\|\cdot\|_2^2$ also separates in its elements, the forth is due to Assumption 106, and the fifth is an equivalent representation of the sum from $i = 1$ to p. □

This alternative approach makes the computation of z-update minimization a tractable one whenever Assumption 106 is in place. This is described in more details in the following remark:

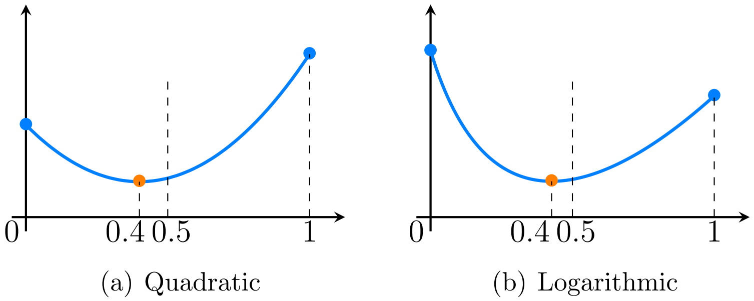
Remark 109 (Per-iteration complexity)**.** *The z-update in the Direct Evaluation method (Method 6) requires $|\mathcal{Z}^{(m)}| = |\mathcal{Z}_1| \times \cdots \times |\mathcal{Z}_m|$ function evaluations (for instance, 2^m in the binary case), whereas when Assumption 106 is satisfied, solving for z by (5.8) only requires $\sum_{j=1}^{m} |\mathcal{Z}_j|$ function evaluations (for instance, $2m$ in the binary case). This alternative step has linear complexity, and is comparable in complexity to the projection step in Method 5.*

Remark 110 (Matrix variables)**.** *This can be easily generalized to handle matrix variables by replacing the 2-norm with the Frobenius norm, which corresponds to the standard inner product in the matrix spaces.*

5.3.1 Discussions

We will provide more intuitions on the suggested modifications to the ADMM-based algorithm described above in this section.

In the Relax-and-Round method (Method 5), the discrete variable is replaced by a continuous one, and the solution of the primal optimization step that is solved with these continuous variables is then projected onto the discrete set in the hope that this projection would still minimize $f(g(x^{(k+1)}, z))$ for the discrete variable z, which might be not the case in general. The quadratic objective (5.4) is symmetrical around its optimal point in each of the directions, and thus the projection would be a best choice when one requires separability in the z-update. In other words, as illustrated in Figure 5.1(a), when we keep all the variables fixed except for a single one-dimensional discrete variable, the discrete value (0 or 1 in here) that minimizes a quadratic function is indeed the one closest to its critical point (0.4 here).



(a) Quadratic

(b) Logarithmic

Figure 5.1: Comparing relax-and-round and Direct Evaluation in a single dimension for different convex functions

However, as illustrated in Figure 5.1(b), this special property might not be in place for a wide variety of convex functions such as piecewise linear, sum of logarithmics or sum of exponential functions. The proposed ADMM-based method also separates in the z-update, and compared to rounding the solution of the relaxed problem, it will actually plug in the binary values and picks the best among them, making it more likely that it would be a better choice for non-quadratic functions. This would be further investigated through numerical examples in the next section.

5.3.2 Numerical Examples

We will investigate the Relax-and-Round (Method 5) and Direct Evaluation (Method 6) for random instances of a problem with fixed dimensions, and compare the run-time and the value that each algorithm obtains in our the following example:

Example 111. *Consider the optimization problem* (5.1) *with a single continuous variable* ($n = 1$) *and where the discrete variables are all in the binary space* ($\mathcal{Z}_i = \mathbb{B}$, *for* $i = 1. \cdots, m$)*, i.e.,* $x \in \mathbb{R}$ *and* $z \in \mathbb{B}^m$*. Let* $g(\cdot, \cdot)$ *be given as:*

$$g(x, z) \;=\; Dz + b + \boldsymbol{1}\, x, \tag{5.9}$$

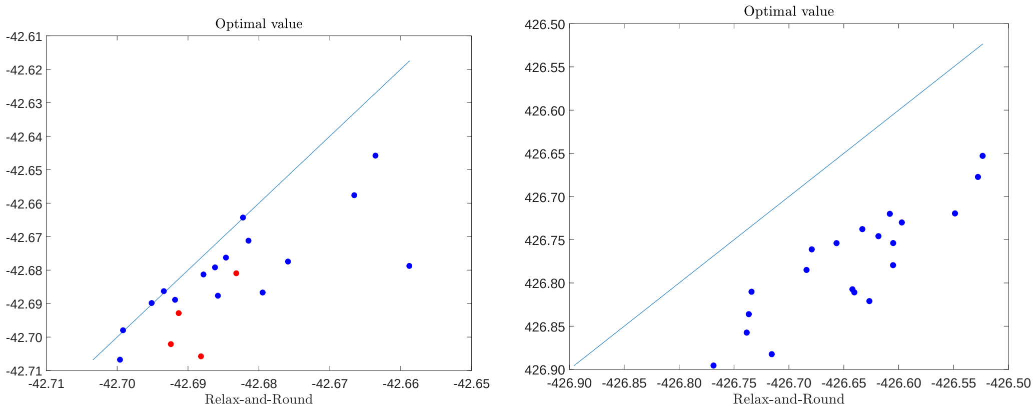
where D *is a diagonal matrix in* $\mathbb{R}^{m \times m}$, $b \in \mathbb{R}^m$ *and* $\boldsymbol{1}$ *is a vector of all ones of compatible dimension. Let* $p = m$ *and also take* $f(\cdot)$ *as:*

$$f(v) \;=\; \sum_{i=1}^{m} -a_1 \log(a_0 v_i + c_1) - a_2 \log(-a_0 v_i + c_2), \tag{5.10}$$

where $a_0, a_1, a_2, c_1,$ *and* c_2 *are all positive real numbers, and* c_1 *and* c_2 *are such that* $[0, 1]$ *is in the domain. This function is convex in its domain and resembles the one illustrated in Figure 5.1(b). In this example we consider two cases of* $m = 10$ *and* $m = 100$*, for each we generate 20 instances of* (5.9) *with random* b *and diagonal* D *of compatible dimension, and solve the optimization problem* (5.1) *by Methods 5 and 6, with* ρ *being fixed to* 0.5*.*

Figure 5.2 shows the optimal value obtained from the Relax-and-Round method versus the Direct Evaluation for $m = 10$ *and 100. The* x*-axis corresponds to the Relax-and-Round method and the* y*-axis is for the Direct Evaluation. The blue dots indicate when Direct Evaluation was faster and the red dots indicate when the Relax-and-Round was faster. Each dot below the* $y = x$ *solid line means that the Di-*

rect Evaluation has obtained a lesser value. This means that the Direct Evaluation method has shown better performance in the 20 considered samples when $m = 100$*, and mostly when* $m = 10$*.*

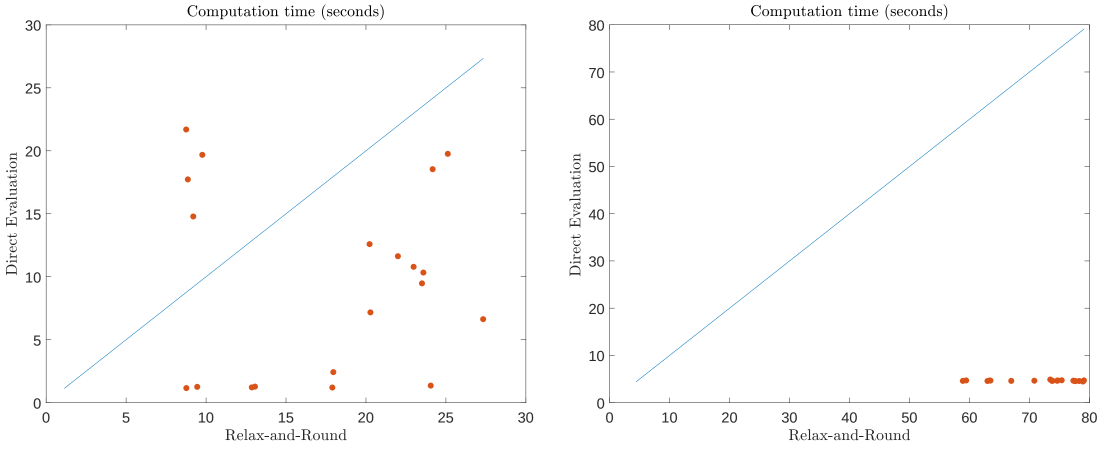


(a) For $m = 10$

(b) For $m = 100$

Figure 5.2: Comparison of the optimal values

Next, we plot the computation time required to get to these values in Figure 5.3. Similar to the previous figure, every point below the $y = x$ *line indicates that the Direct Evaluation has taken less time. Iteration counts that each of the methods take to get to these points are also illustrated in Figure 5.4.*

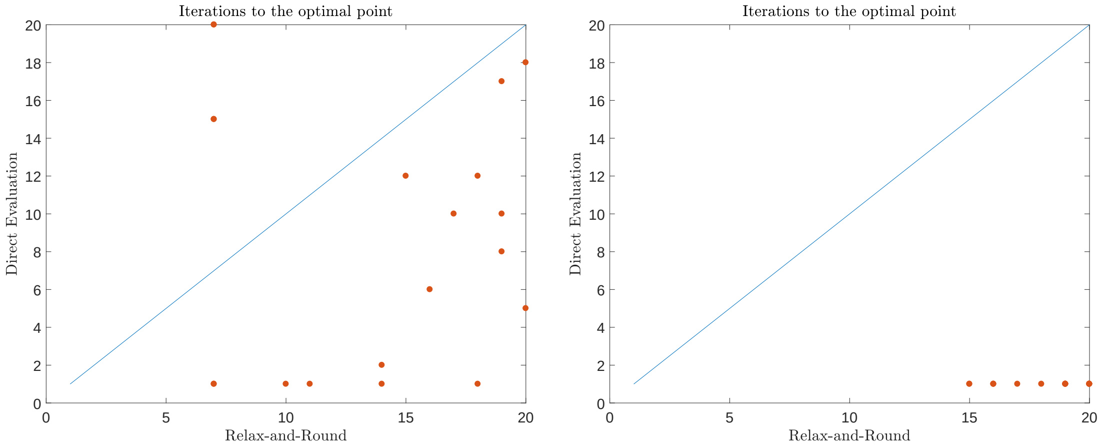


(a) For $m = 10$

(b) For $m = 100$

Figure 5.3: Comparison of the computational time (in seconds)

We will now inspect how these two methods compare to the exact solution.



(a) For $m = 10$

(b) For $m = 100$

Figure 5.4: Iterations to the optimal point

When $m = 10$*, the problem is small enough that we can find the exact solution by exhaustive search over* 2^{10} *instances of* (5.1) *with fixed* z *in each instance. Name the optimal binary solution that corresponds to the exact exhaustive search by* $z^{(\boldsymbol{ex})}$*, the one that corresponds to the Relax-and-Round method by* $z^{(\boldsymbol{round})}$*, and the one that corresponds to the Direct Evaluation method by* $z^{(\boldsymbol{direct})}$*. How much these binary values differ is illustrated in Figure 5.5, where the x-axis denotes the sample index, the blue dots show on how many elements* $z^{(\boldsymbol{ex})}$ *and* $z^{(\boldsymbol{direct})}$ *are different (*$\|z^{(\boldsymbol{direct})} - z^{(\boldsymbol{ex})}\|_1$*), and the red dots show the same for* $z^{(\boldsymbol{round})}$*, i.e.,* $\|z^{(\boldsymbol{round})} - z^{(\boldsymbol{ex})}\|_1$*. As illustrated in Figure 5.5, Direct Evaluation Method 6 has recovered closer discrete variables to the exact solution in most cases, although it might happen that in a few cases Relax-and-Round would be better (as in sample 2).*

In the next example we will vary the problem size and inspect how the two methods compare.

Sum-of-Logs f**:** We consider the optimization problem (5.1) again, and

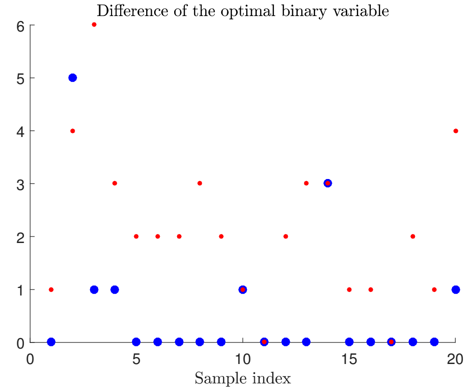


Figure 5.5: Comparison to the exact solution when $m = 10$

take $f(\cdot)$ and $g(\cdot,\cdot)$ as (5.10) and (5.9). In this example, we will vary m from 5 to 100. For each m, we will generate 20 instances of (5.9) with random b and diagonal D of compatible dimension, and then plot the optimal value, computational time, and the iterations to the optimal point.

Figure 5.6 plots the the difference of the optimal value obtained by Relax-and-Round Method 5 from the Direct Evaluation Method 6. The minimum of this difference (purple line) is almost always positive, except for 39 times out of all 1920 simulations (2%). This indicates that the Direct Evaluation has mostly performed better for the considered functions, which satisfy Assumption 106. The black line denotes the average, the solid green denotes the median, whereas the dashed greens denote the 5% and 95% percentiles for this difference.

Figures 5.7(a) and 5.7(b) compare the computational time and iteration count required for each of these methods to reach the optimal point. As shown in these figures, the Direct Evaluation exhibits better performance on average. Also, the number of iterations required to get to a local solution decreases for the Direct Evaluation as the problem size gets bigger. This could be the case as the effect of

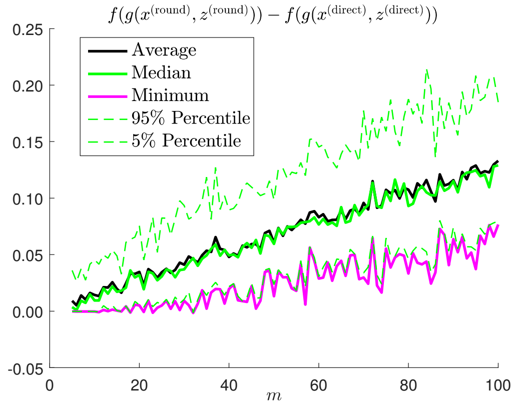
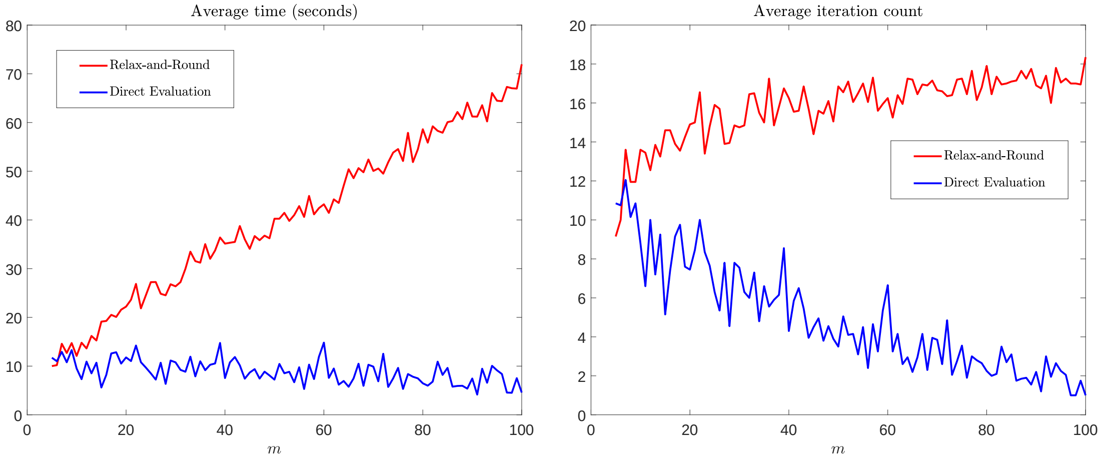


Figure 5.6: Various statistics for the difference of the optimal values

the single continuous variable x decreases as the dimension of the problem increases, and hence the initial iterations that directly solve for the binary variables would be more crucial as m increases.



(a) Time to the optimal point

(b) Required iterations

Figure 5.7: Time and iterations required to obtain a local solution

Finally we compare the exact solution to the these two methods. This was only an option when the problem size was small enough ($m < 15$). We see that, as illustrated in Figure 5.8, Direct Evaluation solutions are closer to the exact value.

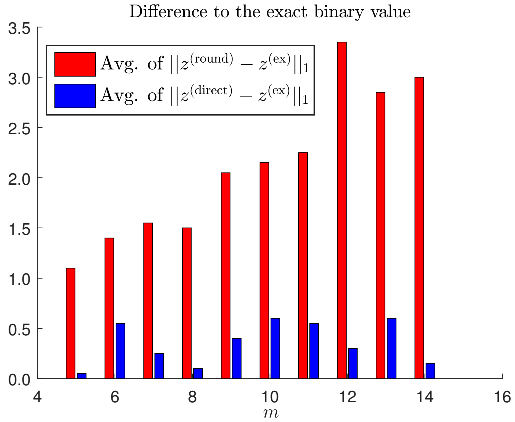


Figure 5.8: Comparison to the exact solution for $5 \leq m \leq 14$

5.4 Presence of a Mixing Matrix

We will discuss generalization of the Direct Evaluation Method 6 for cases when Assumption 106 is not met in this section. It is obvious that mixing the discrete variables as in Az for a general A does not allow one to separately solve the $z-$update for different discrete variables. Similar scenario arises even in the convex cases where one already has a proximal operator that finds the argmin of $h(w) + \frac{\varrho}{2}\|y - w\|_2^2$ with continuous variable w for some convex function h, and wants to approximate the proximal operator to find the argmin of $h(w) + \frac{\varrho}{2}\|y - Aw\|_2^2$. We will benefit from existing suggested ideas [74, 75] that address approximation of proximal operators in presence of a mixing matrix, and adopt them to our problem in Section 5.4.1. This full linearization decouples the effect of the discrete variable even more than what we actually need to preserve the linear per-iteration complexity. Hence, we decompose the coupling matrix into its dominant and non-dominant part and only partially linearize the non-dominant part using a less conservative hybrid approach up to the extent that the linear per-iteration complexity would still be achievable in Section 5.4.2. This method has shown significant improvements in comparison to the other approaches, and to the Relax-and-Round Method 5 in presence of a mixing matrix when the mixing matrix exhibits a mild degree of diagonal dominance. Finally, we will also consider another hybrid approach that applies the relax-and-round algorithm only to the non-dominant part of the coupling matrix, while directly solves the dominant part in Section 5.4.3.

5.4.1 Full Linearization

We will focus on linearizing the effect of the mixing matrix in this section. This approach was first suggested to facilitate approximation of the proximal operators where one wants to obtain the proximal of $h(w) + \frac{\rho}{2}\|y - Aw\|_2^2$ when A is not necessarily the identity matrix. More precisely, we will *linearize* the augmented Lagrangian around the most recent point [75]. This method, also known as Bregmanized Operator Splitting (BOS) [74,76], decouples the effect of the A matrix and allows that the same proximal operator (in absence of A) be applied to obtain an approximate solution in presence of a general A.

Through the rest of this chapter we will assume that g only satisfies Assumption 103, i.e., $g(x, z) = Az + Bx + b$ with a general A. We will adopt the same idea to linearize the augmented Lagrangian $L_\rho^{\mathrm{D}}(x^{(k+1)}, v^{(k+1)}, z, \mu^{(k)})$ for problem (5.7) around the point $z^{(k)}$ to facilitate obtaining an approximate solution for the $z-$update that will have the same desirable separability property (as in Remark 109). To this end, for a parameter $\alpha > 0$ define:

$$\begin{aligned} L_\rho^{\mathrm{FL},z^{(k)}}(x^{(k+1)}, v^{(k+1)}, z, \mu^{(k)}) \;&\triangleq \\ &-\rho\Big(A^T\big(v^{(k+1)} - g(x^{(k+1)}, z^{(k)}) + \mu^{(k)}\big)\Big)^T z + \tfrac{\alpha}{2}\|z - z^{(k)}\|_2^2 + c_1, \end{aligned}$$

where the quadratic term $(\frac{\alpha}{2}\|z - z^{(k)}\|_2^2)$ has been added to preserve the strong convexity, and c_1 captures all the constant terms that do not depend on the optimization variable z. With addition of some constants, this could be equivalently written as:

$$\begin{aligned} L_\rho^{\mathrm{FL},z^{(k)}}(x^{(k+1)}, v^{(k+1)}, z, \mu^{(k)}) \;&=\; c_2 + \\ &\tfrac{\alpha}{2}\|z - z^{(k)} - \tfrac{\rho}{\alpha}A^T\big(v^{(k+1)} - g(x^{(k+1)}, z^{(k)}) + \mu^{(k)}\big)\|_2^2, \end{aligned} \tag{5.11}$$

where c_2 captures all the constant terms that do not depend on the optimization variable z.

It is important to note that this linearized version only replaces the augmented Lagrangian in the z-update, and the first step uses a standard augmented Lagrangian, this is illustrated in the following method.

Method 7 (Fully Linearized)**.**

$$\begin{aligned} (x^{(k+1)}, v^{(k+1)}) \;&=\; \arg\min_{\substack{x\in\mathbb{R}^n \\ v\in\mathbb{R}^p}} \; L_\rho^{\mathrm{D}}(x, v, z^{(k)}, \mu^{(k)}) \\ z^{(k+1)} \;&=\; \arg\min_{z\in\mathcal{Z}^{(m)}} \; L_\rho^{\mathrm{FL},z^{(k)}}\Big(x^{(k+1)}, v^{(k+1)}, z, \mu^{(k)}\Big) \\ \mu^{(k+1)} \;&=\; \mu^{(k)} + v^{(k+1)} - g(x^{(k+1)}, z^{(k+1)}). \end{aligned}$$

It is clear that the z-update step in Method 7 also separated for different discrete variables, i.e.:

$$z_j^{(k+1)} \;=\; \Pi_{\mathcal{Z}_j}\Big(z_j^{(k)} + \frac{\rho}{\alpha}\Big[A^T\big(v^{(k+1)} - g(x^{(k+1)}, z^{(k)}) + \mu^{(k)}\big)\Big]_j\;\Big),$$

for $j \in \{1, \cdots, m\}$. This has the same per-iteration complexity as the Direct Evaluation (Method 6).

The full linearizion of the augmented Lagrangian, in summary, yields a quadratic term of the form $\|z - \bar{z}\|_2^2$, where the $\bar{z}$ denotes the constant terms that do not dependent on z. This is more conservative than what we can actually solve, and as illustrated in Section 6, we can solve the z-update while preserving linear per-iteration complexity even in presence of a more generalized quadratic term as in $\|Dz - \bar{z}\|_2^2$, where D satisfies Assumption 106. Thus we consider a less conservative approach that will exploit this capability in the next two section.

5.4.2 Partial Linearization

We want to approximate the augmented Lagrangian for the z-update up to the extent that we can actually solve the resulting form with linear per-iteration complexity. We will consider a hybrid approach between the Direct Evaluation (Method 6) and Full Linearization (Method 7), for a general A in this section.

We want to separate the effect of the A matrix by decomposing it into two components, where the first component captures the most dominant element in each row, and the second component captures the rest. To this end, let $A = U\Sigma V^T$ be a singular value decomposition for A with $U^TU = V^TV = I$, and define $S \triangleq \Sigma V^T$.

For each row $i \in \{1, \cdots, p\}$, let

$$\ell_i = \arg \max_{j \in \{1, \cdots, m\}} |S_{ij}|$$

denote the index of the maximum-size element in i-th row of S. With a slight abuse of notation, we will assume that ℓ_i remains a singleton set for each i, i.e., whenever multiple elements in a row correspond to the maximum size in that row, we will pick and fix ℓ_i to point to a single one of them. Let $D \in \mathbb{R}^{p \times m}$ denote the dominant part of S and define it as:

$$D_{ij} \triangleq \begin{cases} S_{ij} & j = \ell_i \\ 0 & \text{otherwise,} \end{cases}$$

and let $\tilde{D} = S - D$. Similarly, let $A_D \triangleq UD$ and $A_{\tilde{D}} \triangleq U\tilde{D} = A - A_D$.

The following theorem will utilize this decomposition to rewrite the augmented Lagrangian (when only Assumption 103 is in place) in a form that allows us to perform partial linearization only for needed part.

Theorem 112. *The augmented Lagrangian $L_\rho^{\mathrm{D}}(x^{(k+1)}, v^{(k+1)}, z, \mu^{(k)})$ for problem (5.7) can be equivalently written as:*

$$L_\rho^{\mathrm{D}}(x^{(k+1)}, v^{(k+1)}, z, \mu^{(k)}) = \frac{\rho}{2}\Big(\|\bar{q}_{\mathrm{PL}}^{(k)} - Dz\|_2^2 - 2(\tilde{D}^T \bar{q}_{\mathrm{PL}}^{(k)})^T z + z^T \tilde{D}^T (\tilde{D} + 2D) z\Big) + c_4, \tag{5.12}$$

where $\bar{q}_{\mathrm{PL}}^{(k)} \triangleq U^T \left(v^{(k+1)} - (Bx^{(k+1)} + b) + \mu^{(k)}\right)$, and c_4 captures the terms that do *not depend on the variable z.*

Proof. We can write:

$$\begin{aligned} L^{\mathrm{D}}_{\rho}(x^{(k+1)}, v^{(k+1)}, z, \mu^{(k)}) &= \tfrac{\rho}{2}\|v^{(k+1)} - g(x^{(k+1)}, z) + \mu^{(k)}\|_2^2 + c_3 \\ &= \tfrac{\rho}{2}\|v^{(k+1)} - (Az + Bx^{(k+1)} + b) + \mu^{(k)}\|_2^2 + c_3 \\ &= \tfrac{\rho}{2}\|\bar{w}^{(k)}_{\mathrm{PL}} - Az\|_2^2 + c_3 \\ &= \tfrac{\rho}{2}\|\bar{w}^{(k)}_{\mathrm{PL}} - A_D z - A_{\tilde{D}} z\|_2^2 + c_3 \\ &= \tfrac{\rho}{2}\|\bar{q}^{(k)}_{\mathrm{PL}} - Dz - \tilde{D}z\|_2^2 + c_3 \end{aligned}$$

where the first and second equalities follow due to the definition. The third follows by rearranging the terms and defining $\bar{w}^{(k)}_{\mathrm{PL}} \triangleq v^{(k+1)} - (Bx^{(k+1)} + b) + \mu^{(k)}$, the forth follows as $A = A_D + A_{\tilde{D}}$, and the fifth follows due to the invariance of the 2-norm under unitary multiplications ($\|U^T x\|_2 = \|x\|_2$ for unitary $U^T U = I$). Then, (5.12) follows by expansion of the terms of the last line. □

We only want to linearize the part of the augmented Lagrangian that involves the non-dominant component of A, i.e., linearizing the second line of (5.12) around $z^{(k)}$ to have:

$$L^{\mathrm{PL},z^{(k)}}_{\rho}(x^{(k+1)}, v^{(k+1)}, z, \mu^{(k)}) = \tfrac{\rho}{2}\|\bar{q}^{(k)}_{\mathrm{PL}} - Dz\|_2^2 + (\bar{r}^{(k)})^T z + c_5, \tag{5.13}$$

where $\bar{r}^{(k)} \triangleq 2\Big(\tilde{D}^T(\tilde{D} + 2D)z^{(k)} - \tilde{D}^T\bar{q}^{(k)}_{\mathrm{PL}}\Big)$ represents the gradient of the linearized part, and c_5 captures the constant terms that do not depend on z. This would result in the following method:

Method 8 (Partially Linearized)**.**

$$\begin{aligned}
(x^{(k+1)}, v^{(k+1)}) &= \arg\min_{\substack{x\in\mathbb{R}^n\\ v\in\mathbb{R}^p}} \ L_\rho^{\mathrm{D}}(x, v, z^{(k)}, \mu^{(k)})\\
z^{(k+1)} &= \arg\min_{z\in\mathcal{Z}^{(m)}} \ L_\rho^{\mathrm{PL},z^{(k)}}\Big(x^{(k+1)}, v^{(k+1)}, z, \mu^{(k)}\Big)\\
\mu^{(k+1)} &= \mu^{(k)} + v^{(k+1)} - g(x^{(k+1)}, z^{(k+1)}).
\end{aligned}$$

Corollary 113. *The z-update in Method 8 can be solved separately for each of the discrete variables, i.e., for each $j \in \{1, \cdots, m\}$ we have that:*

$$z_j^{(k+1)} = \begin{cases} \arg\min\limits_{z_j\in\mathcal{Z}_j} \ \bar{r}_j^{(k)} z_j + \sum\limits_{\{i \ | \ \ell_j = i\}} \big((\bar{q}_{\mathrm{PL}}^{(k)})_i - D_{ij} z_j\big)^2 & \text{if } \{i \mid \ell_j = i\} \neq \varnothing,\\[2ex] \arg\min\limits_{z_j\in\mathcal{Z}_j} \ \bar{r}_j^{(k)} z_j & \text{if } \{i \mid \ell_j = i\} = \varnothing. \end{cases}$$

Proof. Proof is very similar to that of Theorem 108. □

Remark 114 (zero row of D)**.** *It follows that if $D_{i\ell_i} = 0$, then the i^{th} row of D (and consequently the i^{th} row of $\tilde{D}$) would be zero too, meaning that there would be no discrete variable present in the i-element of the vector $\bar{q}_{\mathrm{PL}}^{(k)} - Dz - \tilde{D}z$. As this element would be an extra constant term, we can drop that row and equivalently assume that $D_{i\ell_i} \neq 0$ for all $i \in \{1, \cdots, p\}$.*

Remark 115. *When A has only one non-zero element in each of its rows, Partial Linearization (Method 8) recovers the Direct Evaluation (Method 6).*

5.4.3 Partial Rounding

We will also consider a second hybrid algorithm between the full relax-and-rounding (Method 5) and the Direct Evaluation (Method 6) for a general A in this section.

We will decompose the A matrix as in Section 5.4.2 and then introduce a new variable to capture the discrete variable that enters through non-dominant component of A. To this end we will write the optimization problem (5.1) as:

$$\begin{array}{ll} \text{minimize} & f(v) \\ \text{subject to} & v = A_D z + A_{\tilde{D}} y + Bx + b, \\ & y = z, \end{array} \tag{5.14}$$

with variables $x \in \mathbb{R}^n$, $v \in \mathbb{R}^p$, $y \in \mathbb{R}^m$, and $z \in \mathcal{Z}^{(m)}$.

The augmented Lagrangian with $\rho > 0$ for this problem can be written as:

$$\begin{aligned} L_\rho^{\text{PR}}(x, v, y, z, \mu, \eta) = f(v) + \tfrac{\rho}{2}\|y - z + \eta\|_2^2 + \tfrac{\rho}{2}\|v - (A_D z + A_{\tilde{D}} y + Bx + b) + \mu\|_2^2 \\ -\tfrac{\rho}{2}\|\eta\|_2^2 - \tfrac{\rho}{2}\|\mu\|_2^2 \end{aligned}$$

which would result in the following partial rounding method:

Method 9 (Partial Rounding)**.**

$$\begin{aligned}
(x,v,y)^{(k+1)} &= \arg\min_{\substack{x\in\mathbb{R}^n\\ v\in\mathbb{R}^p\\ y\in\mathbb{R}^m}} L_\rho^{\mathrm{PR}}\Big(x,v,y,z^{(k)},\mu^{(k)},\eta^{(k)}\Big)\\
z^{(k+1)} &= \arg\min_{z\in\mathcal{Z}^{(m)}} L_\rho^{\mathrm{PR}}\Big(x^{(k+1)},v^{(k+1)},y^{(k+1)},z,\mu^{(k)},\eta^{(k)}\Big)\\
\mu^{(k+1)} &= \mu^{(k)}+v^{(k+1)}-\big(A_D z^{(k+1)}+A_{\tilde{D}}y^{(k+1)}+Bx^{(k+1)}+b\big)\\
\eta^{(k+1)} &= \eta^{(k)}+y^{(k+1)}-z^{(k+1)}.
\end{aligned}$$

Corollary 116. *The z-update in Method 9 could be separately solved for each of the discrete variables, i.e, we have that:*

$$z_j{}^{(k+1)} = \begin{cases} \arg\min\limits_{z_j\in\mathcal{Z}_j}\ \big(y_j^{(k+1)}-z_j+\eta_j^{(k)}\big)^2+\sum\limits_{\{i\ \mid\ \ell_j=i\}}\big((\bar{q}_{\mathrm{PR}}^{(k)})_i-D_{ij}z_j\big)^2 & \\ & \text{if } \{i\mid \ell_j=i\}\neq\varnothing,\\ \arg\min\limits_{z_j\in\mathcal{Z}_j}\big(y_j^{(k+1)}-z_j+\eta_j^{(k)}\big)^2 & \text{if } \{i\mid \ell_j=i\}=\varnothing,\end{cases}$$

where

$$\bar{q}_{\mathrm{PR}}^{(k)} \triangleq U^T\Big(v^{(k+1)}-(A_{\tilde{D}}y^{(k+1)}+Bx^{(k+1)}+b)+\mu^{(k)}\Big).$$

Proof. Define

$$\bar{w}_{\mathrm{PR}}^{(k)} \triangleq v^{(k+1)}-(A_{\tilde{D}}y^{(k+1)}+Bx^{(k+1)}+b)+\mu^{(k)},$$

then

$$
\begin{aligned}
&\tfrac{2}{\rho} L_\rho^{\mathrm{PR}}(x^{(k+1)}, v^{(k+1)}, y^{(k+1)}, z, \mu^{(k)}, \eta^{(k)}) \\
&\qquad = \|y^{(k+1)} - z + \eta^{(k)}\|_2^2 + \|\bar{w}_{\mathrm{PR}}^{(k)} - A_D z\|_2^2 + c_6 \\
&\qquad = \|y^{(k+1)} - z + \eta^{(k)}\|_2^2 + \|\bar{q}_{\mathrm{PR}}^{(k)} - D z\|_2^2 + c_6 \\
&\qquad = \|y^{(k+1)} - z + \eta^{(k)}\|_2^2 + \sum_{i=1}^{p} \big((\bar{q}_{\mathrm{PR}}^{(k)})_i + D_{i\ell_i} z_{\ell_i}\big)^2 + c_6 \\
&\qquad = \sum_{j=1}^{m} \Big((y_j^{(k+1)} - z_j + \eta_j^{(k)})^2 + \sum_{\{i \,\mid\, \ell_i = j\}} \big((\bar{q}_{\mathrm{PR}}^{(k)})_i + D_{ij} z_j\big)^2\Big) + c_6,
\end{aligned}
$$

where c_6 represents the terms that do not depend on z. The first equality follows as we rearrange the terms, and the second equality follows due to invariance of the 2-norm under multiplication by a unitary matrix, the third follows as it is assumed that $D_{i\ell_i}$ is the only non-zero term in i-th row of D. The forth would be obtained by rearranging the summation, from which the corollary follows. □

5.4.4 Numerical Examples

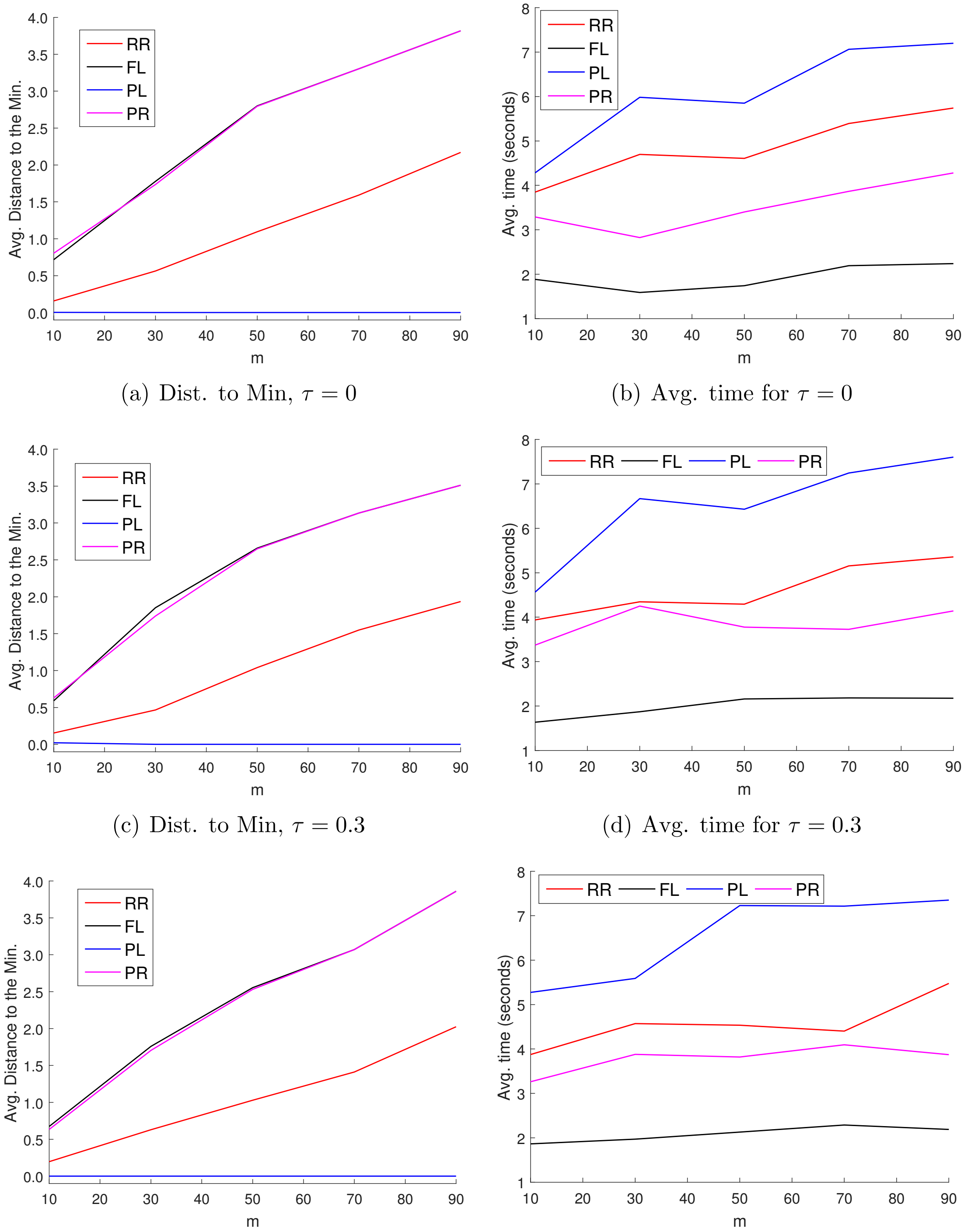
We will investigate the performance of the suggested algorithms in this section.

Early simulations in Section 5.3.2 have indicated that when Assumption 106 is met, the Direct Evaluation method has performed strongly better than the Relax-and-Round method. We will consider cases when g only satisfies Assumption 103 and apply Full Linearization (FL), Partial Linearization (PL), Partial Rounding (PR), and compare them with the Relax-and-Round (RR) algorithm.

Piece-wise linear f: We consider a mixed binary problem ($\mathcal{Z}_j = \mathbb{B}$ for all j) and a synthetic f that is the sum of piecewise linear functions, where each is such that its optimal point is closer to the boundary point with greater value (as in Figure 5.1(b)). We fix $n = 5$, take $p = m$, and generate random instances of A, B, and b to obtain g. We generate A through a parameter $\tau > 0$ that controls how much diagonal-dominant S would be, with $\tau = 0$ meaning that each row of S has only a single non-zero element. As τ gets further away form zero, the structure of S would become more random. For each m and τ, we have generated 20 random sample problems and report the results based on the average of those samples.

We have illustrated the performance of the four methods in Figure 5.9, which indicates that the Relax-and-Round and Partial Linearization (Method 8) have performed better than the other two for different τ, while PL also requires more time to get to its optimal point. There are 20 out of 1300 samples (1.54%) where RR performed better than PL in our examples.

We have compared the two methods that have performed better in closer details (PL and RR) in Figure 5.10, where we plot the their performances versus each other. Their colors indicate how fast the two algorithm were, with bluer meaning that the PL was faster and redder meaning that the RR was faster. We see that as the problem size gets bigger, PL has performed even better (further away from the $y = x$ line).



(a) Dist. to Min, $\tau = 0$

(b) Avg. time for $\tau = 0$

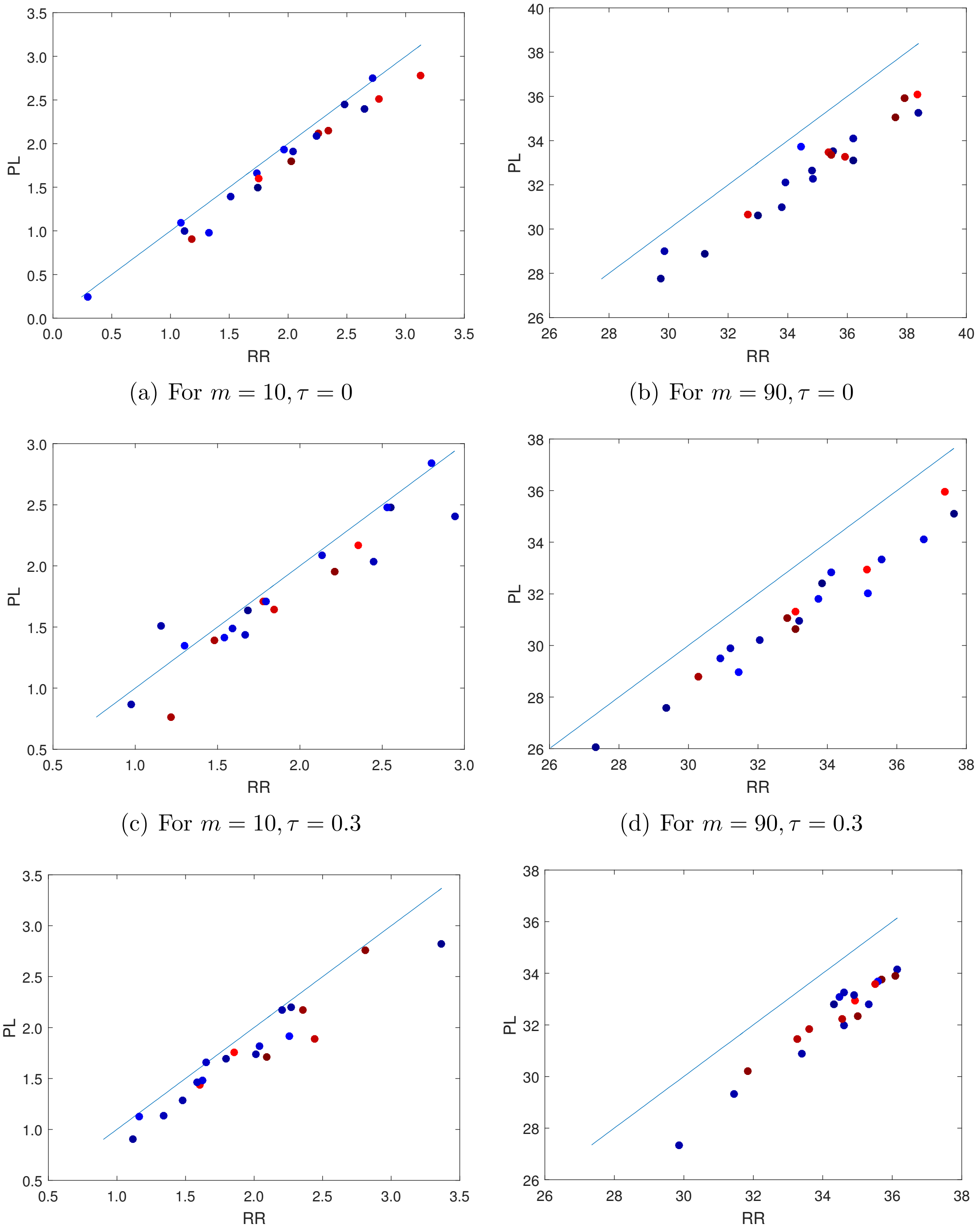
(c) Dist. to Min, $\tau = 0.3$

(d) Avg. time for $\tau = 0.3$

(e) Dist. to Min, $\tau = 0.6$

(f) Avg. time for $\tau = 0.6$

Figure 5.9: Avg. Distance to the optimal value and the Avg. time to reach that point



(a) For $m = 10, \tau = 0$

(b) For $m = 90, \tau = 0$

(c) For $m = 10, \tau = 0.3$

(d) For $m = 90, \tau = 0.3$

(e) For $m = 10, \tau = 0.6$

(f) For $m = 90, \tau = 0.6$

Figure 5.10: Optimal values for the Relax-and-Round and Partial Linearization methods

When $m = 10$ the problem size is small enough that we can exhaustively search the 2^{10} different cases of the discrete variable, and compare the exact discrete variable solution to the ones obtained by the four methods, as illustrated in Figure 5.11. Similarly, Figure 5.12 compares the difference of the optimal discrete variable obtained from the Relax-and-Round and Partial Linearization methods to the exact solution, where the size of circles denote the frequency of that observance in 20 samples.

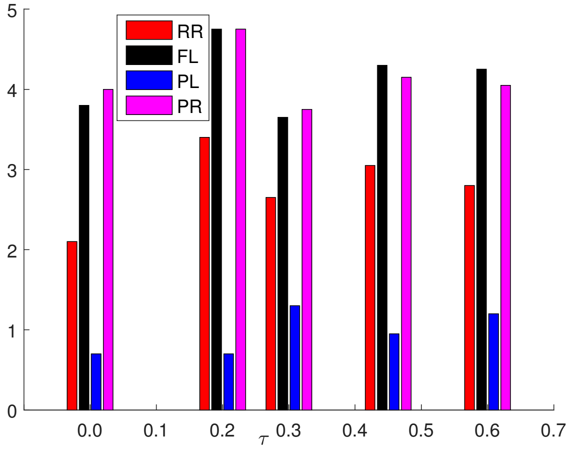
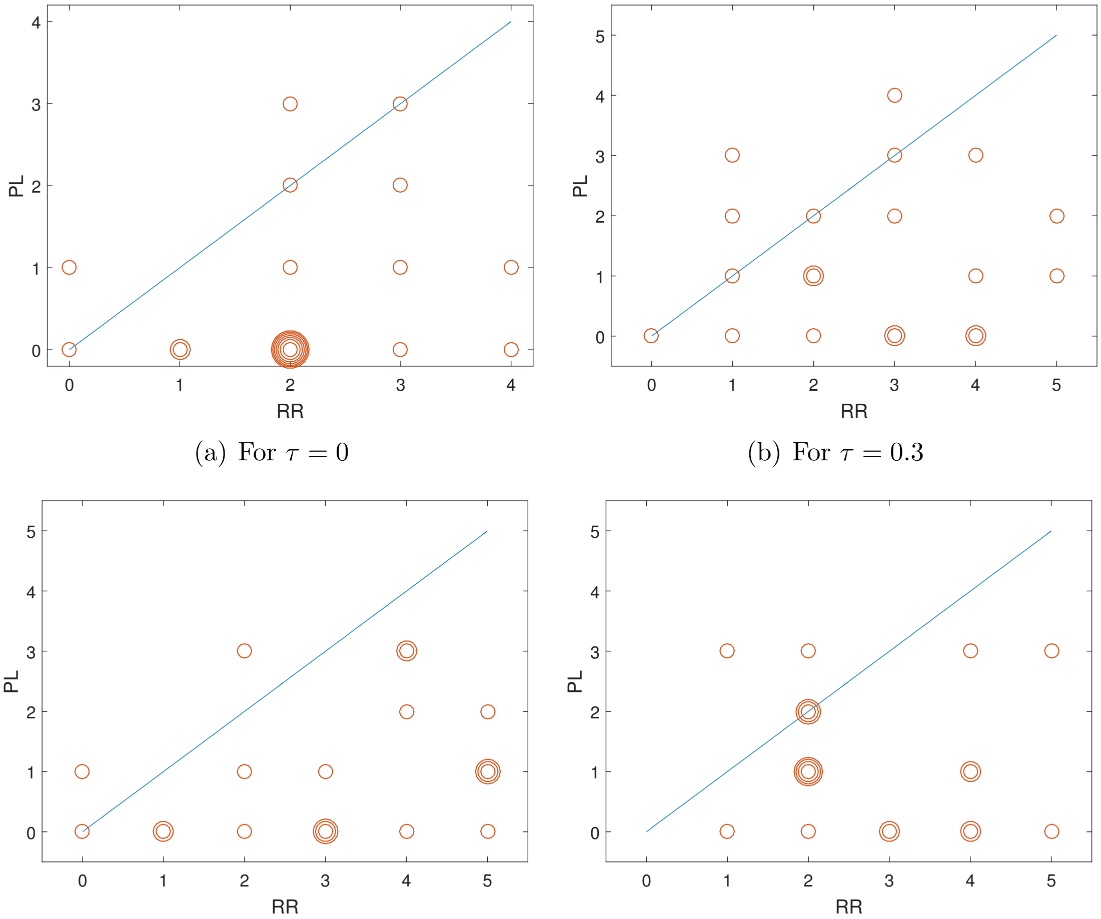


Figure 5.11: Avg. difference of the optimal discrete variable to the exact solution when $m = 10$



(a) For $\tau = 0$

(b) For $\tau = 0.3$

(c) For $\tau = 0.45$

(d) For $\tau = 0.6$

Figure 5.12: Difference of the optimal discrete variable to the exact solution for Relax-and-Round and Partial Linearization methods

Chapter 6: Robustness to Fixed Modes

We studied the a necessary and sufficient condition for stabilizability of LTI systems with respect to LTI controllers in Chapter 2. The condition was a binary one that provides no information on how close a system would be to losing stabilizability. In many cases one needs to know more than just whether or not a fixed mode is present. It could be the case that although the plant is theoretically controllable (i.e., there exist no fixed modes), that a large control effort is required to move the states, and/or that a small perturbation to the plant would result in a fixed mode. These questions have been well answered for the centralized case through controllability, observability, and Hankel operators. In particular, Hankel singular values of a stable plant provide a non-binary measure of how controllable and observable that plant is, and are easy to compute.

A more direct view of such robustness measure in the centralized case is through the notion of the controllability and observability radius. These measures have been first introduced in [77], their connections to the numerical stability of the resulting controller and the notions of the gramians were studied in [78, 79]. More recently there has been progress in finding the exact optimal value of these distances through a series of works including [80–83]. In particular, [80, 83, 84] find the global optimum of the controllability radius through polynomial time algorithms by checking the objective to determine if it can become less than a pre-selected value, which can then be used in conjunction with a bisection method to get as close as desired to the global optimal value. These could also be used to find an upper bound on the μ-norm of an FDLTI system [81]. Lower bounds for this special case has been derived by formulating the problem as a polynomial optimization problem [85], and then using SOS techniques to derive lower bounds on the global minimum.

When each actuator has access to an arbitrary set of sensor measurements, but not the others, the problem of how far we are from losing decentralized controllability becomes more complex. In the decentralized case, Vaz and Davison have defined the decentralized assignability measure based on the distance of the plant from the set of plants that have a fixed mode [2]. They characterized and connected the mobility of an eigenvalue of the plant, which is the change in its location when a decentralized controller of bounded magnitude is applied, to the aforementioned measure. They have also proven that this measure would be non-zero if and only if there exist no fixed modes. However, this metric is hard to compute for all but the smallest problems. This metric can be though of as an extension of the controllability radius to the decentralized settings. As an alternative strategy, the approach taken in [86] has explored the use of the Hankel operator to develop an easily computable metric which could provide information regarding proximity to a fixed mode for decentralized control. The developed metric in [86] combines the controllability gramian, observability gramian, and a cross-gramian that incorporates the information structure. That metric closely tracks the one of the Vaz & Davison near presence of a fixed mode, for some but not all the considered classes of fixed modes.

In this chapter we study two similar approaches to this metric when one considers the smallest complex or real perturbations required to render the plant to have a fixed-mode. Such real perturbations have been considered in [87, 88] for the centralized case, and have been extended to the decentralized case in [3, 89].

The main hurdle that prevents usage of such algorithms for large scale systems remains to be due to the power set minimization involved therein, which is further involved in minimizing a non-convex singular value. The real perturbation case also corresponds to an inner non-concave maximization over a parameter that adds to its complexity.

This perspective of putting these measures on grounds that would make their computation tractable and to provide guarantees of optimality were first considered in [90–92]. This chapter further clarifies and refines those ideas and puts those in a unified framework.

We will review related concepts and original formulation of the complex and real decentralized fixed-mode radius in Section 6.1, and address the power set minimization in those problems by transforming it into a MINLP form in Section 6.2. We will prove that this form would indeed be equivalent to the original metric in general. An approximate simpler form would then be derived that allows us to derive scalable optimization algorithms for its computation, and prove that this approximate form would be an upper bound on the original metric later in that section. We will provide two methods based on the convex relaxation of this MINLP form in Section 6.3 as an initial trial to approximate the complex DFM radius. We will then use the ADMM-based algorithm studied in Section 5.3 and apply it on our MINLP form to derive an efficient method to address the MINLP part. In Section 6.5, we will show that the derived algorithms would also provide an upper bound for the original metric for the complex perturbation case, and could be used with an extra consideration for an upper bound on the real perturbation case. We will then use the polynomial formulation of the singular values studied in Section 4.4.4 to discuss and derive lower bounds on these metrics in Section 6.6. We will provide numerical examples in Section 6.7 to inspect and compare methods of this chapter.

6.1 Review

We will first review an algebraic test on detecting fixed-modes in Section 6.1.1, which would be the basis for all the measures of the robustness in this chapter. A useful method for transforming a problem with a non-diagonal information structure to a new one with a diagonal information structure is reviewed in Section 6.1.2. This method will help in deriving a scalable MINLP form for the power set minimization involved in the original formulation of the considered measures. We will then proceed by reviewing the original formulation of the complex and real decentralized fixed-mode radius in Section 6.1.3.

Assumption 117. *Through the rest of this chapter we will assume that G is a strictly proper state-space system, i.e., we focus on a given minimal state-space representation of G, for which we have $D = 0$.*

Remark 118. *We need G to be strictly proper to derive scalable computational* *algorithms for upper bounds. However the main results of Sections 6.1.2, 6.2.1, and 6.6 would still hold for a proper state-space system.*

6.1.1 An Algebraic Test for Detecting Fixed-Modes

An algebraic test to check for the existence of a fixed mode (similar to the PBH rank test for controllability or observability) is given in [93, Theorem 4.1]. The generalized version of this test is given as follows:

Theorem 119 ([20, Theorem 2])**.** *Given a strictly proper plant G, and a sparsity-induced information structure $\mathcal{S}$, we have that $\lambda \in \mathbb{C}$ is a fixed-mode of G, i.e., $\lambda \in \Lambda\left(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}}\right)$, if and only if there exists a subset $\mathtt{I} \subseteq \{1, \cdots, n_u\}$ such that:*

$$\operatorname{rank} \begin{bmatrix} A - \lambda I & B_{\mathtt{I}} \\ C_{\mathtt{J}_{\bar{\mathtt{I}}}} & 0 \end{bmatrix} < n, \tag{6.1}$$

where n is the dimension of the state, i.e., $A \in \mathbb{R}^{n \times n}$.

6.1.2 Diagonalization

We will briefly review a technique called diagonalization. This technique could be used to transform the non-diagonal sparsity-induced information structures ($\mathcal{S} \neq \mathcal{S}_{\mathrm{d}}$) into a diagonal one by arranging and repeating the columns of B (and rows of C) in a certain manner.

Theorem 120. *Given a strictly proper plant G, and an arbitrary information struc-*

ture $\mathcal{S}$, let G_{d} be the diagonalized plant *given as:*

$$\begin{aligned} A_{\mathrm{d}} &= A, & B_{\mathrm{d}} &= \begin{bmatrix}(B_{\mathrm{d}})_1 & \cdots & (B_{\mathrm{d}})_{n_u}\end{bmatrix} \\ C_{\mathrm{d}} &= \begin{bmatrix}(C_{\mathrm{d}})_1^T & \cdots & (C_{\mathrm{d}})_{n_u}^T\end{bmatrix}^T, & D_{\mathrm{d}} &= 0, \end{aligned} \tag{6.2}$$

where $B_{\mathrm{d}} \in \mathbb{R}^{n\times a}$, $(B_{\mathrm{d}})_i \in \mathbb{R}^{n\times|\mathtt{J}_i|}$, and $(B_{\mathrm{d}})_i = [B_i \quad \dots B_i]$. Also, $C_{\mathrm{d}} \in \mathbb{R}^{a\times n}$, and $(C_{\mathrm{d}})_i = \begin{bmatrix}\dots & C_j^T & \dots\end{bmatrix}^T$, for all $j \in \mathtt{J}_i$, where $\mathtt{J}_i$ is defined as (1.8) on page 21. Then, we have that:

$$\Lambda\left(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}}\right) = \Lambda\left(G_{\mathrm{d}}, \mathcal{S}_{\mathrm{d}}, \mathcal{T}^{\mathrm{s}}\right). \tag{6.3}$$

Proof. The proof would closely follow the one of the [20, Theorem 1]. □

Here, dependence on $\mathcal{S}$ is implicitly through formation of $\mathtt{J}_i$. Whenever this techniques is used through this chapter, we will make it clear by subscripting the state-space matrices by $(\cdot)_{\mathrm{d}}$.

Remark 121. *Given a state-space system G, and a diagonal sparsity-induced information structure $\mathcal{S}_{\mathrm{d}}$, we have that $G_{\mathrm{d}} = G$.*

6.1.3 DFM Radius

We will first state an existing metric on how far a system is from having decentralized fixed modes, and then review some of its properties. The materials in this section are from [2,3], and are adopted to the notation used in this chapter.

We first define the set of plants that have the same dimension as G, and have a fixed mode with respect to $\mathcal{S}$.

Definition 122. *Given the dimension of state-space matrices by* $\dim(G)$*, a sparsity-induced information structure* $\mathcal{S}$*, and a real or complex field* $\mathbb{F}$*, define the set of unassignable systems as:*

$$\begin{aligned}
&\boldsymbol{UNA}\left(\dim\left(G\right),\mathcal{S},\mathbb{F}\right) \triangleq \\
&\{\tilde{G} \mid \tilde{G} = \left(\tilde{A},\tilde{B},\tilde{C},\tilde{D}\right),\ \textit{where } \tilde{A} \in \mathbb{F}^{n\times n}, \tilde{B} \in \mathbb{F}^{n\times n_u}, \\
&\qquad \tilde{C} \in \mathbb{F}^{n_y\times n}, \tilde{D} \in \mathbb{F}^{n_y\times n_u},\ \textit{s.t. } \Lambda\left(\tilde{G},\mathcal{S},\mathcal{T}^{\mathrm{s}}\right) \neq \varnothing\},
\end{aligned} \tag{6.4}$$

where dependence on G *is implicitly through the dimension of its state-space matrices, and the dependence on* $\mathcal{S}$ *comes from all* $\tilde{G}$ *having a fixed mode with respect to* $\mathcal{S}$*.*

We are interested in the minimum distance between G, and the set of plants that have fixed-mode(s) with respect to the information structure $\mathcal{S}$, i.e., we interested in the distance of G from $\mathbf{UNA}\left(\dim\left(G\right),\mathcal{S},\mathbb{F}\right)$. To this end, define the following notion of distance:

$$d\left(G,\mathbf{UNA}\left(\dim\left(G\right),\mathcal{S},\mathbb{F}\right)\right) \triangleq \inf_{\tilde{G}\in\mathbf{UNA}(\dim(G),\mathcal{S},\mathbb{F})} \left\| \begin{bmatrix} A-\tilde{A} & B-\tilde{B} \\ C-\tilde{C} & -\tilde{D} \end{bmatrix} \right\|_2, \tag{6.5}$$

where $(\tilde{A},\tilde{B},\tilde{C},\tilde{D})$ is a state-space representation for $\tilde{G}$.

Vaz & Davison [2] have defined the *decentralized assignability measure* as the above distance when $\mathbb{F} = \mathbb{C}$, and have shown that it can equivalently be written as an another optimization problem:

Theorem 123 ([2, Theorem 3])**.** *Given a strictly proper state-space system G, and a sparsity-induced information structure $\mathcal{S}$, the* decentralized assignability measure *is given by:*

$$\sigma_{\mathrm{VD}}\left(G,\mathcal{S}\right) \;\triangleq\; d\left(G,\boldsymbol{UNA}\left(\dim\left(G\right),\mathcal{S},\mathbb{C}\right)\right) \;=\; \min_{\substack{\lambda\in\mathbb{C},\\ \mathtt{I}\subseteq\{1,\cdots,n_u\}}} \sigma_n\left(\begin{bmatrix} A-\lambda I & B_{\mathtt{I}} \\ C_{\mathtt{J}_{\bar{\mathtt{I}}}} & 0 \end{bmatrix}\right), \tag{6.6}$$

where $\mathtt{I}$ can be any subset, and $\mathtt{J}_{\bar{\mathtt{I}}}$ depends on $\mathcal{S}$ and $\mathtt{I}$ as in (1.8)*.*

Remark 124. *This metric is zero if and only if* (6.1) *is satisfied, which in turn is a necessary and sufficient condition for having a fixed mode.*

Remark 125. *This metric possesses interesting properties, but it is hard to compute due to two reasons. Firstly, minimizing over the n-th singular value is non-convex, and secondly, minimizing over the partitions $\mathtt{I}\subseteq\{1,\cdots,n_u\}$ would involve integer programming ($2^{n_u}-2$ cases). This is our main motivation to approximate* (6.6) *by easily computable methods.*

When the field $\mathbb{F}$ is taken to be the reals, it can be shown that the above metric would be equivalent to the following optimization problem.

Theorem 126 ([3, Theorem 3.1])**.** *Given a strictly proper state-space system G, and a sparsity-induced information structure $\mathcal{S}$, the* real DFM radius *is given by:*

$$d\left(G,\boldsymbol{UNA}\left(\dim\left(G\right),\mathcal{S},\mathbb{R}\right)\right) \;=\; \min_{\substack{\lambda\in\mathbb{C},\\ \mathtt{I}\subseteq\{1,\cdots,n_u\}}} \tau_n\left(\begin{bmatrix} A-\lambda I & B_{\mathtt{I}} \\ C_{\mathtt{J}_{\bar{\mathtt{I}}}} & 0 \end{bmatrix}\right), \tag{6.7}$$

where $\mathtt{I}$ *can be any subset,* $\mathtt{J}_{\bar{\mathtt{I}}}$ *depends on* $\mathcal{S}$ *and* $\mathtt{I}$ *as stated earlier, and* $\tau_n(\cdot)$ *is defined as:*

$$\tau_n(W) \triangleq \sup_{\gamma\in(0,1]} \sigma_{2n-1}\left(\begin{bmatrix} \Re(W) & -\gamma\Im(W) \\ \gamma^{-1}\Im(W) & \Re(W) \end{bmatrix}\right).$$

Also for ease of notation define:

$$\eta_\gamma(W) \triangleq \begin{bmatrix} \Re(W) & -\gamma\Im(W) \\ \gamma^{-1}\Im(W) & \Re(W) \end{bmatrix}.$$

6.2 MINLP Forms

We will transform the power-set minimization in the complex and real DFM optimization problems in this section. We will first derive an equivalent MINLP with a monomial combination of the integers, and show that this form would indeed be exactly equal to the original formulation in Section 6.2.1. We will then consider diagonalization of the plant and derive a simpler MINLP form with affine combination of the integers in Section 6.2.2. We will prove that this latter form provides an upper bounds to the original formulations, and use it for our computational methods in the same section.

6.2.1 An Equivalent MINLP with Integers in Monomials

For the the ease of notation define:

$$G(\lambda, z) \;\triangleq\; \begin{bmatrix} A - \lambda I & B \operatorname{diag}(z) \\ L(z)\, C & 0 \end{bmatrix} \in \mathbb{C}^{(n+n_y)\times(n+n_u)} \tag{6.8}$$

where $\lambda \in \mathbb{C}$, $z \in \mathbb{B}^{n_u}$, and $L(z) \in \mathbb{B}^{n_y \times n_y}$ is a diagonal matrix that is zero everywhere, except for the diagonals that are given by:

$$[L(z)]_{jj} \;=\; 1 - \prod_{\{i \mid K^{\text{bin}}_{ij}=1\}} z_i,$$

for $j = 1, \cdots, n_y$.

We note that through this chapter we always use G with no parameter at all to refer the LTI state-space system. With a slight abuse of notation, whenever $G(\lambda, z)$ is used with two parameters it refers to a complex-valued matrix (6.8).

Remark 127. *When we have a diagonal sparsity-induced information structure, i.e., $\mathcal{S} = \mathcal{S}_{\mathrm{d}}$, the $L(z)$ matrix would be an affine function of the integer variables, i.e., $[L(z)]_{jj} = 1 - z_j$.*

We can then form the following optimization problems based on this binary formulation:

$$\sigma_{\mathbb{B}}(G, \mathcal{S}) \;\triangleq\; \min_{\substack{\lambda \in \mathbb{C} \\ z \in \mathbb{B}^{n_u}}} \; \sigma_n\left(G(\lambda, z)\right), \tag{6.9}$$

for the complex DFM radius, and

$$\xi_{\mathbb{B}}\left(G,\mathcal{S}\right) \triangleq \min_{\substack{\lambda\in\mathbb{C}\\ z\in\mathbb{B}^{n_u}}} \sup_{\gamma\in(0,1]} \sigma_{2n-1}\left(\eta_\gamma(G(\lambda,z))\right), \tag{6.10}$$

for the real DFM radius. The next theorem will prove that this formulation is equal to the original forms of Vaz & Davison and Lam & Davison.

Theorem 128. *Given a strictly proper state-space system G, and an arbitrary sparsity-induced information structure $\mathcal{S}$, we have that:*

$$d\left(G, \boldsymbol{UNA}\left(\dim\left(G\right),\mathcal{S},\mathbb{C}\right)\right) = \sigma_{\mathbb{B}}\left(G,\mathcal{S}\right), \tag{6.11}$$

and

$$d\left(G, \boldsymbol{UNA}\left(\dim\left(G\right),\mathcal{S},\mathbb{R}\right)\right) = \xi_{\mathbb{B}}\left(G,\mathcal{S}\right). \tag{6.12}$$

Proof. Given any λ and $\mathtt{I}$ in (6.1), for each $i\in\{1,\cdots,n_u\}$ take $z_i=1$ if $i\in\mathtt{I}$, and 0 otherwise. Then, the matrix in (6.6) (and (6.7)) would be equal to the $G(\lambda,z)$ in (6.8) except for possibly extra zero rows or columns. These extra zero rows or columns do not affect the the n-th singular value of the $(n+n_y)\times(n+n_u)$ dimensional matrix $G(\lambda,z)$ in the complex case, or the $2n-1$ singular value of the $2(n+n_y)\times 2(n+n_u)$ dimensional matrix $\eta_\gamma(G(\lambda,z))$ in the real case, which in turn render the equality. □

6.2.2 An Approximate MINLP with Integers in Affine Forms

In this section we consider a simpler MINLP that allows us to have the integer variable affinely in the $G(\lambda, z)$ matrix. This comes in the price of losing the exact equality relation that was derived in the previous section. However, we will prove that it would still detect fixed-modes whenever they exist, and that they would indeed be an upper bound on the original metric.

We will first use the diagonalization method of Section 6.1.2 to transform the strictly proper plant G with an arbitrary information structure $\mathcal{S}$ into a diagonalized form, denote by G_{d} with a diagonal information structure $\mathcal{S}_{\mathrm{d}}$. We will then define the following diagonalized counterpart of $G(\lambda, z)$:

$$G_{\mathrm{d}}(\lambda, z) \triangleq \begin{bmatrix} A - \lambda I & B_{\mathrm{d}} \operatorname{diag}(z) \\ \operatorname{diag}(1 - z)\, C_{\mathrm{d}} & 0 \end{bmatrix}, \tag{6.13}$$

and

$$G_{\mathrm{d},\gamma}(\lambda, z) \triangleq \eta_{\gamma}\left(G_{\mathrm{d}}(\lambda, z)\right). \tag{6.14}$$

We would then have the following theorem on rank deficiency of $G_{\mathrm{d}}(\lambda, z)$ and $G_{\mathrm{d},\gamma}(\lambda, z)$ in presence of a fixed mode:

Theorem 129. *Given a strictly proper state-space system G, and an arbitrary sparsity-induced information structure $\mathcal{S}$, let G_{d} denote the diagonalized plant as in (6.2). Let $\lambda \in \mathbb{C}$, and $\gamma \in (0,\ 1]$ be fixed, then the followings are equivalent:*

1. *There exists an $\mathtt{I} \subseteq \{1, \cdots, n_u\}$ such that (6.1) holds.*

2. There exists an $z \in \mathbb{B}^a$ such that:

$$\operatorname{rank}(G_{\mathrm{d}}(\lambda, z)) < n. \tag{6.15}$$

3. There exists an $z \in \mathbb{B}^a$ such that:

$$\operatorname{rank}\left(\eta_\gamma(G_{\mathrm{d}}(\lambda, z))\right) < n. \tag{6.16}$$

i.e., $\lambda \in \Lambda\left(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}}\right)$ if and only if (6.15) (or (6.16)) holds for some $z \in \mathbb{B}^a$.

Proof. We will first prove equality of Theorem 129.1 to 129.2, and then 129.2 to 129.3.

Take $z_i = 1$ if and only if $i \in \mathtt{I}$, and 0 otherwise. Then the matrix $G_{\mathrm{d}}(\lambda, z)$ would have the exact same columns or rows as the matrix in (6.1). It may also have some extra columns or rows, which are either zero, or already have appeared in $G_{\mathrm{d}}(\lambda, z)$. Thus their rank are equal to one another.

Equality of Theorem 129.2 to 129.3 is achieved by inspecting the eigenvalues of a transformed version of $\eta_\gamma(G_{\mathrm{d}}(\lambda, z))$. Take the similarity transformation matrix $T = \frac{1}{\sqrt{2}} \begin{bmatrix} \gamma I & \gamma I \\ -\mathbf{j} I & \mathbf{j} I \end{bmatrix}$ and observe that:

$$\operatorname{eig}\left(G_{\mathrm{d},\gamma}(\lambda, z)\right) \;=\; \operatorname{eig}\left(T^{-1} G_{\mathrm{d},\gamma}(\lambda, z) T\right) \;=\; \operatorname{eig}\Big(\operatorname{blkdiag}(G_{\mathrm{d}}(\lambda, z), \overline{G_{\mathrm{d}}(\lambda, z)})\Big).$$

As rank of a block diagonal matrix is equal to sum of ranks of its blocks, and that $\operatorname{rank}(G_{\mathrm{d}}(\lambda, z)) = \operatorname{rank}(\overline{G_{\mathrm{d}}(\lambda, z)})$, we would have $\operatorname{rank}(G_{\mathrm{d}}(\lambda, z)) \leq n-1$ if and only if $\operatorname{rank}(G_{\mathrm{d},\gamma}(\lambda, z)) \leq 2n-2$. □

We then focus on the following optimization problem:

$$\bar{\sigma}_{\mathbb{B}}\left(G, \mathcal{S}\right) \triangleq \min_{\substack{\lambda\in\mathbb{C}\\ z\in\mathbb{B}^a}} \sigma_n\left(G_{\mathrm{d}}(\lambda, z)\right), \tag{6.17}$$

and its real counter part:

$$\bar{\xi}_{\mathbb{B}}\left(G, \mathcal{S}\right) \triangleq \min_{\substack{\lambda\in\mathbb{C}\\ z\in\mathbb{B}^a}} \sup_{\gamma\in(0,1]} \sigma_{2n-1}\left(\eta_\gamma(G_{\mathrm{d}}(\lambda, z))\right). \tag{6.18}$$

When we have a diagonal information structure ($\mathcal{S} = \mathcal{S}_{\mathrm{d}}$), the diagonalization of a plant is the initial plant itself (Remark 121), and thus matrix $G_{\mathrm{d}}(\lambda, z)$ would be equal to $G(\lambda, z)$, which renders (6.17) and (6.18) to be also equal to the original form of Vaz & Davison and Lam & Davison as in (6.11) and (6.12) in Theorem 128.

Remark 130. *When $\mathcal{S} \neq \mathcal{S}_{\mathrm{d}}$, the equivalence between (6.6) and $\bar{\sigma}_{\mathbb{B}}\left(G, \mathcal{S}\right)$, and between (6.7) and $\bar{\xi}_{\mathbb{B}}\left(G, \mathcal{S}\right)$ does not hold anymore. However, due to Theorem 129, we have that:*

$$\begin{aligned}
\Lambda\left(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}}\right) \neq \varnothing \quad &\Longleftrightarrow\ d\left(G, \boldsymbol{UNA}\left(\dim\left(G\right), \mathcal{S}, \mathbb{C}\right)\right) = 0 \\
&\Longleftrightarrow\ d\left(G, \boldsymbol{UNA}\left(\dim\left(G\right), \mathcal{S}, \mathbb{R}\right)\right) = 0 \\
&\overset{\text{Thm.129.2}}{\Longleftrightarrow}\ \bar{\sigma}_{\mathbb{B}}\left(G, \mathcal{S}\right) = 0 \\
&\overset{\text{Thm.129.3}}{\Longleftrightarrow}\ \bar{\xi}_{\mathbb{B}}\left(G, \mathcal{S}\right) = 0,
\end{aligned}$$

where the first two line follow as the smallest perturbation would trivially be zero in the presence of a fixed mode. This means that in presence of a fixed method both $\bar{\sigma}_{\mathbb{B}}\left(G, \mathcal{S}\right)$ and $\bar{\xi}_{\mathbb{B}}\left(G, \mathcal{S}\right)$ would be zero for any sparsity-induced information structure $\mathcal{S}$.

We can now prove that these simpler forms $\bar{\sigma}_{\mathbb{B}}\left(G, \mathcal{S}\right)$ and $\bar{\xi}_{\mathbb{B}}\left(G, \mathcal{S}\right)$ would indeed be upper bounds for complex and real DFM radius in the following theorem.

Theorem 131. *Given a strictly proper state-space system G, and a sparsity-induced information structure $\mathcal{S}$, we have that:*

$$\bar{\sigma}_{\mathbb{B}}\left(G, \mathcal{S}\right) \geq d\left(G, \boldsymbol{UNA}\left(\dim\left(G\right), \mathcal{S}, \mathbb{C}\right)\right), \tag{6.19}$$

and

$$\bar{\xi}_{\mathbb{B}}\left(G, \mathcal{S}\right) \geq d\left(G, \boldsymbol{UNA}\left(\dim\left(G\right), \mathcal{S}, \mathbb{R}\right)\right). \tag{6.20}$$

Proof. Given a $\lambda \in \mathbb{C}$, $z \in \mathbb{B}^a$, and $\gamma \in (0, 1]$, if we take the set $\mathtt{I}$ as:

$$\mathtt{I} \;=\; \left\{ i \mid z_k = 1, \text{ for some } k \in \left\{ \sum\nolimits_{l=1}^{i-1} |\mathtt{J}_l| + 1, \cdots, \sum\nolimits_{l=1}^{i} |\mathtt{J}_l| \right\} \;\right\},$$

then we would have:

$$\sigma_n(G_{\mathrm{d}}(\lambda, z)) \;\geq\; \sigma_n\left(\begin{bmatrix} A - \lambda I & B_{\mathtt{I}} \\ C_{\mathtt{J}_{\bar{\mathtt{I}}}} & 0 \end{bmatrix}\right),$$

and

$$\sigma_{2n-1}\left(\eta_\gamma\left(G_{\mathrm{d}}(\lambda, z)\right)\right) \;\geq\; \sigma_{2n-1}\left(\eta_\gamma\left(\begin{bmatrix} A-\lambda I & B_{\mathtt{I}} \\ C_{\mathtt{J}_{\bar{\mathtt{I}}}} & 0 \end{bmatrix}\right)\right),$$

as the left hand sides would (possibly) only have extra columns and rows compared to the matrices on their right hand sides, and adding extra columns or rows could not decrease any of the singular values [94, Corollary 8.6.3]. The result then follows by applying the min operator for the complex DFM radius, and sup and min operators consecutively for the real DFM radius. □

6.3 A Convex Approach

In this section, we will propose two methods to approximate the decentralized assignability measure of Vaz & Davison, mainly to compare them to our main method in Section 6.4. We have seen that the ADMM-based method in the next section is out-performing the ones in this section even for the complex DFM radius, and thus have not considered their extension for the real DFM radius.

Remark 132. *Here in this section, and in Section 6.4 we will use the nuclear norm convex surrogate instead of minimization over the n-th singular value for the complex DFM radius, or for when we need to minimize the $(2n-1)$-th singular value for the real DFM radius. We note that better convex heuristics as those studied in Section 4.3 could specially be considered for large-scale systems where n is large.*

We thus first consider using the following relaxation of (6.17) directly as to suggest it as the approximated version of the original metric:

$$\min_{\substack{\lambda\in\mathbb{C}\\ z\in[0,1]^a}} \quad \|G_{\mathrm{d}}(\lambda, z)\|_* \,, \tag{6.21}$$

and then we will use the obtained solution from the first method to fix the binary variable, and then solve another convex optimization problem to find a better continuous variable.

Method 10 (Nuc)**.**

Let a strictly proper G and an arbitrary information structure $\mathcal{S}$ be given.

1. *Construct the diagonalized plant G_{d} as in* (6.2)*.*
2. *Solve the optimization problem* (6.21) *with variables $\lambda \in \mathbb{C}$, and $z \in [0,1]^a$.*
3. *Name the obtained solution by $\lambda^{(\mathrm{Nuc})^\star}$ and $z^{(\mathrm{Nuc})^\star}$, and let $\sigma^{(\mathrm{Nuc})^*}$ denote the n-th singular value of the optimal matrix in* (6.21)*, i.e.:*

$$\sigma^{(\mathrm{Nuc})^*} \;\triangleq\; \sigma_n\left(G_{\mathrm{d}}(\lambda^{(\mathrm{Nuc})^\star}, z^{(\mathrm{Nuc})^\star})\right).$$

The optimization problem (6.21) is convex, and thus can be solved with available software packages such as cvx toolbox [26]. Although it is desirable that z lies in its ideal binary set, i.e., $z \in \mathbb{B}^a$, enforcing this constraint would result in a non-convex problem that could not be readily approached. This is the motivation to consider the following method, in which, we will use the obtained solution from

Method 10 (Nuc), and round $z^{(\mathrm{Nuc})^\star}$ to the closest binary value, and will then solve (6.21) again with fixed $z \in \mathbb{B}^a$.

Method 11 (Nuc+Rounding)**.**

Let a strictly proper G and an arbitrary information structure $\mathcal{S}$ be given.

1. *Construct the diagonalized plant G_d as in (6.2).*

2. *Apply Method 10 (Nuc).*

3. *Set $z^\mathrm{F} \in \mathbb{B}^a$ as: $z^\mathrm{F} \leftarrow \mathrm{round}\left(z^{(\mathrm{Nuc})^\star}\right)$.*

4. *Solve the following optimization problem:*

$$\min_{\lambda \in \mathbb{C}} \left\| G_\mathrm{d}(\lambda, z^\mathrm{F}) \right\|_* . \tag{6.22}$$

5. *Let $\sigma^{(\mathrm{Nuc+R})^*}$ denote the n-th singular value of the optimal matrix in (6.22).*

The Nuc+Rounding Method 11 will ultimately result in a binary z, however, this method would not look at any other binary vector in $\mathbb{B}^a$ other than the rounded one. This could be a lot different from its optimal value, and motivated us to consider an alternative iterative approach to directly address (6.21) with $z \in \mathbb{B}^a$ by ADMM in the next section.

6.4 An ADMM-Based Approach

We lay out our ADMM-based method for the complex DFM radius in Section 6.4.1, and then discuss how to account for the real DFM radius in Section 6.4.2. The core concepts in this section are based on the ones derived in Chapter 5, specially to those in Section 5.3, which we elaborate for the matrix case here.

6.4.1 Complex DFM Radius

In this section we develop an ADMM-based algorithm to approximate the complex DFM radius. Specifically, we consider the following problem:

$$\min_{\substack{\lambda \in \mathbb{C} \\ z \in \mathbb{B}^a}} \left\| G_{\mathrm{d}}(\lambda, z) \right\|_* , \tag{6.23}$$

and use the Direct Evaluation method from Chapter 5 (Method 6 on page 156) to directly address the binary variable.

If we take $f(\cdot) = \|\cdot\|_*$ and $g(x, z) = G_{\mathrm{d}}(\lambda, z)$, then optimization problem (6.23) could be readily written in the form of (5.7) on page 156. It can be shown that this further satisfies Assumption 106, and hence we can apply Direct Evaluation Method 6 on it. To see the full derivation, rewrite (6.23) as:

$$\begin{aligned} &\text{minimize} && \|V\|_* \\ &\text{subject to :} && V = G_{\mathrm{d}}(\lambda, z), \end{aligned}$$

with variables $V \in \mathbb{C}^{(n+a)\times(n+a)}$, $\lambda \in \mathbb{C}$, and $z \in \mathbb{B}^a$. The augmented Lagrangian for this problem, for $\rho > 0$ can be written as:

$$L_\rho\left(V, \lambda, z, \mu\right) \;=\; \left\|V\right\|_* + \langle \mu, V - G_{\mathrm{d}}(\lambda, z)\rangle + \;\; 0.5\rho\left\|V - G_{\mathrm{d}}(\lambda, z)\right\|_{\mathrm{F}}^2$$

where $\mu \in \mathbb{C}^{(n+a)\times(n+a)}$ is the dual variable. This augmented Lagrangian can be equivalently written as:

$$L_\rho\left(V, \lambda, z, \mu\right) \;=\; \left\|V\right\|_* + \frac{\rho}{2}\left\|V - G_{\mathrm{d}}(\lambda, z) + \rho^{-1}\mu\right\|_{\mathrm{F}}^2 - (2\rho)^{-1}\left\|\mu\right\|_{\mathrm{F}}^2, \qquad (6.24)$$

which can be derived by expanding the terms. The ADMM consists of iteration over minimizing z, the pair (V, λ), and updating the dual variable μ. We will first derive minimization over z. To this end, partition V as:

$$V \;=\; \left[\begin{array}{c:ccc} V_A & V_{B1} & \cdots & V_{Ba} \\ \hdashline V_{C1} & & & \\ \vdots & & V_D & \\ V_{Ca} & & & \end{array}\right],$$

where $V_A \in \mathbb{C}^{n\times n}$, $V_{Bi} \in \mathbb{R}^{n\times 1}$, and $V_{Ci} \in \mathbb{R}^{1\times n}$. Similar partitioning also applies to μ. We have that:

$$\begin{aligned} z^{(k+1)} \;&=\; \arg\min_{z\in\mathbb{B}^a} L_\rho(V^{(k)}, \lambda^{(k)}, z, \mu^{(k)}) \\ &=\; \arg\min_{z\in\mathbb{B}^a} \left\|V^{(k)} - G_{\mathrm{d}}(\lambda^{(k)}, z) + \rho^{-1}\mu^{(k)}\right\|_{\mathrm{F}}^2 \\ &\overset{\star}{=}\; \arg\min_{z\in\mathbb{B}^a} \sum_{i=1}^{a} {f_i}^{(k)}(z_i) + c, \end{aligned}$$

where the second equality follows by taking the only term of (6.24) that depends on z. For all $i \in \{1, \cdots, a\}$, we have defined ${f_i}^{(k)} : \mathbb{B} \to \mathbb{R}$ as:

$$ {f_i}^{(k)}(z) \triangleq \left\| V_{Bi}^{(k)} + \rho^{-1}\mu_{Bi}^{(k)} - (B_{\mathrm{d}})_i z \right\|_{\mathrm{F}}^2 + \left\| V_{Ci}^{(k)} + \rho^{-1}\mu_{Ci}^{(k)} - (C_{\mathrm{d}})_i (1-z) \right\|_{\mathrm{F}}^2, \quad (6.25) $$

then, the third equality ($\overset{\star}{=}$) follows since different z_i appear in different rows and columns, and the Frobenius norm can be written as the square sum of the elements. Also c gathers all the terms that do not depend on z. Hence, the z-update separates for the individual elements z_i, and thus we can write ${z_i}^{(k+1)} = \arg\min_{z_i \in \mathbb{B}} {f_i}^{(k)}(z_i)$, which would give the following easily checkable condition:

$$ {z_i}^{(k+1)} = \begin{cases} 1 & \text{if } {f_i}^{(k)}(1) \leq {f_i}^{(k)}(0) \\ 0 & \text{otherwise.} \end{cases} \quad (6.26) $$

Remark 133. *Since the minimization over z separates for different elements, we would only need to check the function value at $2a$ points, rather than 2^a points, where a denotes the cardinality of admissible non-zero elements in the controller as on page 20.*

Next, we formulate the minimization over the pair (V, λ):

$$ \begin{aligned} (V^{(k+1)}, \lambda^{(k+1)}) &= \arg\min_{V,\lambda} \; L_\rho(V, \lambda, z^{(k+1)}, \mu^{(k)}) \\ &= \arg\min_{V,\lambda} \; \|V\|_* + 0.5\rho \left\| V - G_{\mathrm{d}}(\lambda, z^{(k+1)}) + \rho^{-1}\mu^{(k)} \right\|_{\mathrm{F}}^2, \end{aligned} \quad (6.27) $$

with variables $V \in \mathbb{C}^{(n+a)\times(n+a)}$ and $\lambda \in \mathbb{C}$. The minimization (6.27) is a convex optimization problem over the pair $(V,\, \lambda)$.

Lastly, the μ-update would be:

$$\mu^{(k+1)} \;=\; \mu^{(k)} + \rho\left(V^{(k+1)} - G_{\mathrm{d}}(\lambda^{(k+1)}, z^{(k+1)})\right). \tag{6.28}$$

We will further improve the optimal point obtained from ADMM method by applying the subgradient of the k-th singular value. This subgradient was discussed for a matrix variable in Section 4.2, and we will use the rule of subgradient of an affine combination to derive it for $\sigma_n\left(G_{\mathrm{d}}(\lambda, z)\right)$ with respect to λ when z is fixed.

Corollary 134. *Given a fixed z^{F}, let $G_{\mathrm{d}}(\lambda^{(k)}, z^{\mathrm{F}}) = U_F^{(k)}\Sigma_F^{(k)}(V_F^{(k)})^*$ be a SVD decomposition for $G_{\mathrm{d}}(\lambda^{(k)}, z^{\mathrm{F}})$, then a subgradient of $\sigma_n\left(G_{\mathrm{d}}(\lambda, z^{\mathrm{F}})\right)$ at $\lambda^{(k)}$, denoted by $h_{\mathbb{C}\mathrm{DFM}}(\lambda^{(k)})$, is given by:*

$$h_{\mathbb{C}\mathrm{DFM}}(\lambda^{(k)}) \;=\; L^T \mathrm{vec}\left(U_F^{(k)}\mathbf{e}_n\mathbf{e}_n^T(V_F^{(k)})^*\right),$$

where $L \in \mathbb{R}^{2(n+a)^2\times 2}$ is an appropriate matrix such that:

$$\mathrm{vec}\left(G_{\mathrm{d}}(\lambda, z^{\mathrm{F}})\right) = L\begin{bmatrix}\Re(\lambda)\\ \Im(\lambda)\end{bmatrix} + d,$$

for some $d \in \mathbb{R}^{2(n+a)^2}$.

Proof. Since $\mathrm{vec}\left(G_{\mathrm{d}}(\lambda, z^{\mathrm{F}})\right)$ is an affine function of λ, it can be written as:

$$\mathrm{vec}\left(G_{\mathrm{d}}(\lambda, z^{\mathrm{F}})\right) \;=\; L\begin{bmatrix}\Re(\lambda) & \Im(\lambda)\end{bmatrix}^T + d.$$

Hence, we have that $\sigma_n\left(G_{\mathrm{d}}(\lambda, z^{\mathrm{F}})\right) = \sigma_n\left(\mathrm{vec}^{-1}\left(L\begin{bmatrix}\Re(\lambda) & \Im(\lambda)\end{bmatrix}^T + d\right)\right)$, and we can use the affine combination rule for deriving the subgradient in conjunction with Theorem 68 to derive the subgradient with respect to λ. □

The ADMM based algorithm for the complex DFM radius is thus given as:

Method 12 (ADMM-$\mathbb{C}$)**.**

Let a strictly proper G and an arbitrary information structure $\mathcal{S}$ be given.

1. *Construct the diagonalized plant G_{d} as in (6.2).*

2. *We have that $V \in \mathbb{C}^{(n+a)\times(n+a)}$, $\lambda \in \mathbb{C}$, $z \in \mathbb{B}^a$, and $\mu \in \mathbb{C}^{(n+a)\times(n+a)}$. Initialize $k \leftarrow 0$, and let $V^{(0)}, \lambda^{(0)}, z^{(0)}, \mu^{(0)}$ be all initialized to 0 as well.*

3. *Update $z^{(k+1)}$ as (6.26).*

4. *Update $V^{(k+1)}$ and $\lambda^{(k+1)}$ as (6.27).*

5. *Update $\mu^{(k+1)}$ as (6.28).*

6. *Let $\sigma_n^{(k+1)} = \sigma_n\left(G_{\mathrm{d}}(\lambda^{(k+1)}, z^{(k+1)})\right)$.*

7. ***If** $\left|\sigma_n^{(k+1)} - \sigma_n^{(k)}\right| < \epsilon$, go to step 8, **else** let $k \leftarrow k+1$, and go to step 3.*

8. *Subgradient-based fine tunning:*

   (a) *Let $k^* = \arg\min_{k\geq 1} \sigma_n^{(k)}$, $k \leftarrow k+1$. Fix $z^{\mathrm{F}} = z^{(k^*)}$, and set $\lambda^{(k)} = \lambda^{(k^*)}$ to the optimal values given by the ADMM algorithm. .*

   (b) *Find a Clarke subgradient of $\sigma_n\left(G_{\mathrm{d}}(\lambda, z^{(k^*)})\right)$ at $\lambda^{(k)}$ by Corollary 134.*

(c) *Update* $\lambda^{(k+1)} \leftarrow \lambda^{(k)} - \theta h_{\mathbb{C}\mathrm{DFM}}(\lambda^{(k)})$*, and* $k \leftarrow k+1$*.*

(d) ***If*** *the stopping criteria is met, report* $\min_k \sigma_n \left(G_{\mathrm{d}}(\lambda^{(k)}, z^{(k^*)})\right)$*.* ***Else****, go to step 8b.*

6.4.2 Real DFM Radius

Computation of the real DFM radius involves a minimax form (6.18) as on page 194. This prevents us from directly applying method of previous section to approximate this metric in the real case.

We consider the optimization problem (6.18) in this section, and modify the ADMM-$\mathbb{C}$ Method 12 to account for the min-sup part in a sequential scheme. We will first fix γ and only consider the minimization part and apply the ADMM-based algorithm, i.e., we will use the Direct Evaluation method (Method 6 for the following optimization problem:

$$\min_{\substack{\lambda \in \mathbb{C} \\ z \in \mathbb{B}^a}} \left\| \eta_\gamma (G_{\mathrm{d}}(\lambda, z)) \right\|_*, \tag{6.29}$$

If we take $f(\cdot) = \|\eta_\gamma(\cdot)\|_*$ and $g(x, z) = G_{\mathrm{d}}(\lambda, z)$, then optimization problem (6.29), similar to (6.23), satisfies Assumption 106 and could be written in the form of (5.7) as on page 156:

$$\begin{aligned} &\text{minimize} && \left\|\eta_\gamma\left(V\right)\right\|_* \\ &\text{subject to:} && V = G_{\mathrm{d}}(\lambda, z), \end{aligned}$$

with variables $V \in \mathbb{C}^{(n+a)\times(n+a)}$, $\lambda \in \mathbb{C}$, and $z \in \mathbb{B}^a$. Hence we can apply Direct Evaluation Method 6 on this. The z-update, (V, λ)-update, and the dual update formulas for this method would then be exactly as to those in previous section when one replaces the nuclear norm $\|\cdot\|_*$ with $\|\eta_\gamma(\cdot)\|_*$ in equations (6.24) and (6.27).

The ADMM-based method would approximate the binary variable independently of γ. Algorithms in [89] try to find the global optima of (6.7) with variables λ and γ once one has fixed z by gridding the complex plane. We will present a local algorithm that would not depend on gridding the complex plane through the rest of this section.

We will fix z to be z^{F} from the ADMM-based algorithm, and find a descent direction for $\sup_{\gamma\in(0,1]}\sigma_{2n-1}\left(\eta_\gamma(G_{\mathrm{d}}(\lambda, z^{\mathrm{F}}))\right)$ in order to make the sup part smaller in each step of our sequential method. We do this by computing the subgradient of $\sigma_{2n-1}\left(\eta_\gamma(G_{\mathrm{d}}(\lambda, z^{\mathrm{F}}))\right)$ with respect to λ for fixed γ and z^{F}. This would also be a subgradient of $\sup_{\gamma\in(0,1]}\sigma_{2n-1}(\eta_\gamma(\cdot))$ if γ is such that it achieves the supremum for the updated λ. However as the optimal γ could change as we update λ, we iteratively adjust the step-size θ by making it smaller, and check if the updated λ would indeed decrease $\sup_{\gamma\in(0,1]}\sigma_{2n-1}(\eta_\gamma(\cdot))$. By using a continuity-based argument it can be seen that updating λ along its subgradient direction would decrease the sup part for for sufficiently small θ.

We will derive the subgradient of the $\sigma_{2n-1}\left(\eta_\gamma(G_{\mathrm{d}}(\lambda, z^{\mathrm{F}}))\right)$ in the following corollary.

Corollary 135. *Given a fixed z^{F} and a fixed γ, let $\eta_\gamma(G_{\mathrm{d}}(\lambda^{(k)}, z^{\mathrm{F}})) = U_F^{(k)}\Sigma_F^{(k)}(V_F^{(k)})^*$ be a SVD decomposition for $\eta_\gamma\left(G_{\mathrm{d}}(\lambda^{(k)}, z^{\mathrm{F}})\right)$, then a subgradient of $\sigma_{2n-1}\left(\eta_\gamma(G_{\mathrm{d}}(\lambda, z^{\mathrm{F}}))\right)$*

at $\lambda^{(k)}$, denoted by $h_{\mathbb{R}\mathrm{DFM}}(\lambda^{(k)})$, is given by

$$h_{\mathbb{R}\mathrm{DFM}}(\lambda^{(k)}) \;=\; L^T \mathrm{vec}\left(U_F^{(k)} \mathbf{e}_{2n-1} \mathbf{e}_{2n-1}^T (V_F^{(k)})^*\right),$$

where $L \in \mathbb{R}^{(2(n+a))^2 \times 2}$ is an appropriate matrix such that

$$\mathrm{vec}\left(\eta_\gamma(G_\mathrm{d}(\lambda, z^\mathrm{F}))\right) \;=\; L \begin{bmatrix} \Re(\lambda) \\ \Im(\lambda) \end{bmatrix} + d,$$

for some $d \in \mathbb{R}^{(2(n+a))^2}$.

Proof. Proof is very similar to that of Corollary 134. □

Then, our algorithm for computing the real DFM radius can be stated as:

Method 13 (ADMM-$\mathbb{R}$)**.**

1. *Given a strictly proper plant G and an arbitrary information structure $\mathcal{S}$, set $k \leftarrow 0$, $\gamma^{(k)} \leftarrow 1$, $(V, \lambda)^{(k)} \leftarrow 0$.*

2. *Update $z^{(k+1)}$ as in (6.26).*

3. *Update $(V, \lambda)^{(k+1)}$ as in (6.27) with $\|\cdot\|_*$ replaced by $\left\|\eta_{\gamma^{(k)}}(\cdot)\right\|_*$.*

4. *Update $\mu^{(k+1)}$ as in (6.28).*

5. *Set $t \leftarrow 1$, let $\tilde{\lambda}^{(t)} \leftarrow \lambda^{(k+1)}$, and use a ternary search to find*
   $\gamma^{(\mathrm{pre})} \leftarrow \arg\max_{\gamma \in (0,1]} \sigma_{2n-1}\left(\eta_\gamma(G_\mathrm{d}(\tilde{\lambda}^{(t)}, z^{(k+1)}))\right)$*, and*
   $\tau^{(\mathrm{pre})} \leftarrow \sigma_{2n-1}\left(\eta_{\gamma^{(\mathrm{pre})}}(G_\mathrm{d}(\tilde{\lambda}^{(t)}, z^{(k+1)}))\right)$.

6. *Compute the subgradient of* $\sigma_{2n-1}(\eta_{\gamma^{(\text{pre})}}(G_{\text{d}}(\lambda, z^{(k+1)}))$ *with respect to* λ *at* $\tilde{\lambda}^{(t)}$ *as in Corollary 135. Let* $\theta \leftarrow \theta_0$.

7. *Take* $\tilde{\lambda}^{(t+1)} \leftarrow \tilde{\lambda}^{(t)} - \theta h_{\mathbb{R}\text{DFM}}(\tilde{\lambda}^{(t)})$.

8. *Use a ternary search to find*
   $\gamma^{(\text{post})} \leftarrow \arg\max_{\gamma \in (0,1]} \sigma_{2n-1}\left(\eta_{\gamma}(G_{\text{d}}(\tilde{\lambda}^{(t+1)}, z^{(k+1)}))\right)$, *and*
   $\tau^{(\text{post})} \leftarrow \sigma_{2n-1}\left(\eta_{\gamma^{(\text{post})}}(G_{\text{d}}(\tilde{\lambda}^{(t+1)}, z^{(k+1)}))\right)$.

9. ***If*** $\tau^{(\text{post})} > \tau^{(\text{pre})}$, ***then*** $\theta \leftarrow \theta/2$ *and go to step 7.*

10. *Set* $\gamma^{(\text{pre})} \leftarrow \gamma^{(\text{post})}$, *and* $\tau^{(\text{pre})} \leftarrow \tau^{(\text{post})}$. ***If*** $t < t^{\max}$, ***then*** *let* $t \leftarrow t+1$ *and go to step 6,* ***else if*** $k < k^{\max}$ ***then*** *take* $\lambda^{(k+1)} \leftarrow \lambda^{(t+1)}$, *and* $\tau^{(\text{k+1})} \leftarrow \tau^{(\text{post})}$, *let* $k \leftarrow k+1$, *and go to step 2.*

11. *Let* $k^* \leftarrow \arg\min_k \tau^{(k)}$, *and report* $\tau^{(\text{best})} \leftarrow \tau^{(k^*)}$, $\lambda^{(\text{best})} \leftarrow \lambda^{(k^*)}$, $z^{(\text{best})} \leftarrow z^{(k^*)}$, *and* $\gamma^{(\text{best})} \leftarrow \gamma^{(k^*)}$.

It is known that $\sigma_{2n-1}\left(\eta_{\gamma}(G_{\text{d}}(\lambda, z))\right)$ is quasi-concave in γ [95, p. 100], and thus we will use this property and adopt a ternary search to shrink the interval containing the optimal γ arbitrarily small. In the above method the τ serves as to guarantee that the computed subgradient direction would indeed be a descent direction for $\sup_{\gamma \in (0,1]} \sigma_{2n-1}(F_{\gamma}(\tilde{\lambda}^{(t+1)}, z^{(k+1)}))$.

6.5 Upper Bounds

We discuss upper bounds for our approximation methods in this section. In particular, we will show how one can bound the error that arises from not solving the inner sup function exactly.

Theorem 136. *Methods Nuc+Rounding (Method 11) and ADMM-$\mathbb{C}$ (Method 12) give upper bounds for the metric $d\left(G, \boldsymbol{UNA}\left(\dim\left(G\right), \mathcal{S}, \mathbb{C}\right)\right)$.*

Proof. Proof is a direct consequence of Theorem 131, and the fact that each of these methods result in a solution that satisfy any feasibility condition for the minimization problem (6.17) on page 194 due to resulting in a binary z. □

We can prove the following lemma that would be useful for deriving an upper bound on real DFM radius when ADMM-$\mathbb{R}$ (Method 13) is used.

Lemma 137. *Given a normed space V, points $x, y \in V$, and a set $S \subset V$,*

$$\operatorname{dist}(x, S) - \operatorname{dist}(y, S) \;\; \leq \;\; \|x - y\|.$$

Proof. Proof is done by contradiction. Let x^* and y^* be points in S such that $\operatorname{dist}(x, x^*) = \operatorname{dist}(x, S)$ and $\operatorname{dist}(y, y^*) = \operatorname{dist}(y, S)$, and assume that the contrary holds, i.e.,:

$$\operatorname{dist}(x, S) \;\; > \;\; \operatorname{dist}(y, S) + \operatorname{dist}(x, y).$$

Then we have:

$$\begin{aligned} \operatorname{dist}(x, S) \;\; &> \;\; \operatorname{dist}(y, S) + \operatorname{dist}(x, y) \\ &= \;\; \operatorname{dist}(y^*, y) + \operatorname{dist}(y, x) \\ &\geq \;\; \operatorname{dist}(y^*, x) \\ \Longrightarrow \operatorname{dist}(x, S) \;\; &> \;\; \operatorname{dist}(x, y^*), \end{aligned}$$

where the $\geq$ follows due to the triangle inequality, and thus this achieves the contradiction. $\square$

We can derive the following theorem:

Theorem 138. *Given a strictly proper state-space system G and an arbitrary sparsity-induced information structure $\mathcal{S}$, by applying Method 13 (ADMM-$\mathbb{R}$), we have the following upper bound for the real DFM radius:*

$$d\left(G, \boldsymbol{UNA}\left(\dim\left(G\right), \mathcal{S}, \mathbb{R}\right)\right) \leq \min_{k} \left(\tau^{(k)} + \Delta(\gamma^{(k)})^{-1} \cdot \|\Im(G_{\mathrm{d}}(\lambda^{(k)}, z^{(k)}))\|\right), \quad (6.30)$$

where $\Delta\gamma^{-1} = (\gamma + \Delta\gamma)^{-1} - \gamma^{-1}$.

Proof. Use Corollary 86 and represent the singular value of interest as a distance to a set

$$\sigma_k(M) = \min_{\mathrm{rank}(X)=k-1} \|M - X\|_2,$$

where the set S is taken to be the set of all matrices of appropriate dimension that have rank $2n-2$. We can then use Lemma 137 to write:

$$\begin{aligned}
&\sigma_{2n-1}(\eta_{\gamma+\Delta\gamma}(G_{\mathrm{d}}(\lambda,z))) - \sigma_{2n-1}(\eta_{\gamma}(G_{\mathrm{d}}(\lambda,z))) \\
&\quad\le \|\eta_{\gamma+\Delta\gamma}(G_{\mathrm{d}}(\lambda,z)) - \eta_{\gamma}(G_{\mathrm{d}}(\lambda,z))\|_2 \\
&\quad= \left\| \begin{bmatrix} 0 & -\Delta\gamma\Im(G_{\mathrm{d}}(\lambda,z)) \\ \Delta\gamma^{-1}\Im(G_{\mathrm{d}}(\lambda,z)) & 0 \end{bmatrix} \right\|_2 \\
&\quad= \max\{\Delta\gamma, \Delta\gamma^{-1}\}\cdot\|\Im(G_{\mathrm{d}}(\lambda,z))\|_2 \\
&\quad= \Delta\gamma^{-1}\cdot\|\Im(G_{\mathrm{d}}(\lambda,z))\|_2
\end{aligned}$$

where we have let $\Delta\gamma^{-1} = (\gamma+\Delta\gamma)^{-1} - \gamma^{-1} = \gamma^{-1}[(1+\frac{\Delta\gamma}{\gamma})^{-1} - 1]$ and used the fact that this will always be larger than $\Delta\gamma$ when $\gamma \in (0,1]$. We further note that to first order, $\Delta\gamma^{-1} \approx \frac{\Delta\gamma}{\gamma^2}$.

Given our approximate solution λ, z, γ determined as in the previous section, λ and z are feasible points for our minimization, and we know how far γ may be from optimal for the given λ and z based on the experimented point in the ternary search process, so we can then add $\Delta\gamma^{-1}\cdot\|\Im(G_{\mathrm{d}}(\lambda,z))\|$ to our approximately optimal value to obtain an upper bound on the metric. Then (6.30) follows by taking the least of these upper bounds and considering (6.20). □

6.6 Lower Bounds

We provide lower bounds for the complex and real DFM radius in this section. Our approach is based on the Courant-Fischer variational formulation of the singular values in Section 4.4.4 on page 141 to derive a polynomial optimization problem, which is then used with a Sum-of-Squares technique to derive an SDP that provides a lower bound.

We will first form a P.O. for the complex DFM radius in the following corollary:

Corollary 139. *Assume that a strictly proper state-space system G, an arbitrary sparsity-induced information structure $\mathcal{S}$, and an $q \in \mathbb{N}$ are given. Then, the following optimization problem gives a non-trivial lower bound for the (squared of) the complex DFM radius: $(\sigma_{\text{VD}}(G, \mathcal{S}))^2$:*

$$\begin{array}{lll} \min & \kappa & \\ \text{s.t.} & \kappa \geq \underline{v}_i^* \left(G(\lambda, z)\right)^* G(\lambda, z) \underline{v}_i & \text{for} \quad i = 1, \cdots, q \\ & \underline{V}_i \underline{v}_i = 0 & \text{for} \quad i = 1, \cdots, q \\ & \underline{v}_i^* \underline{v}_i = 1 & \text{for} \quad i = 1, \cdots, q \\ & z_i(1 - z_i) = 0 & \text{for} \quad i = 1, \cdots, n_u, \end{array} \tag{6.31}$$

with variables $\kappa \in \mathbb{R}$, $\lambda \in \mathbb{C}$, $z \in \mathbb{R}^{n_u}$, $\underline{v}_1, \cdots, \underline{v}_q \in \mathbb{C}^{n+n_u}$, and where $\underline{V}_1, \cdots, \underline{V}_q$ are fixed matrices in $\mathbb{C}^{n_u \times (n+n_u)}$ that all have rank n_u.

Proof. This can be seen by using the equality in Theorem 128 (on p. 191), and then applying (4.14) (on p. 144) with X in Corollary 98 (on p. 143) replaced by $G(\lambda, z)$, and then enforcing $z \in \mathbb{B}^{n_u}$ by adding the last equality constraint in (6.31) (on p. 211) . □

Similarly for the real DFM radius we have the following corollary.

Corollary 140. *Assume that a strictly proper state-space system G, an arbitrary sparsity-induced information structure $\mathcal{S}$, and $q_1 \in \mathbb{N}$ and $q_2 \in \mathbb{N}$ are given. Then, the following optimization problem gives a non-trivial lower bound for the (squared of) the real DFM radius:*

$$
\begin{aligned}
\min \quad & \kappa \\
\text{s.t.} \quad & \kappa \geq \underline{v}_i^* \left(\eta_{\gamma_j}(G(\lambda, z))\right)^* \eta_{\gamma_j}(G(\lambda, z)) \underline{v}_i && \text{for} \quad i = 1, \cdots, q_1, \\
& && \text{and} \quad j = 1, \cdots, q_2 \\
& \underline{V}_i \underline{v}_i = 0 && \text{for} \quad i = 1, \cdots, q_1 \\
& \underline{v}_i^* \underline{v}_i = 1 && \text{for} \quad i = 1, \cdots, q_1 \\
& z_i(1 - z_i) = 0 && \text{for} \quad i = 1, \cdots, n_u,
\end{aligned}
\tag{6.32}
$$

with variables $\kappa \in \mathbb{R}$, $\lambda \in \mathbb{C}$, $z \in \mathbb{R}^{n_u}$, $\underline{v}_1, \cdots, \underline{v}_{q_1} \in \mathbb{C}^{2(n+n_u)}$, $\underline{V}_1, \cdots, \underline{V}_{q_1}$ that are all fixed matrices in $\mathbb{C}^{(2n_u+1) \times 2(n+n_u)}$ that all have rank $2n_u + 1$, and $\gamma_1, \cdots, \gamma_{q_2}$ that are all in $(0, 1]$.

Proof. Proof is similar to that of Corollary 139, and with using sampling to lower bound the sup over γ as minimization over a finitely many samples of that:

$$
\begin{aligned}
\min_{\lambda,z}\sup_{\gamma}\sigma_{2n-1}^2\left(\eta_\gamma(G(\lambda,z))\right) \;\geq\; & \min_{\lambda,z}\sup_{\gamma}\min \quad \kappa \\
& \qquad\qquad\quad \text{s.t.} \quad \kappa \geq \underline{v}_i^*\left(\eta_\gamma(G(\lambda,z))\right)^*\eta_\gamma(G(\lambda,z))\underline{v}_i, \\
& \qquad\qquad\qquad\qquad \vdots \\
= \; & \min_{\lambda,z}\min_{\kappa,\underline{v}_i} \quad \kappa \\
& \qquad\quad \text{s.t.} \quad \kappa \geq \underline{v}_i^*\left(\eta_\gamma(G(\lambda,z))\right)^*\eta_\gamma(G(\lambda,z))\underline{v}_i, \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \forall\ \gamma \in (0,1] \\
& \qquad\qquad\quad\ \ \vdots \\
\geq \; & \min_{\lambda,z}\min_{\kappa,\underline{v}_i} \quad \kappa \\
& \qquad\quad \text{s.t.} \quad \kappa \geq \underline{v}_i^*\left(\eta_{\gamma_j}(G(\lambda,z))\right)^*\eta_{\gamma_j}(G(\lambda,z))\underline{v}_i, \\
& \qquad\qquad\qquad\qquad\qquad\qquad\quad \text{for } j \in \{1,\cdots,q_2\} \\
& \qquad\qquad\quad\ \ \vdots
\end{aligned}
$$

where the first inequality follows due to the finite sampling from Courant-Fischer subspaces, the equality is a direct reformulation of the sup, and the last inequality also follows due to the finite sampling from $\gamma \in (0,1]$. □

The following theorem establishes that under some mild conditions, lower bounds for the complex and real DFM radius can be found by convex programs.

Theorem 141. *Given a strictly proper state-space system G, and an arbitrary sparsity-induced information structure $\mathcal{S}$, assume that some bounds on the optimal λ, and κ in (6.31) and (6.32) are known (i.e., $\|\lambda\| \leq \bar{\lambda}$, and $\|\kappa\| \leq \bar{\kappa}$), then non-trivial lower bounds for $\left(d\left(G, \boldsymbol{UNA}\left(\dim(G), \mathcal{S}, \mathbb{C}\right)\right)\right)^2$ and $\left(d\left(G, \boldsymbol{UNA}\left(\dim(G), \mathcal{S}, \mathbb{R}\right)\right)\right)^2$ can be obtained by convex programs.*

Proof. Optimization problems (6.31) and (6.32) are polynomial optimization problem where the objective and all the constraints are real valued. These problems satisfy the P.O. Definition 79 (on p. 134). Using the bounds on λ and κ, one can apply Remark 83 (on p. 135), which ensures that Assumption 82 holds. Then, the SOS-based SDP which provide lower bounds can be derived using Theorem 84. □

6.7 Numerical Examples

In this section, we provide numerical examples to compare the proposed methods. All the systems are strictly proper LTI, and are further centrally controllable and observable.

Example 142. *Consider the following state-space system, with parameter $\beta \in \mathbb{R}$:*

$$A = \begin{bmatrix} -1 & 0 \\ 0 & -3 \end{bmatrix}, B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, C = \begin{bmatrix} 0 & \beta \\ 1 & 1 \end{bmatrix}, K^{\text{bin}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

This system has a fixed mode only at $\beta = 0$. We vary β and plot the n-th singular value obtained from the Nuc, Nuc+Rounding, ADMM-$\mathbb{C}$ and its fine tunning by subgradient, and the lower bound Corollary 140 in Figure 6.1. The Vaz & Davison metric ($\sigma_{\text{VD}}(G, \mathcal{S})$ in (6.6)) is computed for the numerical examples by evaluating the singular values over a discrete grid in the complex plane for each of the $2^{n_u} - 2$ possible subsets $\mathtt{I}$, which is clearly only an option for very small problems. In this example, the results of Nuc and Nuc+Rounding methods collide, meaning that the z in Method 10 (on p. 197) was already very close to its binary value. We see that both Nuc and Nuc+Rounding methods are outperformed by the ADMM-$\mathbb{C}$ algorithm of Method 12 (on p. 203) . Also, they all behave similarly near the fixed mode.

We also see that applying the subgradient fine tuning has improved the result of the ADMM-$\mathbb{C}$.

The lower bound obtained is obtained by either sampling two or three $\underline{V}$ in (6.31) *(i.e., $m = 2$, or 3). We have observed that when $m = 1$, the lower bound is zero in the considered range. Also, it is noteworthy that the lower bound is not only dependent on m, but also on the choice of $\underline{V}$ in* (6.31)*, i.e., different choices for $\underline{V}_1, \cdots, \underline{V}_m$ may result in different lower bounds. We have randomly generated these $\underline{V}_i$ in this example. We have used* `gloptipoly` *[96] to form the SDP relaxation corresponding to the lower bound on the polynomial optimization problem* (6.31) *. As illustrated in Figure 6.1, as we increase the number of samples (m) from the rank constrained subspace in* (6.31)*, the lower bound becomes more accurate.*

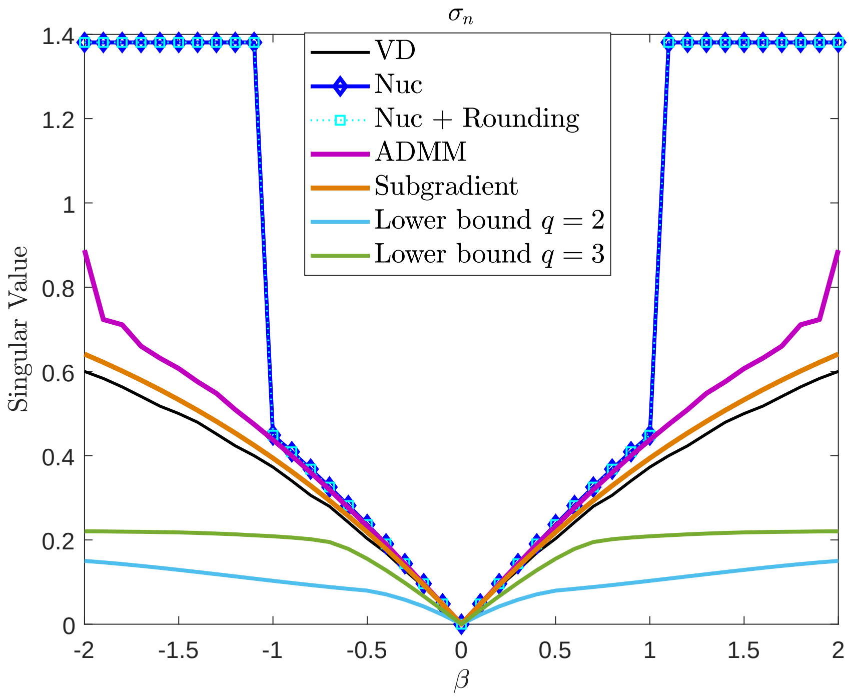


Figure 6.1: Comparison of Singular Values for complex DFM radius in Example 142

Next, we give another example for which the result of Nuc and Nuc+Rounding would be different, yet still similar near a fixed mode.

Example 143. *Consider the following system with parameter $\beta \in \mathbb{R}$:*

$$A = \mathrm{diag}(-1, -1, -1, -5),$$

$$B = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 3 & 1 & 2 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & \beta & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 3 & 0 & 4 \end{bmatrix}, \quad K^{\mathrm{bin}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

This system has a fixed mode at $\beta = 1$. We again vary β and plot the singular value obtained from the Nuc, Nuc+Rounding, the ADMM-$\mathbb{C}$ and its subgradient-based improvement, and the complex DFM radius in Figure 6.2. We see that the Nuc method fails to detect the fixed mode, as it is non-zero at $\beta = 1$. Nuc+Rounding would detect the fixed mode, and would be close to the ideal case of Vaz & Davison near the fixed mode, but will give an unrealistic approximation as we get farther away from the fixed mode. The ADMM approach of Method 12 has the same shape as the ideal case, and closely tracks it. We also see that subgradient-based improvement has been able to close the gap almost fully between the actual complex DFM radius (VD) and the ADMM-based approximation.

We continue by providing two examples for computing the real DFM radius.

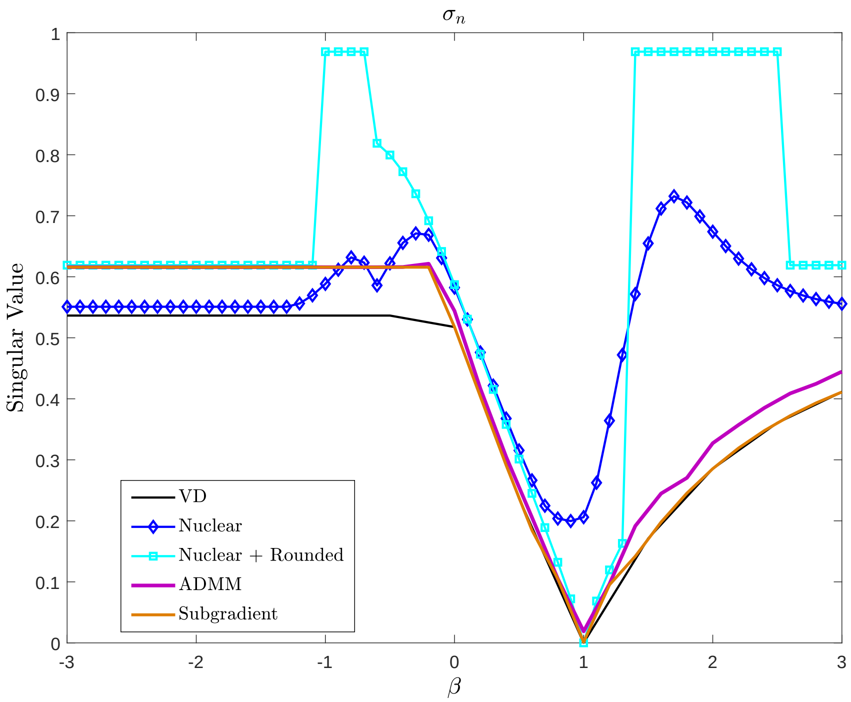


Figure 6.2: Comparison of Singular Values for complex DFM radius in Example 143

Example 144. *Consider the following strictly proper system:*

$$A = \begin{bmatrix} 1 & -0.2 \\ 0.2 & 1 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 & 10 \\ -0.2 & 17 \end{bmatrix}$$
$$C = \begin{bmatrix} 20 & 32 \\ 0.2 & 0 \end{bmatrix} \qquad K^{\text{bin}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

This system does not have a fixed mode. The complex DFM radius for this plant is computed through the ADMM-$\mathbb{C}$ method and was further fine tuned through its subgradient, which resulted in:

$$\bar{\sigma}_{\mathbb{B}}\left(G, \mathcal{S}\right) = 0.1415, \quad \lambda^{(\text{best})} = 1 + \boldsymbol{j}\, 0.14, \quad z^{(\text{best})} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

The resulting perturbation Δ is in the complex plane, and is such that:

$$\frac{\|\Im(\Delta)\|_{\mathrm{F}}}{\|\Re(\Delta)\|_{\mathrm{F}}} = 0.7288.$$

The real DFM radius for this plant was computed to be 0.2 *according to the ADMM-*$\mathbb{R}$ *Method 13 (on p. 206), which also matches the DFM radius when one only considers $\lambda \in \mathbb{R}$. Figure 6.3 depicts the $\sigma_{2n-1}\left(\eta_\gamma(G_{\mathrm{d}}(\lambda^{(\mathrm{best})}, z^{(\mathrm{best})})\right)$ as a function of γ for this example, and verifies its quasi-concavity in γ.*

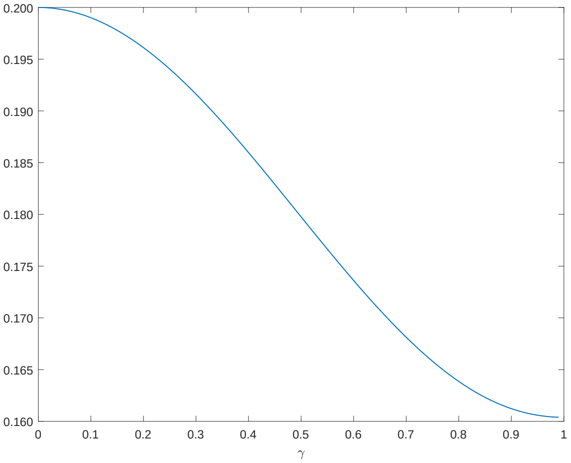


Figure 6.3: Inspecting the effect of γ

Example 145. *Consider the following strictly proper state-space system with parameter $\beta \in \mathbb{R}$, and a sparsity-induced diagonal information structure:*

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ \beta & 1 \end{bmatrix}$$
$$C = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad K^{\mathrm{bin}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

This system has a fixed mode at $\beta = 0$. We have $\Lambda(G, \mathcal{S}, \mathcal{T}^{\mathrm{s}}) = \{2\}$, for which (6.1) *(on p. 185) drops rank when $\lambda = 2$ and $\mathtt{I} = \{1\}$.*

We will vary β and plot the real DFM radius obtained through exhaustive search, the real and complex DFM radius obtained from ADMM-$\mathbb{R}$ and ADMM-$\mathbb{C}$ methods, and the real DFM radius when we further restrict λ to be in the reals in Figure 6.4. The dashed red line denotes the complex DFM radius obtained using the ADMM-$\mathbb{C}$ method and is a lower bound for the real distance, which is also what one expects from inspecting the figure. The black line denotes exhaustive search over all $z \in \mathbb{B}^a$, $\lambda \in \mathbb{C}$, and $\gamma \in (0, 1]$, which is obviously only an option for very small plants, where the search over λ and γ is by gridding. The blue line depicts ADMM-$\mathbb{R}$ Method 13 (on p. 206) when one enforces $\lambda \in \mathbb{R}$, and only proceeds up to step 5 in that method, and reports $\tau^{(\mathrm{pre})}$. Enforcing $\lambda \in \mathbb{R}$ would result in real-only

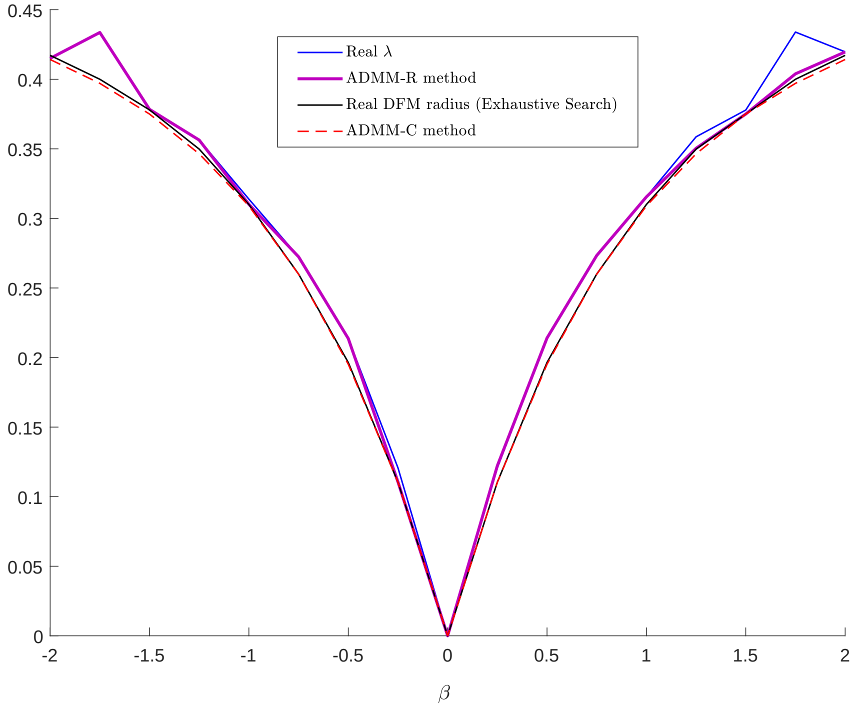


Figure 6.4: Comparing various methods for obtaining the real DFM radius

perturbations and thus would also be an upper for the real DFM radius. The purple line denotes the usage of ADMM-$\mathbb{R}$ method, which is seen to be able to find a real perturbation of a smaller norm.

Chapter 7: Conclusions and Future Works

We will conclude by mentioning possible future directions regarding the topics discussed in this dissertation in this chapter.

Chapter 2 discussed stabilizability of LTI systems with respect to decentralized LTI controllers. The algorithm developed in Section 2.1.5 could be further studied to make it more optimized. It is interesting to know if it could be devised in a way that results in a controller with the lowest possible McMillan order. Also, when designing the controller, each of the subsystems in the considered framework are assumed to be aware of the control law of the other subsystems, and it is interesting to to study if and under what conditions one could proceed with an ad-hoc design paradigm in absence of such knowledge.

Chapter 3 studied designing $\mathcal{H}_\infty$-optimal decentralized controllers when the plant is quadratically invariant. The QI allowed transformation of the decentralized controller design problem into a decentralized mode-matching problem, and the rest of that chapter developed algorithms assuming that this model-matching form with a convex decentralization constraint is either given, or possible to obtain. It is interesting to explore if and how one could extend this framework to non-QI information structures.

Chapter 4 considered a pure optimization problem of minimization of a non-convex singular value of a matrix variable. We provided a family of convex heuristics, and discussed a counter-intuitive observation regarding which is the best. It is interesting to study this problem in more detail to be able to provide when and under what conditions such convex heuristics would actually recover the optimal solution, whether it be in local or global sense. Lower bounds for this metric was also derived in latter sections of that chapter, which were based on the formulation of the singular-values as polynomial optimization problems, then lower bounds for these PO were considered using Sum-of-Squares techniques. The scalability of these lower bounds depend heavily on the underlying PO problems, and it is interesting to explore in more detail to see when and how a scalable PO for this problem could be derived.

Chapter 5 considered a class of MINLP problems that is convex except for a discrete variable and an ADMM-based algorithm was derived that allowed separate update of this discrete variable for each of its elements with a linear per-iteration complexity in the dimension of the discrete variable when the discrete variable appeared in absence of a mixing matrix in an affine part of the objective. Extensions were possible in presence of a mixing matrix that possess a mild degree of mixture level. It is interesting to embed this algorithm in branch and bound frameworks, and then compare it with some of the off the shelves ones that use such techniques. It is also interesting to theoretically quantify how well this algorithm would perform based on the mixture level, and to provide explicit extensions when $g(\cdot, \cdot)$ is not affine in the discrete variable.

Chapter 6 considered the problem of finding a non-binary measure of controllability with respect to the decentralized information structures. We considered the complex and real decentralized fixed-mode radius and transformed them into equivalent forms involving a binary vector, rather than minimization over a power-set. We then derived simpler MINLP forms with the binary vector appearing affinely in the objective, and showed that these forms would indeed upper bound the original metrics. These upper bounds result from the diagonalization procedure and it is interesting to explore this in more detail to quantify how good these upper bounds would be. It is also interesting to consider such robustness measures based on the stabilizability notions for other types of information structures. Furthermore, it is also desirable to see when and how one can embed these robustness measures into a design framework, in which one wants to find a robust and efficient decentralization paradigm by allowing the sparsity constraints to be variable as well.

Bibliography

[1] H. S. Witsenhausen, “A counterexample in stochastic optimum control,” *SIAM Journal of Control*, vol. 6, no. 1, pp. 131–147, 1968. 1

[2] A. Vaz and E. Davison, “A measure for the decentralized assignability of eigenvalues,” *Systems and Control Letters*, vol. 10, no. 3, pp. 191 – 199, 1988. 5, 182, 186, 187, 188

[3] S. Lam and E. J. Davison, “The real decentralized fixed mode radius of LTI systems,” in *Proc. IEEE Conference on Decision and Control*, 2007, pp. 3036–3041. 5, 183, 186, 188

[4] R. Bhatia, “Matrix analysis, volume 169 of graduate texts in mathematics,” 1997. 10

[5] N. Karcanias, G. Halikias, and A. Papageorgiou, “Strong stability of internal system descriptions,” *International Journal of Control*, vol. 83, no. 1, pp. 182–205, 2010. 13, 65, 66, 67

[6] M. Rotkowitz and S. Lall, “A characterization of convex problems in decentralized control,” *IEEE Transactions on Automatic Control*, vol. 51, no. 2, pp. 274–286, February 2006. 21, 22, 71, 80, 84, 85, 99, 103

[7] Y.-C. Ho and K. C. Chu, “Team decision theory and information structures in optimal control problems – Part I,” *IEEE Transactions on Automatic Control*, vol. 17, no. 1, pp. 15–22, January 1972. 22

[8] M. Rotkowitz, “On information structures, convexity, and linear optimality,” in *Proc. IEEE Conference on Decision and Control*, 2008, pp. 1642–1647. 22

[9] S.-H. Wang and E. J. Davison, “On the stabilization of decentralized control systems,” *IEEE Transactions on Automatic Control*, vol. 18, no. 5, pp. 473–478, 1973. 23, 26, 28, 34, 37, 38, 46, 51, 57

[10] A. Alavian and M. C. Rotkowitz, “Constructive stabilization and pole placement by arbitrary decentralized architectures,” *preprint arXiv:1704.01674*, 2017. 24

[11] ——, “Stabilizing decentralized systems with arbitrary information structure,” in *Proc. IEEE Conference on Decision and Control*, 2014, pp. 4032–4038. 24

[12] ——, “Fixed modes of decentralized systems with arbitrary information structure,” in *Proc. Mathematical Theory of Networks and Systems*, 2014, pp. 913–919. 24

[13] H. Kobayashi, H. Hanafusa, and T. Yoshikawa, “Controllability under decentralized information structure,” *IEEE Transactions on Automatic Control*, vol. 23, no. 2, pp. 182–188, 1978. 25

[14] B. Anderson and J. Moore, “Time-varying feedback laws for decentralized control,” *IEEE Transactions on Automatic Control*, vol. 26, no. 5, pp. 1133–1139, 1981. 25

[15] S.-H. Wang, “Stabilization of decentralized control systems via time-varying controllers,” *IEEE Transactions on Automatic Control*, vol. 27, no. 3, pp. 741–744, 1982. 25

[16] Z. Gong and M. Aldeen, “Stabilization of decentralized control systems,” *Journal of Mathematical Systems Estimation and Control*, vol. 7, pp. 111–114, 1997. 25, 26

[17] S. S. Sastry, *Nonlinear systems: analysis, stability, and control.* Springer Science & Business Media, 1999, vol. 10. 25

[18] P. Khargonekar and A. Ozguler, “Decentralized control and periodic feedback,” *IEEE Transactions on Automatic Control*, vol. 39, no. 4, pp. 877–882, 1994. 25

[19] J. L. Willems, “Time-varying feedback for the stabilization of fixed modes in decentralized control systems,” *Automatica*, vol. 25, no. 1, pp. 127–131, 1989. 25

[20] V. Pichai, M. Sezer, and D. Šiljak, “A graph-theoretic characterization of structurally fixed modes,” *Automatica*, vol. 20, no. 2, pp. 247 – 250, 1984. 26, 185, 186

[21] R. E. Kalman, “Canonical structure of linear dynamical systems,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 48, no. 4, p. 596, 1962. 31

[22] E. J. Davison and S.-H. Wang, “Properties of linear time-invariant multivariable systems subject to arbitrary output and state feedback,” *Automatic Control, IEEE Transactions on*, vol. 18, no. 1, pp. 24–32, 1973. 31

[23] D. Serre, *Matrices: Theory and Applications*, 2nd ed. Springer, 2010. 44

[24] E. Davison and T. Chang, "Decentralized stabilization and pole assignment for general proper systems," *Automatic Control, IEEE Transactions on*, vol. 35, no. 6, pp. 652–664, Jun 1990. 64, 65

[25] G. Halikias, A. Papageorgiou, and N. Karcanias, "Non-overshooting stabilisation via state and output feedback," *International Journal of Control*, vol. 83, no. 6, pp. 1232–1247, 2010. 65, 67, 68

[26] CVX Research, Inc., "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Jun. 2014. 75, 102, 197

[27] P. Gahinet and P. Apkarian, "A linear matrix inequality approach to H_∞ control," *International Journal of Robust and Nonlinear Control*, vol. 4, pp. 421–448, 1994. 79, 104, 107

[28] J. C. Doyle, K. Glover, P. P. Khargonekar, and B. A. Francis, "State-space solutions to standard H_2 and H_∞ control problems," *IEEE Transactions on Automatic Control*, vol. 34, no. 8, pp. 831–847, 1989. 79, 104

[29] G. Zhai, M. Ikeda, and Y. Fujisaki, "Decentralized H_∞ controller design: a matrix inequality approach using a homotopy method," *Automatica*, vol. 37, no. 4, pp. 565 – 572, 2001. 79

[30] V. Bompart, D. Noll, and P. Apkarian, "Second-order nonsmooth optimization for H_∞ synthesis," *Numerische Mathematik*, vol. 107, no. 3, pp. 433–454, 2007. 79

[31] G. Scorletti and G. Duc, "A convex approach to decentralized H_∞ control," in *Proc. American Control Conference*, vol. 4, 1997, pp. 2390–2394. 79

[32] C. W. Scherer, "Structured finite-dimensional controller design by convex optimization," *Linear Algebra and its Applications*, vol. 351-352, pp. 639 – 669, 2002. 79

[33] H. Hindi, B. Hassibi, and S. Boyd, "Multi-objective H_2/H_∞-optimal control via finite dimensional Q-parametrization and linear matrix inequalities," in *Proc. American Control Conference*, 1998, pp. 3244–3249. 80, 85

[34] C. W. Scherer, "Structured H_∞-optimal control for nested interconnections: A state-space solution," *Systems and Control Letters*, vol. 62, no. 12, pp. 1105 – 1113, 2013. 80

[35] Y. Wang, J. P. Lynch, and K. H. Law, "Decentralized $\mathcal{H}_\infty$ controller design for large-scale civil structures," *Earthquake Engineering and Structural Dynamics*, vol. 38, no. 3, pp. 377–401, 2009. [Online]. Available: http://dx.doi.org/10.1002/eqe.862 80

[36] C. W. Scherer, “An efficient solution to multiobjective control problems with lmi objectives,” *Systems & control letters*, vol. 40, no. 1, pp. 43–57, 2000. 80, 82, 94, 95

[37] A. Megretski, “H-infinity optimal decentralized matching model is not always rational,” *preprint arXiv:1305.5856*, 2013. 81

[38] A. Alavian and M. C. Rotkowitz, “Q-parametrization and an SDP for $\mathcal{H}_\infty$-optimal decentralized control,” in *Proceedings of the IFAC Workshop on Estimation and Control of Networked Systems*, 2013. 81

[39] ——, “On the pole selection for $\mathcal{H}_\infty$-optimal decentralized control,” in *Proc. American Control Conference*, 2015, pp. 5471–5476. 81

[40] J. A. Tropp, “Algorithms for simultaneous sparse approximation. Part II: Convex relaxation,” *Signal Processing*, vol. 86, no. 3, pp. 589–602, 2006. 81, 109, 111

[41] D. D. Siljak, *Decentralized control of complex systems.* Academic Press, Boston, 1994. 83

[42] X. Qi, M. Salapaka, P. Voulgaris, and M. Khammash, “Structured optimal and robust control with multiple criteria: A convex solution,” *IEEE Transactions on Automatic Control*, vol. 49, no. 10, pp. 1623–1640, 2004. 83

[43] D. Youla, H. Jabr, and J. B. Jr., “Modern Wiener-Hopf design of optimal controllers: part II,” *IEEE Transactions on Automatic Control*, vol. 21, no. 3, pp. 319–338, 1976. 84

[44] M. Rotkowitz and S. Lall, “Affine controller parameterization for decentralized control over Banach spaces,” *IEEE Transactions on Automatic Control*, vol. 51, no. 9, pp. 1497–1500, September 2006. 84

[45] P. Shah and P. A. Parrilo, “H_2-optimal decentralized control over posets: A state space solution for state-feedback,” in *Proc. IEEE Conference on Decision and Control*, Dec. 2010, pp. 6722–6727. 85

[46] S. Boyd and C. Barratt, *Linear Controller Design: Limits of Performance.* Prentice-Hall, 1991. 90

[47] P. Shah, B. N. Bhaskar, G. Tang, and B. Recht, “Linear system identification via atomic norm regularization,” *preprint arXiv:1204.0590*, 2012. 111

[48] M. Sadeghi, M. Babaie-Zadeh, and C. Jutten, “Dictionary learning for sparse representation: A novel approach,” *Signal Processing Letters, IEEE*, vol. 20, no. 12, pp. 1195–1198, Dec 2013. 115

[49] B. Recht, W. Xu, and B. Hassibi, “Necessary and sufficient conditions for success of the nuclear norm heuristic for rank minimization,” in *Proc. IEEE Conference on Decision and Control.* IEEE, 2008, pp. 3065–3070. 121

[50] B. Recht, M. Fazel, and P. A. Parrilo, “Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization,” *SIAM review*, vol. 52, no. 3, pp. 471–501, 2010. 121

[51] E. J. Candès and B. Recht, “Exact matrix completion via convex optimization,” *Foundations of Computational mathematics*, vol. 9, no. 6, pp. 717–772, 2009. 121

[52] A. S. Lewis and H. S. Sendov, “Nonsmooth analysis of singular values. Part I: Theory,” *Set-Valued Analysis*, vol. 13, no. 3, pp. 213–241, 2005. 121, 124, 132

[53] ——, “Nonsmooth analysis of singular values. Part II: Applications,” *Set-Valued Analysis*, vol. 13, no. 3, pp. 243–264, 2005. 121, 126, 132

[54] T.-H. Oh, Y.-W. Tai, J.-C. Bazin, H. Kim, and I. S. Kweon, “Partial sum minimization of singular values in robust pca: Algorithm and applications,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 4, pp. 744–758, 2016. 121

[55] A. Alavian and M. C. Rotkowitz, “Minimization of a particular singular value,” in *Proceedings of the 54th Annual Allerton Conference on Communication Control and Computing*, 2016, pp. 974–981. 122

[56] A. Ben-Tal and A. Nemirovski, “Lectures on modern convex optimization analysis, algorithms, and engineering applications. philadelphia: Siam, 2013,” *URL: http://www2.isye.gatech.edu/˜nemirovs/Lect_ModConvOpt.pdf*. 127

[57] J. B. Lasserre, “Global optimization with polynomials and the problem of moments,” *SIAM Journal on Optimization*, vol. 11, no. 3, pp. 796–817, 2001. 134, 135, 136, 150

[58] C. D. Meyer, *Matrix analysis and applied linear algebra*. Siam, 2000, vol. 2. 136

[59] F. R. Gantmakher, *The theory of matrices*. American Mathematical Soc., 1998, vol. 131. 140

[60] G. T. Gilbert, “Positive definite matrices and sylvester’s criterion,” *The American Mathematical Monthly*, vol. 98, no. 1, pp. 44–46, 1991. 140

[61] R. Horn and C. R. Johnson, Eds., *Matrix Analysis*. New York, NY, USA: Cambridge University Press, 1985. 142

[62] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004. 146

[63] ——, “Semidefinite programming relaxations of non-convex problems in control and combinatorial optimization,” in *Communications, Computation, Control, and Signal Processing*. Springer, 1997, pp. 279–287. 149

[64] F. Alizadeh, "Interior point methods in semidefinite programming with applications to combinatorial optimization," *SIAM Journal on Optimization*, vol. 5, no. 1, pp. 13–51, 1995. 149

[65] L. Lovász and A. Schrijver, "Cones of matrices and set-functions and 0-1 optimization," *SIAM Journal on Optimization*, vol. 1, no. 2, pp. 166–190, 1991. 149

[66] S. Burer and D. Vandenbussche, "Solving lift-and-project relaxations of binary integer programs," *SIAM Journal on Optimization*, vol. 16, no. 3, pp. 726–750, 2006. 149

[67] J. B. Lasserre, "Semidefinite programming vs. LP relaxations for polynomial programming," *Mathematics of operations research*, vol. 27, no. 2, pp. 347–360, 2002. 150

[68] H. D. Sherali and W. P. Adams, "A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems," *SIAM Journal on Discrete Mathematics*, vol. 3, no. 3, pp. 411–430, 1990. 150

[69] R. Takapoui, N. Moehle, S. Boyd, and A. Bemporad, "A simple effective heuristic for embedded mixed-integer quadratic programming," in *Proc. American Control Conference*, 2016, pp. 5619–5625. 150, 154

[70] 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*, vol. 3, no. 1, pp. 1–122, 2011. 150

[71] Y. Wang, W. Yin, and J. Zeng, "Global convergence of ADMM in nonconvex nonsmooth optimization," *preprint arXiv:1511.06324*, 2016. 150

[72] S. Diamond, R. Takapoui, and S. Boyd, "A general system for heuristic solution of convex problems over nonconvex sets," *arXiv preprint:1601.07277*, 2016. 150, 154

[73] A. Alavian and M. C. Rotkowitz, "Improving ADMM-based optimization of mixed integer objectives," in Proceedings of the 51st Annual Conference on Information Sciences and Systems, 2017. 150

[74] X. Zhang, M. Burger, and S. Osher, "A unified primal-dual algorithm framework based on bregman iteration," *Journal of Scientific Computing*, vol. 46, no. 1, pp. 20–46, 2011. 165, 166

[75] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014. 165, 166

[76] X. Zhang, M. Burger, X. Bresson, and S. Osher, "Bregmanized nonlocal regularization for deconvolution and sparse reconstruction," *SIAM Journal on Imaging Sciences*, vol. 3, no. 3, pp. 253–276, 2010. 166

[77] C. Paige, "Properties of numerical algorithms related to computing controllability," *IEEE Transactions on Automatic Control*, vol. 26, no. 1, pp. 130–138, 1981. 181

[78] J. W. Demmel, "On condition numbers and the distance to the nearest ill-posed problem," *Numerische Mathematik*, vol. 51, no. 3, pp. 251–289, 1987. 181

[79] D. Boley and W.-S. Lu, "Measuring how far a controllable system is from an uncontrollable one," *IEEE Transactions on Automatic Control*, vol. 31, no. 3, pp. 249–251, 1986. 181

[80] J. Sreedhar, P. V. Dooren, and A. L. Tits, "A fast algorithm to compute the real structured stability radius," *International Series of Numerical Mathematics*, pp. 219–230, 1996. 181

[81] C. T. Lawrence, A. L. Tits, and P. V. Dooren, "A fast algorithm for the computation of an upper bound on the μ-norm," *Automatica*, vol. 36, no. 3, pp. 449–456, 2000. 181, 182

[82] M. Gu, "New methods for estimating the distance to uncontrollability," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 3, pp. 989–1003, 2000. 181

[83] M. Gu, E. Mengi, M. L. Overton, J. Xia, and J. Zhu, "Fast methods for estimating the distance to uncontrollability," *SIAM journal on matrix analysis and applications*, vol. 28, no. 2, pp. 477–502, 2006. 181

[84] J. V. Burke, A. S. Lewis, and M. L. Overton, "Pseudospectral components and the distance to uncontrollability," *SIAM Journal on Matrix Analysis and Applications*, vol. 26, no. 2, pp. 350–361, 2004. 181

[85] B. Dumitrescu, B. Sicleru, and R. Stefan, "Computing the controllability radius: a semi-definite programming approach," *IET Control Theory & Applications*, vol. 3, no. 6, pp. 654–660, 2009. 182

[86] A. Alavian and M. C. Rotkowitz, "On a Hankel-based measure of decentralized controllability and observability," in *Proceedings of the IFAC Workshop on Estimation and Control of Networked Systems*, 2015, pp. 227 – 232. 182

[87] L. Qiu, B. Bernhardsson, A. Rantzer, E. Davison, P. Young, and J. Doyle, "A formula for computation of the real stability radius," *Automatica*, vol. 31, no. 6, pp. 879–890, 1995. 183

[88] B. Bernhardsson, A. Rantzer, and L. Qiu, “Real perturbation values and real quadratic forms in a complex vector space,” *Linear Algebra and its Applications*, vol. 270, no. 1-3, pp. 131–154, 1998. 183

[89] S. Lam and E. J. Davison, “A fast algorithm to compute the controllability, decentralized fixed-mode, and minimum-phase radius of LTI systems,” in *Proc. IEEE Conference on Decision and Control*, 2008, pp. 508–513. 183, 205

[90] A. Alavian and M. C. Rotkowitz, “An optimization-based approach to decentralized assignability,” in *Proc. American Control Conference*, 2016, pp. 5199–5204. 183

[91] ——, “Enhanced approximation of the decentralized assignability measure by subgradient methods,” in *Proc. Mathematical Theory of Networks and Systems*, 2016, pp. 511–514. 183

[92] ——, “Polynomial optimization methods for determining lower bounds on decentralized assignability,” in *Proceedings of the 54th Annual Allerton Conference on Communication Control and Computing*, 2016, pp. 1054–1059. 183

[93] B. D. Anderson and D. J. Clements, “Algebraic characterization of fixed modes in decentralized control,” *Automatica*, vol. 17, no. 5, pp. 703–712, 1981. 185

[94] G. H. Golub and C. F. V. Loan, *Matrix computations.* The Johns Hopkins University Press, 1996. 196

[95] M. Karow, “Geometry of spectral value sets,” Ph.D. dissertation, University of Bremen, 2003. 207

[96] D. Henrion, J.-B. Lasserre, and J. Löfberg, “Gloptipoly 3: moments, optimization and semidefinite programming,” *Optimization Methods & Software*, vol. 24, no. 4-5, pp. 761–779, 2009. 215