



UNIVERSITÀ DEGLI STUDI DI PADOVA

DEPARTMENT OF INFORMATION ENGINEERING

PH.D. SCHOOL IN INFORMATION ENGINEERING

FROM ONLINE OPTIMIZATION TO NEURAL STATE
SPACE MODELS: A CONTROL-INSPIRED
APPROACH TO LEARNING

SUPERVISOR

SANDRO ZAMPIERI
UNIVERSITY OF PADOVA

CO-SUPERVISOR

GIACOMO BAGGIO
UNIVERSITY OF PADOVA

PH.D. CANDIDATE

UMBERTO CASTI
UNIVERSITY OF PADOVA

SEPTEMBER 20, 2025

ACADEMIC YEAR 2024-2025

TO *ANNA*.

Ringraziamenti

Questa tesi di dottorato si conclude dopo tre anni che hanno segnato profondamente la mia vita. Un periodo intenso, talvolta difficile, ma capace di lasciare un segno profondo nel mio modo di pensare e di rapportarmi al mondo. Ho avuto il privilegio di vivere dall'interno il mondo della ricerca scientifica, della matematica e dell'università come luogo di riflessione e scoperta. Se oggi posso far tesoro di questa esperienza, è grazie alle persone che hanno condiviso con me questo percorso. A loro va la mia sincera gratitudine.

Prima di tutto desidero ringraziare il mio advisor, il Prof. Sandro Zampieri. Non solo per la sua straordinaria intelligenza e capacità di visione, ma soprattutto per la pazienza, la disponibilità e la generosità con cui mi ha guidato. Il suo spirito critico, essenziale e lucido mi ha aiutato a crescere enormemente. Da un punto di vista intellettuale, ciò che sono diventato lo devo al suo ruolo di vero mentore, grande professore e matematico, e persona di eccezionale umanità, cui va tutta la mia riconoscenza. Grazie per l'opportunità concessa fin dall'inizio e per aver reso questo percorso, pur impegnativo, sorprendentemente bello.

Un ringraziamento speciale va al mio co-advisor, il Prof. Giacomo Baggio, presenza fondamentale e sempre disponibile. In ogni difficoltà, anche nelle più piccole, non mi ha mai fatto mancare supporto, chiarezza e vicinanza.

Un pensiero profondo va anche al Prof. Fabio Pasqualetti, sotto la cui supervisione ho trascorso un periodo indimenticabile in California. Quei mesi hanno avuto un impatto enorme sulla mia crescita, scientifica e personale. L'ambiente accogliente e stimolante che ho trovato è merito della sua guida attenta, della sua sensibilità e della sua apertura verso le nuove idee. Grazie per avermi fatto sentire a casa anche così lontano, e per avermi avvicinato a un settore e delle tematiche che oggi sento parte del mio futuro.

Un grazie speciale va ai miei genitori. A mio papà devo la passione per la scienza, la curiosità, l'ingegnosità e la capacità di affrontare i problemi con determinazione e spirito pratico. La voglia e l'orgoglio di portare a termine con successo un percorso universitario di questo tipo mi sono stati trasmessi dal tuo esempio. Questa tesi di dottorato è quindi anche frutto dei tuoi insegnamenti e del tuo sostegno. A mia mamma, per avermi trasmesso il senso del dovere, la disciplina, l'abnegazione e l'importanza delle relazioni umane, insegnandomi a coltivare con cura i rapporti con gli altri. Grazie a te ho capito quanto il lavoro scientifico sia fatto anche di relazioni, scambi e collaborazione, e quanto il confronto con gli altri ricercatori sia essenziale per crescere, ti sono profondamente grato per avermi trasmesso tutto questo.

Un altro ringraziamento di cuore va a te, Diana. Grazie per tutti i bei momenti vissuti insieme, per i viaggi e le esperienze condivise. Per come sono fatto, la tua presenza, capace di infondermi serenità e quella giusta dose di spensieratezza, è stata fondamentale anche per

raggiungere questo traguardo. I momenti condivisi con te hanno reso questo cammino un viaggio più felice, leggero e spensierato.

Ringrazio mia zia Emanuela, che per prima mi ha fatto conoscere lo spirito critico e il rigore dell'ambiente accademico, introducendomi a ciò che significa davvero fare ricerca. È anche grazie a te se mi sono avvicinato a questo mondo e ho imparato ad apprezzarne la bellezza.

Ringrazio i miei amici storici: Cesare, Igor, Matteo e Nicolò. E tutti i compagni di dottorato con cui ho condiviso questa esperienza: Alessandro, Alessio, Antonio, Daniele, Davide, Davide, Elena, Giorgia, Giulio, Giulio, Lorenzo, Luca, Marina, Matteo, Matteo, Michelangelo, Nicola, Riccardo, Sara. Un ringraziamento speciale alla *Prima 326*: Chiara, Lucia, Simone. E ai compagni che hanno reso speciale il periodo californiano: Abed, Darshan, Federico, Taosha.

Infine, ma non per importanza, voglio ringraziare mia sorella Anna, a cui è dedicata questa tesi. La tua presenza ha influenzato profondamente il mio modo di crescere, scegliere e affrontare le relazioni e la vita quotidiana. Anche in questi anni universitari sei sempre stata con me. Grazie.

Abstract

This thesis investigates *online optimization* and its connections to modern *neural architectures* through a control-theoretic approach.

In classical optimization, the goal is to minimize a fixed cost function. In the *online* setting, instead, the problem must be solved as the objective and constraints evolve over time. This transforms the problem into a dynamic one, in which tools and principles from control theory become relevant to ensure convergence of the optimization error. Rather than directly applying standard control techniques, this work proposes new algorithms that reinterpret control-theoretic ideas within the optimization framework and its associated challenges.

One of the most compelling areas of application is in modern deep learning, where recent architectures such as Transformers and State Space Models exhibit *in-context learning* abilities. These mechanisms can be formalized as online regression problems, offering a principled way to analyze and improve such models.

In parallel, the second part of the thesis develops and studies brain-inspired neural architectures. Starting from neuroscience-motivated principles, we investigate properties of both dynamical models of brain activity and artificial brain-inspired architectures, and we leverage control-theoretic tools for their analysis and synthesis.

Overall, the thesis advances a framework for online optimization and, in parallel, offers a complementary investigation of modern neural architectures.

Contents

ABSTRACT	vii
LIST OF FIGURES	xv
LIST OF TABLES	xvii
LIST OF SYMBOLS AND CONVENTIONS	xix
LISTING OF ACRONYMS	xxi
1 INTRODUCTION	I
1.1 Part I: Online Optimization	1
1.2 Part II: Neural Architectures	2
2 NOTATION, DEFINITIONS & CONVENTIONS	5
I Models for online optimization	II
3 INTRODUCTION	15
4 PRELIMINARIES	17
4.1 Background on Optimization	18
5 DETERMINISTIC MODELS FOR UNCONSTRAINED ONLINE OPTIMIZATION	25
5.1 Algorithm	26
5.1.1 Internal Model	27
5.1.2 LTI controller	28
5.1.3 The online algorithm	30
5.2 Convergence	30
5.3 Discussion	31
6 DETERMINISTIC MODELS FOR CONSTRAINED ONLINE OPTIMIZATION: PROJECTED METHODS FOR INEQUALITY CONSTRAINTS	33
6.1 Algorithm	34
6.1.1 The online algorithm	38

6.2	Numerical results	38
6.3	Discussion	39
6.4	Continuous time	40
6.4.1	Algorithm	41
6.4.2	Stabilizing controller - unconstrained case	44
6.4.3	The online algorithm	45
6.4.4	Theoretical analysis & AW synthesis	45
6.4.5	Numerical experiments	49
6.4.6	Discussion	51
7	DETERMINISTIC MODELS FOR CONSTRAINED ONLINE OPTIMIZATION: PRIMAL-DUAL METHODS FOR EQUALITY CONSTRAINTS	53
7.1	Algorithm	54
7.1.1	Internal Model	56
7.1.2	LTI controller	57
7.1.3	The online algorithm	60
7.2	Convergence	61
7.3	Numerical results	61
7.3.1	Cost with fixed $\mathcal{A}$	61
7.3.2	Time-varying quadratic term and constraint	63
7.3.3	Non-quadratic cost	64
7.4	Discussion	65
8	DETERMINISTIC MODELS FOR CONSTRAINED ONLINE OPTIMIZATION: PRIMAL-DUAL METHODS FOR EQUALITY AND INEQUALITY CONSTRAINTS	67
8.1	Algorithm	68
8.1.1	The online algorithm	70
8.2	Numerical results	71
8.3	Discussion	73
9	STOCHASTIC MODELS FOR UNCONSTRAINED ONLINE OPTIMIZATION	75
9.1	Stochastic formulation	76
9.2	Algorithms	77
9.2.1	Kalman filtering inspired algorithm	80
9.2.2	Optimized filter	83
9.2.3	Robust $\mathcal{H}_\infty$ control inspired algorithm	86
9.3	Numerical results	89
9.4	Discussion	92

II	Neural architectures	95
10	BRAIN CONNECTIVITY	99
10.1	Problem formulation	100
10.1.1	Parametrizing systems with prescribed $\mathbf{P}$	101
10.2	Role of $\mathbf{S}$ in the stability of linear DCM	103
10.3	Role of $\mathbf{S}$ in the excitability of linear DCM	107
10.4	$\mathbf{S}$ and frequency-domain behavior of linear DCM	109
10.5	Discussion	113
11	MULTI-TASK ARCHITECTURES	117
11.1	Problem formulation & architecture structure	119
11.2	Numerical analysis	121
11.3	Theoretical analysis	125
11.3.1	Fundamental limitations	125
11.3.2	Case $N \geq nM$	126
11.3.3	Case $N < nM$	127
11.4	Discussion	136
12	CONCLUSIONS	139
	APPENDIX A ONLINE OPTIMIZATION	141
	APPENDIX B NEURAL ARCHITECTURES	145
	BIBLIOGRAPHY	162

List of Figures

2.1	General control scheme.	8
2.2	General framework for static optimization (left) and online optimization (right).	13
5.1	Proposed control scheme for the solution of Problem (5.1).	26
6.1	General optimization scheme derived in this section, represented in the style of Figure 2.2.	34
6.2	Scheme of Figure 6.1 with inactive projection (identity projection operator).	35
6.3	Diagram of Figure 6.2 with $\mathbf{C}_{yq}(z) = \mathbf{C}(z)$	35
6.4	Diagram of Figure 6.3 modified to ensure the feasibility of the solution.	36
6.5	PIMAW architecture with AW mechanism.	37
6.6	The graph of $\text{triang}(t) := 4 \left \frac{t}{2\pi} - \lfloor \frac{t}{2\pi} + 3/4 \rfloor + 1/4 \right - 1, t \in \mathbb{R}$	38
6.7	Comparison among OPGD (6.6), the control-based projected algorithm without anti-windup (Control-based) of Figure 6.4, and the same algorithm with anti-windup compensation (Control-based & AntiWindup) PIMAW of Figure 6.5. Top: trajectories in the (x_1, x_2) plane showing the iterates and the corresponding unconstrained solution. Bottom: semilogarithmic plot of the tracking error versus k	39
6.8	General CT-OPGD scheme rendered in the style of Figure 2.2.	42
6.9	CT-PIMAW scheme proposed in this section.	42
6.10	Equivalent block diagram of Figure 6.9 after the substitutions (6.13), $\mathbf{C}(s) = \frac{c(s)}{d(s)}\mathbf{I}$, and $\mathbf{T} = \rho\mathbf{I}$	43
6.11	Intermediate equivalent block diagram of Figure 6.10, obtained via the eigendecomposition $\mathbf{A} = \mathbf{V}\mathbf{A}\mathbf{V}^\top$. The transformed signals $\tilde{\mathbf{x}}(t)$, $\tilde{\mathbf{b}}(t)$, and $\tilde{\mathbf{w}}(t)$ are defined in (6.15).	43
6.12	Final “decoupled” block diagram, equivalent to Figures 6.10–6.11, in the transformed coordinates $\tilde{\mathbf{x}}(t)$, $\tilde{\mathbf{b}}(t)$, and $\tilde{\mathbf{w}}(t)$	44
6.13	Tracking error $\ \mathbf{x}(t) - \mathbf{x}^*(t)\ $ under triangular-wave evolution of $\mathbf{c}(t)$ and internal model (6.23).	50
6.14	Tracking error $\ \mathbf{x}(t) - \mathbf{x}^*(t)\ $ with $\mathbf{b}(t)$ following a sinusoidal-plus-constant evolution.	51
7.1	Control scheme proposed for solving Problem (7.1).	56

7.2	Comparison between the primal-dual and the control-based algorithm (semilogarithmic scale) for sinusoidal and triangular-wave signals.	62
7.3	Comparison between the OPDGD and the control-based algorithm with models (7.23) ($L = 1, 2, 3, 6$) in the time-varying case.	64
7.4	Comparison between the OPDGD and the control-based algorithm with models (7.23) ($L = 1, 2, 3$) in the non-quadratic case.	65
8.1	Control scheme proposed for solving Problem (8.1).	69
8.2	Comparison among OPDGD (8.3), the algorithm in (8.7) without anti-windup, and the complete algorithm (8.7) with anti-windup compensation, in a semilogarithmic plot where $\mathbf{c}_k$ and $\mathbf{g}_k$ are sine waves.	71
9.1	Control scheme of Figure 5.1 when $\mathbf{C}(z) = \frac{c(z)}{d(z)}\mathbf{I}_n$ and condition (9.3) holds.	78
9.2	Block representation of the i -th transfer function $[\mathbf{H}_{nr}(z)]_{ii}$ in (9.7).	80
9.3	General framework for μ -synthesis.	87
9.4	Reshaped version for a generic i of Figure 9.2 compatible with the μ -synthesis framework in Figure 9.3.	88
9.5	Equivalent control scheme for $\frac{c(z)}{d(z)}$ parametrized as in (9.31).	89
9.6	Root mean square cost (9.4) for the problem in (9.32) under sinusoidal evolution of $\mathbf{y}'_k$, comparing three algorithms: the Online Internal Model Gradient Descent (OIMGD) algorithm (5.18), the Kalman-inspired algorithm from Section 9.2.1, and the algorithm from Section 9.2.2 derived setting $\phi(\cdot)$ to (9.34).	91
9.7	Robustness analysis for the algorithms from Section 9.2.2 (with correct and misspecified prior $\phi(\cdot)$) and Section 9.2.3, tested on the problem (9.32) with $n = 2000$, $\sigma = 0.5$, $\sigma_{\text{MP}} = 0.7$, and sinusoidal evolution of $\mathbf{y}'_k$	92
9.8	Performance comparison of the Kalman-inspired algorithm (Section 9.2.1) with and without correct prior, and the robust $\mathcal{H}_\infty$ algorithm (Section 9.2.3), under noisy linear ramp evolution of $\mathbf{y}'_k$	93
10.1	The figure shows the norm of $\mathbf{S}$ in the parametrization (10.4) for mice ECs matrices and randomly generated Hurwitz stable matrices. The entries of the random matrices are drawn independently from a standard normal distribution. To enable a fair comparison with the mice ECs, these matrices are shifted and rescaled so that they possess the same maximum real part (stability margin) and imaginary part of the eigenvalues of mice ECs (averaged over the N subjects). The box plot shows the results obtained for 20 realizations of the random ensemble and 20 mice subjects with $\mathbf{R} = \mathbf{I}$	103
10.2	Behavior of the eigenvalues of $\mathbf{A}$ for the system in Example 3 and increasing values of $\alpha \geq 0$	105

10.3	Behavior of the eigenvalues of $\mathbf{A}$ for increasing values of $\alpha \geq 0$ for a 4-dimensional system the asymptotic branches corresponds to the limit values of $\text{Re}[\lambda_i]$ of Equation (10.8).	107
10.4	Behavior of $\omega(\mathbf{A})$ for increasing values of $\alpha \geq 0$ for the system in Example 4. The dashed lines are the lower and upper bound in Theorem 5.	108
10.5	Plot in a logarithmic scale of the trace of the frequency response for the $\text{tr}[\Phi(i\omega)]$ and its approximate version $\text{tr}[\tilde{\Phi}(i\omega)]$ for different values of α , in particular $\alpha = \alpha_{\text{th}} - 0.1, 10, 100$, and $d_1 > d_2 > 0$ have been generated random between 0 and 10. α_{th} denotes the threshold at which the system exhibits resonance.	111
11.1	Schematic of the multi-task approximator considered in this chapter. A library of LTI systems $\{\Sigma_i^{\text{D}}\}_{i=1}^M$ is integrated by a single RNN architecture Σ in (11.1).	123
11.2	Impulse responses of the feedback interconnections between the desired controllers Σ_i^{D} (blue) and their implementations via the multi-task neural architecture (11.1) (red), for the plants in Figure 11.1. The “approximating controllers” are obtained by linearizing the backbone around the selected equilibria.	124
11.3	Box plot of the final cost (11.5) across 17 simulations with distinct sets ($M = 5$) of randomly generated task solvers Σ_i^{D} , as the architecture dimension N increases from 1 to 8. The red line denotes the median; box edges mark the 25 th and 75 th percentiles; whiskers cover approximately 99.3% of the data.	124
11.4	Box plot (formatted as in Figure 11.3) of the final cost (11.5) across 100 simulations for 100 randomly selected sets of task solvers Σ_i^{D} . We evaluate the cost while increasing the number of systems M from 3 to 10.	125
11.5	Comparison between the minimum of the cost (11.5) obtained via gradient descent using the explicit gradient of Section 11.2, the upper bound (11.19) and the upper bound (11.21). Results are generated random with a set of task solvers $\{\Sigma_i^{\text{D}}\}_{i=1}^M$ of dimension $n = 2$ and a multi-task neural architecture of dimension $N = 2$	133

List of Tables

4.1	Summary of the problem classes and methods covered in this thesis.	24
7.1	Asymptotic errors of the OPDGD compared to the control-based algorithm when $\mathbf{c}_k$ follows model 2 (the “Sine” model) as described in Section 7.3.1 with $p(z) = z^2 - 2 \cos(\omega)z + 1$. The control-based algorithm in (7.22) is designed with $p(z) = \tilde{p}(z) := z^2 - 2 \cos(\tilde{\omega})z + 1$ where $\tilde{\omega} = 0.96\omega, 0.92\omega, 0.88\omega, 0.84\omega, 0.8\omega$. The OPDGD yields a single error value since its design is independent of the oscillation frequency of $\mathbf{c}_k$	63

List of symbols and conventions

Symbol	Description
$\mathbb{N}$	Set of natural numbers $\{0, 1, 2, 3, \dots\}$
$\mathbb{R}$	Set of real numbers
$\mathbb{R}_{\geq 0}$	Set of non-negative real numbers
$\mathbb{R}_{> 0}$	Set of positive real numbers
$\mathbb{C}$	Set of complex numbers
$\mathbb{R}^n$	Set of vectors of dimension n with real entries
$\mathbb{R}^{n \times m}$	Set of $n \times m$ matrices with real entries
$\mathbb{R}[z]$	Set polynomial function in z with real coefficients
$\mathbb{R}^{n \times m}[z]$	Set polynomial matrices $n \times m$ in z with real coefficients
$\mathbb{R}(z)$	Set rational function in z with real coefficients
$\mathbb{R}^{n \times m}(z)$	Set rational matrices $n \times m$ in z with real coefficients
$\mathcal{C}^1(\mathbb{R}^n)$	Set of continuously differentiable functions on $\mathbb{R}^n$
$\mathbf{X}$	Bold uppercase letters denote matrices
$\mathbf{x}$	Bold lowercase letters denote vectors
x	Plain (non-bold) lowercase letters denote scalars
$\mathbf{X}^\top(\mathbf{x}^\top)$	Denotes the transposition of matrix $\mathbf{X}$ (vector $\mathbf{x}$)
$\mathbf{X}^*(\mathbf{x}^*)$	Denotes the conjugate transposition of matrix $\mathbf{X}$ (vector $\mathbf{x}$)
$[\mathbf{x}]_i$ or x_i	Denotes the i -th entry of the vector $\mathbf{x}$
$(\mathbf{X})_{ij}$	Denotes the (i, j) -th entry of the matrix $\mathbf{X}$
$\lambda_{\min}(\mathbf{X})$	Denotes the minimum eigenvalue of the symmetric matrix $\mathbf{X}$
$\lambda_{\max}(\mathbf{X})$	Denotes the maximum eigenvalue of the symmetric matrix $\mathbf{X}$
$\sigma_{\min}(\mathbf{X})$	Denotes the minimum singular value of the matrix $\mathbf{X}$
$\sigma_{\max}(\mathbf{X})$	Denotes the maximum singular value of the matrix $\mathbf{X}$
$\kappa(\mathbf{X})$	Denotes the condition number of the matrix $\mathbf{X}$
$\mathbf{I}_n$	Identity matrix of order n (subscript omitted when clear from context)
$\otimes$	Denotes the Kronecker product
$\odot$	Denotes the Hadamard product
$\text{Re}[\cdot]$	Denotes the real part of a complex number
$\text{Im}[\cdot]$	Denotes the imaginary part of a complex number
$\mathcal{Z}[\cdot]$	Denotes the unilateral $\mathcal{Z}$ transform of a signal in time
$\mathcal{L}[\cdot]$	Denotes the unilateral Laplace transform of a signal in time
$\mathbb{E}[\cdot]$	Denotes the expectation of a random variable or vector

Listing of acronyms

AI	Artificial Intelligence
ANN	Artificial Neural Network
AW	Anti-Windup
AWBT	Anti-Windup Bumpless Transfer
BOLD	Blood-Oxygen-Level-Dependent
CT-OPGD	Continuous-Time Online Projected Gradient Descent
CT-PIMAW	Continuous-Time Projected Internal Model Anti-Windup
DCM	Dynamic Causal Modeling
EC	Effective Connectivity
EEG	Electro-Encephalo-Graphy
FC	Functional Connectivity
fMRI	Functional Magnetic Resonance Imaging
HiPPO	High-order Polynomial Projection Operators
IMP	Internal Model Principle
KKT	Karush–Kuhn–Tucker
LICQ	Linear Independence Constraint Qualification
LMI	Linear Matrix Inequality
LQR	Linear Quadratic Regulator
MEG	Magneto-Encephalo-Graphy
MFCQ	Mangasarian-Fromovitz Constraint Qualification
MIMO	Multi-Input Multi-Output

MoE	Mixture Of Experts
MPC	Model Predictive Control
OGD	Online Gradient Descent
OIMGD	Online Internal Model Gradient Descent
OO	Online Optimization
OPDGD	Online Primal-Dual Gradient Descent
OPGD	Online Projected Gradient Descent
OPPDGD	Online Projected Primal–Dual Gradient Descent
PIMAW	Projected Internal Model Anti-Windup
QP	Quadratic Program
RNN	Recurrent Neural Network
rsFC	Resting-State Functional Connectivity
SC	Structural Connectivity
SISO	Single-Input Single-Output
SSM	State Space Model

1

Introduction

This thesis is structured in two main parts, each spanning a distinct but closely interconnected area of research. These areas are strongly linked through both theoretical development and practical relevance.

The first part focuses on *online optimization*, introducing novel methods and algorithms designed for time-varying optimization problems. The second part investigates some of the main challenges arising in the field of *neural network architectures*, and explores how new design techniques can be used to address and improve upon these issues. Finally, we show how concepts from *online optimization* naturally emerge in this domain, and how they can be addressed using the framework introduced in the first part of the thesis.

1.1 PART I: ONLINE OPTIMIZATION

The first part of this thesis is devoted to the study of *online optimization*, a rapidly developing area of research concerned with solving time-varying optimization problems in a sequential and adaptive fashion. This framework is motivated by a wide range of applications, including Model Predictive Control (MPC), signal processing, reinforcement learning, and more recently, deep learning and sequence modeling.

A central focus of this part is the development and analysis of algorithms for solving online optimization problems with quadratic cost functions and linear constraints in discrete time. We begin by considering equality-constrained problems, and we propose an algorithm

that combines elements from control theory and optimization to track the time-varying minimizer as the objective and constraints evolve. We formally prove the convergence of the algorithm under suitable regularity assumptions, and we highlight its structural simplicity and computational efficiency, which make it suitable for real-time applications.

We then extend the methodology to handle inequality constraints. This requires introducing projection steps and adapting the algorithmic structure to ensure feasibility and stability over time. The transition from equality to inequality constraints is addressed rigorously, and we provide a theoretical convergence analysis in the continuous-time setting.

In the final part of this part, we consider a stochastic formulation of the problem, where the objective functions are affected by additive noise and no explicit constraints are imposed. We present an extension of the previous algorithms tailored to this setting and analyze its convergence properties in expectation. This extension highlights the flexibility of the proposed approach and its applicability in uncertain or data-driven environments.

Overall, this part of the thesis introduces an innovative and modular framework for addressing dynamic optimization problems, leveraging control-theoretic approaches to improve algorithm design, and rigorously validating their performance through theoretical analysis.

1.2 PART II: NEURAL ARCHITECTURES

The second part of this thesis focuses on the study and design of novel Artificial Intelligence (AI) neural architectures, drawing inspiration from both biological systems and recent advances in deep learning. In particular, this part explores different research directions that highlight structural and functional properties relevant to multitask learning and dynamic inference.

First, we investigate the structural properties of connectivity matrices in neural networks, motivated by parallels with brain connectivity observed in neuroscience. This perspective aims to uncover how specific patterns of connectivity can influence the functional capabilities of artificial systems, potentially guiding the design of more efficient and interpretable architectures.

Second, we propose a new class of multitask control architectures that extend beyond traditional linear layers. By incorporating structured control mechanisms into the network design, we aim to enhance the system's ability to manage multiple tasks and maintain robust performance over extended temporal horizons.

Overall, this part adopts a systems-oriented perspective on neural architecture design, where ideas such as memory, structure, and adaptability through online regression resonate with the principles developed in the first part of the thesis. The final aim is to develop models and design principles that are both theoretically grounded and capable of addressing a diverse set of learning and inference tasks.

2

Notation, definitions & conventions

This chapter collects the basic notation and the main definitions used throughout the thesis. Unless otherwise stated, all vectors and matrices are real-valued. Bold lowercase letters (e.g., $\mathbf{x}$) denote vectors, bold uppercase letters (e.g., $\mathbf{X}$) denote matrices, and plain letters (e.g., x) denote scalars. The sets $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{R}$, and $\mathbb{C}$ denote the natural, integer, real, and complex numbers, respectively. The identity matrix of order n is $\mathbf{I}_n$ (the subscript is omitted when clear from context). Transpose and conjugate-transpose are denoted by $(\cdot)^\top$ and $(\cdot)^*$, respectively. The (i, j) -entry of a matrix $\mathbf{X}$ is $(\mathbf{X})_{ij}$, and the i -th entry of a vector $\mathbf{x}$ is x_i or $[\mathbf{x}]_i$.

OPERATORS, TRANSFORMS, SIGNALS AND PRINCIPLES. We begin by defining commonly used operations and operators.

Definition 1 (Kronecker and Hadamard products). For $\mathbf{X} \in \mathbb{R}^{n \times m}$ and $\mathbf{Y} \in \mathbb{R}^{p \times q}$, the Kronecker product $\mathbf{X} \otimes \mathbf{Y} \in \mathbb{R}^{np \times mq}$ is the block matrix whose (i, j) -block equals $x_{ij}\mathbf{Y}$. For conformable matrices $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n \times m}$, the Hadamard product $\mathbf{X} \odot \mathbf{Y} \in \mathbb{R}^{n \times m}$ is $(\mathbf{X} \odot \mathbf{Y})_{ij} = x_{ij}y_{ij}$.

Definition 2 (Trace and diagonal operators). The trace is $\text{tr}(\mathbf{X}) := \sum_i (\mathbf{X})_{ii}$. We use $\text{diag}(\mathbf{X}) \in \mathbb{R}^n$ for the vector of diagonal entries of $\mathbf{X}$, and $\text{diag}(\mathbf{x}) \in \mathbb{R}^{n \times n}$ for the diagonal matrix with diagonal $\mathbf{x}$.

Definition 3 (Convex function). Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a scalar-valued function. We say that

f is convex if, for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ and for all $\theta \in [0, 1]$, it holds that

$$f(\theta \mathbf{x} + (1 - \theta) \mathbf{y}) \leq \theta f(\mathbf{x}) + (1 - \theta) f(\mathbf{y}). \quad (2.1)$$

We say that f is strictly convex if

$$f(\theta \mathbf{x} + (1 - \theta) \mathbf{y}) < \theta f(\mathbf{x}) + (1 - \theta) f(\mathbf{y}). \quad (2.2)$$

Definition 4 ($\mathcal{C}^1$ -smooth function). Let $f : \mathbb{R}^n \rightarrow \mathbf{R}$ be a scalar-valued function. We say that f is $\mathcal{C}^1$ -smooth (or simply of class $\mathcal{C}^1(\mathbb{R}^n)$, or equivalently continuously differentiable) if f is differentiable on $\mathbb{R}^n$ and its gradient $\nabla_{\mathbf{x}} f(\mathbf{x})$ is continuous on $\mathbb{R}^n$. For vector-valued functions, the same definition applies component-wise.

Definition 5 (Gradient of a scalar function). For $f : \mathbb{R}^n \rightarrow \mathbb{R}$, the gradient is the column vector

$$\nabla_{\mathbf{x}} f(\mathbf{x}) := \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix}.$$

We now introduce the standard transforms for signals.

Definition 6 (Unilateral $\mathcal{Z}$ -transform). For a discrete-time sequence $\{x_k\}_{\mathbb{N}}$ with $x_k \in \mathbb{R}$, its unilateral $\mathcal{Z}$ -transform is

$$\mathcal{Z}[x_k] := \hat{x}(z) = \sum_{k=0}^{\infty} x_k z^{-k}, \quad z \in \mathbb{C}.$$

For vector/matrix signals we adopt the hat convention, e.g., $\hat{\mathbf{x}}(z) := \mathcal{Z}[\mathbf{x}_k]$.

Definition 7 (Unilateral Laplace transform). For a continuous-time signal $x : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$, its unilateral Laplace transform is

$$\mathcal{L}[x(t)] := \int_0^{\infty} x(t) e^{-st} dt, \quad s \in \mathbb{C}.$$

For vector/matrix signals we use the same convention, e.g., $\hat{\mathbf{x}}(s) := \mathcal{L}[\mathbf{x}(t)]$.

In addition, in the depicted block diagrams, we label the systems blocks in the transform domain, while input signals are denoted in the time domain, following a common control-

theoretic convention. This notation emphasizes that the controller is an Linear Time Invariant (LTI) system acting on the input sequences. Consequently, its outputs are obtained by a linear combinations of the convolutions of the system impulse responses with the inputs. Instead, concerning properties in the transform domain, we recall the following stability notions.

Definition 8 (Schur stability). A monic polynomial $p(z) = z^n + p_{n-1}z^{n-1} + \dots + p_0$ is *Schur stable* if all its roots lie strictly inside the unit disk, i.e., $|z_i| < 1$ for every root z_i . It is *marginally stable* if its roots lie in the closed unit disk.

Definition 9 (Hurwitz stability). A polynomial $p(s) = s^n + p_{n-1}s^{n-1} + \dots + p_0$ is *Hurwitz stable* if all its roots lie strictly in the open left half-plane, i.e., $\text{Re}(s_i) < 0$ for every root s_i . It is *marginally stable* if its roots lie in the closed left half-plane.

We next define fundamental concepts related to controllability and observability of LTI systems. In particular, consider the LTI state-space system

$$\Sigma : \begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t), \end{cases}$$

and assume that $\mathbf{A}$ is Hurwitz. The *controllability* and *observability* Gramians are the unique positive-definite solutions of

$$\begin{aligned} \mathbf{A}\mathbf{P} + \mathbf{P}\mathbf{A}^\top + \mathbf{B}\mathbf{B}^\top &= 0, \\ \mathbf{A}^\top \mathbf{Q} + \mathbf{Q}\mathbf{A} + \mathbf{C}^\top \mathbf{C} &= 0. \end{aligned}$$

In the following we frequently use the notion of a Linear Matrix Inequality (LMI); thus we recall it here.

Definition 10 (LMI). An LMI is an inequality of the form

$$\mathbf{F}_0 + \sum_{i=1}^p \theta_i \mathbf{F}_i \geq 0,$$

where $\theta \in \mathbb{R}^p$ collects decision variables and the symmetric matrices $\mathbf{F}_i$ are given. We write > 0 (≥ 0) for (semi)definiteness.

For a symmetric block matrix, we write

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ * & \mathbf{C} \end{bmatrix} \succ 0 \iff \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^\top & \mathbf{C} \end{bmatrix} \succ 0,$$

i.e., the symbol “*” denotes the transpose of the corresponding off-diagonal block.

In the end, we present a very important principle, relevant for the following derivations, namely the Internal Model Principle (IMP). This principle is central in control theory and can be described as follows. Suppose we consider the classical control scheme depicted in Figure 2.1. Here, η_k represents a reference signal with $\mathcal{Z}$ -transform $\hat{r}(z) = n_r(z)/d_r(z)$, where

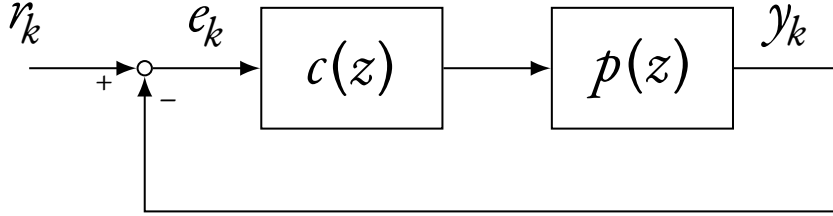


Figure 2.1: General control scheme.

$d_r(z)$ is an unstable polynomial. The asymptotic tracking requirement

$$y_k \xrightarrow[k \rightarrow \infty]{} \eta_k$$

is fulfilled if the series connection $c(z)p(z)$ contains the polynomial $d_r(z)$ as denominator. For this reason, one usually chooses $c(z) = n_c(z)/d_r(z)$ and designs $n_c(z)$ so as to keep the $\mathcal{Z}$ -transform of the error

$$\hat{e}(z) = \frac{d_p(z)n_r(z)}{d_r(z)d_p(z) + n_c(z)n_p(z)},$$

stable by designing $n_c(z)$ such that the polynomial $d_r(z)d_p(z) + n_c(z)n_p(z)$ is stable. Surprisingly, $\hat{e}(z)$ does not contain the poles of $d_r(z)$; this is due to a pole-zero cancellation ensured by choosing $c(z) = n_c(z)/d_r(z)$, that is, with $d_r(z)$ as denominator polynomial.

In essence, these steps encode the IMP: the name comes from the fact that setting $c(z) = n_c(z)/d_r(z)$ is equivalent to embedding the model of the reference, described by $d_r(z)$, inside the controller action, hence the term *internal model principle*. The IMP is equally valid in continuous time as well.

NORMS, SINGULAR VALUES, GAINS, AND CONDITION NUMBER. For vectors, $\|\mathbf{x}\|_2 = \sqrt{\sum_i x_i^2}$ and $\|\mathbf{x}\|_1 = \sum_i |x_i|$. For matrices, the induced (spectral) 2-norm is $\|\mathbf{X}\|_2 = \sigma_{\max}(\mathbf{X})$, where $\sigma_{\max}(\mathbf{X})$ and $\sigma_{\min}(\mathbf{X})$ denote the largest and smallest singular values, and the Frobenius norm is $\|\mathbf{X}\|_F = \sqrt{\text{tr}(\mathbf{X}^\top \mathbf{X})}$. The (matrix) condition number in the 2-norm is

$$\kappa(\mathbf{X}) := \|\mathbf{X}\|_2 \|\mathbf{X}^{-1}\|_2.$$

For symmetric positive definite $\mathbf{X}$, $\kappa(\mathbf{X}) = \frac{\lambda_{\max}(\mathbf{X})}{\lambda_{\min}(\mathbf{X})}$, where $\lambda_{\max}(\mathbf{X})$ and $\lambda_{\min}(\mathbf{X})$ denote the largest and smallest eigenvalues.

Regarding system norms, let $\mathbf{G}$ be a stable LTI transfer matrix. In continuous time, i.e. $\mathbf{G} \in \mathbb{R}^{n \times m}[s]$,

$$\|\mathbf{G}\|_2^2 := \frac{1}{2\pi} \int_{-\infty}^{\infty} \text{tr}(\mathbf{G}(j\omega)^* \mathbf{G}(j\omega)) d\omega, \quad \|\mathbf{G}\|_\infty := \sup_{\omega \in \mathbb{R}} \sigma_{\max}(\mathbf{G}(j\omega)),$$

and in discrete time, i.e. $\mathbf{G} \in \mathbb{R}^{n \times m}[z]$,

$$\|\mathbf{G}\|_2^2 := \frac{1}{2\pi} \int_0^{2\pi} \text{tr}(\mathbf{G}(e^{j\omega})^* \mathbf{G}(e^{j\omega})) d\omega, \quad \|\mathbf{G}\|_\infty := \sup_{\omega \in [0, 2\pi)} \sigma_{\max}(\mathbf{G}(e^{j\omega})).$$

Let $\mathbf{u} \mapsto \mathbf{y}$ be the input–output map of a stable system with zero initial condition. The system has a *finite* ℓ_2 -gain or *finite* $\mathcal{L}_2$ -gain $\gamma \in \mathbb{R}_{>0}$ from input $\mathbf{u}$ to output $\mathbf{y}$ if

$$\|\mathbf{y}\|_2 \leq \gamma \|\mathbf{u}\|_2,$$

for all admissible inputs $\mathbf{u}$. For LTI systems, the smallest such γ equals the $\mathcal{H}_\infty$ norm of the transfer matrix $\mathbf{G}$.

To conclude, we introduce a few concepts from probability and statistics. We denote expectation by $\mathbb{E}[\cdot]$. Random vectors are in bold (e.g., $\mathbf{w}$), and their covariance is $\mathbb{E}[(\mathbf{w} - \mathbb{E}[\mathbf{w}])(\mathbf{w} - \mathbb{E}[\mathbf{w}])^\top]$.

Definition 11. A random vector $\mathbf{x} \in \mathbb{R}^n$ is *Gaussian with mean* $\mathbf{m} \in \mathbb{R}^n$ *and covariance* $\mathbf{S} \in \mathbb{R}^{n \times n}$, denoted $\mathbf{x} \sim \mathcal{N}(\mathbf{m}, \mathbf{S})$. If $\mathbf{S} > 0$, $\mathbf{x}$ admits the density

$$\phi(\mathbf{x}) = \frac{1}{(2\pi)^{n/2} \det(\mathbf{S})^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{m})^\top \mathbf{S}^{-1}(\mathbf{x} - \mathbf{m})\right), \quad \mathbf{x} \in \mathbb{R}^n.$$

As a special case, for $n = 1$, $x \sim \mathcal{N}(m, \sigma^2)$ has density

$$\phi(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-m)^2}{2\sigma^2}\right), \quad x \in \mathbb{R}.$$

Part I

Models for online optimization

The *online optimization* is an emerging and increasingly relevant field at the intersection of mathematics, optimization, and control theory. Optimization is a well-established area in mathematics, and many concepts from control theory have been successfully applied to it.

However, much of the existing literature focuses on the *static* setting, where the optimization problem remains fixed across the iterations of the algorithm. Control-theoretic ideas have been leveraged to design such algorithms in several works, for instance, in distributed optimization; see [117, 23, 114]. In contrast, this part of the thesis is primarily concerned with a different, yet increasingly important, class of problems: time-varying formulations. This shift in focus is motivated not only by innovative and practically relevant applications, such as MPC [75, 90], feedback optimization [21], signal processing [63, 50, 85], and machine learning [100, 43, 35, 118, 66], but also by the observation that tools from dynamical systems and control theory are naturally suited to address the challenges posed by this setting.

Indeed, *time-varying* problems evolve over time, and as such, benefit from algorithmic designs that account for temporal dynamics. In this context, expertise from control theory can be successfully adapted to improve performance guarantees and convergence properties of online optimization algorithms. The class of algorithms considered in this thesis builds upon *first-order* methods [25], that is, methods that rely on access to first-order information on the cost function from an oracle, such as gradients, Lagrangian gradients, and related quantities.

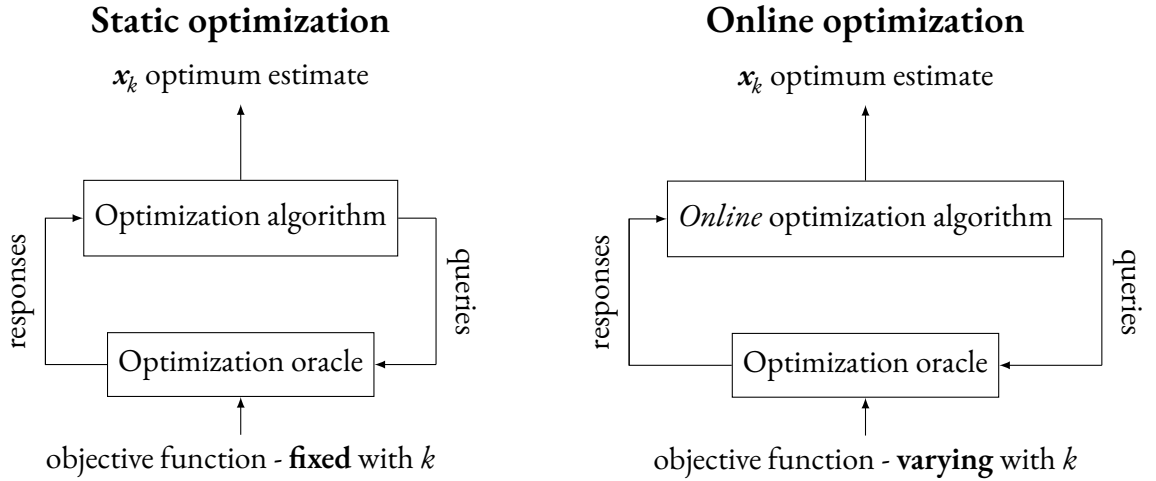


Figure 2.2: General framework for static optimization (left) and online optimization (right).

The scheme depicted in the right part of Figure 2.2 highlights the conceptual connection between control and optimization. The key insight is that the fundamental components in the figure mirror those in a feedback control loop. The first block, labeled online optimization algorithm, receives real-time information from the oracle at time k and returns an estimate $\mathbf{x}_k$ of the current optimal variable $\mathbf{x}_k^*$. In a control-theoretic setting, this algorithm can be interpreted as a Multi-Input Multi-Output (MIMO) discrete-time controller. Conversely, the oracle, which supplies problem data via one or more queries (e.g., gradient evaluations), acts as the plant to be controlled.

From this perspective, the goal of the optimization algorithm becomes analogous to that of a controller: driving the optimization error (or the gradient) to zero. Consequently, techniques from control theory can be tailored and leveraged to synthesize online optimization algorithms, using the same design principles and performance specifications typically adopted in control applications.

In this thesis, we adopt this viewpoint and develop a family of online optimization algorithms by leveraging the analogy between control and optimization. The algorithms are tailored to different problem formulations, determined by the structure and information pattern of the oracle, and are designed to meet distinct performance objectives.

Part I is mostly based on the following works:

- [OO1] Casti, U., Bastianello, N., Carli, R., & Zampieri, S. (2024). A control theoretical approach to online constrained optimization. *Automatica*.
- [OO2] Casti, U., & Zampieri, S. (2025). Anti-windup design for internal model online constrained optimization. In *2025 64th IEEE Conference on Decision and Control (CDC)*.
- [OO3] Casti, U., & Zampieri, S. (2024). Stochastic models for online optimization. In *2025 European Control Conference (ECC)*.

3

Introduction

Recent advances in modern technology have made online optimization problems increasingly relevant in a wide range of dynamic applications, including control [75, 90], signal processing [63, 50, 85], and machine learning [100, 43, 35]. These problems are characterized by objective functions and constraints that evolve over time due to external disturbances, underlying dynamics, or streaming data. As a result, solving a single static problem is insufficient, and the focus shifts to solving a sequence of optimization problems in an online fashion commonly in real time.

A classical strategy to address general online optimization problems is to extend static optimization techniques such as gradient descent, projected gradient descent, or primal-dual gradient descent, to the dynamic setting. These *unstructured* methods treat each time instance of the objective function independently and do not account for temporal coupling in the evolution of cost functions and constraints [100, 63, 98, 29, 124, 78]. While simple and broadly applicable, unstructured methods generally provide limited tracking accuracy, especially in rapidly changing or non-stationary environments [103].

To overcome these limitations, recent works have explored *structured* algorithms that incorporate prior knowledge or assumptions about the temporal evolution of the problem. These include prediction-correction schemes [102, 12], interior-point methods [48], and virtual queue-based algorithms [30]. Structured approaches often assume some regularity in the optimum trajectory, such as bounded time derivatives, and generally offer improved performance over unstructured methods. Nevertheless, most structured approaches do not

fully exploit a precise model of the dynamics governing the evolution of the problem.

In contrast, model-based approaches explicitly embed the time variation of the cost function within a control-theoretic framework. Notably, [13] proposes a deterministic algorithm based on the Internal Model Principle (IMP), reformulating the optimization task as a tracking problem for a linear dynamical system. More recently, control-theoretic tools have been employed to design structured algorithms in both deterministic and stochastic settings [32, 33, 104, 39, 105]. These contributions rely on techniques such as Kalman filtering [104], robust $\mathcal{H}_\infty$ control [33], and contraction theory [39] to enhance the robustness and performance of online optimization methods.

This thesis builds upon and unifies these ideas, advancing the control-theoretic approach to constrained and stochastic online optimization. The central idea is to reinterpret the online optimization problem, illustrated in Figure 2.2, as a control problem. More specifically, the time-varying objective function is viewed as the plant to be controlled, for which we possess either a deterministic or a stochastic model. Leveraging robust control techniques such as the IMP, Anti-Windup (AW) mechanisms, Kalman filtering, optimal control, and robust $\mathcal{H}_\infty$ control, we develop novel online optimization algorithms that achieve strong theoretical guarantees and practical effectiveness in tracking performance. This perspective aligns with a growing body of work that applies control-theoretic analysis and synthesis to the design of optimization algorithms [20, 73, 97, 39].

4

Preliminaries

We are interested in online optimization problems with equality and inequality constraints of the form

$$\begin{aligned} \mathbf{x}_k^* = \arg \min_{\mathbf{x} \in \mathbb{R}^n} \quad & f_k(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{h}_k(\mathbf{x}) = 0, \\ & \mathbf{g}_k(\mathbf{x}) \leq 0 \end{aligned} \tag{4.1}$$

where $k \in \mathbb{N}$, $f_k : \mathbb{R}^n \times \mathbb{N} \rightarrow \mathbb{R}$ is a convex objective function, $\mathbf{h}_k : \mathbb{R}^n \times \mathbb{N} \rightarrow \mathbb{R}^l$ collects equality constraints, and $\mathbf{g}_k : \mathbb{R}^n \times \mathbb{N} \rightarrow \mathbb{R}^p$ the inequality constraints. All these mappings are assumed to be continuously differentiable as stated in the preliminaries 2.

In the central analysis of this thesis, we study problems with the following structure:

$$f_k(\mathbf{x}) := \frac{1}{2} \mathbf{x}^\top \mathbf{A} \mathbf{x} + \mathbf{b}_k^\top \mathbf{x} + \tilde{d}_k \tag{4.2a}$$

$$\mathbf{h}_k(\mathbf{x}) := \mathbf{H} \mathbf{x} - \mathbf{h}_k, \tag{4.2b}$$

$$\mathbf{g}_k(\mathbf{x}) := \mathbf{G} \mathbf{x} - \mathbf{g}_k, \tag{4.2c}$$

where $\mathbf{A} = \mathbf{A}^\top \succ 0 \in \mathbb{R}^{n \times n}$, $\mathbf{H} \in \mathbb{R}^{l \times n}$, and $\mathbf{G} \in \mathbb{R}^{p \times n}$ are fixed but unknown matrices and $[\mathbf{H}^\top, \mathbf{G}^\top]^\top$ is full row rank, while $\mathbf{b}_k \in \mathbb{R}^n$, $\mathbf{h}_k \in \mathbb{R}^l$, and $\mathbf{g}_k \in \mathbb{R}^p$ are generated by known deterministic or stochastic LTI systems, and $\tilde{d}_k$ is a general time-varying term.

Remark 1. We highlight that the objective function (4.2a) can be equivalently written as

$$f_k(\mathbf{x}) = \frac{1}{2} (\mathbf{x} - \mathbf{c}_k)^\top \mathbf{A} (\mathbf{x} - \mathbf{c}_k) + d_k, \quad (4.3)$$

with $\mathbf{c}_k := -\mathbf{A}^{-1}\mathbf{b}_k$ and $d_k := \tilde{d}_k - (1/2) \mathbf{b}_k^\top \mathbf{A}^{-1} \mathbf{b}_k$. While this form makes the optimizer $\mathbf{c}_k$ explicit, in online optimization one typically does not know the full objective function: the algorithm assumes structural properties (e.g., quadratic, convex) but the exact parameters $(\mathbf{A}, \mathbf{b}_k, \tilde{d}_k)$ or $(\mathbf{A}, \mathbf{c}_k, d_k)$ are unknown and can only be inferred through oracle queries, such as gradient evaluations. For this reason, studying Problem (4.1) with objective (4.2a) or with its equivalent representation (4.3) is equally meaningful. In what follows, and throughout the thesis, we leverage this equivalence to clarify the problem structure and inform algorithm design.

4.1 BACKGROUND ON OPTIMIZATION

Here we summarize essential results on *static* optimization; see, e.g., the standard textbooks [25, 86]. We consider the static equivalent problem of Problem (4.1), that is

$$\begin{aligned} \mathbf{x}^* = \arg \min_{\mathbf{x} \in \mathbb{R}^n} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{h}(\mathbf{x}) = 0, \\ & \mathbf{g}(\mathbf{x}) \leq 0 \end{aligned} \quad (4.4)$$

where, as before, $f : \mathbb{R}^n \times \mathbb{N} \rightarrow \mathbb{R}$ is a convex objective function, $\mathbf{h} : \mathbb{R}^n \times \mathbb{N} \rightarrow \mathbb{R}^l$ collects equality constraints, and $\mathbf{g} : \mathbb{R}^n \times \mathbb{N} \rightarrow \mathbb{R}^p$ the inequality constraints*. All these mappings are assumed to be continuously differentiable as stated in the preliminaries 2.

We begin by recalling the Lagrangian function.

Definition 12 (Lagrangian function). The Lagrangian associated with Problem (4.4) is the mapping $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^l \times \mathbb{R}^p \rightarrow \mathbb{R}$ defined as

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\eta}, \boldsymbol{\gamma}) := f(\mathbf{x}) + \boldsymbol{\eta}^\top \mathbf{h}(\mathbf{x}) + \boldsymbol{\gamma}^\top \mathbf{g}(\mathbf{x}), \quad (4.5)$$

where $\boldsymbol{\eta} \in \mathbb{R}^l$ and $\boldsymbol{\gamma} \in \mathbb{R}^p$ are the vectors of Lagrange multipliers associated with the equality and inequality constraints, respectively.

*We called *feasible set* all the points $\mathbf{x} \in \mathbb{R}^n$ such that the constraints are satisfied.

In the remainder of this subsection we present standard first-order optimality conditions for the static case.

Theorem 1 (KKT conditions). *Suppose that $\mathbf{x}^*$ is a solution to (4.4) and that a standard constraint qualification holds at $\mathbf{x}^*$ (for example, the Linear Independence Constraint Qualification (LICQ), see [86, Definition 12.4]), or when $\mathbf{h}$ and $\mathbf{g}$ are linear affine functions with linearly independent rows). Then there exist Lagrange multipliers $\boldsymbol{\eta}^* \in \mathbb{R}^l$ and $\boldsymbol{\gamma}^* \in \mathbb{R}^p$ such that the following conditions are satisfied at $(\mathbf{x}^*, \boldsymbol{\eta}^*, \boldsymbol{\gamma}^*)$:*

$$\text{stationarity:} \quad \nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}^*, \boldsymbol{\eta}^*, \boldsymbol{\gamma}^*) = 0, \quad (4.6a)$$

$$\text{primal feasibility (i):} \quad \mathbf{h}(\mathbf{x}^*) = 0, \quad (4.6b)$$

$$\text{primal feasibility (ii):} \quad \mathbf{g}(\mathbf{x}^*) \leq 0, \quad (4.6c)$$

$$\text{dual feasibility:} \quad \boldsymbol{\gamma}^* \geq 0, \quad (4.6d)$$

$$\text{complementary slackness:} \quad \boldsymbol{\gamma}^* \odot \mathbf{g}(\mathbf{x}^*) = 0. \quad (4.6e)$$

The proof of Theorem 1 can be found in classical references; see, e.g., [86, Theorem 12.1]. We now briefly explain the meaning of each of the conditions (4.6). The relations in (4.6) play the following roles:

- *Stationarity* (4.6a): the optimizer $\mathbf{x}^*$ is a critical point of the Lagrangian with respect to $\mathbf{x}$.
- *Primal feasibility* (4.6b)&(4.6c): the constraints are satisfied at $\mathbf{x}^*$.
- *Dual feasibility* (4.6d): the multipliers of the inequality constraints are nonnegative.
- *Complementary slackness* (4.6e): an inequality constraint is either active at $\mathbf{x}^*$ (in which case the corresponding multiplier may be positive) or inactive (in which case the corresponding multiplier is zero).

We say that an *inequality* constraint is *active* at $\mathbf{x}^*$ if it holds with equality, and *inactive* if it holds with strict inequality. The Karush–Kuhn–Tucker (KKT) conditions are, in general, *necessary* for local optimality under mild constraint qualifications (e.g., LICQ, Mangasarian–Fromovitz Constraint Qualification (MFCQ), or Slater’s condition). In the setting of this

thesis, namely, with

$$f(\mathbf{x}) := \frac{1}{2} (\mathbf{x} - \mathbf{c})^\top \mathbf{A} (\mathbf{x} - \mathbf{c}) + d \quad (4.7a)$$

$$\mathbf{h}(\mathbf{x}) := \mathbf{H}\mathbf{x} - \mathbf{h}, \quad (4.7b)$$

$$\mathbf{g}(\mathbf{x}) := \mathbf{G}\mathbf{x} - \mathbf{g}, \quad (4.7c)$$

where $\mathbf{A} = \mathbf{A}^\top > 0 \in \mathbb{R}^{n \times n}$, $\mathbf{H} \in \mathbb{R}^{l \times n}$, and $\mathbf{G} \in \mathbb{R}^{p \times n}$ and $[\mathbf{H}^\top, \mathbf{G}^\top]^\top$ is full row rank, while $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{h} \in \mathbb{R}^l$, and $\mathbf{g} \in \mathbb{R}^p$, and $d \in \mathbb{R}$, they are also sufficient.

Theorem 2 (Sufficiency in the convex quadratic/affine case). *Consider Problem (4.4) with f strictly convex as in (4.7a) and affine constraints as in (4.7b)–(4.7c). If $(\mathbf{x}^*, \boldsymbol{\eta}^*, \boldsymbol{\gamma}^*)$ satisfies (4.6), that is,*

$$\mathbf{A} (\mathbf{x}^* - \mathbf{c}) + \mathbf{H}^\top \boldsymbol{\eta}^* + \mathbf{G}^\top \boldsymbol{\gamma}^* = 0,$$

$$\mathbf{H}\mathbf{x}^* - \mathbf{h} = 0,$$

$$\mathbf{G}\mathbf{x}^* - \mathbf{g} \leq 0,$$

$$\gamma_i^* = 0 \text{ if } i\text{-th constraint is active and } \gamma_i^* > 0 \text{ if } i\text{-th constraint is inactive,}$$

then $\mathbf{x}^*$ is the unique global minimizer of (4.4).

EQUALITY-CONSTRAINED CASE ONLY. If only equality constraints are present (i.e., $p = 0$), conditions (4.6c)–(4.6e) disappear, and the KKT system reduces to

$$\begin{bmatrix} \nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \boldsymbol{\eta}) \\ \nabla_{\boldsymbol{\eta}} \mathcal{L}(\mathbf{x}, \boldsymbol{\eta}) \end{bmatrix} = \begin{bmatrix} \mathbf{A}(\mathbf{x} - \mathbf{c}) + \mathbf{H}^\top \boldsymbol{\eta} \\ \mathbf{H}\mathbf{x} - \mathbf{h} \end{bmatrix} = 0. \quad (4.8)$$

That is, the gradient of the Lagrangian must vanish on $\mathbb{R}^n \times \mathbb{R}^l$. Moreover, it can be shown that any such point is a saddle point of the original Lagrangian $\mathcal{L}$. This characterization will be used later to establish convergence of the equality-constrained online method (see Chapter 7).

UNCONSTRAINED CASE.

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = 0, \quad (4.9)$$

which is well known to be *necessary and sufficient* for optimality when f is strictly convex.

While the structure specified in Problem (4.1) and (4.2) is central to the synthesis and analysis of the proposed algorithms, we also assess the methods in broader settings through numerical simulations, including cases with non-quadratic objective functions $f_k(\mathbf{x})$ and time-varying matrix $\mathbf{A}$.

The imposed structure plays a similar role to that of linearization in control theory: it simplifies analysis and algorithm design, while still offering useful insights into more complex systems. If the algorithm is well designed, the optimizer can remain close to the true solution, even when the actual problem deviates from the linearized regime. Motivated by this analogy with control theory, we develop our optimization strategies based on the proposed case, and validate their robustness and effectiveness in more general scenarios.

Remark 2. *The time-varying form of the objective function (4.2a) can be interpreted, and extended, to a broader class of problems. Indeed, it can be seen as a local quadratic approximation of a more general time-varying cost function of the form*

$$f_k(\mathbf{x}) = f(\mathbf{x} - \mathbf{c}_k) + d_k, \quad (4.10)$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a fixed, smooth, strongly convex function with a unique minimum at $\mathbf{x} = 0$, and where $\mathbf{c}_k \in \mathbb{R}^n$, $d_k \in \mathbb{R}$ are time-varying.

A second-order Taylor expansion of $f(\mathbf{x})$ around its minimizer $\mathbf{x} = 0$ yields

$$f(\mathbf{x}) \approx \frac{1}{2} \mathbf{x}^\top \mathbf{A} \mathbf{x} + f(0),$$

where $\mathbf{A}$ is the Hessian of $f(\mathbf{x})$ evaluated at $\mathbf{x} = 0$. The linear term vanishes because f attains its minimum at the origin. Consequently, $f(\mathbf{x}_k)$ is well approximated by (4.2a), provided that $\mathbf{x}_k$ remains close to $\mathbf{c}_k$.

This construction mirrors the classical control strategy of linearizing a nonlinear system around a reference trajectory, which is valid only if the control action keeps the system state sufficiently close to that trajectory.

Analogously, the cost function in (4.2a) can be interpreted as a local approximation of a generic time-varying, smooth, strongly convex function $f_k(\mathbf{x})$, under the assumption that the Hessian at its minimizer $\mathbf{c}_k$ is constant or slowly varying over time.

In addition to these hypotheses, we align ourselves with the existing literature on the on-line optimization framework by not assuming an exact knowledge of the optimization problem. This aspect is central to our derivation: the variables and parameters in (4.2) are par-

tially unknown, subject to uncertainty, or exact computation may be infeasible in real time. For this reason, we relax our derivation by not assuming full knowledge of the optimization problem and instead consider different levels of prior information.

In particular, we formalize a hierarchy of assumptions, ranging from minimal to more restrictive, that reflect increasing knowledge about the problem structure and its temporal evolution. These assumptions will be used throughout the subsequent analysis.

Assumption 1.1 (Known eigenvalues). *The matrix $\mathbf{A}$ is symmetric positive definite and has known eigenvalues λ_i , for $i = 1, \dots, n$.*

This assumption is commonly used as a starting point. In practice, full knowledge of the matrix $\mathbf{A}$, especially its eigenvectors or exact structure, is often unrealistic. Hence, assuming only the eigenvalues are known provides a more general framework.

Assumption 1.2 (Known eigenvalue distribution). *The matrix $\mathbf{A}$ is symmetric positive definite, and its eigenvalues are not known individually, but are assumed to be drawn from a known probability density function $\phi(\cdot)$.*

Even knowledge of individual eigenvalues may be too strong in many practical scenarios. A more reasonable and application-driven assumption is to know the statistical distribution of the eigenvalues. For instance, in the case of high dimensional regression problems, the eigenvalue distribution may be characterized by the Marchenko–Pastur law [81].

Assumption 1.3 (Known bounds on eigenvalues). *The matrix $\mathbf{A}$ is symmetric positive definite and unknown, but its eigenvalues λ_i lie within known bounds $0 < \lambda_{\min} \leq \lambda_{\max}$, such that $\lambda_i \in [\lambda_{\min}, \lambda_{\max}]$ for all i .*

This is the least restrictive assumption and is widely adopted in the literature. It is equivalent to assuming that the cost function (4.3) (or (4.2a)) is $\lambda_{\min}$ -strongly convex and $\lambda_{\max}$ -smooth for every $k \in \mathbb{N}$ [13].

Similarly to the case of the matrix $\mathbf{A}$, the information available about the constraints is also incomplete. We formalize the oracle knowledge about the constraints with the following assumption.

Assumption 1.4 (Known bounds on singular values). *The matrix $[\mathbf{H}^\top, \mathbf{G}^\top]^\top$ is unknown, but its singular values σ_i lie within known bounds $0 < \sigma_{\min} \leq \sigma_{\max}$, such that $\sigma_i \in [\sigma_{\min}, \sigma_{\max}]$ for all i .*

Regarding the online nature of the optimization problem, we assume that the time-varying components are driven by an underlying model. In particular, we make a fundamental assumption on the evolution of the parameters $\mathbf{c}_k$, $\mathbf{g}_k$, and $\mathbf{h}_k$ in (4.2), which are assumed to follow specific dynamics described in the next assumption:

Assumption 1.5 (Models of $\mathbf{c}_k$, $\mathbf{h}_k$, and $\mathbf{g}_k$). *The sequences $\{\mathbf{c}_k\}_{k \in \mathbb{N}}$, $\{\mathbf{h}_k\}_{k \in \mathbb{N}}$, and $\{\mathbf{g}_k\}_{k \in \mathbb{N}}$ are generated by the following LTI models:*

$$\begin{cases} \mathbf{z}_{k+1} &= (\mathbf{I}_n \otimes \mathbf{F}) \mathbf{z}_k, \\ \mathbf{w}_{k+1} &= (\mathbf{I}_l \otimes \mathbf{F}) \mathbf{w}_k, \\ \mathbf{y}_{k+1} &= (\mathbf{I}_p \otimes \mathbf{F}) \mathbf{y}_k, \\ \mathbf{c}_k &= (\mathbf{I}_n \otimes \mathbf{H}) \mathbf{z}_k, \\ \mathbf{h}_k &= (\mathbf{I}_l \otimes \mathbf{H}) \mathbf{w}_k, \\ \mathbf{g}_k &= (\mathbf{I}_p \otimes \mathbf{H}) \mathbf{y}_k. \end{cases} \quad (4.11)$$

where $\mathbf{F} \in \mathbb{R}^{m \times m}$ and $\mathbf{H} \in \mathbb{R}^{1 \times m}$ are known, have marginally Schur-stable poles, and the initial states $\mathbf{z}_0$, $\mathbf{w}_0$, and $\mathbf{y}_0$ are unknown. The signals generated by (4.11) admit rational $\mathcal{Z}$ -transforms:

$$\hat{\mathbf{c}}(z) := \mathcal{Z}[\mathbf{c}_k] = \frac{\mathbf{c}_N(z)}{p(z)}, \quad (4.12a)$$

$$\hat{\mathbf{h}}(z) := \mathcal{Z}[\mathbf{h}_k] = \frac{\mathbf{h}_N(z)}{p(z)}, \quad (4.12b)$$

$$\hat{\mathbf{g}}(z) := \mathcal{Z}[\mathbf{g}_k] = \frac{\mathbf{g}_N(z)}{p(z)}, \quad (4.12c)$$

where, consistent with (4.11), the denominator $p(z) = z^m + \sum_{i=0}^{m-1} p_i z^i = \det(z\mathbf{I} - \mathbf{F})$ is known, while the numerators $\mathbf{c}_N(z) \in \mathbb{R}^n[z]$, $\mathbf{h}_N(z) \in \mathbb{R}^l[z]$, and $\mathbf{g}_N(z) \in \mathbb{R}^p[z]$ are unknown.

Remark 3. *The requirements in (4.11) and (4.12) are equivalent. In particular, (4.12) states that all three models can be expressed as transfer functions with a common polynomial denominator. This is not reductive but simplify the following analysis, and can be always guaranteed as it is always possible taking the least common denominator among (4.12a), (4.12b), and (4.12c). The resulting rational representations describe the same time-domain signals.*

In the stochastic setting, the same model structure given in (4.12) applies. However, in

this case, the signals $\mathbf{c}_k$, $\mathbf{b}_k$ and $\mathbf{g}_k$ in (4.2) should be interpreted as the outputs of stochastic LTI systems (4.11) with transfer matrices defined in (4.12), driven by white Gaussian noise.

Remark 4. *We maintain the general problem structure defined in (4.1), and then specialize the objective function and constraints as in (4.2) to preserve a decoupled formulation. The algorithms and synthesis methods proposed in this work are developed specifically for the structure given in (4.2), and consequently, the convergence analysis is formally valid only under these assumptions.*

Nevertheless, this modeling approach is common in control theory. For instance, classical controllers such as PID controllers are typically designed under simplifying assumptions (e.g., linear dynamics), while in practice, the controlled system may exhibit significant nonlinearities. If the controller is sufficiently robust, it often performs well even outside its nominal design conditions.

The same philosophy applies here: while the algorithms and most convergence results are derived under the assumption that the objective and constraint functions conform to the model structure in (4.2) and (4.12), we also evaluate the algorithms numerically in more complex scenarios. These simulations demonstrate that the proposed methods exhibit good convergence properties even in settings that deviate from the assumed model.

Table 4.1 summarizes the attacked problems by mapping problem classes (unconstrained, equality-, and inequality-constrained) against deterministic and stochastic settings, instead topics not treated are explicitly marked as out of scope.

	Deterministic	Stochastic
Unconstrained	(Chap. 5)	(Chap. 9)
Inequality constraints projected methods	(Chap. 6)	(OUT OF SCOPE)
Equality constraints	(Chap. 7)	(OUT OF SCOPE)
Inequality constraints primal/dual methods	(Chap. 8)	(OUT OF SCOPE)

Table 4.1: Summary of the problem classes and methods covered in this thesis.

5

Deterministic models for unconstrained online optimization

In this chapter, we consider the first entry of Table 4.1. For completeness, and because it serves as the foundation for all later developments, we summarize the results reported in [13]. Specifically, the authors of [13] study and design an algorithm for Problem (4.1) with objective (4.3), under Assumption 1.5 with the time-varying term $\mathbf{c}_k$ as in (4.12a), and without equality or inequality constraints (i.e., $p = l = 0$ in (4.1)). The additive term d_k is omitted, as it does not affect the minimizer. We thus focus on the following optimization problem:

$$\mathbf{x}_k^* = \arg \min_{\mathbf{x} \in \mathbb{R}^n} \underbrace{\frac{1}{2} (\mathbf{x} - \mathbf{c}_k)^\top \mathbf{A} (\mathbf{x} - \mathbf{c}_k)}_{f_k(\mathbf{x})}. \quad (5.1)$$

Following [13], we adopt an algorithmic design that assumes access to an oracle (Figure 2.2), which provides the gradient $\nabla f_k(\mathbf{x})$ of the cost function $f_k(\mathbf{x})$ evaluated at the estimated optimum $\mathbf{x}$. The design assumes knowledge of bounds on the eigenvalues of $\mathbf{A}$, i.e., Assumption 1.3, and on the model driving the time-varying term $\mathbf{c}_k$.

Remark 5. *The solution to Problem 5.1 is trivially $\mathbf{x}_k^* = \mathbf{c}_k$, and it can be computed in closed form from the oracle evaluation at each k as $\mathbf{c}_k = \mathbf{x}_k - \mathbf{A}^{-1} \nabla f_k(\mathbf{x}_k)$. The fundamental difficulty is that $\mathbf{A}$ may be unknown (e.g., due to noisy gradient measurements or estimation procedures), or computing/applying $\mathbf{A}^{-1}$ may be prohibitively expensive in high-dimensional settings for*

real-time operation. For these reasons, iterative optimization methods that estimate the optimum $\mathbf{c}_k$ from gradient information are preferable.

One of the most widely used algorithms to solve Problem 5.1 is the Online Gradient Descent (OGD), namely:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \mathbf{e}_k, \quad (5.2)$$

where

$$\mathbf{e}_k := \nabla f_k(\mathbf{x}_k) = \mathbf{A}(\mathbf{x}_k - \mathbf{c}_k). \quad (5.3)$$

However, when $\mathbf{c}_k$ is time-varying, algorithm (5.2) generally does not converge to the true optimum. In contrast, in this chapter we present a control-inspired algorithm based on the IMP that guarantees

$$\lim_{k \rightarrow +\infty} \|\mathbf{x}_k - \mathbf{x}_k^*\| = 0. \quad (5.4)$$

5.1 ALGORITHM

In this section, we provide a control-inspired algorithm achieving (5.4). This result is obtained by leveraging control-theoretic ideas, specifically, by merging robust control, optimization, and the IMP. In particular, by adopting the control scheme illustrated in Figure 5.1, which refines the general structure in Figure 2.2 for this specific problem, with the oracle providing the gradient of the objective function and the online optimization algorithm implemented as a linear filter, we achieve this goal.

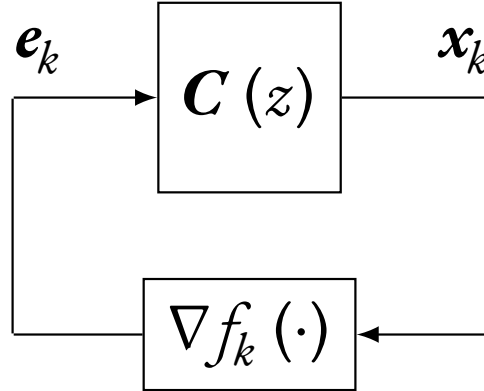


Figure 5.1: Proposed control scheme for the solution of Problem (5.1).

The objective, from a control perspective, is to synthesize a controller $\mathbf{C}(z) \in \mathbb{R}^{n \times n}(z)$

such that (5.4) is satisfied, or equivalently (see Equation (4.9)), that

$$\mathbf{e}_k \xrightarrow[k \rightarrow \infty]{} 0. \quad (5.5)$$

5.1.1 INTERNAL MODEL

Specifically, let us define the following $\mathcal{Z}$ -transforms:

$$\hat{\mathbf{x}}(z) := \mathcal{Z}[\mathbf{x}_k], \quad (5.6a)$$

$$\hat{\mathbf{e}}(z) := \mathcal{Z}[\mathbf{e}_k], \quad (5.6b)$$

$$\hat{\mathbf{c}}(z) := \mathcal{Z}[\mathbf{c}_k], \quad (5.6c)$$

that always exist given $\hat{\mathbf{c}}(z)$ as in Equation (4.12a). Then, the control relation among the relevant signals in Figure 5.1 is equivalently expressed by the matrix equation

$$\hat{\mathbf{x}}(z) = \mathbf{C}(z) \hat{\mathbf{e}}(z). \quad (5.7)$$

From (5.3), we also have

$$\hat{\mathbf{e}}(z) = \mathbf{A} (\hat{\mathbf{x}}(z) - \hat{\mathbf{c}}(z)). \quad (5.8)$$

Let,

$$\hat{\mathbf{b}}(z) := -\mathbf{A}\hat{\mathbf{c}}(z) = \frac{\mathbf{b}_N(z)}{p(z)} \quad (5.9)$$

where $\mathbf{b}_N(z) := -\mathbf{A}\mathbf{c}_N(z)$. In this way we obtain*

$$\hat{\mathbf{e}}(z) = [\mathbf{I} - \mathbf{A}\mathbf{C}(z)]^{-1} \hat{\mathbf{b}}(z). \quad (5.10)$$

Observe that, by Assumption 1.5, $\hat{\mathbf{c}}(z)$ is rational and hence $\hat{\mathbf{b}}(z)$ and $\hat{\mathbf{e}}(z)$ are rational as well. Therefore, if we prove that their poles are stable (Definition 8), namely inside the unit circle, then we can conclude that the signal $\mathbf{e}_k$ converges to zero; i.e., (5.5) holds.

To this objective, we apply the well-known IMP [27]. The first step is to choose the controller $\mathbf{C}(z)$ so as to cancel the marginally stable poles of $\hat{\mathbf{b}}(z)$, i.e., the zeros of $p(z)$ introduced in Assumption 1.5, which are not directly influenced by the feedback loop in (5.10).

*The matrix is invertible for values of z different from the poles of the transfer function.

Since we assume that all the roots of $p(z)$ are marginally stable, it is sufficient to choose

$$\mathbf{C}(z) = \frac{\mathbf{C}_N(z)}{p(z)}, \quad (5.11)$$

where $\mathbf{C}_N(z) \in \mathbb{R}^{n \times n}[z]$ is a polynomial matrix. Indeed, this choice yields

$$\hat{\mathbf{e}}(z) = [p(z) \mathbf{I} - \mathbf{A}\mathbf{C}_N(z)]^{-1} \mathbf{b}_N(z). \quad (5.12)$$

Hence the poles of $\hat{\mathbf{e}}(z)$ coincide with the poles of $[p(z) \mathbf{I} - \mathbf{A}\mathbf{C}_N(z)]^{-1}$, and the goal is now to determine $\mathbf{C}_N(z)$ such that all the poles of $[p(z) \mathbf{I} - \mathbf{A}\mathbf{C}_N(z)]^{-1}$ are stable.

5.1.2 LTI CONTROLLER

In general, to achieve (5.4) we can exploit the degrees of freedom afforded by the MIMO controller $\mathbf{C}(z) \in \mathbb{R}^{n \times n}(z)$. However, we must account for the lack of knowledge of $\mathbf{A}$. For this reason, we fix

$$\mathbf{C}_N(z) = c(z) \mathbf{I}, \quad (5.13)$$

where $c(z) = \sum_{i=0}^{m-1} c_i z^i \in \mathbb{R}[z]$ is a scalar polynomial of degree $m-1$ ensuring strict properness of the controller transfer matrix. This choice permits useful simplifications and makes the control design (the algorithm design) more tractable and suitable for a robust approach. With this choice,

$$\begin{aligned} [p(z) \mathbf{I} - \mathbf{A}\mathbf{C}_N(z)]^{-1} &= [p(z) \mathbf{I} - c(z) \mathbf{A}]^{-1} = [p(z) \mathbf{I} - c(z) \mathbf{V}\mathbf{A}\mathbf{V}^\top]^{-1} \\ &= \mathbf{V} [p(z) \mathbf{I} - c(z) \mathbf{A}]^{-1} \mathbf{V}^\top, \end{aligned} \quad (5.14)$$

where we leveraged the diagonalizability of $\mathbf{A}$ via the eigendecomposition $\mathbf{A} = \mathbf{V}\mathbf{A}\mathbf{V}^\top$ due to its symmetry. Hence, the poles of $[p(z) \mathbf{I} - \mathbf{A}\mathbf{C}_N(z)]^{-1}$ coincide with those of the diagonal rational matrix $[p(z) \mathbf{I} - c(z) \mathbf{A}]^{-1}$. Denoting by λ_i the diagonal entries of $\mathbf{A}$, we conclude that the set of poles is the union of the zeros of the polynomials

$$p(z) - c(z) \lambda_i, \quad (5.15)$$

for $i = 1, \dots, n$. At this point, depending on Assumptions 1.1, 1.2, or 1.3, one can derive different controllers $c(z)$ based on the prior knowledge about uncertainty. We consider the least restrictive Assumption 1.3, which allows us to derive the most general and robust controller.

To achieve this, we employ a robust-control technique to obtain stabilizing controllers for uncertain systems. We rely on the results of [41], which use a LMI approach. Fixing the controller denominator in (5.11) to $p(z)$ by the IMP, we parametrize the controller in the state-space form:

$$\begin{aligned} \mathbf{F} &= \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \ddots & 1 \\ -p_0 & \dots & \dots & \dots & -p_{m-1} \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \\ \mathbf{K} &= [c_0 \dots \dots \dots c_{m-1}]. \end{aligned} \quad (5.16)$$

In this sense, the stabilization problem in (5.15) is equivalent to finding the output matrix $\mathbf{K}$ that stabilizes the polynomial (5.15), i.e., to stabilizing $\mathbf{F}_c(\lambda) := \mathbf{F} + \lambda \mathbf{C}\mathbf{K}$ with $\lambda \in [\lambda_{\min}, \lambda_{\max}]$. This problem is addressed in [41]. In particular, if $\mathbf{F}_c(\lambda)$ is a convex polytope in the unknown parameter λ , the feasibility conditions of the following LMIs (evaluated at the vertices of the polytope) are sufficient to guarantee stability of the closed-loop matrix $\mathbf{F}_c(\lambda)$ over the unknown domain $[\lambda_{\min}, \lambda_{\max}]$ and provide a stabilizing choice for $\mathbf{K}$. Since $\mathbf{F}_c(\lambda)$ forms a convex polytope, applying Theorem 15, specified from [41, Theorem 3], in the Appendix A yields the following result, which provides $\mathbf{K}$:

Corollary 1 (Stabilizing controller design). *The eigenvalues of $\mathbf{F}_c(\lambda)$ lie strictly inside the unit circle for any $\lambda \in [\lambda_{\min}, \lambda_{\max}]$ if there exist symmetric matrices $\mathbf{P}_{\min}, \mathbf{P}_{\max} > 0$, a matrix $\mathbf{Q} \in \mathbb{R}^{m \times m}$, and a row vector $\mathbf{R} \in \mathbb{R}^{1 \times m}$ such that*

$$\begin{aligned} \begin{bmatrix} \mathbf{P}_{\min} & \mathbf{F}\mathbf{Q} + \lambda_{\min}\mathbf{C}\mathbf{R} \\ \star & \mathbf{Q} + \mathbf{Q}^\top - \mathbf{P}_{\min} \end{bmatrix} &> 0, \\ \begin{bmatrix} \mathbf{P}_{\max} & \mathbf{F}\mathbf{Q} + \lambda_{\max}\mathbf{C}\mathbf{R} \\ \star & \mathbf{Q} + \mathbf{Q}^\top - \mathbf{P}_{\max} \end{bmatrix} &> 0. \end{aligned}$$

A stabilizing controller is then $\mathbf{K} = \mathbf{R}\mathbf{Q}^{-1}$.

Finally, we remark that the computational complexity of solving these LMIs depends also on the degree of $p(z)$. However, the LMIs are solved only once, offline before deployment of the algorithm; thus, their complexity does not affect real-time operation.

5.1.3 THE ONLINE ALGORITHM

The actual implementation of the scheme in Figure 2.2 involves translating the relationships between the signals $\mathbf{e}_k$ and $\mathbf{x}_k$ from the $\mathcal{Z}$ -domain to the time domain. The relation in the $\mathcal{Z}$ -domain is

$$\hat{\mathbf{x}}(z) = \left(\frac{c(z)}{p(z)} \mathbf{I}_n \right) \hat{\mathbf{e}}(z). \quad (5.17)$$

The algorithm recursion is then expressed by directly deriving a realization of the controller transfer matrix, following standard control-theoretic practice. A state-space realization is

$$\text{Online Optimization algorithm} \quad \begin{cases} \mathbf{z}_{k+1} = (\mathbf{I}_n \otimes \mathbf{F}) \mathbf{z}_k + (\mathbf{I}_n \otimes \mathbf{C}) \mathbf{e}_k, \\ \mathbf{x}_k = (\mathbf{I}_n \otimes \mathbf{K}) \mathbf{z}_k, \end{cases} \quad (5.18a)$$

$$(5.18b)$$

$$\text{Online optimization oracle } \mathbf{e}_k = \nabla_{\mathbf{x}} f_k(\mathbf{x}_k), \quad (5.18c)$$

where the matrices $\mathbf{F}$, $\mathbf{C}$, and $\mathbf{K}$ are defined in (5.16), and $\mathbf{z}_k \in \mathbb{R}^{nm}$ is the controller state vector. Hence, equations (5.18) describe the online algorithm in the scheme of Figure 5.1.

Remark 6. *We highlight that the structure of the MIMO controller in (5.18) follows from choosing $\mathbf{C}_N(z)$ as in (5.13), i.e., a scalar controller times the identity. This MIMO extension does not require any knowledge of the eigenvectors of $\mathbf{A}$. However, the Kronecker-product structure in (5.18) can be refined if information on eigenvector/eigenvalue pairs is available. For example, if the eigenvectors can be partitioned into two known subspaces with different eigenvalue bounds than $[\lambda_{\min}, \lambda_{\max}]$, one can design two different controllers acting on the respective subspaces, improving the performance of the optimization algorithm.*

5.2 CONVERGENCE

In this section, we prove convergence of (5.18) for the optimization problem (5.1). Convergence is established by the following proposition, which employs classical arguments based on $\mathcal{Z}$ -transforms.

Proposition 1 (Convergence of algorithm 5.18). *Consider the optimization problem (5.1), with $\{\mathbf{c}_k\}_{k \in \mathbb{N}}$ modeled as in (4.12a). Choose the controller (5.13). Let $c(z)$ be such that $\mathbf{F}_c(\lambda)$, defined in (5.16), is Schur stable for all $\lambda \in [\lambda_{\min}, \lambda_{\max}]$. Let $\{\mathbf{x}_k\}_{k \in \mathbb{N}}$ be the output of the online algorithm (5.18). Then*

$$\lim_{k \rightarrow \infty} \|\mathbf{x}_k - \mathbf{x}_k^*\| = 0.$$

Proof. Choosing a controller that stabilizes $\mathbf{F}_c(\lambda)$ for all $\lambda \in [\lambda_{\min}, \lambda_{\max}]$ implies that the poles of $\hat{\mathbf{e}}(z)$ (cf. (5.10)) are Schur-stable. Therefore, $\mathbf{e}_k$ is asymptotically driven to zero as in (5.5), which establishes the thesis. $\square$

5.3 DISCUSSION

In this chapter, we presented the fundamental steps and rationale underlying the rest of Part I. We completed the first case, the simplest one in Table 4.1. However, the core steps that drive our subsequent developments are all embedded here. These are:

1. decoupled analysis of the original problem, moving from a robust-control MIMO problem to a SISO robust problem (e.g., exploiting eigenvector/eigenvalue decompositions in Chapters 5, 9, and Section 6.4 or the Schur decomposition in Section 7);
2. exploitation of the IMP to track the model driving the time-varying part of the original optimization problem (in the deterministic setting in Chapters 5, 7, 8, and 6, and in the stochastic one in Chapter 9);
3. leveraging robust-control techniques to deal with incomplete knowledge of the optimization problem structure (LMI-based approach generally, and Kalman filtering, optimal control and robust $\mathcal{H}_\infty$ control in Chapter 9).

In this chapter, we do not present numerical results nor a full experimental analysis; these can be found in [13]. However, in the remaining part of the thesis we address most of the cases left open in Table 4.1, providing an exhaustive numerical evaluation.

6

Deterministic models for constrained online optimization: projected methods for inequality constraints

In the case of constrained optimization problems with inequality constraints, one can adopt a different strategy to derive an algorithm: the *projected gradient method*, an alternative to the primal–dual approach that we will introduce in Chapter 8.

In this chapter, we propose a new optimization algorithm applicable to problems of the form

$$\begin{aligned} \mathbf{x}_k^* = \arg \min_{\mathbf{x} \in \mathbb{R}^n} \quad & \underbrace{\frac{1}{2} (\mathbf{x} - \mathbf{c}_k)^\top \mathbf{A} (\mathbf{x} - \mathbf{c}_k)}_{f_k(\mathbf{x})}, \\ \text{subject to} \quad & \mathbf{g}_k(\mathbf{x}) \leq 0, \end{aligned} \tag{6.1}$$

where $\mathbf{c}_k$ follows the evolution described in Assumption 1.5 and $\mathbf{g}_k(\mathbf{x})$ is defined as in (4.1).

Defining the feasible set as

$$\mathcal{G}_k := \{\mathbf{x} \in \mathbb{R}^n : \mathbf{g}_k(\mathbf{x}) \leq 0\}, \tag{6.2}$$

and the corresponding projection operator as $\text{proj}_{\mathcal{G}_k} : \mathbb{R}^n \rightarrow \mathcal{G}_k$, the general structure of the proposed algorithm is depicted in Figure 6.1. This scheme can be viewed as an instance of

the general framework in Figure 2.2, where the optimization algorithm is described by the transfer matrices $C_{xp}(z)$, $C_{yp}(z)$, $C_{xq}(z)$, and $C_{yq}(z)$. Equivalently, the online optimization algorithm acts as a linear filter that maps the input signals x_k and y_k to the outputs p_k and q_k , mirroring the viewpoint adopted in Chapter 5. Relative to that chapter, in addition to the gradient-evaluation block $\nabla f_k(\cdot)$, the present scheme explicitly includes the nonlinear projection operator $\text{proj}_{\mathcal{G}_k}(\cdot)$.

The rationale is as follows. The signals p_k and q_k constitute two queries (see Figure 2.2) to an oracle, which returns (i) x_k by projecting the pre-projection optimal estimate p_k onto the feasible set $\mathcal{G}_k$, and (ii) the gradient evaluation y_k at the query point q_k . In this sense, the oracle operates in two distinct steps, projection and gradient evaluation, thereby providing a general mechanism to extend the unconstrained algorithm to the constrained setting.

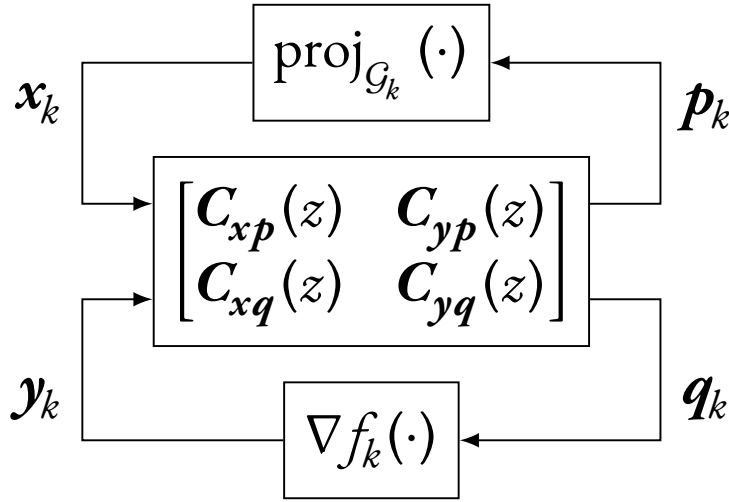


Figure 6.1: General optimization scheme derived in this section, represented in the style of Figure 2.2.

6.1 ALGORITHM

In this section, we propose an algorithm to address problems of the form (6.1). Our design follows the standard rationale of extending the results from the previous chapter to the setting in (6.1). Hence, we consider the case in which x_k^* lies in the interior of $\mathcal{G}$, that is,

$$x_k^* \in \{x \in \mathbb{R}^n : g_k(x) < 0\}.$$

Under this condition, and assuming that the algorithm's estimate remains sufficiently close to the true optimum $\mathbf{x}_k^*$, the projection in Figure 6.1 acts effectively as the identity operator. Hence, the projection step can be removed from the scheme, as illustrated in Figure 6.2. This is equivalent to closing the feedback loop around the gradient-evaluation block only and designing the controller $\mathbf{C}(z) \in \mathbb{R}^{n \times n}(z)$ as in the previous chapter. In the resulting scheme

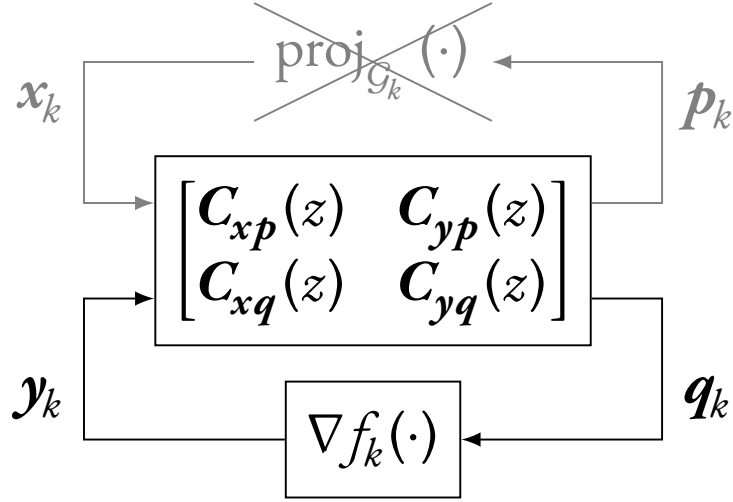


Figure 6.2: Scheme of Figure 6.1 with inactive projection (identity projection operator).

of Figure 6.3, our objective is to study and derive a control design that operates effectively in this regime.

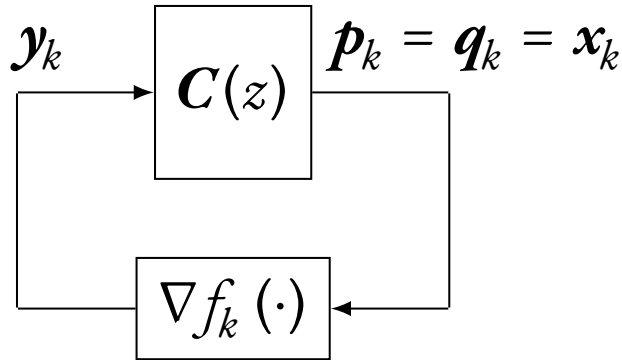


Figure 6.3: Diagram of Figure 6.2 with $\mathbf{C}_{yq}(z) = \mathbf{C}(z)$.

INTERNAL MODEL. Such an algorithm was already derived in Chapter 5. More specifically, as in Chapter 5, we set

$$\mathbf{C}(z) = \frac{\mathbf{C}_N(z)}{p(z)}, \quad (6.3)$$

with

$$\mathbf{C}_N(z) = c(z) \mathbf{I}, \quad (6.4)$$

where $c(z) = \sum_{i=0}^{m-1} c_i z^i \in \mathbb{R}[z]$ is a scalar polynomial of degree $m - 1$ ensuring strict properness of the controller transfer matrix. In this way, $\mathbf{C}(z)$ is chosen to satisfy the IMP. Using the state-space realization given by the triplet $(\mathbf{F}, \mathbf{C}, \mathbf{K})$, the remaining design freedom, captured by the polynomial $c(z)$ (equivalently, by the gain $\mathbf{K}$ in state space), is selected following exactly the same steps as in Subsection 5.1.2.

IMPOSING A FEASIBILITY SOLUTION. With the previous derived controller, we already know that the scheme in Figure 6.3 guarantees algorithmic convergence in the unconstrained case, i.e., when $\text{proj}_{\mathcal{G}_k}(\mathbf{x}_k^*) = \mathbf{x}_k^*$.

However, when $\text{proj}_{\mathcal{G}_k}(\mathbf{x}_k^*) \neq \mathbf{x}_k^*$, the algorithm in Figure 6.3 does not ensure feasibility. To enforce feasibility, we insert a projection step in the feedback loop, namely:

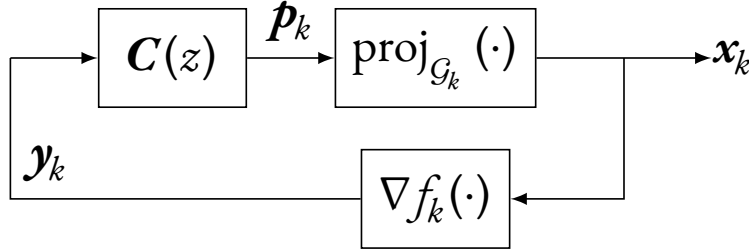


Figure 6.4: Diagram of Figure 6.3 modified to ensure the feasibility of the solution.

Although the algorithm in Figure 6.4 maintains feasibility, empirical evidence shows poor performance due to the *windup* effect [70]. To mitigate this issue, we introduce a static AW mechanism $\mathbf{T} \in \mathbb{R}^{n \times n}$, known as back-calculation [5], and propose the algorithm Projected Internal Model Anti-Windup (PIMAW) represented by the scheme in Figure 6.5.

Example 1 (Projected Gradient Descent). *With the choice*

$$\mathbf{C}(z) = \frac{-\alpha}{z - 1} \mathbf{I}, \quad (6.5)$$

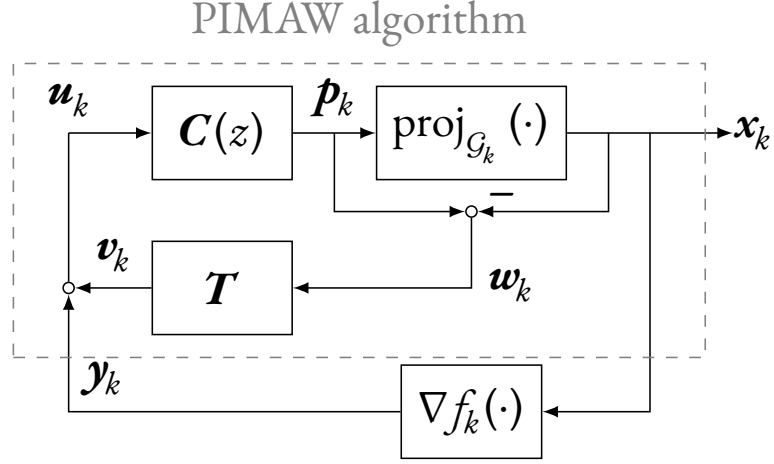


Figure 6.5: PIMAW architecture with AW mechanism.

in the scheme of Figure 6.4, it is immediate to obtain the following algorithm:

$$\begin{cases} \mathbf{p}_{k+1} &= \mathbf{p}_k - \alpha \mathbf{y}_k, \\ \mathbf{y}_k &= \nabla f_k(\mathbf{x}_k), \\ \mathbf{x}_k &= \text{proj}_{\mathcal{G}_k}(\mathbf{p}_k). \end{cases}$$

By adding an AW mechanism as in Figure 6.5, we obtain

$$\begin{cases} \mathbf{p}_{k+1} &= \mathbf{p}_k - \alpha (\nabla f_k(\mathbf{x}_k) + \mathbf{T} \mathbf{p}_k - \mathbf{T} \mathbf{x}_k), \\ \mathbf{x}_k &= \text{proj}_{\mathcal{G}_k}(\mathbf{p}_k). \end{cases}$$

Setting $\mathbf{T} = 1/\alpha$, we recover exactly the Online Projected Gradient Descent (OPGD) algorithm:

$$\mathbf{x}_{k+1} = \text{proj}_{\mathcal{G}_k}(\mathbf{x}_k - \alpha \nabla f_k(\mathbf{x}_k)), \quad (6.6)$$

where $\alpha > 0$ is a chosen step size.

6.1.1 THE ONLINE ALGORITHM

A state-space representation of the proposed algorithm is:

$$\text{Online optimization algorithm} \begin{cases} \mathbf{z}_{k+1} = (\mathbf{I}_n \otimes \mathbf{F}) \mathbf{z}_k + (\mathbf{I}_n \otimes \mathbf{C}) \mathbf{u}_k, & (6.7a) \\ \mathbf{p}_k = (\mathbf{I}_n \otimes \mathbf{K}) \mathbf{z}_k, & (6.7b) \\ \mathbf{u}_k = \mathbf{y}_k + \underbrace{\mathbf{T}(\mathbf{x}_k - \mathbf{p}_k)}_{\text{AW}}, & (6.7c) \end{cases}$$

$$\text{Optimization oracle} \begin{cases} \mathbf{x}_k = \text{proj}_{\mathcal{G}_k}(\mathbf{p}_k), & (6.7d) \\ \mathbf{y}_k = \nabla f_k(\mathbf{x}_k). & (6.7e) \end{cases}$$

6.2 NUMERICAL RESULTS

In this subsection, we illustrate the results of a numerical experiment designed to assess the performance of the proposed algorithm and to highlight the effects induced by the saturation element in the control loop (6.7). Specifically, we randomly generate an orthogonal matrix $\mathbf{V}$ with dimension $n = 2$ and select random eigenvalues in the range $[10, 50]$. We then construct the matrix $\mathbf{A} = \mathbf{V}\mathbf{A}\mathbf{V}^\top$. The signal $\mathbf{c}_k$ is generated as a triangular wave with the same shape as in Figure 6.6. Finally, we constrain only the second coordinate (just for

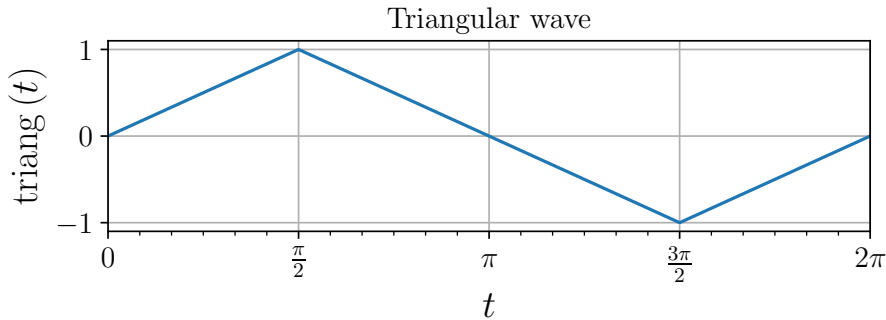


Figure 6.6: The graph of $\text{triang}(t) := 4 \left| \frac{t}{2\pi} - \lfloor \frac{t}{2\pi} + 3/4 \rfloor + 1/4 \right| - 1, t \in \mathbb{R}$.

simplicity), enforcing $\mathcal{G}_k = \{\mathbf{x} : x_1 \in \mathbb{R}, x_2 \geq l\}$.

We apply OPGD and the algorithm in (6.7), both with and without the AW mechanism, and compare their performance in time and in state space (hence the choice $n = 2$).

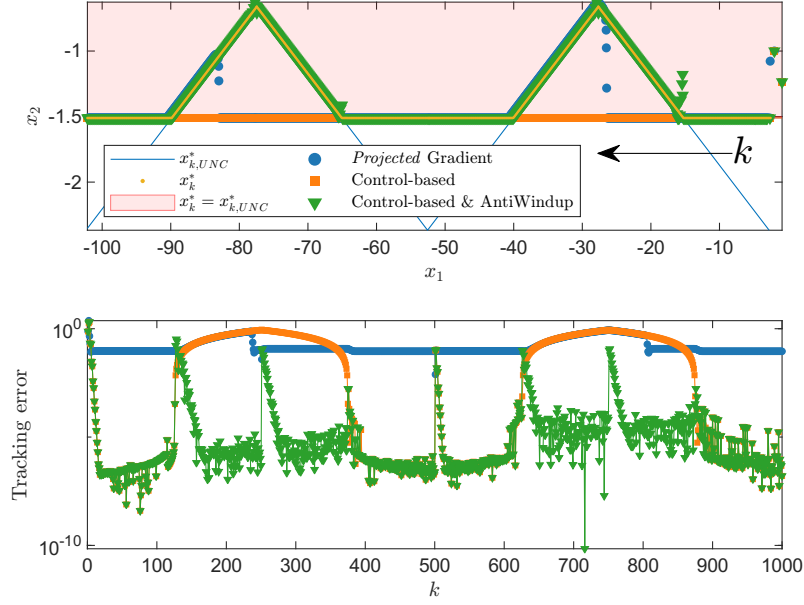


Figure 6.7: Comparison among OPGD (6.6), the control-based projected algorithm without anti-windup (Control-based) of Figure 6.4, and the same algorithm with anti-windup compensation (Control-based & AntiWindup) PIMAW of Figure 6.5. Top: trajectories in the (x_1, x_2) plane showing the iterates and the corresponding unconstrained solution. Bottom: semilogarithmic plot of the tracking error versus k .

From the bottom panel of Figure 6.7, the proposed algorithm with AW compensation achieves the lowest tracking error, outperforming both OPGD and the same control-based algorithm without AW. The “space” plot (top panel) further clarifies the contribution of the AW mechanism: it counteracts the degradation caused by the activation of the inequality constraint, producing trajectories that remain closer to the constrained optimizer. Note also that the benefit of AW compensation is most pronounced during intervals in which the constraint is active, where the improvement is substantial.

6.3 DISCUSSION

We introduced a heuristic algorithm that naturally extends the ideas of Chapter 5 to the constrained problem (6.1) by leveraging projected gradient methods. Although we do not provide an explicit convergence proof in discrete time, the algorithm, motivated by the previous section, performs well empirically (see Section 6.2). While convergence guarantees for this discrete-time, nonlinear setting remain elusive, we will show that favorable performance properties can be established for the continuous-time counterpart of (6.1). The next section develops this continuous-time extension.

6.4 CONTINUOUS TIME

In this section, we derive a continuous-time version of algorithm (6.7), motivated by the richer set of techniques available in continuous time for analyzing saturated systems and AW schemes. To prove convergence of the previous method, we reformulate both the problem and the algorithmic derivation in continuous time, which streamlines the analysis. Accordingly, we introduce the continuous-time extension of Problem (6.1), namely

$$\begin{aligned} \mathbf{x}^*(t) = \arg \min_{\mathbf{x} \in \mathbb{R}^n} \quad & \frac{1}{2} (\mathbf{x} - \mathbf{c}(t))^\top \mathbf{A} (\mathbf{x} - \mathbf{c}(t)), \\ \text{subject to} \quad & \mathbf{g}_t(\mathbf{x}) \leq 0, \end{aligned} \quad (6.8)$$

where the objective function is the continuous-time counterpart of (4.3), and $\mathbf{g}_t : \mathbb{R}^n \rightarrow \mathbb{R}^p$, with $t \in \mathbb{R}_{\geq 0}$, denotes a possibly time-varying constraint. In particular, $\mathbf{A} = \mathbf{A}^\top > 0$ satisfies Assumption 1.3, and $\mathbf{c}(t)$ is generated by the continuous-time counterpart of model (4.12a), that is,

$$\begin{cases} \dot{\mathbf{z}}(t) &= (\mathbf{I}_n \otimes \mathbf{F}) \mathbf{z}(t), \\ \mathbf{c}(t) &= (\mathbf{I}_n \otimes \mathbf{H}) \mathbf{z}(t), \end{cases} \quad (6.9)$$

with

$$\mathbf{z}(t) = \begin{bmatrix} \mathbf{z}^{(1)}(t) \\ \vdots \\ \mathbf{z}^{(n)}(t) \end{bmatrix} \in \mathbb{R}^{mn}, \quad (6.10)$$

and where each component $\mathbf{z}^{(i)}(t)$ is the state of the following Single-Input Single-Output (SISO) LTI system:

$$\begin{cases} \dot{\mathbf{z}}^{(i)}(t) &= \mathbf{F} \mathbf{z}^{(i)}(t), \\ [\mathbf{c}(t)]_i &= \mathbf{H} \mathbf{z}^{(i)}(t), \end{cases} \quad (6.11)$$

with $\mathbf{F} \in \mathbb{R}^{m \times m}$ marginally stable and known, $\mathbf{H} \in \mathbb{R}^{1 \times m}$ known, and unknown initial state $\mathbf{z}^{(i)}(0) \in \mathbb{R}^m$. In addition, for the remainder of this section we simplify the analysis by fixing $\mathbf{g}_t(\mathbf{x}) = -\mathbf{x}$, which translates into a nonnegativity constraint on the decision variable $\mathbf{x}$. We adopt this choice for simplicity, noting that the following analysis remains valid for more general constraints.

As a first step, we present the standard online optimization algorithm* used to solve this

*Strictly speaking, “flow” would be more appropriate in continuous time; however, we slightly abuse terminology and keep using “algorithm” for consistency.

class of problems. The algorithm is the Continuous-Time Online Projected Gradient Descent (CT-OPGD), the natural continuous-time extension of the discrete-time OPGD introduced in (6.6); see also [55]:

$$\begin{aligned}\dot{\mathbf{q}}(t) &= -\mathbf{q}(t) + \text{proj}_{\geq 0}(\mathbf{p}(t)), \\ \mathbf{p}(t) &= \mathbf{q}(t) - \alpha \nabla f_t(\mathbf{q}(t)),\end{aligned}\tag{6.12}$$

where $\alpha > 0$ is the stepsize, and $\text{proj}_{\geq 0}(\cdot)$ denotes the projection onto the nonnegative orthant.

In the context of constrained optimization, demanding exact convergence of the algorithmic state to the true (time-varying) optimum, as in (5.4), is difficult to guarantee: activation and deactivation of inequality constraints generally induce unavoidable transients, e.g. Figure 6.7. For this reason, we adopt a different, control-inspired performance metric closely related to the satisfaction of the KKT conditions introduced in Theorem 1.

6.4.1 ALGORITHM

The proposed algorithm follows the general control scheme in Figure 6.1, here reported for completeness and adapted to continuous time in Figure 6.8. Moreover, similarly to what was done before, we pursue a two-step synthesis procedure aligned with the widely adopted unified framework of [70], based on the paradigm: *design a linear controller while ignoring input nonlinearities, then add Anti-Windup/Bumpless-Transfer (AWBT) compensation to mitigate the adverse effects of those nonlinearities on closed-loop performance*.

To this end, by introducing the rational transfer matrix $\mathbf{C}(s) \in \mathbb{R}^{n \times n}(s)$ and the static AW mechanism $\mathbf{T} \in \mathbb{R}^{n \times n}$, we propose the following Continuous-Time Projected Internal Model Anti-Windup (CT-PIMAW) scheme, presented in Figure 6.9, which is the continuous-time counterpart of the scheme shown in Figure 6.5.

This reproduces the discrete-time structure in Figure 6.5: a standard AW system with an LTI controller $\mathbf{C}(s)$ and a static AW element $\mathbf{T}$.

DECOUPLING ANALYSIS

Similarly to what is done in the previous chapters and sections, we restrict the structures of $\mathbf{C}(s)$ and $\mathbf{T}$ in Figure 6.9 to $\mathbf{C}(s) = \frac{c(s)}{d(s)}\mathbf{I}$ and $\mathbf{T} = \rho\mathbf{I}$, where $c(s), d(s) \in \mathbb{R}[s]$ are polynomials of degree $m - 1$ and m , respectively, and $\rho \in \mathbb{R}$. By standard block-diagram manipulations, mirroring the algebraic steps used for the unconstrained case in Section 5.1.2,

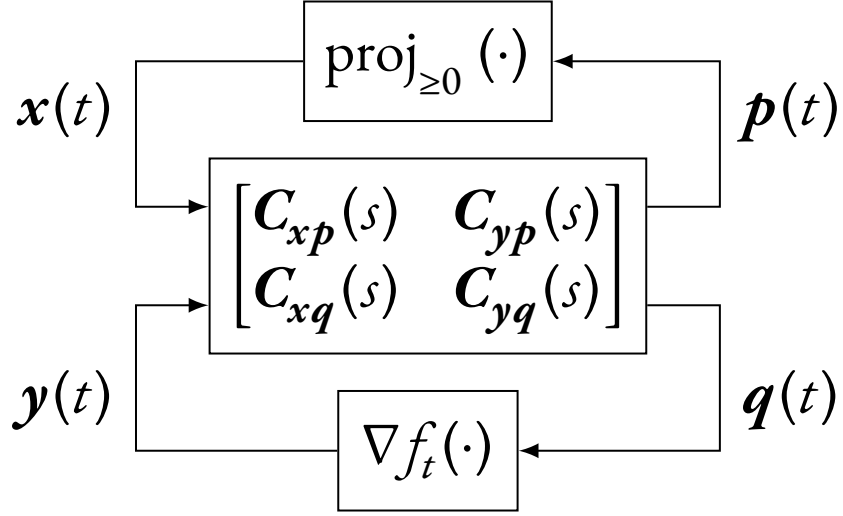


Figure 6.8: General CT-OPGD scheme rendered in the style of Figure 2.2.

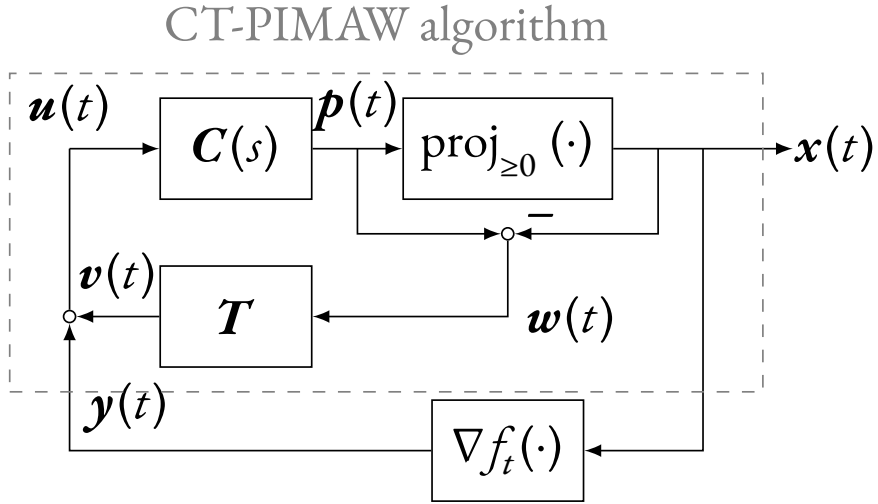


Figure 6.9: CT-PIMAW scheme proposed in this section.

the diagram in Figure 6.9 is equivalent to Figure 6.10. Noting that in Figure 6.9 we have $\mathbf{u}(t) = \mathbf{y}(t) + \mathbf{v}(t)$, we obtain

$$\mathbf{u}(t) = \mathbf{A}(\mathbf{x}(t) - \mathbf{c}(t)) + \mathbf{v}(t) = (\mathbf{A}\mathbf{x}(t) + \mathbf{v}(t)) + \underbrace{-\mathbf{A}\mathbf{c}(t)}_{:=\mathbf{b}(t)}, \quad (6.13)$$

where $\mathbf{b}(t)$ has the same evolution as $\mathbf{c}(t)$, i.e., it is generated by the same model (6.9); hence $\mathbf{b}(t)$ and $\mathbf{c}(t)$ share the same modes and poles.

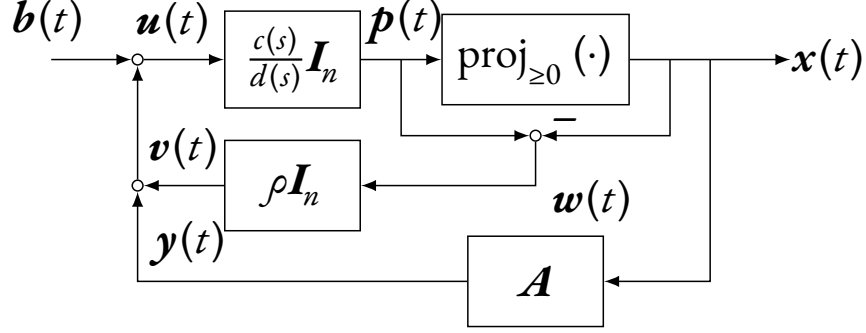


Figure 6.10: Equivalent block diagram of Figure 6.9 after the substitutions (6.13), $\mathbf{C}(s) = \frac{c(s)}{d(s)} \mathbf{I}_n$, and $\mathbf{T} = \rho \mathbf{I}$.

Exploiting the symmetry of $\mathbf{A}$, as done in the unconstrained case (Section 5.1.2), write $\mathbf{A} = \mathbf{V} \mathbf{\Lambda} \mathbf{V}^\top$, with $\mathbf{V}$ orthogonal and $\mathbf{\Lambda}$ diagonal (eigenvalues of $\mathbf{A}$ on the diagonal). The scheme in Figure 6.10 can then be rewritten as in Figure 6.11.

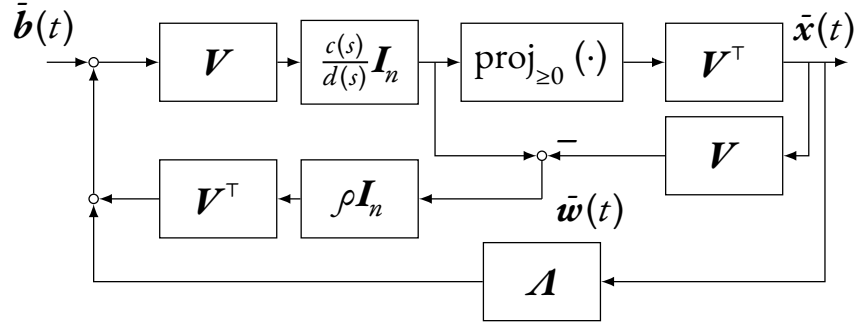


Figure 6.11: Intermediate equivalent block diagram of Figure 6.10, obtained via the eigendecomposition $\mathbf{A} = \mathbf{V} \mathbf{\Lambda} \mathbf{V}^\top$. The transformed signals $\tilde{\mathbf{x}}(t)$, $\tilde{\mathbf{b}}(t)$, and $\tilde{\mathbf{w}}(t)$ are defined in (6.15).

Using the controller/AW structure, we can move $\mathbf{V}$ after the controller block $\frac{c(s)}{d(s)} \mathbf{I}$ and $\mathbf{V}^\top$ before the AW block $\rho \mathbf{I}$. Introducing the nonlinear mapping $\phi : \mathbb{R}^n \rightarrow \mathbb{R}^n$, defined for all $\mathbf{u} \in \mathbb{R}^n$ as

$$\phi(\mathbf{u}) := \mathbf{V}^\top \text{proj}_{\geq 0}(\mathbf{V} \mathbf{u}), \quad (6.14)$$

and defining the transformed signals

$$\bar{\mathbf{x}}(t) := \mathbf{V}^\top \mathbf{x}(t), \quad (6.15a)$$

$$\bar{\mathbf{b}}(t) := \mathbf{V}^\top \mathbf{b}(t), \quad (6.15b)$$

$$\bar{\mathbf{w}}(t) := \mathbf{V}^\top \mathbf{w}(t), \quad (6.15c)$$

yields the decoupled scheme in Fig. 6.12.

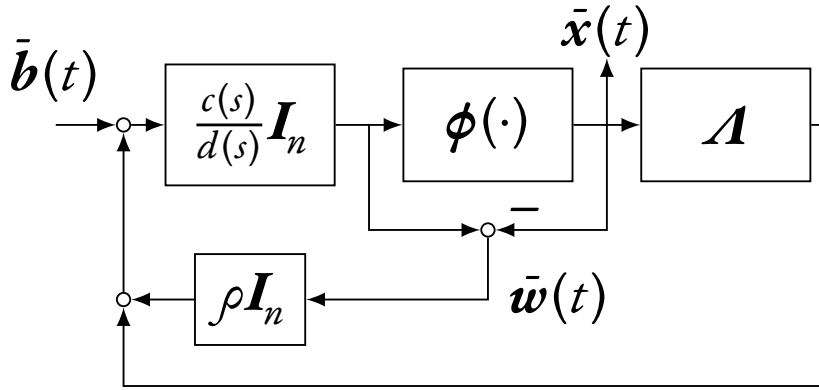


Figure 6.12: Final “decoupled” block diagram, equivalent to Figures 6.10–6.11, in the transformed coordinates $\bar{\mathbf{x}}(t)$, $\bar{\mathbf{b}}(t)$, and $\bar{\mathbf{w}}(t)$.

The scheme in Fig. 6.12 differs fundamentally from the unconstrained decomposition in Equation (5.14). In the unconstrained case, the system becomes fully decoupled in an Input–Output (IO) sense. Here, all blocks are diagonal/decoupled *except* for the nonlinearity $\phi(\cdot)$, which cannot be simplified further, cf. (6.14). Nevertheless, Figure 6.12 is particularly well-suited to the analysis developed next; importantly, the controller and the AW synthesis can be carried out in two separate, sequential steps.

6.4.2 STABILIZING CONTROLLER - UNCONSTRAINED CASE

We now recall the first step of the unified framework in [70], in which the nonlinearity is *removed* by assuming

$$\phi(\mathbf{u}) = \mathbf{u}. \quad (6.16)$$

The block $\phi(\cdot)$ is precisely the part of the algorithm that enforces feasibility of the inequality constraints. Designing the controller under (6.16) amounts to guaranteeing good perfor-

mance when constraints are *inactive*, i.e., in the unconstrained regime (equivalently, when $\mathbf{x}^*(t) > 0$). In this regime, continuous-time arguments mirror the discrete-time ones in Chapter 5: for the quadratic objective (4.3), the condition $\nabla f_t(\mathbf{x}(t)) \rightarrow 0$ as $t \rightarrow +\infty$ is sufficient to ensure $\mathbf{x}(t) \rightarrow \mathbf{x}^*(t)$.

Accordingly, we apply the same design developed for the unconstrained case to obtain a controller that guarantees the desired performance when constraints are inactive. In particular, we set $d(s) = p(s)$ (as in (5.11)) to satisfy the IMP in continuous time, and we choose $c(s)$, as in the previous sections, so that $c(s)/d(s)$ admits a state-space realization of the form (5.16). Thus, the triplet $(\mathbf{F}, \mathbf{C}, \mathbf{K})$ incorporates the internal model of (6.11) via $\mathbf{F}$ and $\mathbf{C}$, whereas $\mathbf{K}$ is obtained by robust stabilization under the unknowledge due to Assumption 1.3. Several classical synthesis methods apply; in this work we adopt the continuous-time LMI-based procedure of [95], which generalizes the discrete-time Theorem 15.

6.4.3 THE ONLINE ALGORITHM

The implementation of the online algorithm mirrors exactly that described in Subsection 6.1.1, translated to continuous time, and is encoded by the following state-space representation:

$$\text{Online Optimization algorithm} \begin{cases} \dot{\mathbf{z}}(t) = (\mathbf{I}_n \otimes \mathbf{F}) \mathbf{z}(t) + (\mathbf{I}_n \otimes \mathbf{C}) \mathbf{u}(t), & (6.17a) \\ \mathbf{p}(t) = (\mathbf{I}_n \otimes \mathbf{K}) \mathbf{z}(t), & (6.17b) \\ \mathbf{u}(t) = \mathbf{y}(t) + \underbrace{\rho (\mathbf{p}(t) - \mathbf{x}(t))}_{\text{AW}}, & (6.17c) \end{cases}$$

$$\text{Optimization oracle} \begin{cases} \mathbf{x}(t) = \text{proj}_{\geq 0}(\mathbf{p}(t)), & (6.17d) \\ \mathbf{y}(t) = \nabla f_t(\mathbf{x}(t)). & (6.17e) \end{cases}$$

where $\mathbf{z}(t) \in \mathbb{R}^{nm}$ is the controller state, $\mathbf{p}(t) \in \mathbb{R}^n$ is the output of the linear controller $\mathbf{C}(s)$, and $\rho \in \mathbb{R}$ is the AW parameter, whose selection will be discussed below.

6.4.4 THEORETICAL ANALYSIS & AW SYNTHESIS

In this subsection, we jointly prove performance guarantees for the algorithm in (6.17) under a suitable choice of the AW parameter ρ . In particular, we focus on the actual nonlinear control system of Figure 6.12 in the *constrained* case, i.e., when $\phi(\mathbf{u}) \neq \mathbf{u}$.

First, we clarify the performance objective. In the unconstrained case, objective (5.4) is achieved by proving convergence to zero of the gradient $\nabla f_t(\mathbf{x}(t))$. In the constrained setting of Problem (6.8), this is no longer sufficient: if the optimum activates the constraints, then $\mathbf{x}^*(t)$ is such that $\nabla f_t(\mathbf{x}(t))^* \neq 0$.

However, it is known from Theorem 1 that the optimal solution to Problem (6.8) satisfies the KKT conditions (4.6); that is, for any fixed $t \geq 0$, $\mathbf{x}^*(t) \in \mathbb{R}^n$ is optimal if there exists $\boldsymbol{\gamma}^*(t) \in \mathbb{R}^n$ such that

$$\text{stationarity (4.6a):} \quad \nabla f_t(\mathbf{x}^*(t)) - \boldsymbol{\gamma}^*(t) = 0, \quad (6.18a)$$

$$\text{primal feasibility (ii) (4.6c):} \quad \mathbf{x}^*(t) \geq 0, \quad (6.18b)$$

$$\text{dual feasibility (4.6d):} \quad \boldsymbol{\gamma}^*(t) \geq 0, \quad (6.18c)$$

$$\text{complementary slackness (4.6e):} \quad \boldsymbol{\gamma}^*(t) \odot \mathbf{x}^*(t) = 0, \quad (6.18d)$$

in both the *constrained* and *unconstrained* settings. In particular, the complementary slackness condition (6.18d), together with (6.18b) and (6.18c), implies that for each component i either $x_i^* = 0$ and $\gamma_i^* > 0$, or $x_i^* > 0$ and $\gamma_i^* = 0$. Simultaneous positivity of both x_i^* and γ_i^* is not admissible for any $i = 1, \dots, n$.

Remarkably, the chosen controller structure defined in Figure 6.9 implicitly inherits and imposes these properties on the algorithm output $\mathbf{x}(t)$. Specifically, for each i , the quantities $-w_i(t)$ and $x_i(t)$ are mutually exclusive:

- If the constraint is inactive, then

$$x_i(t) = \text{proj}_{\geq 0}(p_i(t)) = p_i(t) > 0,$$

$$\text{and thus } -w_i(t) = x_i(t) - p_i(t) = 0;$$

- If the constraint is active (i.e., $p_i(t) < 0$), then $x_i(t) = 0$ and

$$-w_i(t) = x_i(t) - p_i(t) > 0.$$

Hence, the algorithm in Figure 6.9 structurally enforces conditions (6.18b)–(6.18d) at all times, provided that $-\mathbf{w}(t)$ is interpreted as the vector of Lagrange multipliers $\boldsymbol{\gamma}(t)$ associated with the nonnegativity constraint in Problem (6.8).

Under this interpretation, we define the performance output as

$$\mathbf{e}(t) = \nabla f_t(\mathbf{x}(t)) - (-\mathbf{w}(t)), \quad (6.19)$$

which measures the deviation from the stationarity condition (6.18a). Consequently, minimizing the norm of $\mathbf{e}(t)$ (or its integral over time) is a natural and meaningful performance objective, analogous to requiring (5.5) in the unconstrained case.

To this end, we first categorize the nonlinearity (6.14) within the standard nonlinear framework introduced in [70]. Hence, we introduce the following definition from [58].

Definition 13. A function $\phi : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is said to *belong to* Φ_I if:

1. $\phi(0) = 0$;
2. ϕ is locally Lipschitz continuous;
3. ϕ satisfies

$$\left(\frac{\partial \phi}{\partial \mathbf{u}}(\mathbf{u}) \right)^\top \left(\mathbf{I}_n - \frac{\partial \phi}{\partial \mathbf{u}}(\mathbf{u}) \right) \geq 0 \quad (6.20)$$

for almost all $\mathbf{u} \in \mathbb{R}^n$, where $\frac{\partial \phi}{\partial \mathbf{u}}(\mathbf{u})$ denotes the Jacobian of ϕ evaluated at $\mathbf{u}^*$.

Remark 7. *Definition 13 introduces an incremental sector condition, which is useful to show that a nonlinearity satisfies a standard sector condition. Sector conditions are classical tools in nonlinear control to characterize static nonlinearities lying in a prescribed sector, and play a central role in the analysis of Lur'e-type systems, see, e.g., [68].*

The next proposition establishes that the nonlinearity defined in (6.14) satisfies Definition 13. This alignment ensures that our analysis is compatible with the hypotheses introduced in [58], enabling application of the same theoretical framework.

Proposition 2. *The function $\phi(\mathbf{u})$ defined in (6.14) satisfies Definition 13.*

Proof. Condition 1 is immediate. Condition 2 holds because $\text{proj}_{\geq 0}(\cdot)$ corresponds to the element-wise application of the ReLU function [45], which is Lipschitz continuous, and compositions of Lipschitz functions are Lipschitz [47].

For Condition 3, compute the Jacobian of ϕ at $\mathbf{u}$:

$$\frac{\partial \phi}{\partial \mathbf{u}}(\mathbf{u}) = \mathbf{V}^\top \frac{\partial \text{proj}_{\geq 0}(\mathbf{V}\mathbf{u})}{\partial \mathbf{u}} = \mathbf{V}^\top \mathbf{X}\mathbf{V},$$

*For a nonsymmetric matrix $\mathbf{A}$, the symbol $\mathbf{A} \geq 0$ means $\mathbf{A} + \mathbf{A}^\top \geq 0$.

where $\mathbf{X} = \text{diag}(\mathcal{X}_{\geq 0}(v_1), \dots, \mathcal{X}_{\geq 0}(v_n))$ and $\mathbf{v} := \mathbf{V}\mathbf{u}$, with $\mathcal{X}_{\geq 0}(\cdot)$ denoting the nonnegative indicator function. Evaluating the left-hand side of (6.20) gives

$$\begin{aligned} \mathbf{V}^\top \mathbf{X}^\top \mathbf{V} (\mathbf{I}_n - \mathbf{V}^\top \mathbf{X} \mathbf{V}) &= \mathbf{V}^\top (\mathbf{X}^\top - \mathbf{X}^\top \mathbf{X}) \mathbf{V} \\ &= 0 \geq 0, \end{aligned}$$

since $\mathbf{X}$ is diagonal with entries in $\{0, 1\}$. Thus, Condition 3 holds. $\square$

At this point, we can leverage the results of [58] to establish that the block scheme in Figure 6.12 guarantees a *quadratic performance level* γ for all ϕ belonging to Φ_I .

Definition 14. Given $\gamma > 0$, the system described in Fig. 6.12 and Equations (6.17a)–(6.17b) achieves a *quadratic performance of level* γ for all $\phi \in \Phi_I$ if there exist a scalar $\epsilon > 0$ and a quadratic Lyapunov function $V(\mathbf{z}(t)) = \mathbf{z}^\top(t) \mathbf{P} \mathbf{z}(t)$ such that the time derivative $\dot{V}(\mathbf{z}(t))$ along the dynamics induced by Figure 6.12 and (6.17a)–(6.17b) satisfies

$$\begin{aligned} \dot{V}(\mathbf{z}(t)) &< -\epsilon \mathbf{z}^\top(t) \mathbf{z}(t) - \frac{1}{\gamma} \mathbf{e}^\top(t) \mathbf{e}(t) + \gamma \lambda_{\max}^2 \mathbf{c}^\top(t) \mathbf{c}(t) \\ \forall (\mathbf{z}(t), \mathbf{c}(t)) &\in \mathbb{R}^{nm} \times \mathbb{R}^n, \end{aligned} \quad (6.21)$$

where $\mathbf{e} \in \mathbb{R}^n$ is defined in (6.19); this term quantifies the approximation accuracy of $\mathbf{x}(t)$ to $\mathbf{x}^*(t)$, and $\mathbf{c}(t)$ represents the exogenous input (6.9).*

Verifying inequality (6.21) is sufficient to ensure stability and to certify a finite $\mathcal{L}_2$ gain γ from the exogenous input $\mathbf{c}$ to the performance output $\mathbf{z}$, for all nonlinearities $\phi \in \Phi_I$ [58].

We are now in a position to provide a formal result for computing the AW gain ρ .

Proposition 3. *Given a desired performance level $\gamma > 0$, suppose there exist a matrix $\bar{\mathbf{Q}} = \bar{\mathbf{Q}}^\top > 0$, a scalar $\delta > 0$, and $\xi \neq 0$ such that the following LMIs hold for $\lambda = \lambda_{\min}$ and $\lambda_{\max}$:*

$$\begin{bmatrix} \bar{\mathbf{Q}} \mathbf{F}(\lambda)^\top + \mathbf{F}(\lambda) \bar{\mathbf{Q}} & \mathbf{H} (\xi - \delta \lambda) + \bar{\mathbf{Q}} \mathbf{K}^\top & \mathbf{H} & \bar{\mathbf{Q}} \mathbf{K}^\top \lambda \\ \star & -2\delta & 0 & \delta (1 - \lambda) \\ \star & \star & -\gamma & 1 \\ \star & \star & \star & -\gamma \end{bmatrix} < 0 \quad (6.22)$$

*Imposing (6.21) using $\mathbf{z}, \mathbf{c}$ or their decoupled counterparts $\bar{\mathbf{z}} := \mathbf{V}^\top \mathbf{z} = \mathbf{A}(\bar{\mathbf{x}} - \bar{\mathbf{c}}) + \bar{\mathbf{w}}$ and $\bar{\mathbf{c}} := \mathbf{V}^\top \mathbf{c}$ is equivalent due to the orthogonality of $\mathbf{V}$.

where $F(\lambda) := F + \lambda HK$. Then, the algorithm in Figure 6.12 with $\rho := \xi/\delta$ satisfies Definition 14.

Proof. Define $P(\lambda, \bar{Q}, \delta, \xi)$ as the block matrix in (6.22), viewed as a function of λ , $\bar{Q}$, δ , and ξ . By assumption, $P(\lambda_{\min}; \bar{Q}, \delta, \xi) < 0$ and $P(\lambda_{\max}; \bar{Q}, \delta, \xi) < 0$.

It is well known that, for fixed $\bar{Q}$, δ , and ξ , the set of λ such that $P(\lambda, \bar{Q}, \delta, \xi) < 0$ is convex [34]. Consequently, $P(\lambda; \bar{Q}, \delta, \xi) < 0$ holds for any $\lambda \in [\lambda_{\min}, \lambda_{\max}]$ and hence for all (unknown) values $\lambda = \lambda_i$, where λ_i are the n eigenvalues of A .

We now apply [58, Proposition 1]. The LMI in [58, Proposition 1], when applied to the system of Figure 6.12 with the controller given by (6.17a)–(6.17b), admits the solution $Q = I_n \otimes \bar{Q}$, since this LMI is equivalent to the n independent LMIs $P(\lambda_i; \bar{Q}, \delta, \xi) < 0$ for each $i = 1, \dots, n$.

Since, by the previous argument, these LMIs are satisfied for all $\lambda_{\min} \leq \lambda_i \leq \lambda_{\max}$, the system in Figure 6.12 with dynamics (6.17a)–(6.17b) satisfies the performance condition of Definition 14, as guaranteed by [58, Proposition 1]. $\square$

Thus, the online optimization algorithm in (6.17), with the anti-windup gain ρ computed via Proposition 3, ensures that the closed-loop algorithm of Figure 6.12 satisfies the performance criterion of Definition 14, as desired.

6.4.5 NUMERICAL EXPERIMENTS

In this section, we compare the performance of three methods applied to Problem (6.8): the standard CT-OPGD algorithm (6.12) with stepsize $\alpha = 1/\lambda_{\max}$; the continuous-time counterpart of the heuristic PIMAW proposed in (6.7), with $\rho = 1$; and the CT-PIMAW introduced in this section in Fig. 6.9, with ρ chosen through Section 6.4.4.

For both PIMAW algorithms, the robust controller is designed according to Subsection 6.4.2, leveraging the result in [95, Theorem 4 with $\xi = 0.54, \epsilon = 1$]. The parameter ρ is computed via Proposition 3 with $\gamma = 10$.

In the first simulation, shown in Figure 6.13, we consider Problem (6.8) with $n = 10$ and $A = V\Lambda V^\top$, where Λ is diagonal with eigenvalues randomly sampled from the uniform distribution over $[1, 10]$, and V is a random orthogonal matrix. The signal $c(t)$ is generated as the triangular wave of Figure 6.6, with angular frequency $\omega = 0.25$ rad/s, thus inducing several activations and deactivations of the constraints and allowing us to appreciate the anti-windup behavior.

The internal model adopted for the design is associated with the polynomial

$$p(s) = s^2, \quad (6.23)$$

and the pair $(\mathbf{F}, \mathbf{H})$ is assumed in controllable canonical form with $\mathbf{F}$ having characteristic polynomial $p(s)$, corresponding to a linear ramp evolution.

The resulting tracking error $\|\mathbf{x}(t) - \mathbf{x}^*(t)\|$ is plotted over a 45 s time window. As expected, the triangular wave introduces unavoidable transients due to constraint switching and only partial satisfaction of the IMP. Nonetheless, the improved performance of the proposed CT-PIMAW is evident, particularly during switching events generating windup effects.

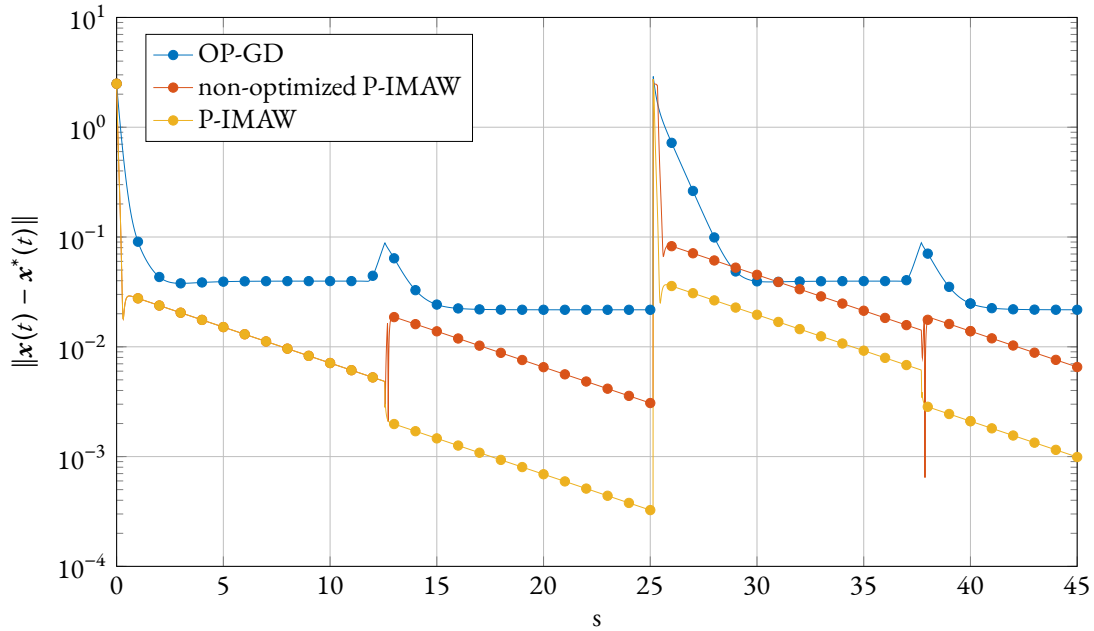


Figure 6.13: Tracking error $\|\mathbf{x}(t) - \mathbf{x}^*(t)\|$ under triangular-wave evolution of $\mathbf{c}(t)$ and internal model (6.23).

In Figure 6.14, we report a second simulation under similar conditions. The objective-function matrix $\mathbf{A}$ and the algorithm design are the same as in the previous experiment. This time, the signal $\mathbf{c}(t)$ is generated according to (6.11), with $\mathbf{H}$ and $\mathbf{F}$ as before but with

$$p(s) = (s^2 + \omega^2)s, \quad (6.24)$$

which corresponds to a sinusoidal signal with angular frequency $\omega = 0.25$ rad/s and an added constant component.

In this case, the improved tracking performance of CT-PIMAW is even more evident, especially during transients triggered by constraint activation/deactivation.

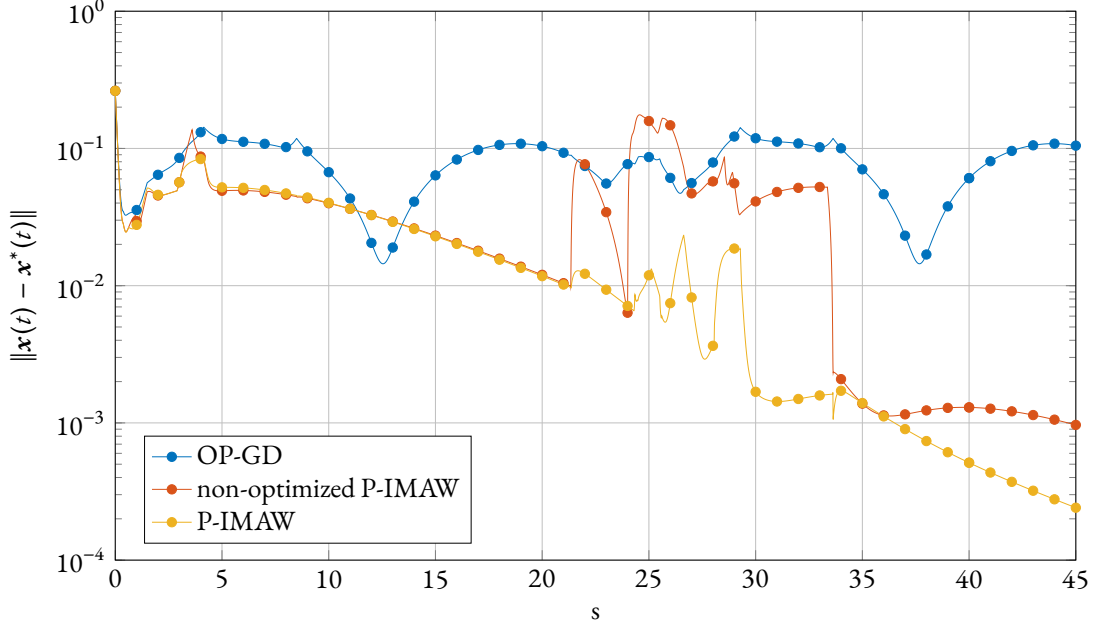


Figure 6.14: Tracking error $\|x(t) - x^*(t)\|$ with $b(t)$ following a sinusoidal-plus-constant evolution.

6.4.6 DISCUSSION

In this section, we complete the work introduced in Section 6. We provide a continuous-time version of the algorithm proposed in (6.7), together with a selection criterion for the AW gain and stability/performance guarantees. Extensions of this work include translating the performance guarantees of CT-PIMAW to its discrete-time counterpart PIMAW, integrating the same concepts into the primal–dual heuristic algorithm proposed in (8.7) and testing the proposed algorithm in less structure scenarios, for example scenarios with an objective function subjects to nonlienaar exogenous system with known linearization.

7

Deterministic models for constrained online optimization: primal–dual methods for equality constraints

We consider the third case in Table 4.1, namely online optimization with equality constraints. In particular, we study Problem (4.1) with objective function specified in (4.3) and equality constraints as in (4.2b). Specifically, we focus on the specific problem:

$$\begin{aligned} \mathbf{x}_k^* = \arg \min_{\mathbf{x} \in \mathbb{R}^n} \quad & \underbrace{\frac{1}{2} (\mathbf{x} - \mathbf{c}_k)^\top \mathbf{A} (\mathbf{x} - \mathbf{c}_k)}_{f_k(\mathbf{x})}, \\ \text{subject to} \quad & \mathbf{H}\mathbf{x} - \mathbf{h}_k = 0. \end{aligned} \tag{7.1}$$

Adopting the design philosophy of Chapter 5 for the unconstrained case, we derive an algorithmic procedure to compute the optimizer of (7.1). Unlike the previous chapter, the formulation (7.1) includes an equality-constraint term, which is assumed to satisfy Assumption 1.4 and to evolve according to (4.12b). As in Chapter 5, the objective function satisfies Assumption 1.3 and evolves according to (4.12a).

Therefore, we develop an online optimization algorithm of the form in Figure 2.2, where the algorithm remains a linear filter while the optimization oracle evaluates the Lagrangian

associated with the constrained problem, namely $\mathcal{L}_k : \mathbb{R}^n \times \mathbb{R}^l \rightarrow \mathbb{R}$,

$$\mathcal{L}_k(\mathbf{x}, \boldsymbol{\eta}) := f_k(\mathbf{x}) + \boldsymbol{\eta}^\top (\mathbf{H}\mathbf{x} - \mathbf{b}_k), \quad k \in \mathbb{N}, \quad (7.2)$$

in which $\boldsymbol{\eta} \in \mathbb{R}^l$ is the vector of Lagrange multipliers (see Definition 12). A standard approach to such constrained convex problems is the primal-dual gradient method [19], which evaluates the gradient of the Lagrangian. Its natural online extension [18, 29] is the Online Primal-Dual Gradient Descent (OPDGD)

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k - \alpha \mathbf{e}_k, \\ \boldsymbol{\eta}_{k+1} &= \boldsymbol{\eta}_k + \beta \mathbf{f}_k, \end{aligned} \quad (7.3)$$

where $k \in \mathbb{N}$ and $\alpha, \beta > 0$ are step sizes, and

$$\begin{bmatrix} \mathbf{e}_k \\ \mathbf{f}_k \end{bmatrix} := \begin{bmatrix} \nabla_{\mathbf{x}} \mathcal{L}_k(\mathbf{x}_k, \boldsymbol{\eta}_k) \\ \nabla_{\boldsymbol{\eta}} \mathcal{L}_k(\mathbf{x}_k, \boldsymbol{\eta}_k) \end{bmatrix} = \begin{bmatrix} \mathbf{A}(\mathbf{x}_k - \mathbf{c}_k) + \mathbf{H}^\top \boldsymbol{\eta}_k \\ \mathbf{H}\mathbf{x}_k - \mathbf{b}_k \end{bmatrix}, \quad (7.4)$$

are the gradients associated to the Lagrangian function.

As in Chapter 5, the iteration above does not, in general, guarantee asymptotic convergence. We therefore derive an alternative algorithm that ensures the standard performance property

$$\lim_{k \rightarrow +\infty} \|\mathbf{x}_k - \mathbf{x}_k^*\| = 0. \quad (7.5)$$

7.1 ALGORITHM

In this section, we propose a new optimization algorithm for Problem (7.1), assuming access to an oracle for the Lagrangian gradient (7.4). Moving to the primal–dual formulation, described in Section 4.1, renders (7.1) an unconstrained problem. One might therefore ask whether the analysis of Chapter 5 applies directly in the Lagrangian setting. Unfortunately, additional work is required before a functioning online algorithm can be derived. The main difficulty arises because, although the objectives in the unconstrained case and here are similar, both seek critical points at which the gradients of the quadratic functions $f_k(\mathbf{x})$ and

$\mathcal{L}_k(\mathbf{x}, \boldsymbol{\eta})$ vanish (see Equations (4.8) and (4.9)),

Chapter 5	Chapter 7
$\nabla f_k(\mathbf{x}_k) = 0$	$\nabla \mathcal{L}_k(\mathbf{x}_k, \boldsymbol{\eta}_k) = 0$

they are not equivalent.

Although both problems are Quadratic Programs (QPs) with a linear, time-varying term governed by Assumption 1.5, they differ fundamentally: in $f_k(\mathbf{x}_k)$ the quadratic form is induced by a symmetric positive definite matrix, whereas in $\mathcal{L}_k(\mathbf{x}_k, \boldsymbol{\eta}_k)$ the saddle-point structure renders the quadratic form indefinite, because stationarity is enforced with respect to both $\mathbf{x}_k$ and $\boldsymbol{\eta}_k$. This indefiniteness introduces additional technical challenges, which we address below.

We aim to design an online optimization algorithm of the form illustrated in Figure 2.2, where the oracle provides the gradients discussed above and the algorithm is represented by a transfer matrix $\mathbf{C}(z) \in \mathbb{R}^{(n+l) \times (n+l)}(z)$, i.e.,

$$\mathbf{C}(z) = \begin{bmatrix} \mathbf{C}_{11}(z) & \mathbf{C}_{12}(z) \\ \mathbf{C}_{21}(z) & \mathbf{C}_{22}(z) \end{bmatrix} \in \mathbb{R}^{(n+l) \times (n+l)}(z), \quad (7.6)$$

of larger dimension than in Chapter 5 to interface with the Lagrangian oracle then the n decision variables and the l equality constraints. The objective is to enforce the optimality condition for the constrained problem (7.1) as stated in (4.8), namely

$$\begin{bmatrix} \mathbf{e}_k \\ \mathbf{f}_k \end{bmatrix} \xrightarrow{k \rightarrow \infty} \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \quad (7.7)$$

To visualize the control structure, the complete scheme is depicted in Figure 7.1. From a control perspective, it is similar to Figure 5.1, except for the additional signals associated with the gradient of the Lagrangian with respect to the Lagrange multipliers.

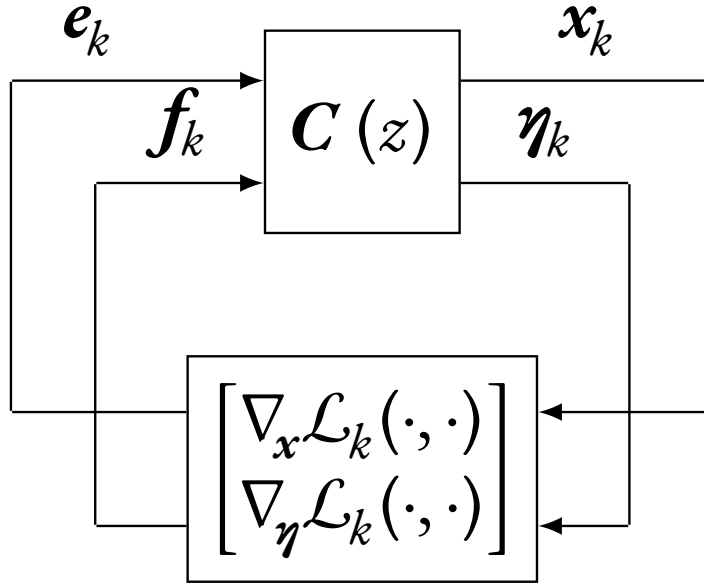


Figure 7.1: Control scheme proposed for solving Problem (7.1).

7.1.1 INTERNAL MODEL

Let us define the $\mathcal{Z}$ -transforms

$$\hat{\mathbf{x}}(z) := \mathcal{Z}[\mathbf{x}_k], \quad (7.8a)$$

$$\hat{\boldsymbol{\eta}}(z) := \mathcal{Z}[\boldsymbol{\eta}_k], \quad (7.8b)$$

$$\hat{\mathbf{e}}(z) := \mathcal{Z}[\mathbf{e}_k], \quad (7.8c)$$

$$\hat{\mathbf{f}}(z) := \mathcal{Z}[\mathbf{f}_k]. \quad (7.8d)$$

These are the transforms associated with the primal problem, i.e., (7.8a) and (7.8c), plus those for the dual variables, i.e., (7.8b) and (7.8d).

From the scheme in Figure 7.1, we obtain

$$\begin{bmatrix} \hat{\mathbf{x}}(z) \\ \hat{\boldsymbol{\eta}}(z) \end{bmatrix} = \mathbf{C}(z) \begin{bmatrix} \hat{\mathbf{e}}(z) \\ \hat{\mathbf{f}}(z) \end{bmatrix}. \quad (7.9)$$

Thanks to simple algebraic manipulation we obtain

$$\begin{bmatrix} \hat{\mathbf{e}}(z) \\ \hat{\mathbf{f}}(z) \end{bmatrix} = \begin{bmatrix} \mathbf{A}(\hat{\mathbf{x}}(z) - \hat{\mathbf{c}}(z)) + \mathbf{H}^\top \hat{\boldsymbol{\eta}}(z) \\ \mathbf{H} \hat{\mathbf{x}}(z) - \hat{\mathbf{b}}(z) \end{bmatrix}.$$

If we define $\hat{\mathbf{b}}(z) := -\mathbf{A}\hat{\mathbf{c}}(z)$, we obtain

$$\begin{aligned} \begin{bmatrix} \hat{\mathbf{e}}(z) \\ \hat{\mathbf{f}}(z) \end{bmatrix} &= \underbrace{\begin{bmatrix} \mathbf{A} & \mathbf{H}^\top \\ \mathbf{H} & 0 \end{bmatrix}}_{:=\mathbf{B}} \begin{bmatrix} \hat{\mathbf{x}}(z) \\ \hat{\mathbf{y}}(z) \end{bmatrix} + \begin{bmatrix} \hat{\mathbf{b}}(z) \\ -\hat{\mathbf{b}}(z) \end{bmatrix} \\ &= \mathbf{B}\mathbf{C}(z) \begin{bmatrix} \hat{\mathbf{e}}(z) \\ \hat{\mathbf{f}}(z) \end{bmatrix} + \begin{bmatrix} \hat{\mathbf{b}}(z) \\ -\hat{\mathbf{b}}(z) \end{bmatrix}, \end{aligned} \quad (7.10)$$

where $\mathbf{B} \in \mathbb{R}^{(n+l) \times (n+l)}$. Hence,

$$\begin{bmatrix} \hat{\mathbf{e}}(z) \\ \hat{\mathbf{f}}(z) \end{bmatrix} = [\mathbf{I} - \mathbf{B}\mathbf{C}(z)]^{-1} \begin{bmatrix} \hat{\mathbf{b}}(z) \\ -\hat{\mathbf{b}}(z) \end{bmatrix}. \quad (7.11)$$

Compared with the unconstrained counterpart (5.10), the key difference is that the unknown matrix $\mathbf{B}$, the analogue of $\mathbf{A}$ in (5.10), is a *saddle matrix* [16, 17], i.e., an indefinite matrix. This makes a direct application of previous results nontrivial. As in the previous chapter, we fix the poles of the controller $\mathbf{C}(z)$ (of the form (5.11)) to be $p(z)$ so as to satisfy the IMP, yielding

$$\begin{bmatrix} \hat{\mathbf{e}}(z) \\ \hat{\mathbf{f}}(z) \end{bmatrix} = [p(z)\mathbf{I} - \mathbf{B}\mathbf{C}_N(z)]^{-1} \begin{bmatrix} \mathbf{b}_N(z) \\ -\mathbf{b}_N(z) \end{bmatrix}. \quad (7.12)$$

Consequently, ensuring (7.7) reduces to selecting $\mathbf{C}_N(z)$ such that all poles of

$$[p(z)\mathbf{I} - \mathbf{B}\mathbf{C}_N(z)]^{-1}$$

lie strictly inside the unit disk.

7.1.2 LTI CONTROLLER

We now derive a controller that stabilizes the unstable poles arising from (7.12). As before, we impose a specific structure on the controller to (i) deal with the lack of knowledge of $\mathbf{A}$ and $\mathbf{H}$ and (ii) render the matrix $\mathbf{B}$ in (7.12) with real positive eigenvalues, thereby enabling the previous reasoning in this setting. The chosen structure is

$$\mathbf{C}_N(z) = c(z) \begin{bmatrix} \mathbf{I} & 0 \\ 0 & -\tau\mathbf{I} \end{bmatrix}, \quad (7.13)$$

where $c(z) = \sum_{i=0}^{m-1} c_i z^i \in \mathbb{R}[z]$ is a scalar polynomial of degree $m - 1$ ensuring strict properness of the controller transfer matrix, and τ is a positive parameter. We highlight the difference with (5.13), where the additional factor $-\tau \mathbf{I}$ does not appear. At this point, as in the previous chapter, we can reduce the MIMO control design to a SISO problem.

With this choice we have

$$[p(z) \mathbf{I} - \mathbf{B} \mathbf{C}_N(z)]^{-1} = [p(z) \mathbf{I} - c(z) \tilde{\mathbf{B}}(\tau)]^{-1}, \quad (7.14)$$

where

$$\tilde{\mathbf{B}}(\tau) = \mathbf{B} \begin{bmatrix} \mathbf{I} & 0 \\ 0 & -\tau \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & -\tau \mathbf{H}^\top \\ \mathbf{H} & 0 \end{bmatrix}. \quad (7.15)$$

We will see that robust-control arguments are simpler if $\tilde{\mathbf{B}}(\tau)$ has real positive eigenvalues, since the robust stability condition mirrors that of Corollary 1.

Remark 8. *One might ask whether the matrix structure in (7.13) can be leveraged to enhance performance. A straightforward approach is to introduce a new scalar parameter into the design, resulting in*

$$\mathbf{C}_N(z) = c(z) \begin{bmatrix} \alpha \mathbf{I} & 0 \\ 0 & -\beta \mathbf{I} \end{bmatrix},$$

with $\alpha, \beta > 0$. Nevertheless, this controller can be rewritten as

$$\mathbf{C}_N(z) = \underbrace{c(z) \alpha}_{:=\tilde{c}(z)} \begin{bmatrix} \mathbf{I} & 0 \\ 0 & -\frac{\beta}{\alpha} \mathbf{I} \end{bmatrix}.$$

Since $c(z)$ is a design polynomial, the extra degree of freedom introduced by α can be absorbed into it. Consequently, addressing (7.13) entails no loss of generality. In addition, we impose that the controller has the structure of a scalar transfer function times the identity matrix, since we only have partial knowledge of the matrix $\mathbf{A}$. Other controller structures could in principle be adopted, but the subsequent mode-decomposition analysis would no longer be valid if the controller had different transfer functions for each entry of the matrix.

The matrix $\tilde{\mathbf{B}}(\tau)$ in (7.15) admits real positive eigenvalues (for suitable τ shown below), but it is generally not diagonalizable, so (5.14) does not hold in general. Replacing the eigendecomposition with the Schur decomposition $\tilde{\mathbf{B}}(\tau) = \mathbf{Q} \mathbf{U} \mathbf{Q}^\top$, where $\mathbf{Q}$ is unitary and $\mathbf{U}$ is

upper triangular, yields

$$\left[p(z) \mathbf{I} - c(z) \tilde{\mathbf{B}}(\tau) \right]^{-1} = \mathbf{Q} \left[p(z) \mathbf{I} - c(z) \mathbf{U} \right]^{-1} \mathbf{Q}^\top. \quad (7.16)$$

Hence, the poles of $\left[p(z) \mathbf{I} - \mathbf{B} \mathbf{C}_N(z) \right]^{-1}$ coincide with those of $\left[p(z) \mathbf{I} - c(z) \mathbf{U} \right]^{-1}$. The (i, j) entry of this matrix has the form

$$\left(\left[p(z) \mathbf{I} - c(z) \mathbf{U} \right]^{-1} \right)_{ij} = \frac{(\text{adj} [p(z) \mathbf{I} - c(z) \mathbf{U}])_{ij}}{\det [p(z) \mathbf{I} - c(z) \mathbf{U}]}. \quad (7.17)$$

Therefore, the poles are zeros of $\det [p(z) \mathbf{I} - c(z) \mathbf{U}]$. Since $\mathbf{U}$ is triangular, these zeros coincide with the union of the zeros of the polynomials

$$p(z) - c(z) \mu_i, \quad (7.18)$$

where $\mu_i, i = 1, \dots, n$, are the eigenvalues of $\tilde{\mathbf{B}}(\tau)$. As in Subsection 5.1.2, we reuse the LMIs of Corollary 1 upon replacing $\lambda \in [\lambda_{\min}, \lambda_{\max}]$ with $\mu \in [\mu_{\min}, \mu_{\max}]$, thereby obtaining the same state-space controller triplet $(\mathbf{F}, \mathbf{C}, \mathbf{K})$ as in (5.16). The bounds $\mu_{\min}$ and $\mu_{\max}$ can be derived from the oracle quantities $\lambda_{\min}, \lambda_{\max}, \sigma_{\min}, \sigma_{\max}$ and further tightened via the design parameter τ .

This degree of freedom and the resulting bounds on the eigenvalues of $\tilde{\mathbf{B}}(\tau)$ are formalized in Corollary 2, a corollary of Lemma 1 in the Appendix A.

Corollary 2 (Bounds on the eigenvalues of $\tilde{\mathbf{B}}(\tau)$). *Given $\tilde{\mathbf{B}}(\tau)$ as in (7.15), if τ is such that*

$$0 < \tau \leq \tau_{\text{est}}^* := \frac{\lambda_{\min}^2}{4\sigma_{\max}^2}, \quad (7.19)$$

then the eigenvalues of $\tilde{\mathbf{B}}(\tau)$ are real and belong to the interval

$$\left[\tau \frac{\sigma_{\min}^2}{\lambda_{\max}}, \lambda_{\max} \right]. \quad (7.20)$$

Proof. The statement follows immediately from Lemma 1 and the bounds on the eigenvalues of $\mathbf{H} \mathbf{A}^{-1} \mathbf{H}^\top$:

$$\frac{\sigma_{\min}^2(\mathbf{H})}{\lambda_{\max}(\mathbf{A})} \leq \lambda_{\min}(\mathbf{H} \mathbf{A}^{-1} \mathbf{H}^\top) \leq \lambda_{\max}(\mathbf{H} \mathbf{A}^{-1} \mathbf{H}^\top) \leq \frac{\sigma_{\max}^2(\mathbf{H})}{\lambda_{\min}(\mathbf{A})}.$$

□

Corollary 2 provides values of τ for which $\tilde{\mathbf{B}}(\tau)$ has real positive eigenvalues and bounds them within (7.20). For the control specification, τ is not yet fixed; it is an additional design parameter that we set to τ_{est}^* . The rationale is that the smaller the ratio between the upper and lower bounds in (7.20), the easier it is to satisfy the LMIs of Theorem 15 and obtain a robust stabilizing controller with the desired optimization performance. Thus the natural choice, ensuring positivity of the eigenvalues and minimizing this ratio, is $\tau = \tau_{\text{est}}^*$. In that case, the eigenvalues of $\tilde{\mathbf{B}}(\tau)$ lie in

$$\left[\frac{\lambda_{\min}^2 \mu_{\min}^2}{4\lambda_{\max} \mu_{\max}^2}, \lambda_{\max} \right].$$

7.1.3 THE ONLINE ALGORITHM

The online algorithm mirrors Subsection 5.1.3, except that together with the dynamics in (5.17) for the primal variable we include the dual dynamics

$$\hat{\boldsymbol{\eta}}(z) = -\tau \left(\frac{c(z)}{p(z)} \mathbf{I} \right) \hat{\mathbf{f}}(z). \quad (7.21)$$

As before, the complete algorithm in state space is

$$\text{Online optimization algorithm} \begin{cases} \mathbf{z}_{k+1} = (\mathbf{I}_n \otimes \mathbf{F}) \mathbf{z}_k + (\mathbf{I}_n \otimes \mathbf{C}) \mathbf{e}_k, & (7.22a) \\ \mathbf{w}_{k+1} = (\mathbf{I}_l \otimes \mathbf{F}) \mathbf{w}_k + (\mathbf{I}_l \otimes \mathbf{C}) \mathbf{f}_k, & (7.22b) \\ \mathbf{x}_k = (\mathbf{I}_n \otimes \mathbf{K}) \mathbf{z}_k, & (7.22c) \\ \boldsymbol{\eta}_k = -\tau (\mathbf{I}_l \otimes \mathbf{K}) \mathbf{w}_k, & (7.22d) \end{cases}$$

$$\text{Optimization oracle} \begin{cases} \mathbf{e}_k = \nabla_{\mathbf{x}} \mathcal{L}_k(\mathbf{x}_k, \boldsymbol{\eta}_k), & (7.22e) \\ \mathbf{f}_k = \nabla_{\boldsymbol{\eta}} \mathcal{L}_k(\mathbf{x}_k, \boldsymbol{\eta}_k), & (7.22f) \end{cases}$$

where the matrices $\mathbf{F}$, $\mathbf{C}$ and $\mathbf{K}$ are defined in Subsection 7.1.2, and $\mathbf{z}_k \in \mathbb{R}^{nm}$, $\mathbf{w}_k \in \mathbb{R}^{lm}$ are the controller state vectors.

7.2 CONVERGENCE

As in Section 5.2, we prove convergence of (7.22). The proof, omitted here as it mirrors Proposition 1, relies on the convergence to zero of the Lagrangian gradient (7.2) thanks to the IMP.

Proposition 4 (Convergence of algorithm 7.22). *Consider the optimization Problem (7.1), with $\{\mathbf{c}_k\}_{k \in \mathbb{N}}$ and $\{\mathbf{b}_k\}_{k \in \mathbb{N}}$ modeled as in (4.12a) and (4.12b). Choose the controller (7.13) as described in Subsection 7.1.2 and let $\{\mathbf{x}_k\}_{k \in \mathbb{N}}$ be the output of the online algorithm (7.22). Then it holds*

$$\lim_{k \rightarrow \infty} \|\mathbf{x}_k - \mathbf{x}_k^*\| = 0.$$

Remark 9. *If the cost includes a time-varying matrix $\mathbf{A}$ or is non-quadratic,* convergence of the tracking error to zero is no longer guaranteed. This limitation arises because our approach reformulates Problem (7.1) as a linear control problem; in more general cases, the resulting formulation is nonlinear.*

To analyze performance under these conditions, one may resort to the small-gain theorem. However, this typically leads to conservative estimates and tends to overestimate the tracking error. An alternative would be to design the online algorithm based on a nonlinear version of the IMP.

In any case, in the next subsection we present (promising) numerical evaluations of the proposed algorithm applied to more general problem instances.

7.3 NUMERICAL RESULTS

We now compare the performance of the proposed algorithms on Problem (7.1). In addition to the baseline setting, we provide numerical examples showing that the algorithm performs well even when the matrices $\mathbf{A}$ and $\mathbf{H}$ are time-varying and in more general non-quadratic cases.

7.3.1 COST WITH FIXED $\mathbf{A}$

First, consider Problem (7.1) with $\mathbf{x} \in \mathbb{R}^n$, where $n = 10$. The matrix $\mathbf{A}$ is defined as $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$, where $\mathbf{V}$ is a randomly generated orthogonal matrix, and $\mathbf{\Lambda}$ is diagonal with elements in $[1, 10]$. The matrix $\mathbf{H} \in \mathbb{R}^{l \times n}$ is randomly generated with orthogonal rows.

*Similarly, if the constraints involve a time-varying matrix $\mathbf{H}$ or are non-linear.

For the linear terms $\{\mathbf{c}_k\}_{k \in \mathbb{N}}$ and $\{\mathbf{b}_k\}_{k \in \mathbb{N}}$, we use the following models:

1. Triangular wave (Figure 6.6): $\mathbf{c}_k = -\text{triang}(\omega k) \mathbf{A}^{-1} \mathbf{1}$, $\mathbf{b}_k = \text{triang}(\omega k) \mathbf{1}$ with $\omega = 10^{-4}\pi$;
2. Sine: $\mathbf{c}_k = -\sin(\omega k) \mathbf{A}^{-1} \mathbf{1}$, $\mathbf{b}_k = \sin(\omega k) \mathbf{1}$ with $\omega = 10^{-4}\pi$.

In Figure 7.2, we compare our control-based algorithm, designed with $p(z) = z^2 - 2 \cos(\omega)z + 1$ for the sine and $p(z) = (z - 1)^2$ for the triangular wave, to the OPDGD (7.3). From a control perspective, the classical OPDGD can be interpreted as a controller with $p(z) = (z - 1)$ as its internal model. Consequently, for the two waves, the OPDGD fails to ensure perfect tracking because it does not satisfy the IMP. By contrast, the control-based design with the correct internal model guarantees perfect tracking. For the triangular wave, the IMP is verified piecewise: tracking is guaranteed except at slope changes, where a transient is required for the controller to re-track the reference.

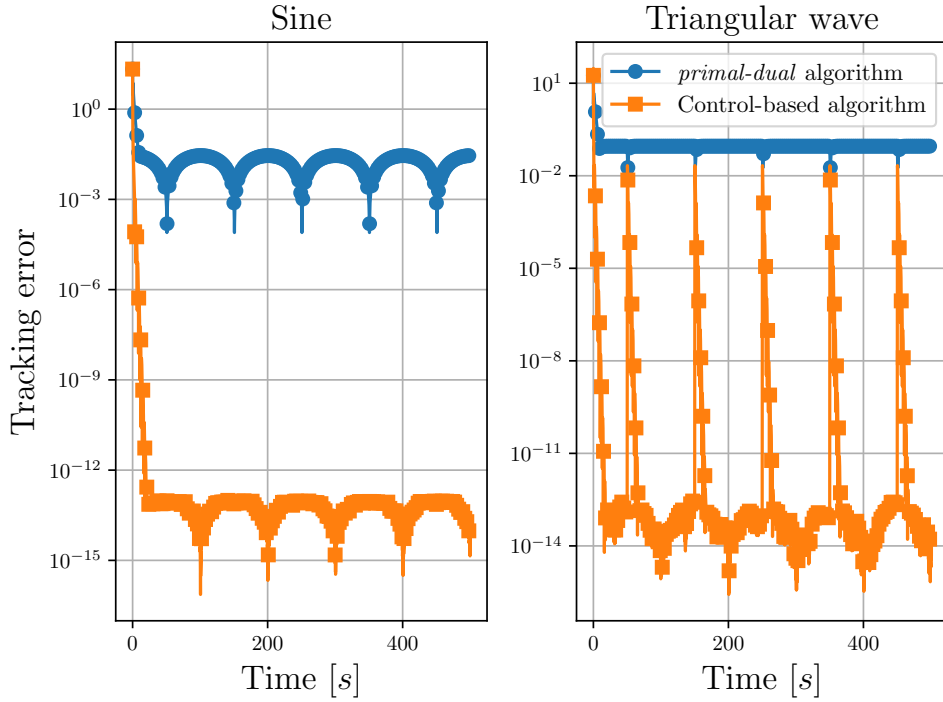


Figure 7.2: Comparison between the primal-dual and the control-based algorithm (semilogarithmic scale) for sinusoidal and triangular-wave signals.

We conclude this part with a simulation in which the model is “slightly” misspecified. Specifically, we assume that $\mathbf{c}_k$ and $\mathbf{b}_k$ follow the “Sine” evolution of Figure 7.2, and we

$\frac{ \omega - \tilde{\omega} }{\omega} =$	4%	8%	12%	16%	20%
OPDGD	3.73	–	–	–	–
control-based algorithm	0.20	0.41	0.61	0.82	1.02

Table 7.1: Asymptotic errors of the OPDGD compared to the control-based algorithm when $\mathbf{c}_k$ follows model 2 (the “Sine” model) as described in Section 7.3.1 with $p(z) = z^2 - 2 \cos(\omega)z + 1$. The control-based algorithm in (7.22) is designed with $p(z) = \tilde{p}(z) := z^2 - 2 \cos(\tilde{\omega})z + 1$ where $\tilde{\omega} = 0.96\omega, 0.92\omega, 0.88\omega, 0.84\omega, 0.8\omega$. The OPDGD yields a single error value since its design is independent of the oscillation frequency of $\mathbf{c}_k$.

measure the asymptotic error when the algorithm is designed with an incorrect oscillation frequency $\tilde{\omega}$ (while the true frequency is ω). The results, reported in Table 7.1, show that although perfect tracking is lost, the control-based algorithm outperforms the OPDGD even when the design frequency deviates by up to 20%.

7.3.2 TIME-VARYING QUADRATIC TERM AND CONSTRAINT

Following [13], we assess performance in more complex scenarios by experimental validation with $n = 10$ and $l = 2$. We consider a time-varying quadratic term $\mathbf{A}_k = \mathbf{A}_1 + \sin(\omega k) \mathbf{A}_2$, where $\mathbf{A}_1$ is symmetric positive definite with eigenvalues in $[1, 10]$, and $\mathbf{A}_2$ is a symmetric sparse matrix of ones with ten percent nonzero entries. We also consider time-varying constraints, setting $\mathbf{H}_k = \mathbf{H}_1 + \sin(\omega k) \mathbf{H}_2$, where $\mathbf{H}_1$ is a random rectangular matrix with singular values in $[1, 3]$, and $\mathbf{H}_2$ is sparse with ten percent ones. The sinusoidal term uses $\omega = 1/2$. The remaining terms are held constant, namely $\mathbf{c}_k = \bar{\mathbf{c}} \in \mathbb{R}^n$ and $\mathbf{h}_k = \bar{\mathbf{h}} \in \mathbb{R}^l$.

To implement (7.22) we require an internal model that captures the dynamics of $f_k(\mathbf{x})$. Given the periodic evolution, reasonable choices are

$$p(z) = (z - 1) \prod_{j=1}^L [z^2 - 2 \cos(j\omega)z + 1], \quad L \in \mathbb{N}. \quad (7.23)$$

Since $f_k(\mathbf{x})$ is nonlinear, the internal model is only approximate. Using (7.23), we aim to capture the minimum periodicity and fast variations.

The results are shown in Figure 7.3. Algorithm (7.22) exhibits better tracking than the OPDGD. Increasing the internal-model complexity improves asymptotic tracking at the cost of slower convergence, an expected trade-off of the IMP, as the controller requires more time to reach steady state with more complex internal models. Since our primary interest is asymptotic performance, this trade-off is acceptable.

The proposed algorithm performs well in time-varying (and in the non-quadratic case of Section 7.3.3). However, as highlighted in Remark 9, we have not formally proved convergence in these settings because the linear-control reduction no longer holds. It may still be possible to bound the error via small-gain arguments similar to [13].

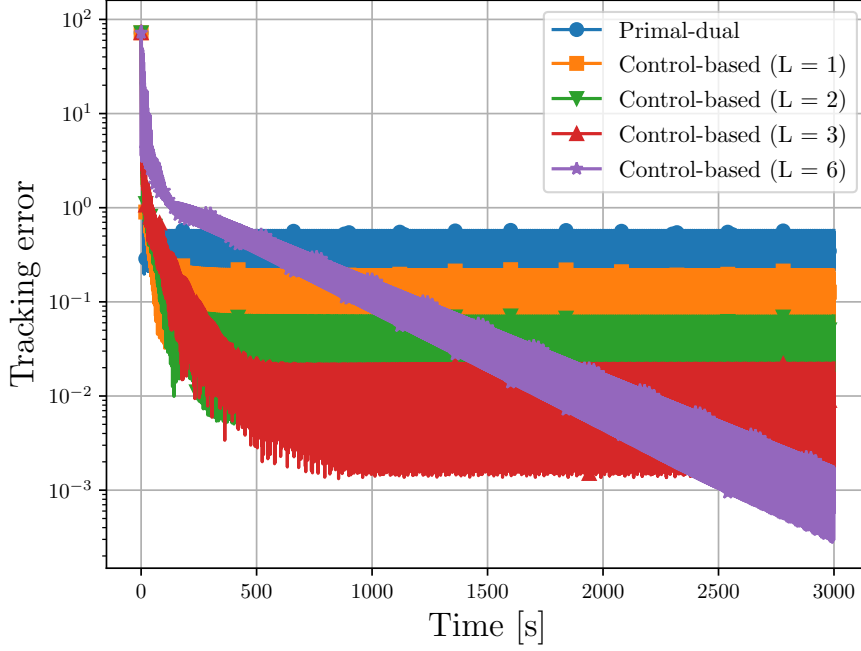


Figure 7.3: Comparison between the OPDGD and the control-based algorithm with models (7.23) ($L = 1, 2, 3, 6$) in the time-varying case.

7.3.3 NON-QUADRATIC COST

We now consider a setting with a non-quadratic objective, using the function from [106]

$$f_k(\mathbf{x}) = \frac{1}{2} \mathbf{x}^\top \mathbf{A} \mathbf{x} - \mathbf{c}^\top \mathbf{A} \mathbf{x} + \sin(\omega k) \log[1 + \exp(\mathbf{b}^\top \mathbf{x})],$$

subject to the constraint $\mathbf{H} \mathbf{x} = \mathbf{b}_k$. We take $\omega = 1/2$, $n = 10$, $\mathbf{b}, \mathbf{c} \in \mathbb{R}^n$ with $\|\mathbf{b}\| = 1$. The matrices $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{H} \in \mathbb{R}^{l \times n}$, with $l = 2$, are generated as in Section 7.3.1, and $\mathbf{b}_k$ is as in case 2. Figure 7.4 reports the tracking performance of the OPDGD alongside three control-based variants implementing (7.23). The control-based method achieves superior tracking, albeit with slower convergence.

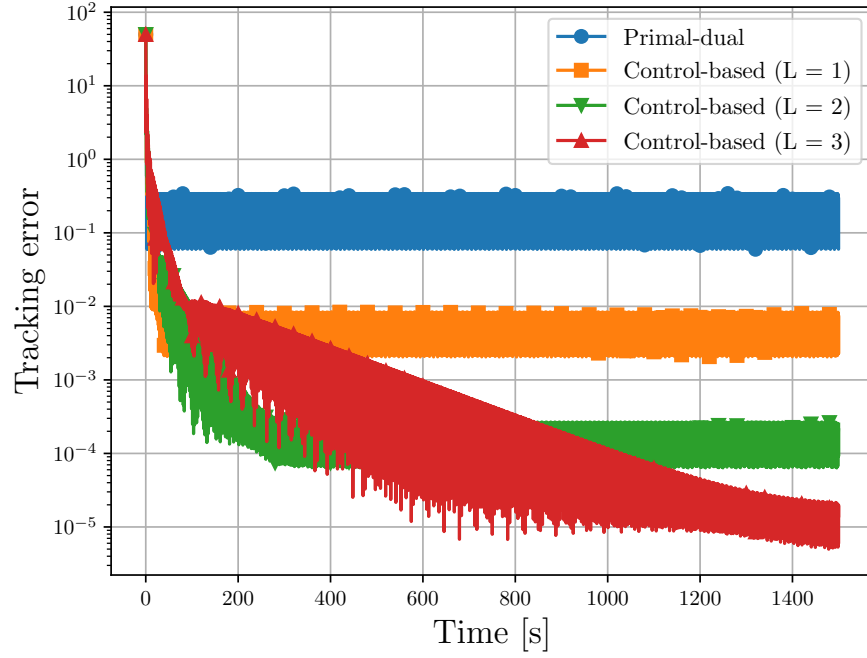


Figure 7.4: Comparison between the OPDGD and the control-based algorithm with models (7.23) ($L = 1, 2, 3$) in the non-quadratic case.

7.4 DISCUSSION

We derived an online optimization algorithm based on the scheme in Figure 2.2 that asymptotically converges to the solution of Problem (7.1) under minimal assumptions on the objective (4.3) and the equality constraints (4.2b).

As in the previous chapter, where we presented the natural model time-varying extension of the OGD algorithm, here we derived the natural extension of the classical OPDGD to the model time-varying evolution, incorporating a structured model for the dynamics of the problem data. Although new complications arise, most notably that $\tilde{\mathbf{B}}(\tau)$ is no longer positive definite, we still obtain an algorithm with exact convergence guarantees in the nominal case and strong empirical performance more broadly.

In the next section, we adopt a similar approach to develop a heuristic version of (7.22) for optimization problems (4.1) with quadratic objective (4.3), equality constraints (4.2b), and inequality constraints (4.2c).

8

Deterministic models for constrained online optimization: primal–dual methods for equality and inequality constraints

In this section we focus on the fourth case of Table 4.1, and we develop a different algorithm-design procedure to derive an online optimization algorithm for problems of the form (4.1) with objective function (4.3), and both equality (4.2b) and inequality (4.2c) constraints, with all time-varying terms described in Assumption 1.5. That is:

$$\begin{aligned} \mathbf{x}_k^* = \arg \min_{\mathbf{x} \in \mathbb{R}^n} \quad & \underbrace{\frac{1}{2} (\mathbf{x} - \mathbf{c}_k)^\top \mathbf{A} (\mathbf{x} - \mathbf{c}_k)}_{f_k(\mathbf{x})}, \\ \text{subject to} \quad & \mathbf{H}\mathbf{x} - \mathbf{h}_k = 0, \\ & \mathbf{G}\mathbf{x} - \mathbf{g}_k \leq 0. \end{aligned} \tag{8.1}$$

We derive an algorithm to solve Problem (8.1) following the approach of Section 7 and Chapter 5. Unfortunately, in this section we are not able to prove the convergence of the proposed algorithm, for reasons clarified below. Nevertheless, we propose a heuristic algorithmic procedure that mirrors what is done in Section 7.

As in the equality-only case, we adopt a primal–dual approach. In particular, the on-

line oracle in Figure 2.2 evaluates the gradient of the Lagrangian associated with the general QP (8.1), namely

$$\mathcal{L}_k(\mathbf{x}, \boldsymbol{\eta}, \boldsymbol{\gamma}) := f_k(\mathbf{x}) + \boldsymbol{\eta}^\top (\mathbf{H}\mathbf{x} - \mathbf{b}_k) + \boldsymbol{\gamma}^\top (\mathbf{G}\mathbf{x} - \mathbf{g}_k), \quad k \in \mathbb{N}, \quad (8.2)$$

where $\boldsymbol{\eta} \in \mathbb{R}^l$ is the vector of Lagrange multipliers associated with the equality constraints, and $\boldsymbol{\gamma} \in \mathbb{R}^p$ is the vector of Lagrange multipliers associated with the inequality constraints. A commonly employed approach to address this online optimization problem is the natural online extension of the primal–dual algorithm [29], which yields the Online Projected Primal–Dual Gradient Descent (OPPDGD) algorithm:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k - \alpha \nabla_{\mathbf{x}_k} \mathcal{L}_k(\mathbf{x}_k, \boldsymbol{\eta}_k, \boldsymbol{\gamma}_k), \\ \boldsymbol{\eta}_{k+1} &= \boldsymbol{\eta}_k + \beta \nabla_{\boldsymbol{\eta}} \mathcal{L}_k(\mathbf{x}_k, \boldsymbol{\eta}_k, \boldsymbol{\gamma}_k), \\ \boldsymbol{\gamma}_{k+1} &= \text{proj}_{\geq 0} \left(\boldsymbol{\gamma}_k + \gamma \nabla_{\boldsymbol{\gamma}} \mathcal{L}_k(\mathbf{x}_k, \boldsymbol{\eta}_k, \boldsymbol{\gamma}_k) \right), \end{aligned} \quad (8.3)$$

where $k \in \mathbb{N}$ and $\alpha, \beta, \gamma > 0$ are step sizes. The fundamental difference with respect to OPDGD (7.3) is the presence of the projection (saturation) nonlinearity on the Lagrange multipliers associated with the inequality constraints, which is related to satisfaction of the dual feasibility condition (4.6d) of the KKT conditions (4.6).

8.1 ALGORITHM

The nonlinearity introduced in (8.3) makes a significant difference with respect to the algorithm developed in the previous section. Because of this projection step, the online optimization algorithm in Figure 2.2 is no longer a linear filter. Therefore, the linear convergence analysis and the linear robust-design approach based on linear control analysis used previously are no longer applicable here.

Following the same reasoning as before, we provide a new online optimization algorithm that implements the control scheme in Figure 8.1. This scheme recasts the one for the equality-only case in Figure 7.1 of the previous section, but it includes a projection step (required to enforce the KKT conditions at the optimizer) and a control-inspired AW mechanism designed to improve performance in the presence of the nonlinear projection.

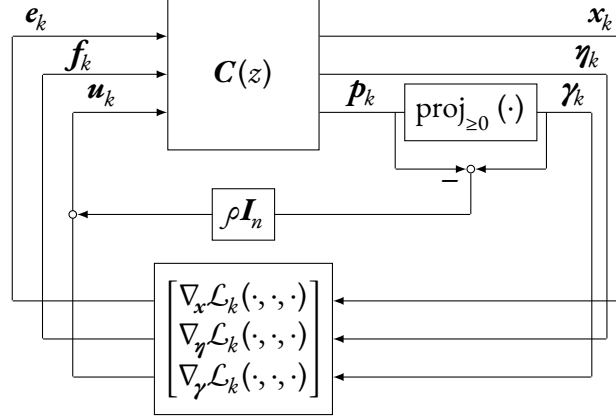


Figure 8.1: Control scheme proposed for solving Problem (8.1).

As in Subsection 7.1, the algorithm leverages the same transfer matrix block

$$\mathbf{C}(z) = \begin{bmatrix} \mathbf{C}_{11}(z) & \mathbf{C}_{12}(z) & \mathbf{C}_{13}(z) \\ \mathbf{C}_{21}(z) & \mathbf{C}_{22}(z) & \mathbf{C}_{23}(z) \\ \mathbf{C}_{31}(z) & \mathbf{C}_{32}(z) & \mathbf{C}_{33}(z) \end{bmatrix} \in \mathbb{R}^{(n+l+p) \times (n+l+p)}(z), \quad (8.4)$$

which is larger than in the previous case, as we must also account for the p inequality constraints. These constraints are temporarily treated as equalities; the inequality-constraint behavior is enforced by the saturation nonlinearity $\text{proj}_{\geq 0}(\cdot)$ (i.e., projection onto the non-negative orthant) acting on the associated Lagrange multipliers. The derivation of the controller $\mathbf{C}(z)$ proceeds exactly as in the previous section thus fixing a first structure to satisfy the IMP:

$$\mathbf{C}(z) = \frac{\mathbf{C}_N(z)}{p(z)}, \quad (8.5)$$

where $\mathbf{C}_N(z) \in \mathbb{R}^{n \times n}[z]$ is a polynomial matrix fixed by treating the inequality constraints as equality constraints, so that

$$\mathbf{C}_N(z) = c(z) \begin{bmatrix} \mathbf{I} & 0 & 0 \\ 0 & -\tau \mathbf{I} & 0 \\ 0 & 0 & -\tau \mathbf{I} \end{bmatrix}, \quad (8.6)$$

with $c(z) = \sum_{i=0}^{m-1} c_i z^i \in \mathbb{R}[z]$ a scalar polynomial of degree $m-1$ ensuring strict properness of the controller transfer matrix, and $\tau > 0$ derived as in Subsection 7.1.2.

8.1.1 THE ONLINE ALGORITHM

The online algorithm is then implemented exactly as in Subsection 7.1.3, adding the inequality-constraint variables as if they were equality constraints. The complete state-space representation of the algorithm is:

$$\text{Online optimization algorithm} \left\{ \begin{array}{ll} \mathbf{z}_{k+1} = (\mathbf{I}_n \otimes \mathbf{F}) \mathbf{z}_k + (\mathbf{I}_n \otimes \mathbf{C}) \mathbf{e}_k, & (8.7a) \\ \mathbf{w}_{k+1} = (\mathbf{I}_l \otimes \mathbf{F}) \mathbf{w}_k + (\mathbf{I}_l \otimes \mathbf{C}) \mathbf{f}_k, & (8.7b) \\ \mathbf{s}_{k+1} = (\mathbf{I}_p \otimes \mathbf{F}) \mathbf{s}_k + (\mathbf{I}_p \otimes \mathbf{C}) \mathbf{u}_k, & (8.7c) \\ \mathbf{x}_k = (\mathbf{I}_n \otimes \mathbf{K}) \mathbf{z}_k, & (8.7d) \\ \boldsymbol{\eta}_k = -\tau (\mathbf{I}_l \otimes \mathbf{K}) \mathbf{w}_k, & (8.7e) \\ \mathbf{p}_k = -\tau (\mathbf{I}_p \otimes \mathbf{K}) \mathbf{s}_k, & (8.7f) \\ \boldsymbol{\gamma}_k = \text{proj}_{\geq 0}(\mathbf{p}_k), & (8.7g) \\ \mathbf{u}_k = \boldsymbol{\gamma}_k + \underbrace{\rho(\boldsymbol{\gamma}_k - \mathbf{p}_k)}_{\text{AW}}, & (8.7h) \end{array} \right.$$

$$\text{Optimization oracle} \left\{ \begin{array}{ll} \mathbf{e}_k = \nabla_{\mathbf{x}} \mathcal{L}_k(\mathbf{x}_k, \boldsymbol{\eta}_k, \boldsymbol{\gamma}_k), & (8.7i) \\ \mathbf{f}_k = \nabla_{\boldsymbol{\eta}} \mathcal{L}_k(\mathbf{x}_k, \boldsymbol{\eta}_k, \boldsymbol{\gamma}_k), & (8.7j) \\ \boldsymbol{\gamma}_k = \nabla_{\boldsymbol{\gamma}} \mathcal{L}_k(\mathbf{x}_k, \boldsymbol{\eta}_k, \boldsymbol{\gamma}_k). & (8.7k) \end{array} \right.$$

Here, the matrices $\mathbf{F}$, $\mathbf{C}$ and $\mathbf{K}$ are the same as defined in Subsection 7.1.2, and $\mathbf{z}_k \in \mathbb{R}^{nm}$, $\mathbf{w}_k \in \mathbb{R}^{lm}$, and $\mathbf{s}_k \in \mathbb{R}^{pm}$ are the controller state vectors.

Algorithm in Equation (8.7) introduces two main differences with respect to (7.22). First, the projection step in (8.7g) yields a nonlinear behavior in the feedback loop, which represents the principal challenge. Second, we include the AW scheme in (8.7h). The AW term $\rho(\boldsymbol{\gamma}_k - \mathbf{p}_k)$ mitigates transient effects occurring when the inequality constraints in Problem (8.1) switch between active and inactive. Because of the projection operator in (8.7g), these transitions resemble the effects arising when a control system with input saturation enters/exits the saturation region. It is well known that such scenarios can lead to undesirable dynamics, AW is introduced precisely to mitigate them. By analogy, we include the same scheme in our optimization architecture and validate its performance in the next section, highlighting the emergence of these adverse effects and how AW mitigates them in the optimization context.

However, analyzing the tracking error of (8.7) for inequality constraints, with and without anti-windup, is not straightforward. In particular, the transient phenomena that occur whenever inequality constraints switch from active to inactive preclude exact tracking without additional (restrictive) assumptions. A theoretical analysis is therefore particularly challenging; nonetheless, we illustrate the effectiveness of the proposed approach through numerical simulations.

8.2 NUMERICAL RESULTS

In this section we evaluate the performance of algorithm in Equation (8.7) in the presence of only inequality constraints, since the equality-only case is treated extensively, both theoretically and numerically, in Chapter 7.

For the simulations we set $n = 10$ and $p = 2$, generate a random orthogonal matrix V and a diagonal matrix A with diagonal entries in the range $[1, 10]$ such that $A = VAV^\top$, and generate G as a random matrix with orthogonal rows. The signals c_k and g_k are sinusoidal waves with angular frequency $\omega = 10^{-4}\pi$, with corresponding internal model $p(z) = z^2 - 2\cos(\omega)z + 1$. In addition, let $x_{k,\text{UNC}}^*$ be the optimal solution of the unconstrained version of (8.1), i.e., $x_{\text{UNC},k}^* = \arg \min_{x \in \mathbb{R}^n} f_k(x)$. The simulation is set up so that, periodically, $x_k^* = x_{\text{UNC},k}^*$ and $x_k^* \neq x_{\text{UNC},k}^*$.

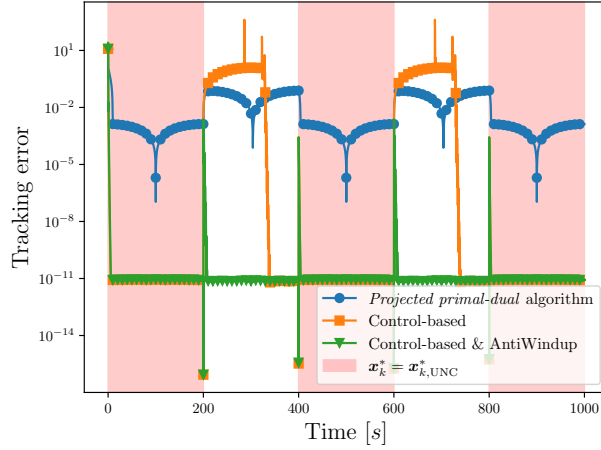


Figure 8.2: Comparison among OPPDGD (8.3), the algorithm in (8.7) without anti-windup, and the complete algorithm (8.7) with anti-windup compensation, in a semilogarithmic plot where c_k and g_k are sine waves.

We compare three algorithms:

- OPPDGD (8.3).
- Algorithm (8.7) including the projection block but without anti-windup ($\rho = 0$).
- Algorithm (8.7) with anti-windup compensation and $\rho = 1$.

Figure 8.2 depicts the tracking errors of these algorithms, with shaded intervals highlighting the times when $\mathbf{x}_k^* = \mathbf{x}_{\text{UNC},k}^*$.

First, similarly to the equality-constrained case in Figure 7.2, OPPDGD is outperformed by both proposed algorithms, in the sense that they attain much smaller tracking errors. In particular, the algorithms based on (8.7) converge towards the optimal trajectory during intervals when $\mathbf{x}_k^* = \mathbf{x}_{\text{UNC},k}^*$ and when $\mathbf{x}_k^* \neq \mathbf{x}_{\text{UNC},k}^*$, while OPPDGD only reaches a bounded neighborhood of the trajectory. This again highlights the effectiveness of a control-theoretic algorithm design.

Comparing the control-based algorithms, we notice that both exhibit transients* when the optimization problem switches between the modes $\mathbf{x}_k^* = \mathbf{x}_{\text{UNC},k}^*$ and $\mathbf{x}_k^* \neq \mathbf{x}_{\text{UNC},k}^*$. However, the algorithms differ significantly in the duration and magnitude of these transients: the AW mechanism ensures much shorter transients and smaller tracking errors.

The main issue when switching between $\mathbf{x}_k^* = \mathbf{x}_{\text{UNC},k}^*$ and $\mathbf{x}_k^* \neq \mathbf{x}_{\text{UNC},k}^*$ is that the dual variable γ_k becomes saturated. In such cases, the algorithm keeps integrating the error $\mathbf{f}_k'$ and increases the internal states associated with the integral actions. Ideally, when saturation is active we are within the bounds of the inequality constraints, and the optimization should behave as an unconstrained problem. Consequently, integrating $\mathbf{u}_k$ becomes not only redundant but also detrimental: the integral actions retain memory of past errors and adversely affect performance even outside the saturation region. The anti-windup mechanism reduces these integral actions when unnecessary, directly compensating the input $\mathbf{u}_k$.

Finally, whenever $\mathbf{x}_k^* = \mathbf{x}_{\text{UNC},k}^*$, the algorithms based on (8.7) yield $\gamma_k = 0$ owing to complementary slackness (4.6e). However, when the problem switches to $\mathbf{x}_k^* \neq \mathbf{x}_{\text{UNC},k}^*$ (and vice versa), a transient occurs. This implies that, as long as the constraints are activated/deactivated throughout the online optimization, exact convergence to the optimal solution cannot be guaranteed.

*We remark that the transients of “Control-based” in Figure 8.2 at times 400 and 800 are present but hidden by the other trajectory.

8.3 DISCUSSION

We proposed a new online optimization algorithm based on classical control principles. The algorithm in this section is a natural extension of the primal–dual optimization scheme presented in Section 7. The new scheme introduces a significant challenge compared to previous ones: the projection step acts as a nonlinearity in the closed loop. For this reason, we are not able to prove convergence (i.e., (7.5) and (5.4)) or performance guarantees for the primal–dual algorithm in (8.7). However, in the next sections we present another constrained-optimization algorithm based on different principles, projection methods rather than primal–dual. In that context, we are also able to prove favorable properties of the algorithm in continuous time case.

9

Stochastic models for unconstrained online optimization

The previous results derived in the first part of this thesis are obtained in the deterministic setting. In this section, we change perspective and derive a new design procedure that is valid in stochastic and non-deterministic settings. The earlier results leverage knowledge of a model that drives the evolution of the objective function to be optimized. Here we relax Assumption 1.5 and, in particular, generalize the results presented in the unconstrained case to a *noisy environment* in which stochastic disturbances affect the models determining the optimum of the objective function. We draw the reader’s attention to the fact that these disturbances include both *measurement* noise, additive terms corrupting the output in the feedback loop, and *process* noise, uncertainties that capture partial lack of knowledge about the model. We account for these undesirable effects and propose control-inspired algorithms that seek to improve the optimizer’s performance in a mean squared error sense.

This analysis points toward extending and applying these concepts to practical scenarios. In particular, this line of work is especially relevant in the context of modern deep learning architectures, where an online optimization problem, cast as online regression, has been identified as a mechanism for in-context learning [44]. For this reason, we conclude by presenting experiments that evaluate the performance of our algorithms in high-dimensional linear regression settings.

In this setting, we propose three classes of algorithms, each inspired by a distinct control

concept: a *Kalman inspired algorithm*, an *optimized control inspired algorithm*, and a *robust $\mathcal{H}_\infty$ control inspired algorithm*. The three algorithms are derived under different priors on the knowledge of the matrix $\mathbf{A}$, following Assumptions 1.1, 1.2, and 1.3. Each algorithm is suitable for different applications and purposes. In particular, the Kalman inspired algorithm is typically the easiest to implement and offers a favorable compromise between implementation simplicity and performance; the optimized control inspired algorithm is generally the most performance oriented solution and is often the preferred choice. However, these two algorithms can be sensitive to uncertainty in the eigenvalues of the matrix $\mathbf{A}$. In such cases, a more robust alternative is the robust $\mathcal{H}_\infty$ control inspired algorithm. We substantiate these claims through numerical experiments in representative practical scenarios.

These algorithms are obtained by extending the ideas proposed in [13] and discussed in the previous sections.

9.1 STOCHASTIC FORMULATION

We now particularize the optimization problem of interest. Specifically, we focus on solving Problem (4.1) in the unconstrained case, under the same setup as in (4.2), with the objective function (4.3).

To introduce stochasticity into the problem, we extend the deterministic model in Assumption 1.5 as follows.

Assumption 1.6 (Model of $\mathbf{c}_k$). *The sequence $\{\mathbf{c}_k\}_{k \in \mathbb{N}}$ is generated according to the following stochastic LTI model:*

$$\begin{cases} \mathbf{z}_{k+1} = (\mathbf{I}_n \otimes \mathbf{F}) \mathbf{z}_k + (\mathbf{I}_n \otimes \mathbf{G}) \mathbf{n}_k, \\ \mathbf{c}_k = (\mathbf{I}_n \otimes \mathbf{H}) \mathbf{z}_k + (\mathbf{I}_n \otimes o) \mathbf{n}_k, \end{cases} \quad (9.1)$$

where $\mathbf{F} \in \mathbb{R}^{m \times m}$ has known marginally Schur-stable poles, $\mathbf{H} \in \mathbb{R}^{1 \times m}$, $\mathbf{G} \in \mathbb{R}^{m \times 1}$, and $o \in \mathbb{R}$ are known, while the initial states $\mathbf{z}_0$ is unknown, and $\mathbf{n}_k \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_n)$ is a Gaussian white noise.

Equivalently, Equation (9.1) describes the time evolution of each component $[\mathbf{c}_k]_i$ of the vector $\mathbf{c}_k$:

$$\begin{cases} \mathbf{z}_{k+1}^{(i)} &= \mathbf{F} \mathbf{z}_k^{(i)} + \mathbf{G} [\mathbf{n}_k]_i, \\ [\mathbf{c}_k]_i &= \mathbf{H} \mathbf{z}_k^{(i)} + o [\mathbf{n}_k]_i, \end{cases} \quad (9.2)$$

where

$$\mathbf{z}_k = \begin{bmatrix} \mathbf{z}_k^{(1)} \\ \vdots \\ \mathbf{z}_k^{(n)} \end{bmatrix} \in \mathbb{R}^{mn},$$

with $\mathbf{z}_k^{(i)} \in \mathbb{R}^m$ for all i .

This assumption generalizes the framework described in Assumption 1.5 used in the previous chapters. In particular, when $\sigma^2 = 0$, the model reduces to a deterministic evolution of (4.3), equivalent to the one considered in Chapter 5. The stochastic extension in Assumption 1.6 allows us to capture unmodeled dynamics. Specifically, the process-noise term $\mathbf{G}[\mathbf{n}_k]_i$ in (9.2) accounts for unknown stochastic dynamics or environmental variations affecting the state propagation, while the measurement-noise term $\mathbf{o}[\mathbf{n}_k]_i$ introduces an additive perturbation in the evolution of $\mathbf{c}_k$.

Remark 10. Assumption 1.6 is equivalent to requiring the existence of a scalar rational transfer function $h(z)$ such that $\mathbf{c}_k$ is the output of a MIMO discrete-time system with transfer matrix $h(z)\mathbf{I}_n$, driven by Gaussian white noise $\mathbf{n}_k$, where $h(z) := \mathbf{H}(z\mathbf{I} - \mathbf{F})^{-1}\mathbf{G} + \mathbf{o}$.

9.2 ALGORITHMS

We propose a set of algorithms to address the online optimization problem described in (4.1), in the unconstrained case and with the objective function (4.3). As anticipated in the introduction, the design of these algorithms is driven by different assumptions on the available knowledge of the cost function, in particular on the eigenvalues structure of the matrix $\mathbf{A}$.

Specifically, we focus on the three assumptions introduced in Assumptions 1.1-1.3, and we develop three distinct algorithmic design procedures. The proposed methods fall into two main families, depending on the prior knowledge about the optimization problem at hand:

- The first two algorithms assume increasingly relaxed yet informative knowledge about the spectrum of $\mathbf{A}$, either exact eigenvalues (Assumption 1.1) or a known distribution (Assumption 1.2). These approaches are performance-oriented and aim to improve the tracking accuracy of the optimizer.
- The third algorithm is designed under minimal assumptions (Assumption 1.3), where only bounds on the eigenvalues are known. It follows a robust control approach based on robust $\mathcal{H}_\infty$ control design. While more conservative, this method is preferable in scenarios with significant uncertainty in the spectrum of $\mathbf{A}$.

We now introduce the general setup within which the proposed optimization algorithms are developed, following the previous derivations. Specifically, our objective is to design a controller with transfer matrix $\mathbf{C}(z) \in \mathbb{R}^{n \times n}(z)$, as in Chapter 5, operating under the same control scheme depicted in Figure 5.1, with the only difference that the online oracle now provides a stochastic gradient $\nabla f_k(\cdot)$.

For the same reasons motivating the analogous choice in the preceding sections, we restrict $\mathbf{C}(z)$ to the form $\mathbf{C}(z) = \frac{c(z)}{d(z)} \mathbf{I}_n$, where $c(z), d(z) \in \mathbb{R}[z]$ are polynomials of degree $m - 1$ and m , respectively.

From

$$\mathbf{e}_k := \nabla f_k(\mathbf{x}_k) = \underbrace{\mathbf{A}(\mathbf{x}_k - \mathbf{c}_k)}_{\mathbf{n}_k}, \quad (9.3)$$

where $\mathbf{n}_k$ is the estimation error, the control scheme in Figure 5.1 in the stochastic setting simplifies to that shown in Figure 9.1.

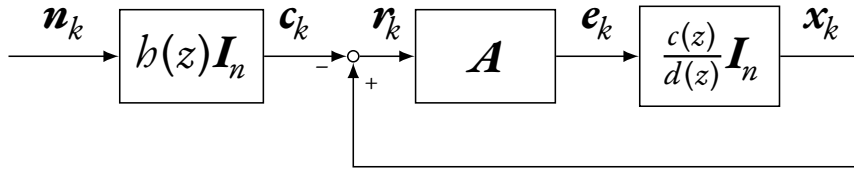


Figure 9.1: Control scheme of Figure 5.1 when $\mathbf{C}(z) = \frac{c(z)}{d(z)} \mathbf{I}_n$ and condition (9.3) holds.

Under Assumption 1.6, the optimizer $\mathbf{c}_k$ is driven by white noise. Consequently, in this stochastic setting the estimation error $\mathbf{n}_k = \mathbf{x}_k - \mathbf{c}_k$ cannot converge to zero as required in the deterministic case (see, e.g., (5.4)). As a result, a purely asymptotic convergence analysis is not meaningful, and we instead assess the algorithm's performance through an alternative metric: the asymptotic expected value of the quadratic norm of the estimation error:

$$\mathcal{J} := \lim_{k \rightarrow +\infty} \mathbb{E} [\mathbf{n}_k^\top \mathbf{n}_k]. \quad (9.4)$$

Remark 11. In this thesis we define different performance metrics that represent what the algorithm seeks to minimize or optimize in an online manner. The indices (5.5), and (7.7) linear deterministic setting, (6.21) for the nonlinear deterministic setting, and (9.4) for the stochastic setting are all measures of algorithmic quality. When we require satisfaction or optimization of one of these performance criteria, we first solve the corresponding problem offline. This yields a fixed algorithm that is then implemented in an online fashion to address the on-line optimization problem. The rationale is that, while optimizing the performance index may

be computationally expensive, it is performed once; the resulting algorithm can then be deployed without incurring the same computational cost repeatedly. This mirrors the distinction between training and inference in machine learning. A key difference here is that the minimization of the loss is not based on empirical samples alone, but leverages the relationship between the cost and a principled model of the data-generation process.

Our objective in designing an online optimization algorithm is to determine a controller $\mathbf{C}(z)$ that minimizes the cost (9.4). To this end, it is convenient to use the eigendecomposition $\mathbf{A} = \mathbf{V}\mathbf{A}\mathbf{V}^\top$ and perform the change of variables

$$\begin{aligned}\bar{\mathbf{n}}_k &:= \mathbf{V}^\top \mathbf{n}_k, & \bar{\mathbf{c}}_k &:= \mathbf{V}^\top \mathbf{c}_k, \\ \bar{\mathbf{r}}_k &:= \mathbf{V}^\top \mathbf{r}_k, & \bar{\mathbf{x}}_k &:= \mathbf{V}^\top \mathbf{x}_k.\end{aligned}\tag{9.5}$$

Under this transformation,

$$\begin{aligned}\mathcal{J} &= \lim_{k \rightarrow +\infty} \mathbb{E} [\mathbf{r}_k^\top \mathbf{r}_k] = \lim_{k \rightarrow +\infty} \mathbb{E} [\bar{\mathbf{r}}_k^\top \mathbf{V} \mathbf{V}^\top \bar{\mathbf{r}}_k] \\ &= \lim_{k \rightarrow +\infty} \mathbb{E} [\bar{\mathbf{r}}_k^\top \bar{\mathbf{r}}_k] = \sum_{i=1}^n \lim_{k \rightarrow +\infty} \mathbb{E} [\bar{\mathbf{r}}_k]_i^2,\end{aligned}\tag{9.6}$$

since $\mathbf{V}$ is orthogonal. Moreover,

$$\begin{aligned}\bar{\mathbf{r}}_k &= \mathbf{V}^\top (\mathbf{x}_k - \mathbf{c}_k) = \mathbf{V}^\top \left(\frac{c(z)}{d(z)} \mathbf{A} \mathbf{e}_k - \mathbf{c}_k \right) \\ &= \frac{c(z)}{d(z)} \mathbf{V}^\top \mathbf{A} \mathbf{e}_k - \bar{\mathbf{c}}_k = \frac{c(z)}{d(z)} \mathbf{A} \bar{\mathbf{r}}_k - \bar{\mathbf{c}}_k.\end{aligned}$$

Using the identity $\bar{\mathbf{c}}_k = b(z) \bar{\mathbf{n}}_k$, the error dynamics in the transformed coordinates are

$$\bar{\mathbf{r}}_k = \underbrace{- \left(\mathbf{I}_n - \frac{c(z)}{d(z)} \mathbf{A} \right)^{-1}}_{:= \mathbf{H}_{nr}(z)} b(z) \bar{\mathbf{n}}_k.\tag{9.7}$$

Since the transfer matrix $\mathbf{H}_{nr}(z)$ is diagonal, each component $[\bar{\mathbf{r}}_k]_i$ is affected solely by $[\bar{\mathbf{n}}_k]_i$ and depends only on the corresponding eigenvalue λ_i (and not on any λ_j for $j \neq i$).

This decoupling, illustrated in Figure 9.2, applied to (9.6) yields the decomposition

$$\mathcal{J} = \sigma^2 \|\mathbf{H}_{nr}(z)\|_2^2 = \sum_{i=1}^n \sigma^2 \left\| [\mathbf{H}_{nr}(z)]_{ii} \right\|_2^2 = \sum_{i=1}^n \tilde{\mathcal{J}}(\lambda_i),\tag{9.8}$$

where

$$\tilde{\mathcal{J}}(\lambda_i) := \sigma^2 \left\| [\mathbf{H}_{nr}(z)]_{ii} \right\|_2^2 = \sigma^2 \left\| \frac{b(z)}{1 - \lambda_i c(z)/d(z)} \right\|_2^2, \quad (9.9)$$

and where we use the classical relationship between the $\mathcal{H}_2$ norm and the asymptotic value of the output covariance.

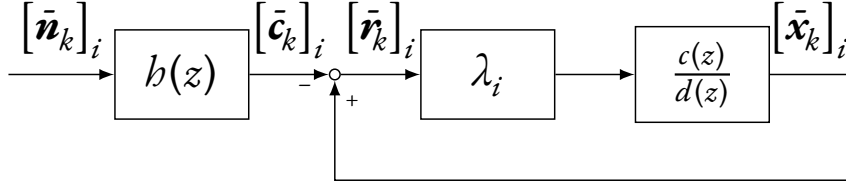


Figure 9.2: Block representation of the i -th transfer function $[\mathbf{H}_{nr}(z)]_{ii}$ in (9.7).

In the following section we propose various algorithms characterized by different levels of complexity and performance. We start from the simplest one, based on Kalman filtering.

9.2.1 KALMAN FILTERING INSPIRED ALGORITHM

Consider the stochastic system of the form (9.1) with input n_k and output c_k , where $n_k = [\mathbf{n}_k]_i$ and $c_k = [\mathbf{c}_k]_i$ for a generic i . It is well known that the system with input c_k and output r_k which minimizes the $\mathcal{H}_2$ norm of their interconnection is given by the Kalman filter, namely

$$\begin{aligned} \hat{\mathbf{z}}_{k+1} &= \mathbf{F}\hat{\mathbf{z}}_k + \bar{\mathbf{K}}(\mathbf{H}\hat{\mathbf{z}}_k - c_k), \\ r_k &= \mathbf{H}\hat{\mathbf{z}}_k - c_k, \end{aligned}$$

where $\hat{\mathbf{z}}_k \in \mathbb{R}^m$, $\mathbf{F}$ and $\mathbf{H}$ are the same matrices as in (9.2), and $\bar{\mathbf{K}}$ is the steady-state Kalman filter gain [113, Theorem 8.5.3], given by

$$\bar{\mathbf{K}} = -(\mathbf{FPH}^\top + \mathbf{G}\sigma^2 j)(\mathbf{HPH}^\top + \sigma^2)^{-1}, \quad (9.10)$$

where $\mathbf{P}$ is the solution of the algebraic Riccati equation

$$\mathbf{P} = \mathbf{FPF}^\top + \sigma^2 \mathbf{GG}^\top - (\mathbf{FPH}^\top + \mathbf{G}\sigma^2 j)(\mathbf{HPH}^\top + \sigma^2)^{-1}(\mathbf{FPH}^\top + \mathbf{G}\sigma^2 j)^\top.$$

Suppose now that we perturb the optimal gain $\bar{\mathbf{K}}$ by scaling it with a factor $\gamma \in \mathbb{R}$, so that

the filter becomes

$$\begin{aligned}\hat{\mathbf{z}}_{k+1} &= \mathbf{F}\hat{\mathbf{z}}_k + \gamma\bar{\mathbf{K}}(\mathbf{H}\hat{\mathbf{z}}_k - c_k), \\ \hat{r}_k &= \mathbf{H}\hat{\mathbf{z}}_k - c_k.\end{aligned}$$

In general, the $\mathcal{H}_2$ norm of the interconnection is no longer minimized, except when $\gamma = 1$. We will see that this perturbation is unavoidable, since in general we do not know the full structure of $\mathbf{A}$ and therefore we must apply the same transfer function to all modes. As a consequence, the closed-loop responses coincide with those in the previous equation with $\gamma = 1$ only if all eigenvalues are identical; in all other cases, we must account for the perturbation introduced by γ .

We denote this $\mathcal{H}_2$ norm by $\tilde{\mathcal{J}}(\gamma)$.

This observation motivates the construction of a suboptimal controller $c(z)/d(z)$ with state space realization (5.16), where the gain is chosen as $\mathbf{K} = a\bar{\mathbf{K}}$ with $a \in \mathbb{R}$ a scalar parameter to be determined. This approach greatly simplifies the controller synthesis, since it reduces the design to tuning a single scalar parameter. With this choice, the overall cost becomes

$$\sum_{i=1}^n \tilde{\mathcal{J}}(a\lambda_i).$$

Note that $\tilde{\mathcal{J}}(\gamma)$ is finite if and only if the transfer function from n_k to $\hat{r}_k$ is stable, which is equivalent to requiring that the matrix $\mathbf{F}_c(\gamma) := \mathbf{F} + \gamma\bar{\mathbf{K}}\mathbf{H}$ is stable. This condition can be verified following the same steps as in Corollary 1; this time we leverage Theorem 15 from [41]. In particular, $\mathbf{F}_c(\gamma)$ is stable for all $\gamma \in [\gamma_{\min}, \gamma_{\max}]$ if the following corollary holds.

Corollary 3. *The eigenvalues of $\mathbf{F}_c(\gamma)$ lie strictly inside the unit circle for any $\gamma \in [\gamma_{\min}, \gamma_{\max}]$ if there exist symmetric matrices $\mathbf{P}_{\min}, \mathbf{P}_{\max} \succ 0$, and a matrix $\mathbf{L}$ such that*

$$\begin{aligned}\begin{bmatrix} \mathbf{P}_{\min} & \mathbf{F}_c^\top(\gamma_{\min})\mathbf{L}^\top \\ \star & \mathbf{L} + \mathbf{L}^\top - \mathbf{P}_{\min} \end{bmatrix} &\succ 0, \\ \begin{bmatrix} \mathbf{P}_{\max} & \mathbf{F}_c^\top(\gamma_{\max})\mathbf{L}^\top \\ \star & \mathbf{L} + \mathbf{L}^\top - \mathbf{P}_{\max} \end{bmatrix} &\succ 0.\end{aligned}$$

Proof. The matrix $\mathbf{F}_c(\gamma)$ can be written as

$$\mathbf{F}_c(\gamma) = \mathbf{F} + \gamma\bar{\mathbf{K}}\mathbf{H} = (1 - \alpha)(\mathbf{F} + \gamma_{\min}\bar{\mathbf{K}}\mathbf{H}) + \alpha(\mathbf{F} + \gamma_{\max}\bar{\mathbf{K}}\mathbf{H}), \quad (9.11)$$

for some $\alpha \in [0, 1]$ such that $\gamma = (1 - \alpha)\gamma_{\min} + \alpha\gamma_{\max}$. Hence (9.11) describes a convex

polytope with vertices $\mathbf{F}_1 = \mathbf{F}_c(\gamma_{\min})$ and $\mathbf{F}_2 = \mathbf{F}_c(\gamma_{\max})$. The result then follows by direct application of Theorem 14. $\square$

Using Corollary 3, we can find $0 \leq \gamma_{\min} \leq \gamma_{\max}$ such that $\mathbf{F}_c(\gamma)$ is stable, and hence $\tilde{J}(\gamma)$ is finite for all $\gamma \in [\gamma_{\min}, \gamma_{\max}]$. Notice that $a\lambda \in [\gamma_{\min}, \gamma_{\max}]$ for all $\lambda \in [\lambda_{\min}, \lambda_{\max}]$ if and only if $a \in [a_{\min}, a_{\max}]$, where

$$a_{\min} := \frac{\gamma_{\min}}{\lambda_{\min}}, \quad a_{\max} := \frac{\gamma_{\max}}{\lambda_{\max}}.$$

This interval is non-empty if and only if

$$\frac{\gamma_{\max}}{\gamma_{\min}} \geq \frac{\lambda_{\max}}{\lambda_{\min}}. \quad (9.12)$$

Thus, when the ratio $\lambda_{\max}/\lambda_{\min}$ is too large, the method becomes infeasible. Otherwise, when (9.12) holds, the optimization problem reduces to

$$\min_{a \in [a_{\min}, a_{\max}]} \sum_{i=1}^n \tilde{J}(a\lambda_i). \quad (9.13)$$

If the eigenvalues λ_i are known (Assumption 1.1), this can be solved by a simple line search.

An even simpler approximation is obtained by expanding $\tilde{J}(\gamma)$ around its minimizer $\gamma = 1$ via a second-order Taylor expansion:

$$\tilde{J}(\gamma) \approx \tilde{J}(1) + \underbrace{\frac{\partial \tilde{J}}{\partial \gamma} \Big|_{\gamma=1}}_{=0} (\gamma - 1) + \underbrace{\frac{\partial^2 \tilde{J}}{\partial \gamma^2} \Big|_{\gamma=1}}_{\beta} (\gamma - 1)^2, \quad (9.14)$$

where the linear term vanishes by optimality of $\gamma = 1$. This approximation is accurate when γ is close to 1. Assuming all $\gamma_i := a\lambda_i$ are close to 1, we obtain

$$\begin{aligned} \sum_{i=1}^n \tilde{J}(a\lambda_i) &\approx \sum_{i=1}^n \beta (a\lambda_i - 1)^2 + \text{const.} \\ &= \beta \left(a^2 \sum_{i=1}^n \lambda_i^2 - 2a \sum_{i=1}^n \lambda_i \right) + \text{const.}, \end{aligned} \quad (9.15)$$

which is quadratic in a . Minimizing (9.15) yields

$$a^* = \frac{\sum_{i=1}^n \lambda_i}{\sum_{i=1}^n \lambda_i^2}, \quad (9.16)$$

provided that $a^* \in [a_{\min}, a_{\max}]$.

The value a^* can be computed when the eigenvalues of $\mathbf{A}$ are known. If instead only a probability distribution $\phi(\lambda)$ of the eigenvalues is available (Assumption 1.2), one can use

$$a^* = \frac{\int \lambda \phi(\lambda) d\lambda}{\int \lambda^2 \phi(\lambda) d\lambda}. \quad (9.17)$$

Finally, if it is only known that $\lambda_i \in [\lambda_{\min}, \lambda_{\max}]$ (Assumption 1.3), we may assume they are uniformly distributed in this interval. In that case, (9.17) gives

$$a^* = \frac{3}{2} \frac{\lambda_{\max} + \lambda_{\min}}{\lambda_{\max}^2 + \lambda_{\max} \lambda_{\min} + \lambda_{\min}^2}. \quad (9.18)$$

In all cases, the resulting controller is

$$\text{Online Optimization algorithm} \begin{cases} \hat{\mathbf{z}}_{k+1} = (\mathbf{I}_n \otimes \mathbf{F}) \hat{\mathbf{z}}_k + (\mathbf{I}_n \otimes \mathbf{K}) \mathbf{e}_k, \\ \mathbf{x}_k = (\mathbf{I}_n \otimes \mathbf{H}) \hat{\mathbf{z}}_k, \end{cases} \quad (9.19a)$$

$$(9.19b)$$

$$\text{Online optimization oracle } \mathbf{e}_k = \nabla_{\mathbf{x}} f_k(\mathbf{x}_k), \quad (9.19c)$$

and with $\mathbf{K} = a^* \bar{\mathbf{K}}$ and $\bar{\mathbf{K}}$ given by (9.10).

9.2.2 OPTIMIZED FILTER

The method proposed in the previous section yields the controller (9.19) with a gain $\mathbf{K}$ that is not optimal. In fact, the choice of this gain can be refined through an optimization procedure. Indeed, the problem of optimizing the cost (9.8) with respect to $\mathbf{K}$ has already been investigated in the literature (see [89]). Here, we propose a tailored first-order optimization algorithm based on gradient descent.

From (9.8) we observe that the gradient of the cost decomposes as

$$\frac{\partial \mathcal{J}(\mathbf{K})}{\partial \mathbf{K}} = \sum_{i=1}^n \frac{\partial \tilde{\mathcal{J}}(\lambda_i, \mathbf{K})}{\partial \mathbf{K}}, \quad (9.20)$$

where each term $\tilde{\mathcal{J}}(\lambda_i, \mathbf{K})$ is defined in (9.9) and corresponds to the $\mathcal{H}_2$ norm associated with the i -th scalar subsystem relative to eigenvalue λ_i . Notice that, since the gradient is taken with respect to $\mathbf{K}$, the dependence of $\tilde{\mathcal{J}}(\lambda_i)$ on $\mathbf{K}$ is made explicit. Following [82], a closed-form expression for each component of the gradient is

$$\frac{\partial \tilde{\mathcal{J}}(\lambda_i, \mathbf{K})}{\partial \mathbf{K}} = 2\lambda_i \mathbf{Q}_i \left(-\lambda_i \mathbf{K} o^2 \sigma^2 - \mathbf{G} \sigma^2 o - (\mathbf{F} + \lambda_i \mathbf{K} \mathbf{H}) \mathbf{P}_i \mathbf{H}^\top \right), \quad (9.21)$$

where, for each $i = 1, \dots, n$, the matrices $\mathbf{P}_i$ and $\mathbf{Q}_i$ are the solutions of the discrete-time Lyapunov equations

$$\begin{aligned} \mathbf{P}_i &= (\mathbf{F} + \lambda_i \mathbf{K} \mathbf{H})^\top \mathbf{P}_i (\mathbf{F} + \lambda_i \mathbf{K} \mathbf{H}) + \mathbf{R}_i \mathbf{R}_i^\top, \\ \mathbf{Q}_i &= (\mathbf{F} + \lambda_i \mathbf{K} \mathbf{H}) \mathbf{Q}_i (\mathbf{F} + \lambda_i \mathbf{K} \mathbf{H})^\top + \mathbf{H}^\top \mathbf{H}, \end{aligned} \quad (9.22)$$

and $\mathbf{R}_i$ is defined as

$$\mathbf{R}_i = \sigma (\mathbf{G} + \lambda_i o \mathbf{K}).$$

The resulting first-order algorithm for gain optimization is

$$\mathbf{K}^{(t+1)} = \mathbf{K}^{(t)} - \alpha \left. \frac{\partial \mathcal{J}(\mathbf{K})}{\partial \mathbf{K}} \right|_{\mathbf{K}=\mathbf{K}^{(t)}}, \quad (9.23)$$

where $\alpha > 0$ is a step size selected at each iteration by a line search. The procedure is as follows:

1. Initialize the gain $\mathbf{K}^{(0)}$ using the method presented in Chapter 5, which guarantees stability.
2. At each iteration t , starting from a stabilizing $\mathbf{K}^{(t)}$, perform:
 - (a) compute the gradient $\left. \frac{\partial \mathcal{J}(\mathbf{K})}{\partial \mathbf{K}} \right|_{\mathbf{K}=\mathbf{K}^{(t)}}$ using (9.20), (9.21), and (9.22);
 - (b) using Corollary 4, determine the maximum admissible step size $\alpha_{\max} > 0$ such that for any $\alpha \in [0, \alpha_{\max}]$ the updated gain $\mathbf{K}^{(t+1)}(\alpha)$ preserves stability of $[\mathbf{H}_{nr}(z)]_{ii}$ for all i ;

- (c) select $\alpha \in [0, \alpha_{\max}]$ that minimizes the cost $\mathcal{J}$, and use this value in (9.23) to update the gain $\mathbf{K}^{(t+1)}(\alpha)$;
- (d) repeat until a stopping criterion is satisfied.

Note that the stability of $[\mathbf{H}_{nr}(z)]_{ii}$ for a given gain $\mathbf{K}$ is equivalent to the stability of $\mathbf{F}_c(\lambda_i, \alpha) := \mathbf{F} + \lambda_i \mathbf{K}(\alpha) \mathbf{H}$. The line search in step (c) thus requires computing the largest admissible step size $\alpha_{\max}$ that ensures stability, as determined in step (b). To this end, we leverage the same LMI results used in Corollary 3, and stated in the following corollary.

Corollary 4. *The eigenvalues of $\mathbf{F}_c(\lambda, \alpha)$, where $\mathbf{K}(\alpha) := \mathbf{K}_1 + \alpha \mathbf{K}_2$, lie strictly inside the unit circle for any $\lambda \in [\lambda_1, \lambda_2]$ and for any $\alpha \in [\alpha_1, \alpha_2]$ if there exist four positive definite matrices $\mathbf{P}_{i,j}$, $i, j \in \{1, 2\}$, and a matrix $\mathbf{L}$ such that*

$$\begin{bmatrix} \mathbf{P}_{i,j} & (\mathbf{F} + \lambda_i \mathbf{K}(\alpha_j) \mathbf{H})^\top \mathbf{L}^\top \\ \star & \mathbf{L} + \mathbf{L}^\top - \mathbf{P}_{i,j} \end{bmatrix} \succ 0, \quad i, j \in \{1, 2\}. \quad (9.24)$$

Proof. First, we note that the matrix $\mathbf{F}_c(\lambda, \alpha)$ forms a convex polytope in $\lambda \in [\lambda_1, \lambda_2]$ and $\alpha \in [\alpha_1, \alpha_2]$. This is a consequence of Proposition 5, with $x = \lambda$, $y = \alpha$, $\mathbf{F}_1 = \mathbf{F}$, $\mathbf{F}_2 = \mathbf{K}_1 \mathbf{H}$, and $\mathbf{F}_3 = -\mathbf{K}_2 \mathbf{H}$. The claim then follows by direct application of Theorem 14. $\square$

Assumption 1.1, which requires exact knowledge of the eigenvalues λ_i of $\mathbf{A}$, is restrictive and often impractical. In real applications, computing the spectrum of $\mathbf{A}$ is rarely feasible, especially when the dimension n is large. In such cases, evaluating the exact gradient expression in (9.21) becomes computationally prohibitive, and an alternative approach is necessary.

A more realistic assumption is that the eigenvalues of $\mathbf{A}$ follow a known probability distribution, as in Assumption 1.2. For instance, when $\mathbf{A}$ represents a random covariance matrix [11], the empirical spectral distribution of $\mathbf{A}$ converges to the Marchenko–Pastur distribution, which provides a statistical description of the eigenvalue distribution.

Under this assumption, the cost (9.8) is naturally modified by replacing the discrete sum over eigenvalues with an expectation with respect to the density $\phi(\lambda)$, yielding

$$\hat{\mathcal{J}} := \mathbb{E} [\bar{\mathcal{J}}(\lambda)] = \int_{-\infty}^{+\infty} \bar{\mathcal{J}}(\lambda) \phi(\lambda) d\lambda, \quad (9.25)$$

where $\phi(\lambda)$ denotes the probability density of the eigenvalues λ_i .

This reformulation ensures the well-posedness of the gain optimization problem and allows the same gradient-based algorithm to be applied. The gradient of the modified cost with respect to $\mathbf{K}$ is

$$\frac{\partial \hat{J}(\mathbf{K})}{\partial \mathbf{K}} = \int_{-\infty}^{+\infty} \frac{\partial \bar{J}(\lambda, \mathbf{K})}{\partial \mathbf{K}} \phi(\lambda) d\lambda. \quad (9.26)$$

The controller in (9.19) can then be implemented using a gain $\mathbf{K}$ computed by the same gradient descent procedure as in (9.23), but with the gradient evaluated according to (9.26).^{*} This provides a practical method for designing the controller when the exact spectrum of $\mathbf{A}$ is unknown but its distribution is known.

9.2.3 ROBUST $\mathcal{H}_\infty$ CONTROL INSPIRED ALGORITHM

In case we are in the least restrictive case, corresponding to Assumption 1.3, one potential approach could be to use the previous methods assuming that the distribution of the eigenvalues is uniform distribution in $[\lambda_{\min}, \lambda_{\max}]$.

Here we propose an alternative a control-theoretic strategy based on a robustness-performance trade-off. In particular, we formulate a new algorithm design using robust $\mathcal{H}_\infty$ control theory. While our original objective is to minimize the cost (9.8), the steady-state covariance error trace, a known fact states that the $\mathcal{H}_\infty$ norm provides an upper bound on the $\mathcal{H}_2$ norm [40]

$$\|\mathbf{H}_{nr}(z)\|_2 \leq \|\mathbf{H}_{nr}(z)\|_\infty.$$

The $\mathcal{H}_\infty$ norm is defined as

$$\|\mathbf{H}_{nr}(z)\|_\infty := \sup_{\theta \in [-\pi, \pi)} \sigma_{\max}(\mathbf{H}_{nr}(e^{i\theta})), \quad (9.27)$$

which simplifies to

$$\begin{aligned} \|\mathbf{H}_{nr}(z)\|_\infty &= \max_{i=1, \dots, n} \sup_{\theta \in [-\pi, \pi)} \left| \frac{b(e^{i\theta})}{1 - \lambda_i \frac{c(e^{i\theta})}{d(e^{i\theta})}} \right|, \\ &= \max_{i=1, \dots, n} \|\mathbf{H}_{nr}(z)\|_{ii}. \end{aligned}$$

Since the eigenvalues λ_i are unknown but known to lie in $[\lambda_{\min}, \lambda_{\max}]$, a robust design

^{*}The integral (9.26) can be computed numerically given $\phi(\lambda)$ and the expression of $\frac{\partial \bar{J}(\lambda, \mathbf{K})}{\partial \mathbf{K}}$. The former is assumed to be known, while the latter can be computed for any λ and $\mathbf{K}$ using (9.21) and (9.22).

must account for the worst-case scenario over this interval. This leads us to the following cost function

$$\begin{aligned}\hat{\mathcal{J}} &= \sup_{\lambda \in [\lambda_{\min}, \lambda_{\max}]} \sup_{\theta \in [-\pi, \pi)} \left| \frac{b(e^{i\theta})}{1 - \lambda \frac{c(e^{i\theta})}{d(e^{i\theta})}} \right| \\ &= \sup_{\lambda \in [\lambda_{\min}, \lambda_{\max}]} \left\| -\frac{b(z)}{1 - \lambda \frac{c(z)}{d(z)}} \right\|_{\infty}.\end{aligned}\tag{9.28}$$

The optimization problem thus becomes finding a controller $\frac{c(z)}{d(z)}$ that minimizes $\hat{\mathcal{J}}$, accounting for parametric uncertainty in λ .*

This formulation is a standard robust $\mathcal{H}_{\infty}$ control problem, typically addressed via μ -synthesis [125, Section 10]. While we do not detail the complete μ -synthesis algorithm here (see [125, Section 10.4]), we now explain how our problem setup maps into the robust control framework.

When F in (9.2) is stable, μ -synthesis can be directly applied. However, when F is only marginally stable, a two-step design procedure is required. This is because the unstable poles of $b(z)$ become uncontrollable in the closed-loop system (as in Figure 9.2), thereby violating the assumptions of classical robust $\mathcal{H}_{\infty}$ control theory [125, Sections 14.7–14.8].

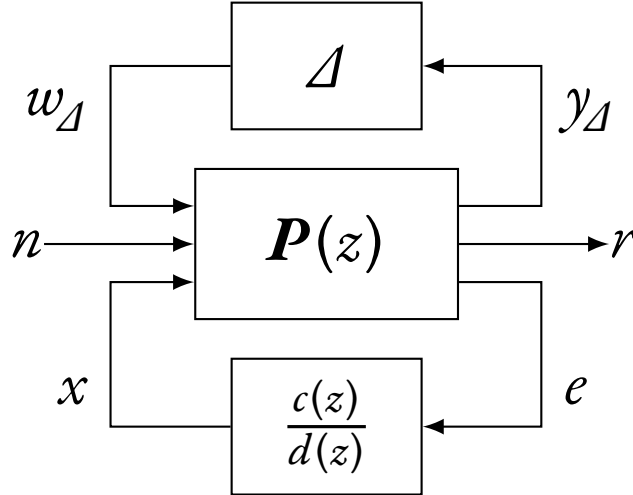


Figure 9.3: General framework for μ -synthesis.

To address this, we align our setup with the general μ -synthesis architecture described

*Solved via bisection on $\gamma > 0$: test whether a controller $\frac{c(z)}{d(z)}$ satisfies (9.28) for all $\lambda \in [\lambda_{\min}, \lambda_{\max}]$, and select the largest feasible γ .

in [125]. We reshape the scheme of Figure 9.2 to fit the general robust control framework depicted in Figure 9.4.

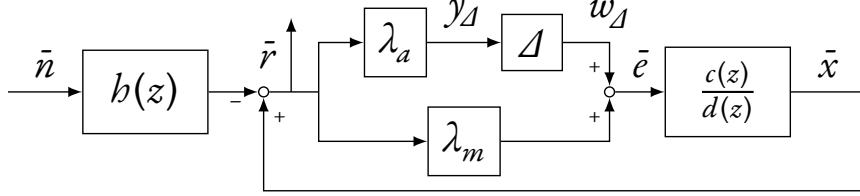


Figure 9.4: Reshaped version for a generic i of Figure 9.2 compatible with the μ -synthesis framework in Figure 9.3.

Here, $\Delta \in [-1, 1]$ represents the uncertainty block, and w_Δ and y_Δ are auxiliary signals capturing the uncertainty structure [96, Section 3.4]. We define

$$\lambda_m = \frac{\lambda_{\max} - \lambda_{\min}}{2}, \quad \lambda_a = \frac{\lambda_{\max} + \lambda_{\min}}{2}, \quad (9.29)$$

so that $\lambda = \lambda_m + \lambda_a \Delta$. The entries of the interconnection matrix $\mathbf{P}(z)$ are then derived from the scheme in Figure 9.4.

Since the uncontrollable poles of $b(z)$ violate robust $\mathcal{H}_\infty$ control conditions, we

1. introduce a fixed precompensator to move the uncontrollable poles of $b(z)$ inside the control loop;
2. perform robust $\mathcal{H}_\infty$ control design on the precompensated system.

For the first step, we factorize $b(z)$ as

$$b(z) = \frac{n(z)}{d_u(z)d_s(z)}, \quad (9.30)$$

where $d_u(z)$ collects the unstable poles, and $d_s(z)$ the stable ones. We then impose the following structure on the controller

$$\frac{c(z)}{d(z)} = \underbrace{\frac{n_s(z)}{d_u(z)}}_{\text{precompensator}} \bar{c}(z), \quad (9.31)$$

where $n_s(z)$ is a stable polynomial of the same degree as $d_u(z)$ and $\bar{c}(z)$ is a strictly proper transfer function. This leads to the equivalent scheme shown in Figure 9.5. In this setup,

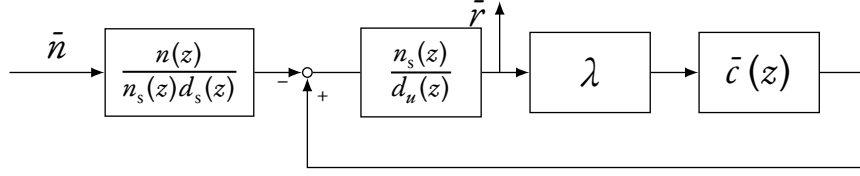


Figure 9.5: Equivalent control scheme for $\frac{c(z)}{d(z)}$ parametrized as in (9.31).

all uncontrollable poles are stable, enabling us to apply standard μ -synthesis to design $\bar{c}(z)$, which is taken to be strictly proper to avoid algebraic loops. Subsequently, the final controller is implemented using a state-space realization of the controller $\mathbf{C}(z) = \frac{c(z)}{d(z)} \mathbf{I}_n$, as done in (9.19).

Finally, we note that the choice of $n_s(z)$ in (9.31) introduces a degree of freedom. For simplicity, we set $n_s = z^{n_u}$,^{*} where n_u is the degree of $d_u(z)$.

9.3 NUMERICAL RESULTS

In this section, we compare the proposed algorithms under various scenarios and conduct a detailed performance evaluation. We begin by highlighting that our proposed methods are compared against standard baselines, including the classical OGD algorithm (5.2), and the algorithm (5.18), optimized for fast convergence in deterministic settings. Note, however, that in our experimental setup, the OGD algorithm yields an infinite cost $\mathcal{J}$ as defined in (9.4), and thus its results are not displayed in subsequent plots.

We consider a simulation scenario inspired by practical applications such as online sensor calibration or portfolio return tracking [74]. Specifically, we analyze the following least-squares problem:

$$\mathbf{x}_k^* = \arg \min_{\mathbf{x} \in \mathbb{R}^n} \underbrace{\frac{1}{p} \|\mathbf{Z}^\top \mathbf{x}_k - \mathbf{y}_k\|_2^2}_{\mathcal{J}_{\text{MP}}}, \quad (9.32)$$

where $\mathbf{Z} = [\mathbf{z}_1, \dots, \mathbf{z}_p] \in \mathbb{R}^{n \times p}$, with $p \geq n$, is a random matrix with entries sampled i.i.d. from a Gaussian distribution $\mathcal{N}(0, \sigma_{\text{MP}}^2)$. With probability one, such a matrix is full row rank. The cost function can be rewritten as:

$$\mathcal{J}_{\text{MP}} = (\mathbf{x}_k - \mathbf{y}_k')^\top \mathbf{A} (\mathbf{x}_k - \mathbf{y}_k') + d_k, \quad (9.33)$$

^{*}This choice is motivated by the empirical observation that different choices of $n_s(z)$ have negligible impact on the closed-loop performance.

where $\mathbf{A} = \mathbf{Z}\mathbf{Z}^\top / p$, $\mathbf{y}'_k = (\mathbf{Z}\mathbf{Z}^\top)^{-1} \mathbf{Z}\mathbf{y}_k$, and d_k is a scalar term independent of $\mathbf{x}_k$.

A key property of this setup is that the matrix $\mathbf{A}$ is a random matrix whose eigenvalues, in the asymptotic regime $n, p \rightarrow +\infty$ with $n/p \rightarrow \nu \in (0, 1)$, follow the Marchenko–Pastur distribution [81]. Its probability density function is given by

$$\phi(\lambda) = \frac{1}{2\pi\sigma_{\text{MP}}^2} \frac{\sqrt{(\lambda_{\text{max}} - \lambda)(\lambda - \lambda_{\text{min}})}}{\nu\lambda}, \quad (9.34)$$

supported on the interval $[\lambda_{\text{min}}, \lambda_{\text{max}}]$, where

$$\lambda_{\text{min}} = \sigma_{\text{MP}}^2 (1 - \sqrt{\nu})^2, \quad \lambda_{\text{max}} = \sigma_{\text{MP}}^2 (1 + \sqrt{\nu})^2.$$

Assuming a known evolution of $\mathbf{y}'_k$ as in (9.2), we apply the optimization algorithms from subsections 9.2.1, 9.2.2, and 9.2.3 to this setting.

We first consider the case in which the vector $\mathbf{y}'_k$ follows a noisy sinusoidal evolution with frequency $1/12$, perturbed by Gaussian noise. This allows us to evaluate algorithm performance under realistic online regression dynamics. We vary the distribution parameter $\nu \in [0.02, 0.055]$, while fixing the parameters to $n = 500$, $\sigma = 0.1$, and $\sigma_{\text{MP}} = 0.7$.

From Figure 9.6, we observe that the proposed methods outperform the OIMGD baseline (5.18), with the algorithm derived in Section 9.2.2, where $\phi(\cdot)$ is set according to (9.34), achieving the lowest cost. The Kalman-inspired method also performs well and, due to its simplicity, may be preferable for practical implementation despite its slightly lower performance.

We now evaluate the robustness of the algorithms from Sections 9.2.2 and 9.2.1 with respect to model mismatch in the eigenvalue distribution, and compare them to the robust $\mathcal{H}_\infty$ approach from Section 9.2.3.

In Figure 9.7, we compare:

1. The algorithm from Section 9.2.2 using the true Marchenko–Pastur distribution (9.34);
2. The same algorithm assuming a uniform eigenvalue distribution in $[\lambda_{\text{min}}, \lambda_{\text{max}}]$;
3. The robust $\mathcal{H}_\infty$ control algorithm from Section 9.2.3.

We observe that while the algorithm from Section 9.2.2 achieves the best performance when the prior is accurate, its performance degrades under model mismatch. In such cases,

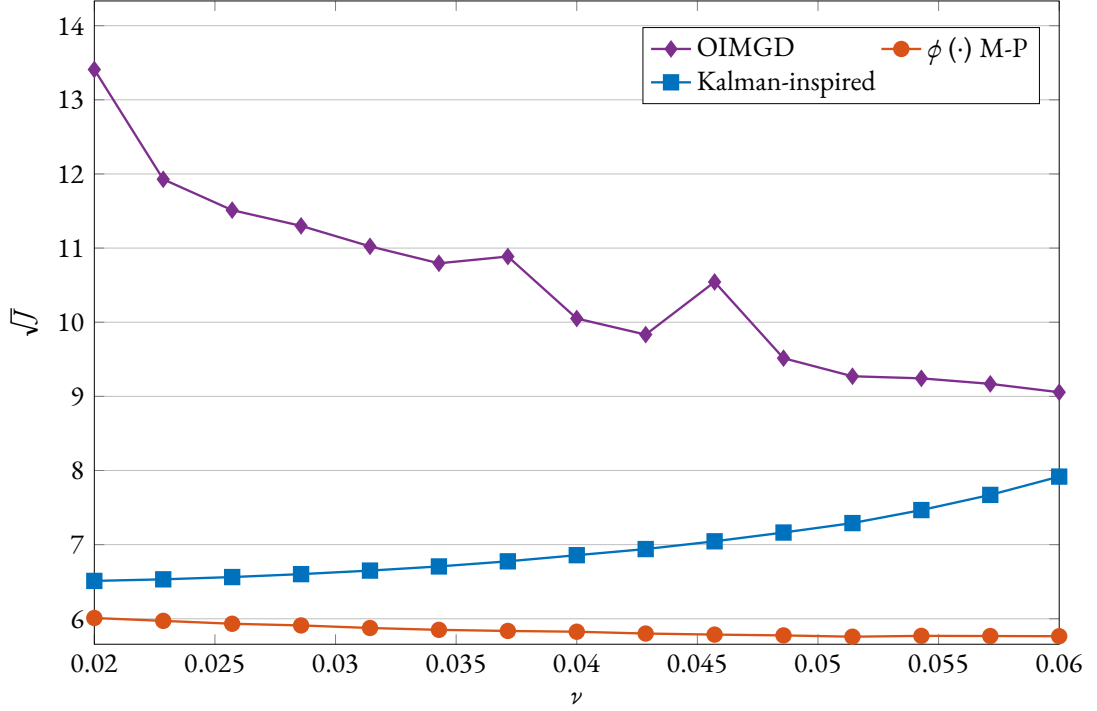


Figure 9.6: Root mean square cost (9.4) for the problem in (9.32) under sinusoidal evolution of $\mathbf{y}'_k$, comparing three algorithms: the Online Internal Model Gradient Descent (OIMGD) algorithm (5.18), the Kalman-inspired algorithm from Section 9.2.1, and the algorithm from Section 9.2.2 derived setting $\phi(\cdot)$ to (9.34).

the robust $\mathcal{H}_\infty$ control approach offers superior robustness and is preferable when the spectral distribution of $\mathbf{A}$ is uncertain.

Finally, we compare the Kalman-inspired and robust $\mathcal{H}_\infty$ control algorithms. For these tests we set $\mathbf{y}'_k$ as a noisy linear ramp, introducing unbounded growth. Parameters are set as $n = 500$, $\sigma = 0.1$, and $\sigma_{\text{MP}} = 0.7$.

In Figure 9.8, we include:

1. the Kalman-inspired algorithm with the true (9.34) prior;
2. the same algorithm assuming the eigenvalues uniformly distributed;
3. the robust $\mathcal{H}_\infty$ control algorithm.

Results show that the Kalman-inspired method performs well when the spectral prior is accurate. However, in the presence of prior mismatch or model uncertainty, the robust $\mathcal{H}_\infty$ control algorithm maintains better performance, making it a reliable default choice in practice.

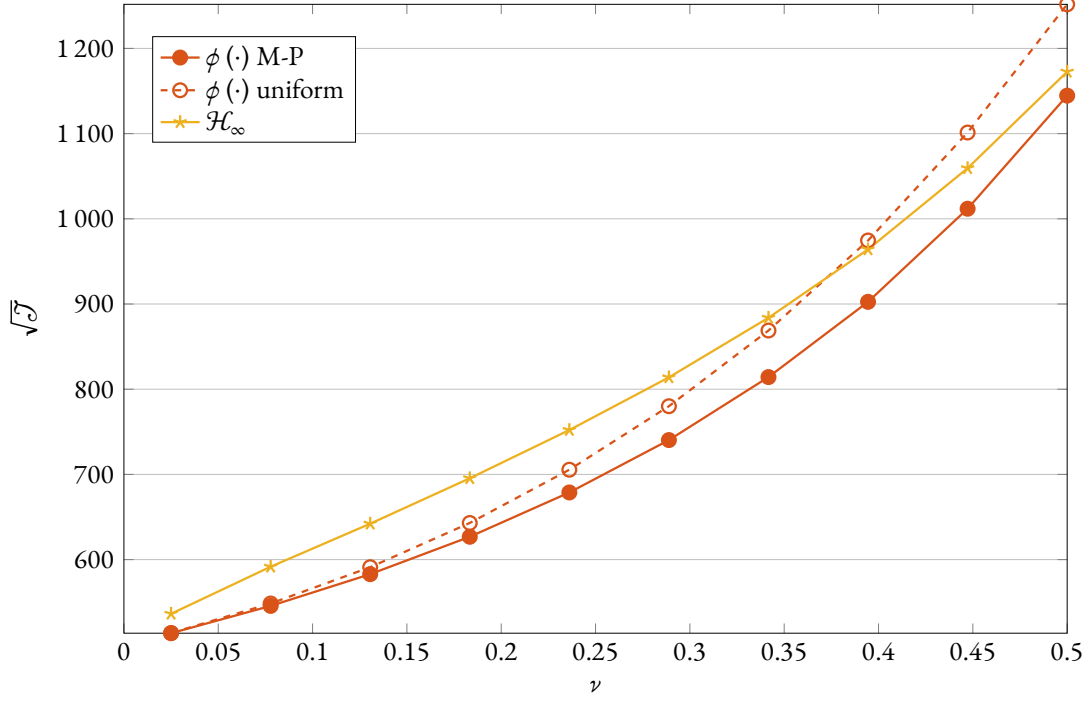


Figure 9.7: Robustness analysis for the algorithms from Section 9.2.2 (with correct and misspecified prior $\phi(\cdot)$) and Section 9.2.3, tested on the problem (9.32) with $n = 2000$, $\sigma = 0.5$, $\sigma_{\text{MP}} = 0.7$, and sinusoidal evolution of $\mathbf{y}'_k$.

9.4 DISCUSSION

In this section, we extended the deterministic framework introduced in Chapter 5 to stochastic environments and developed several online optimization algorithms, each inspired by a distinct control-theoretic principle and tailored to different application scenarios. Through numerical experiments, we show the effectiveness of the proposed methods in challenging, uncertain settings. In this way, we further robustify the approaches introduced in the previous chapters to deal with increased uncertainties.

Promising research directions include extending the methodology to time-varying and nonquadratic objective functions, as well as integrating system-identification techniques to estimate unknown model parameters during *online* execution.

In addition, similarly to what has been done in the previous sections, we would like to extend the results derived here to the constrained case as well.

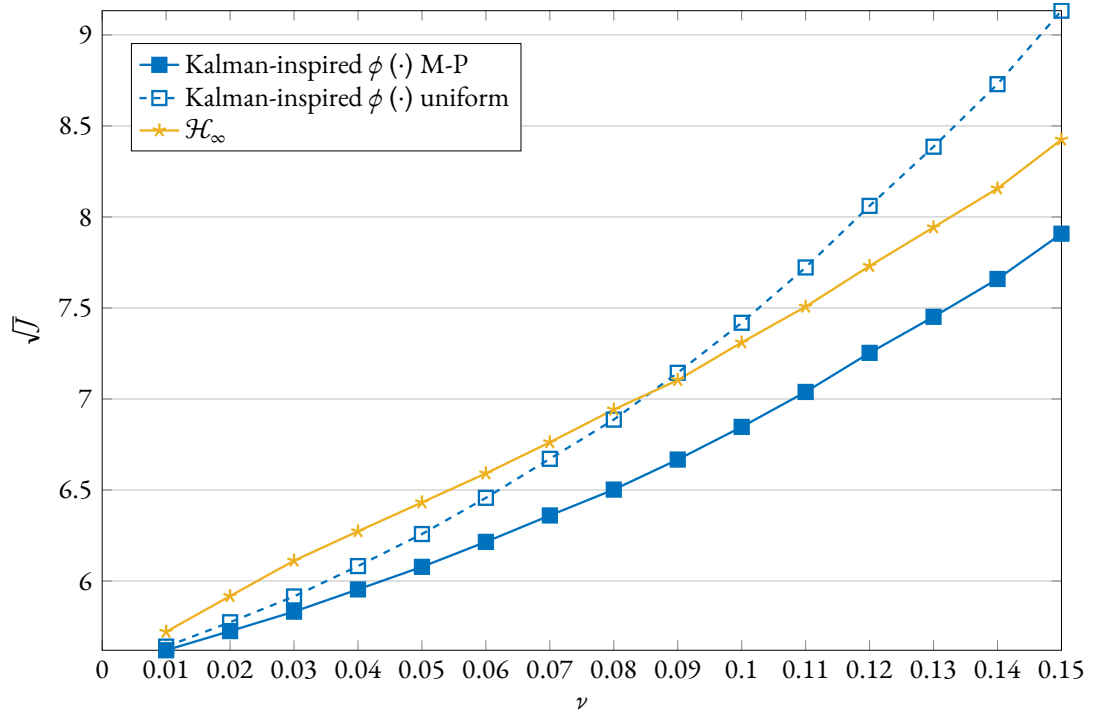


Figure 9.8: Performance comparison of the Kalman-inspired algorithm (Section 9.2.1) with and without correct prior, and the robust $\mathcal{H}_\infty$ algorithm (Section 9.2.3), under noisy linear ramp evolution of $\mathbf{y}'_k$.

Part II

Neural architectures

The objective of Part II is to develop principles for neural network design that are inspired by neuroscience and grounded in dynamical system theory. We begin from biologically motivated structures and interaction rules, formalize these ideas in a mathematical framework, and connect them to contemporary machine learning. With this ordering, we extract mechanisms and constraints relevant to computational neuroscience and identify those that are transferable to artificial systems.

From biological networks to formal models, we first examine properties of neural circuits as described in contemporary neurobiology and biomedicine. Using experimentally informed descriptions of connectivity and dynamics, we build mathematical models that capture essential organizational principles while accepting the necessary simplifications and possible misspecification. The primary aim at this stage is to identify working principles that plausibly underlie network dynamics and behavior in the brain. A secondary aim is to expose parameterizations and structural constraints that can later inform artificial neural architectures.

We then develop neuro-inspired mechanisms for artificial design. Rather than viewing the brain as a purely neuronal network, we leverage recent evidence on the functional role of glia, and *astrocytes* in particular, as modulators that provide slow, context-sensitive control over neuronal populations. We encode key ingredients suggested by this literature (astrocytes slow dynamics, deep interconnection with neuronal circuits, and nonlinear neuronal responses) into a dynamical module that can steer a network toward task-appropriate operating points. The resulting construction naturally supports *multi-task* behavior via contextual guidance, offering a principled alternative to conventional machine-learning schemes.

The connection to modern deep learning is made after the biological and mathematical analyses. In the concluding part of the following chapters, we interpret the proposed neuro-inspired mechanisms through the lens of sequence modeling. Among current architectures, State Space Models (SSMs) are particularly well aligned with this approach: they are intrinsically dynamical, amenable to control-theoretic analysis, and combine efficient training with fast streaming inference. They were proposed as an alternative to Transformers [116] and have performed strongly in this direction. Very recent results [122, 118, 66] also indicate that online optimization mechanisms can be embedded in modern networks to enable in-context (i.e., prompt-driven, online) learning [87]. Building on these insights, we outline how the neuro-inspired modules developed here can interface with SSMs at the macroscale, with the goal of producing architectures that are both theory-grounded and empirically competitive.

The scope and organization of Part II are as follows. The first chapter distills neuroscience-

informed parametrizations for brain connectivity, casting them into a tractable dynamical formalism. The second chapter introduces an artificial module that couples neuronal dynamics with astrocyte-like contextual control, analyzes its properties as a dynamical system, and illustrates how multi-task behavior can emerge from these principles. A concluding bridge then makes the late-stage connection to machine learning by showing how the above modules can be integrated with SSMs and, more broadly, with modern sequence models that implement online adaptation via embedded optimization.

In summary, Part II pursues a bidirectional program: from biology to mathematics (to clarify principles) and from mathematics to machine learning (to inform design).

Part II is mostly based on the following works:

- [NAs1] Casti, U., Zampieri, S., & Pasqualetti, F. (2025). Selection mechanisms for sequence modeling using linear state space models. [arXiv:2505.17932](#).
- [NAs2] Casti, U., Baggio, G., Zampieri, S., & Pasqualetti, F. (2024). Controllable neural architectures for multi-task control. In *2025 European Control Conference (ECC)*.
- [NAs3] Casti, U., Baggio, G., Benozzo, D., Zampieri, S., Bertoldo, A., & Chiuso, A. (2023). Dynamic brain networks with prescribed functional connectivity. In *2023 62nd IEEE Conference on Decision and Control (CDC)* (pp. 709–714).

10

Brain connectivity

Various network models have been proposed in neuroscience to describe brain organization and activity [26, 52]. A commonly used representation is Structural Connectivity (SC), which is grounded in brain anatomy. SC builds a graph whose nodes are brain regions (i.e., populations of strongly interconnected neurons) and whose edges encode anatomical connections between them. While SC has proved extremely useful, its interpretive power is limited by its inherently static nature.

Modern noninvasive imaging techniques such as Functional Magnetic Resonance Imaging (fMRI), Electro-Encephalo-Graphy (EEG), and Magneto-Encephalo-Graphy (MEG) produce time series that quantify the activity of brain regions [24]. Empirically, some regions exhibit correlated activity; this motivates a second type of connectivity, Functional Connectivity (FC), obtained by linking regions that exhibits statistical dependence. Because these correlations vary with task and context, much work has focused on Resting-State Functional Connectivity (rsFC), computed from measurements acquired in the absence of external stimuli [22]. The degree of correlation between regions can be mathematically specified in different ways. One simple and natural way is by means of the statistical correlation between the brain activity signals. In whatever way it is obtained, the FC matrix is always symmetric and hence unable to capture the causal relationship between brain regions.

A third type of connectivity, the Effective Connectivity (EC), explicitly aims to uncover causal interactions. EC is generative: it specifies a dynamical model that, in principle, reproduces the observed signals. Several model classes can be used, an important and tractable

example is the linear stochastic model known as (linear) Dynamic Causal Modeling (DCM) [53]. In this setting, estimating EC amounts to identifying the linear system that best explains the data, a standard system identification problem, see [51, 93, 56].

The starting point of our analysis is a parametrization of the state matrix $\mathbf{A}$ (the EC) of a linear stochastic system in terms of a positive definite matrix $\mathbf{P} \succ 0$, representing its steady-state covariance (the FC), and a skew-symmetric matrix $\mathbf{S} = -\mathbf{S}^\top$. This parametrization arises from decomposing the EC into a symmetric component, determined by the FC and capturing undirected correlation structure among brain areas, and a skew-symmetric component $\mathbf{S}$ that encodes directionality in the EC. We then examine how the “size” of $\mathbf{S}$ influences key dynamical properties of the system. Specifically, we show, through analytical and numerical results, that as $\mathbf{S}$ increases: (i) the stability margin of the system, as measure by the modulus of the largest real part of the eigenvalues of $\mathbf{A}$, grows; (ii) the transient amplification (excitability), quantified by the numerical abscissa of $\mathbf{A}$, increases; and (iii) the spectral energy shifts from low to high frequencies. Altogether, these findings suggest that $\mathbf{S}$ plays a crucial role in shaping brain dynamics.

A few works in neuroscience tried to extract, interpret, and analyze information from EC that goes beyond functional relationships between brain regions. In particular, in [54] the authors use the skew-symmetric part of the EC matrix to build a hierarchy between brain regions. More closely related to our work are [76, 37], which propose the Differential Covariance (DC) as a tool to infer directionality of interactions between brain regions. The DC is a matrix whose entries describe the correlation between the activity of a brain area and the variation of the activity occurring in another one. As it will be clear later, the DC matrix is strongly related to the matrix $\mathbf{S}$ investigated in this chapter.

10.1 PROBLEM FORMULATION

We consider the continuous-time linear time-invariant stochastic system

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{w}(t), \quad (10.1)$$

where $\mathbf{x} \in \mathbb{R}^n$ is the vector containing the states of the network nodes, $\mathbf{A} \in \mathbb{R}^{n \times n}$ is the state interaction matrix, and $\mathbf{w}$ is a zero-mean white noise vector with positive definite covariance matrix $\mathbf{R} := \mathbb{E} [\mathbf{w}(t)\mathbf{w}^\top(t)] \succ 0$.*

*Rigorously speaking, the equation (10.1) should be intended in the differential form $d\mathbf{x} = \mathbf{A}\mathbf{x}dt + d\mathbf{w}$, where $d\mathbf{w}$ is a Wiener process, i.e., a process with stationary orthogonal increments, e.g. see [6, Chap. 3, Sec.

When modeling resting-state brain activity, (10.1) is referred to as linear DCM [53], in which $\mathbf{x}$ contains the states of brain regions, $\mathbf{A}$ is the EC matrix encoding causal relationships between brain regions, and $\mathbf{R} = \sigma^2 \mathbf{I}$, $\sigma \in \mathbb{R}$. The DCM model also contains an output equation of the form

$$\mathbf{y}(t) = [\mathbf{h} * \mathbf{x}](t) + \mathbf{v}(t),$$

where $\mathbf{y} \in \mathbb{R}^n$ is the Blood-Oxygen-Level-Dependent (BOLD) signal, $\mathbf{h} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is the hemodynamic function mapping brain activity to the BOLD signal, the symbol $*$ stands for convolution, and $\mathbf{v}$ is white measurement noise.

In this chapter, $\mathbf{A}$ is assumed to be Hurwitz stable. This ensures the existence of a positive definite steady state or stationary state covariance matrix:

$$\mathbf{P} := \lim_{t \rightarrow \infty} \mathbb{E} [\mathbf{x}(t) \mathbf{x}^\top(t)] > 0,$$

which can be computed as the solution of the algebraic Lyapunov equation [6, Thm. 6.1]:

$$\mathbf{A}\mathbf{P} + \mathbf{P}\mathbf{A}^\top + \mathbf{R} = 0. \quad (10.2)$$

In linear DCM, $\mathbf{P}$ can be seen as a measure of functional strength between brain regions since it is related to the steady-state covariance of the output which (up to a normalization) coincides with the FC. Therefore, $\mathbf{P}$ can be thought of as a proxy of the FC.

10.1.1 PARAMETRIZING SYSTEMS WITH PRESCRIBED $\mathbf{P}$

Consider the system (10.1) and define the sets

$$\begin{aligned} \mathcal{A} &:= \{\mathbf{A} \in \mathbb{R}^{n \times n} : \mathbf{A} \text{ Hurwitz stable}\}, \\ \mathcal{P} &:= \{\mathbf{P} \in \mathbb{R}^{n \times n} : \mathbf{P} = \mathbf{P}^\top > 0\}, \\ \mathcal{S} &:= \{\mathbf{S} \in \mathbb{R}^{n \times n} : \mathbf{S} = -\mathbf{S}^\top\}. \end{aligned} \quad (10.3)$$

Our first result provides a parametrization of Hurwitz stable state matrices $\mathbf{A}$ in terms of a pair of matrices, namely a positive definite matrix and a skew-symmetric one.

4].

Theorem 3. *Parametrization of Hurwitz stable $\mathbf{A}$'s* Consider the system (10.1) and the sets defined in (10.3). For any $\mathbf{P} \in \mathcal{P}$ and $\mathbf{S} \in \mathcal{S}$,

$$\mathbf{A} = \left(-\frac{1}{2}\mathbf{R} + \mathbf{S}\right)\mathbf{P}^{-1} \in \mathcal{A}, \quad (10.4)$$

and $\mathbf{P}$ solves (10.2). Further, the map

$$\begin{aligned} f: \mathcal{P} \times \mathcal{S} &\rightarrow \mathcal{A}, \\ (\mathbf{P}, \mathbf{S}) &\mapsto \mathbf{A} = \left(-\frac{1}{2}\mathbf{R} + \mathbf{S}\right)\mathbf{P}^{-1}, \end{aligned} \quad (10.5)$$

is a bijection.

Proof. By construction, the matrices $\mathbf{P}$ and $\mathbf{A}$ in (10.4) satisfy the Lyapunov equation (10.2). Since $\mathbf{P} \succ 0$ and $\mathbf{R} \succ 0$, it follows that $\mathbf{A} \in \mathcal{A}$ [64, Thm. 8.2]. To prove that (10.5) is a bijection observe that, for any $\mathbf{A} \in \mathcal{A}$, there exists a unique $\mathbf{P} \in \mathcal{P}$ satisfying (10.2) [64, Thm. 8.2]. Further, it holds

$$\mathbf{A}\mathbf{P} = \frac{1}{2}(\mathbf{A}\mathbf{P} + \mathbf{P}\mathbf{A}^\top) + \frac{1}{2}(\mathbf{A}\mathbf{P} - \mathbf{P}\mathbf{A}^\top) \quad (10.6)$$

$$= -\frac{1}{2}\mathbf{R} + \mathbf{S}, \quad (10.7)$$

where $\mathbf{S} := \frac{1}{2}(\mathbf{A}\mathbf{P} - \mathbf{P}\mathbf{A}^\top) \in \mathcal{S}$. Thus, for any $\mathbf{A} \in \mathcal{A}$ there exists $(\mathbf{P}, \mathbf{S}) \in \mathcal{P} \times \mathcal{S}$ such that (10.4) holds, that is, (10.5) is surjective. Finally, consider $\mathbf{A}_1, \mathbf{A}_2 \in \mathcal{A}$ and let $\mathbf{P}_1, \mathbf{P}_2 \in \mathcal{P}$ be the corresponding solutions to (10.2) and $\mathbf{S}_1 := \frac{1}{2}(\mathbf{A}_1\mathbf{P}_1 - \mathbf{P}_1\mathbf{A}_1^\top)$, $\mathbf{S}_2 := \frac{1}{2}(\mathbf{A}_2\mathbf{P}_2 - \mathbf{P}_2\mathbf{A}_2^\top)$. If $\mathbf{A}_1 = \mathbf{A}_2$, then $\mathbf{P}_1 = \mathbf{P}_2$ by the uniqueness of the solution to (10.2) and, from (10.7), $\mathbf{A}_1\mathbf{P}_1 = \mathbf{A}_2\mathbf{P}_2$ implies $\mathbf{S}_1 = \mathbf{S}_2$. This proves injectivity of (10.5) and concludes the proof. $\square$

As a consequence of the previous result, all state matrices of (10.1) generating a fixed steady state covariance $\mathbf{P} \succ 0$ are parametrized by the skew-symmetric matrix $\mathbf{S}$ of Equation (10.4). Further, for a fixed $\mathbf{A} \in \mathcal{A}$ in (10.1) it is possible to obtain the corresponding parameters $\mathbf{P}$ and $\mathbf{S}$ as follows: $\mathbf{P}$ corresponds to the (unique) solution to (10.2) and $\mathbf{S}$ to the skew-symmetric part of $\mathbf{A}\mathbf{P}$, i.e., $\mathbf{S} = \frac{1}{2}(\mathbf{A}\mathbf{P} - \mathbf{P}\mathbf{A}^\top)$.

Remark 12 (Statistical interpretation of $\mathbf{S}$). *The skew-symmetric matrix $\mathbf{S}$ in (10.5) can be*

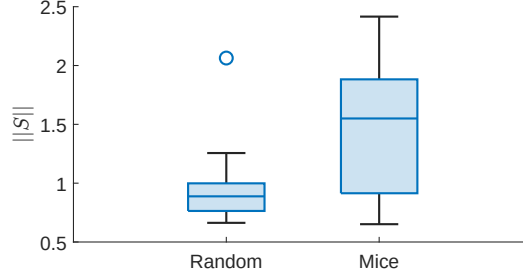


Figure 10.1: The figure shows the norm of $\mathbf{S}$ in the parametrization (10.4) for mice ECs matrices and randomly generated Hurwitz stable matrices. The entries of the random matrices are drawn independently from a standard normal distribution. To enable a fair comparison with the mice ECs, these matrices are shifted and rescaled so that they possess the same maximum real part (stability margin) and imaginary part of the eigenvalues of mice ECs (averaged over the N subjects). The box plot shows the results obtained for 20 realizations of the random ensemble and 20 mice subjects with $\mathbf{R} = \mathbf{I}$.

also characterized from a statistical viewpoint. In fact, it can be shown that

$$\lim_{t \rightarrow \infty} \mathbb{E} [\dot{\mathbf{x}}(t) \mathbf{x}^\top(t)] = \mathbf{A}\mathbf{P} = -\frac{1}{2}\mathbf{R} + \mathbf{S},$$

The latter equation yields

$$\mathbf{S} = \frac{1}{2}\mathbf{R} + \lim_{t \rightarrow \infty} \mathbb{E} [\dot{\mathbf{x}}(t) \mathbf{x}^\top(t)].$$

In [76] the matrix $\lim_{t \rightarrow \infty} \mathbb{E} [\dot{\mathbf{x}}(t) \mathbf{x}^\top(t)]$ is termed *Differential Covariance (DC)*. Note in particular that when $\mathbf{R}$ is diagonal (as in the DCM model) the off-diagonal entries of $\mathbf{S}$ coincide with those of the DC.

Example 2 (Matrices $\mathbf{S}$ in EC mice data). We consider the experimental EC matrices employed in [15] which are estimated from BOLD recordings in mice. In Fig. 10.1, we plot the norm of the $\mathbf{S}$ matrix constructed from the parametrization in Eq. (10.4) for a sample of $N = 20$ subjects, and compare it with an $\mathbf{S}$ extracted from randomly generated stable matrices (see the caption of Fig. 10.1 for details). The mice ECs exhibit a norm of $\mathbf{S}$ substantially larger than that of the random case. In the next sections, we elucidate the role played by a “large” $\mathbf{S}$ on some relevant dynamic properties of the system. $\triangle$

10.2 ROLE OF $\mathbf{S}$ IN THE STABILITY OF LINEAR DCM

We consider the effect of large perturbations on the skew-symmetric term $\mathbf{S}$ in (10.4) on the stability of $\mathbf{A}$. We first note that, for $\mathbf{S} = 0$, the eigenvalues of $\mathbf{A}$ are real since $\mathbf{R}\mathbf{P}^{-1}$ is similar

to a symmetric matrix. Further, the average of the eigenvalues is independent of $\mathbf{S}$, namely

$$\frac{1}{n} \operatorname{tr} [\mathbf{A}] = -\frac{1}{2n} \operatorname{tr} [\mathbf{R}\mathbf{P}^{-1}],$$

since $\mathbf{S}\mathbf{P}^{-1}$ is similar to a skew-symmetric matrix and therefore has trace equal to zero. When the entries of $\mathbf{S}$ increase, the eigenvalues of $\mathbf{A}$ become complex and their real part tend to get closer to the average $-\frac{1}{2n} \operatorname{tr} (\mathbf{R}\mathbf{P}^{-1})$, which means that the system becomes more stable, i.e., the largest real part of the eigenvalues of $\mathbf{A}$ becomes more negative. This is clearly seen in the 2-dimensional case, as discussed below.

Example 3 (2-dimensional system). *Consider the parametrization of $\mathbf{A} \in \mathbb{R}^{2 \times 2}$ given in Theorem 3 and let*

$$\mathbf{S} = \alpha \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \alpha \geq 0.$$

We fix $\mathbf{R} = \sigma^2 \mathbf{I}$ and we take $\mathbf{P}^{-1} = \operatorname{diag}(d_1, d_2)$ with $d_1 \geq d_2 > 0$. Then

$$\mathbf{A} = \begin{bmatrix} -\frac{\sigma^2}{2}d_1 & \alpha d_2 \\ -\alpha d_1 & -\frac{\sigma^2}{2}d_2 \end{bmatrix}.$$

The eigenvalues of $\mathbf{A}$ are the roots of its characteristic polynomial

$$p(\lambda) = \lambda^2 + \frac{\sigma^2}{2}(d_1 + d_2)\lambda + \left(\frac{\sigma^4}{4} + \alpha^2\right)d_1d_2.$$

It follows that the eigenvalues are real if $0 \leq \alpha \leq (d_1 - d_2) / \sqrt{d_1 d_2}$, and complex conjugate with identical real part $-\frac{\sigma^2}{4}(d_1 + d_2)$, otherwise. The behavior of the eigenvalues as a function of $\alpha \geq 0$ is illustrated in Fig. 10.2. $\triangle$

The next result provides asymptotic formulas for the eigenvalues of $\mathbf{A}$ when the entries of $\mathbf{S}$ grow, in the general n -dimensional case.

Theorem 4 (Behavior of eigenvalues of $\mathbf{A}$ for large $\mathbf{S}$). *Consider the parametrization of $\mathbf{A}$ given in Theorem 3 and let $\mathbf{S} = \alpha \bar{\mathbf{S}}$, where $\alpha \in \mathbb{R}$ and $\bar{\mathbf{S}} \in \mathcal{S}$, $\bar{\mathbf{S}} \neq 0$. Let $\{\lambda_i\}_{i=1}^n$ be the eigenvalues of $\mathbf{A}$ and μ_i, u_i be the (imaginary) eigenvalues and the (normalized) eigenvectors*

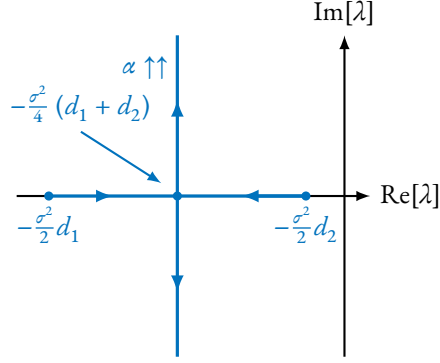


Figure 10.2: Behavior of the eigenvalues of $\mathbf{A}$ for the system in Example 3 and increasing values of $\alpha \geq 0$.

of $\mathbf{P}^{-1/2} \bar{\mathbf{S}} \mathbf{P}^{-1/2}$. Assume that the eigenvalues of $\mathbf{P}^{-1/2} \bar{\mathbf{S}} \mathbf{P}^{-1/2}$ are simple. Then, as $\alpha \rightarrow \infty$,

$$\begin{aligned} \operatorname{Re}[\lambda_i] &\rightarrow -\frac{1}{2} u_i^* \mathbf{P}^{-1/2} \mathbf{R} \mathbf{P}^{-1/2} u_i, \\ \alpha^{-1} \operatorname{Im}[\lambda_i] &\rightarrow \operatorname{Im}[\mu_i]. \end{aligned} \quad (10.8)$$

Proof. Observe that $\mathbf{A}$ is similar to the matrix

$$\begin{aligned} \mathbf{A}' &= -\frac{1}{2} \mathbf{P}^{-1/2} \mathbf{R} \mathbf{P}^{-1/2} + \alpha \mathbf{P}^{-1/2} \bar{\mathbf{S}} \mathbf{P}^{-1/2} \\ &= \frac{1}{\varepsilon} \left(-\frac{\varepsilon}{2} \mathbf{P}^{-1/2} \mathbf{R} \mathbf{P}^{-1/2} + \mathbf{P}^{-1/2} \bar{\mathbf{S}} \mathbf{P}^{-1/2} \right), \end{aligned}$$

with $\varepsilon := \alpha^{-1}$. Consider the matrix $-\frac{\varepsilon}{2} \mathbf{P}^{-1/2} \mathbf{R} \mathbf{P}^{-1/2} + \mathbf{P}^{-1/2} \bar{\mathbf{S}} \mathbf{P}^{-1/2}$ and let $\{\mu_i(\varepsilon)\}$ and $\{u_i(\varepsilon)\}$ be its eigenvalues and corresponding eigenvectors. The latter eigenvalues and eigenvectors are analytic functions of ε in a neighbourhood of $\varepsilon = 0$, since the eigenvalues of $\mathbf{P}^{-1/2} \bar{\mathbf{S}} \mathbf{P}^{-1/2}$ are simple by assumption [79], and we can consider their Taylor expansion around $\varepsilon = 0$:

$$\mu_i(\varepsilon) = \mu_{i,0} + \mu_{i,1}\varepsilon + \dots, \quad u_i(\varepsilon) = u_{i,0} + u_{i,1}\varepsilon + \dots,$$

where $\mu_{i,0} = \mu_i$ is imaginary. Then,

$$\begin{aligned} &\left(-\frac{\varepsilon}{2} \mathbf{P}^{-1/2} \mathbf{R} \mathbf{P}^{-1/2} + \mathbf{P}^{-1/2} \bar{\mathbf{S}} \mathbf{P}^{-1/2} \right) (u_{i,0} + u_{i,1}\varepsilon + \dots) \\ &= (\mu_{i,0} + \mu_{i,1}\varepsilon + \dots) (u_{i,0} + u_{i,1}\varepsilon + \dots). \end{aligned}$$

By equating the terms of the same order in the previous expression, we obtain

$$\begin{aligned}\mathbf{P}^{-1/2}\bar{\mathbf{S}}\mathbf{P}^{-1/2}u_{i,0} &= \mu_{i,0}u_{i,0}, \\ \mathbf{P}^{-1/2}\bar{\mathbf{S}}\mathbf{P}^{-1/2}u_{i,1} - \frac{1}{2}\mathbf{P}^{-1/2}\mathbf{R}\mathbf{P}^{-1/2}u_{i,0} &= \mu_{i,1}u_{i,0} + \mu_{i,0}u_{i,1}.\end{aligned}$$

The first equation is equivalent to $u_{i,0}^*\mathbf{P}^{-1/2}\bar{\mathbf{S}}\mathbf{P}^{-1/2} = \mu_{i,0}u_{i,0}^*$, which substituted in the second one (premultiplied by $u_{i,0}^*$) yields $-\frac{1}{2}u_{i,0}^*\mathbf{P}^{-1/2}\mathbf{R}\mathbf{P}^{-1/2}u_{i,0} = \mu_{i,1}u_{i,0}^*u_{i,0}$. The latter identity in turn implies

$$\begin{aligned}\mu_{i,1} &= -\frac{u_{i,0}^*\mathbf{P}^{-1/2}\mathbf{R}\mathbf{P}^{-1/2}u_{i,0}}{2\|u_{i,0}\|^2} \\ &= -\frac{1}{2}u_{i,0}^*\mathbf{P}^{-1/2}\mathbf{R}\mathbf{P}^{-1/2}u_{i,0} \in \mathbb{R}.\end{aligned}\tag{10.9}$$

Consider now the matrix $\frac{1}{\varepsilon}(-\frac{\varepsilon}{2}\mathbf{P}^{-1/2}\mathbf{R}\mathbf{P}^{-1/2} + \mathbf{P}^{-1/2}\bar{\mathbf{S}}\mathbf{P}^{-1/2})$. As $\varepsilon \rightarrow 0$ the eigenvalues of this matrix satisfy $\frac{1}{\varepsilon}\mu_i(\varepsilon) \rightarrow \frac{1}{\varepsilon}(\mu_{i,0} + \varepsilon\mu_{i,1}) = \mu_{i,1} + \frac{1}{\varepsilon}\mu_{i,0}$. Then,

$$\alpha^{-1}\text{Im}[\lambda_i] \rightarrow \text{Im}[\mu_{i,0}] = \text{Im}[\mu_i].$$

Moreover, since $\mu_{i,0}$ is purely imaginary, as $\varepsilon \rightarrow 0$, from (10.9) we have

$$\begin{aligned}\text{Re}\left[\frac{1}{\varepsilon}\mu_i(\varepsilon)\right] &\rightarrow \text{Re}\left[\frac{1}{\varepsilon}(\mu_{i,0} + \varepsilon\mu_{i,1})\right] = \mu_{i,1} \\ &= -\frac{1}{2}u_{i,0}^*\mathbf{P}^{-1/2}\mathbf{R}\mathbf{P}^{-1/2}u_{i,0}.\end{aligned}$$

The thesis follows by noting that $\varepsilon \rightarrow 0$ is equivalent to $\alpha \rightarrow \infty$ and $\{u_{i,0}\}$ coincide with the (normalized) eigenvectors of $\mathbf{P}^{-1/2}\bar{\mathbf{S}}\mathbf{P}^{-1/2}$. $\square$

Theorem 4 states that while the real parts of the eigenvalues of $\mathbf{A}$ tends to a constant as the size of $\mathbf{S}$ grows, their imaginary part diverge to infinity with rate given by the eigenvalues of $\mathbf{P}^{-1/2}\bar{\mathbf{S}}\mathbf{P}^{-1/2}$. It can also be shown that the limiting values of the real parts of the eigenvalues of $\mathbf{A}$ lie within the interval spanned by the minimum and maximum eigenvalues of $-\frac{1}{2}\mathbf{P}^{-1/2}\mathbf{R}\mathbf{P}^{-1/2}$, that is, the spectrum of $\mathbf{A}$ when $\alpha = 0$ so with a matrix $\mathbf{S}$ of “size” zero. Consequently, the real parts of the eigenvalues of $\mathbf{A}$ remain confined within the bounds determined by the eigenvalues of the matrix associated with $\alpha = 0$. Moreover, numerical simulations suggest that the real parts of the eigenvalues of $\mathbf{A}$ tend to concentrate around

the mean as α increases (see also Figure 10.3 for a 4-dimensional example). This observation typically implies that, as α increases, the stability margin of $\mathbf{A}$ improves (equivalently, $-\max_i \operatorname{Re} \lambda_i(\mathbf{A})$ increases).

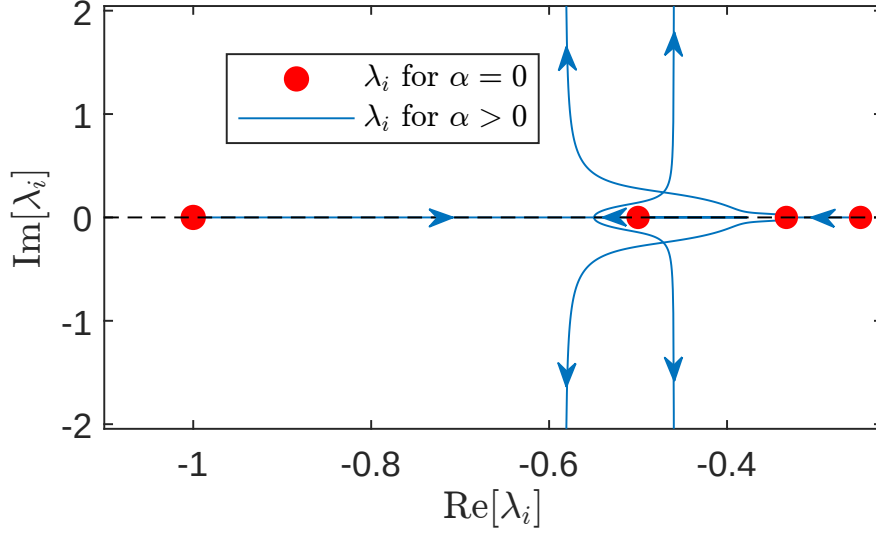


Figure 10.3: Behavior of the eigenvalues of $\mathbf{A}$ for increasing values of $\alpha \geq 0$ for a 4-dimensional system the asymptotic branches corresponds to the limit values of $\operatorname{Re}[\lambda_i]$ of Equation (10.8).

10.3 ROLE OF $\mathbf{S}$ IN THE EXCITABILITY OF LINEAR DCM

While the previous analysis is strictly linked to the asymptotic behavior of the system (10.1), in this section we will focus on a different metric that characterizes the transient evolution of the system. This metric is the numerical abscissa of $\mathbf{A}$, defined as:

$$\omega(\mathbf{A}) := \lambda_{\max} \left(\frac{\mathbf{A} + \mathbf{A}^T}{2} \right).$$

The numerical abscissa quantifies the initial growth rate of the trajectories generated by (10.1). Precisely, it holds [112]:

$$\left. \frac{d\|e^{\mathbf{A}t}\|}{dt} \right|_{t=0} = \omega(\mathbf{A}). \quad (10.10)$$

In view of (10.10), we say that a system is excitable when $\omega(\mathbf{A}) > 0$, and non-excitable otherwise. Excitability has been shown to be beneficial for information transmission and

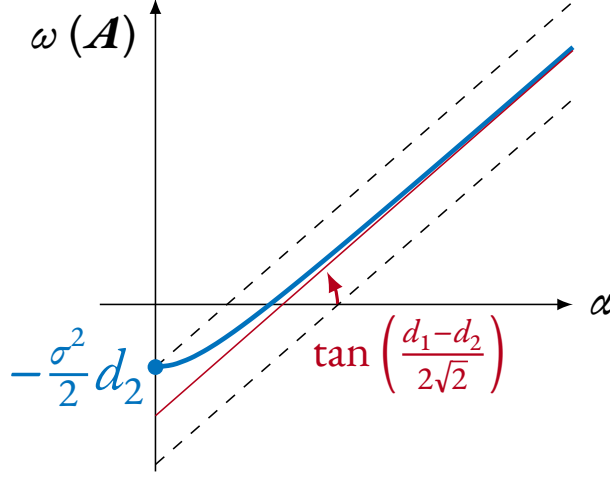


Figure 10.4: Behavior of $\omega(\mathbf{A})$ for increasing values of $\alpha \geq 0$ for the system in Example 4. The dashed lines are the lower and upper bound in Theorem 5.

controllability of linear dynamical systems [10, 9].

The following example provides insights into the relationship between the numerical abscissa of $\mathbf{A}$ and the size of $\mathbf{S}$.

Example 4 (2-dimensional system, cont'd). *Consider the system of Example 3. The numerical abscissa $\omega(\mathbf{A})$ can be expressed in closed form as:*

$$\omega(\mathbf{A}) = -\frac{\sigma^2}{4}(d_1 + d_2) + \frac{\sqrt{\sigma^4 + 2\alpha^2}}{4}(d_1 - d_2).$$

When $d_1 = d_2$ the system is non-excitable for all α . When $d_1 \neq d_2$ it is excitable for $\alpha > \sigma^2 \frac{\sqrt{d_1 d_2}}{d_1 - d_2}$. Further, as α increases, the numerical abscissa grows unbounded as a linear function of α , namely $\omega(\mathbf{A}) \approx \frac{d_1 - d_2}{2\sqrt{2}} \alpha$. $\triangle$

The following theorem generalizes the results of the previous example to the n -dimensional case.

Theorem 5 (Relation between $\omega(\mathbf{A})$ and size of $\mathbf{S}$). *Consider the parametrization of $\mathbf{A}$ given in Theorem 3 and let $\mathbf{S} = \alpha \tilde{\mathbf{S}}$, where $\alpha \in \mathbb{R}$, $\tilde{\mathbf{S}} \in \mathcal{S}$, $\tilde{\mathbf{S}} \neq 0$. It holds*

$$\lambda_{\min}(\mathbf{N}) + \alpha \lambda_{\max}(\mathbf{M}) \leq \omega(\mathbf{A}) \leq \lambda_{\max}(\mathbf{N}) + \alpha \lambda_{\max}(\mathbf{M}), \quad (10.11)$$

where

$$\mathbf{N} := -\frac{1}{4} (\mathbf{R}\mathbf{P}^{-1} + \mathbf{P}^{-1}\mathbf{R}), \quad \mathbf{M} := \frac{1}{2} (\bar{\mathbf{S}}\mathbf{P}^{-1} - \mathbf{P}^{-1}\bar{\mathbf{S}}).$$

Proof. First, note that $\mathbf{N}$ and $\mathbf{M}$ are symmetric, and

$$\frac{\mathbf{A} + \mathbf{A}^\top}{2} = \mathbf{N} + \alpha \mathbf{M}.$$

The bounds in (10.11) now follows by Weyl's inequality [65, Thm 4.3.1], which states that for symmetric matrices $\mathbf{X}, \mathbf{Y}$, $\lambda_{\max}(\mathbf{X} + \mathbf{Y}) \geq \lambda_{\min}(\mathbf{X}) + \lambda_{\max}(\mathbf{Y})$, $\lambda_{\max}(\mathbf{X} + \mathbf{Y}) \leq \lambda_{\max}(\mathbf{X}) + \lambda_{\max}(\mathbf{Y})$. $\square$

From the bounds in Theorem 5, it follows that $\omega(\mathbf{A})$ grows linearly with the size of $\mathbf{S}$, as quantified by the parameter α , if $\lambda_{\max}(\mathbf{M}) > 0$. The latter condition is satisfied if and only if $\mathbf{M} \neq 0$. In fact, the skew-symmetry of $\bar{\mathbf{S}}$ yields $\text{tr}(\mathbf{M}) = 0$, which implies that $\mathbf{M}$ is either the zero matrix or has at least a strictly positive eigenvalue. Fig. 10.4 shows the behavior of $\omega(\mathbf{A})$ together with the bounds in Theorem 5 as a function of α for the 2-dimensional system of Example 4.

To conclude, we note that the lower bound in Theorem 5 yields a sufficient condition on $\mathbf{S}$ to guarantee excitability. Namely, if $\mathbf{M} \neq 0$ and $\alpha > -\lambda_{\min}(\mathbf{P}) / \lambda_{\max}(\mathbf{M})$, then $\omega(\mathbf{A}) > 0$ and the system is excitable.

10.4 $\mathbf{S}$ AND FREQUENCY-DOMAIN BEHAVIOR OF LINEAR DCM

In this section, we examine the effect of $\mathbf{S}$ on the frequency-domain properties of (10.1). Specifically, we analyze how the size of $\mathbf{S}$ affects the energy distribution of the system across different frequencies, as determined by its power spectral density [77, Chap. 10]:

$$\Phi(\omega) = (\omega \mathbf{I} - \mathbf{A})^{-1} \mathbf{R} (\omega \mathbf{I} - \mathbf{A})^{-*}, \quad \omega \in \mathbb{R}.$$

We first observe that $\mathbf{S}$ has no effect on the overall energy of the system since it holds [6, Chap. 5, Sec. 2]

$$\mathbf{P} = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \Phi(\omega) d\omega, \tag{10.12}$$

and the steady-state covariance of the system $\mathbf{P}$ is unaffected by $\mathbf{S}$. However, $\mathbf{S}$ can in principle redistribute the energy across different frequency bands. This is indeed the case for two-dimensional systems, where $\mathbf{S}$ shifts the energy from low to high frequencies, as we illustrate in the next example.

Example 5 (2-dimensional system, cont'd). *Consider the system of Example 3. The trace of the power spectral density of the system is given by*

$$\text{tr} [\Phi(i\omega)] = \sigma^2 \frac{2\omega^2 + \zeta(\alpha^2)}{[\omega^2 - d_1 d_2 \zeta(\alpha^2)]^2 + \omega^2 \frac{\sigma^4}{4} (d_1 + d_2)^2}, \quad (10.13)$$

with $\zeta(\alpha^2) := \alpha^2 + \sigma^4/4$. From (10.13), it follows that for any fixed $\omega \in \mathbb{R}$, $\text{tr} [\Phi(i\omega)] \rightarrow 0$ as $\alpha \rightarrow \infty$, which means that as α grows, the energy of the system moves from low to high frequencies. To better understand how α modifies the shape of $\text{tr} [\Phi(i\omega)]$, we note that (10.13) can be viewed as the modulus of the frequency response of a fictitious second-order scalar system featuring a zero at $-\sqrt{\zeta(\alpha^2)}/2$. As standard practice, we can investigate the emergence of resonance phenomena in the system by neglecting the effect of the zero. That is, we can approximate (10.13) as

$$\text{tr} [\tilde{\Phi}(i\omega)] = \frac{\sigma^2}{[\omega^2 - d_1 d_2 \zeta(\alpha^2)]^2 + \omega^2 \frac{\sigma^4}{4} (d_1 + d_2)^2}. \quad (10.14)$$

Fig. 10.5 shows a comparison between (10.13) and (10.14). Using the approximation in (10.14) and restricting the analysis to nonnegative frequencies, it can be shown, after some calculations, that for $\alpha > \alpha_{\text{th}} := \sigma^2 \sqrt{d_1^2 + d_2^2} / (2\sqrt{2d_1 d_2})$, $\text{tr} [\Phi(i\omega)]$ exhibits a resonance peak at frequency $\omega_r = \sqrt{8d_1 d_2 \alpha^2 - \sigma^4 (d_1^2 + d_2^2)} / (2\sqrt{2})$. We conclude that as α (that is, the size of $\mathbf{S}$) increases the system tends to amplify input signals at one particular frequency, and this frequency grows linearly with α . $\triangle$

The observations made in the previous example can be generalized as follows.

Theorem 6 (High-pass filtering effect of $\mathbf{S}$). *Consider the parametrization of $\mathbf{A}$ given in Theorem 3, with $\mathbf{S} = \alpha \bar{\mathbf{S}}$, $\alpha \in \mathbb{S}$, $\bar{\mathbf{S}} \in \mathbb{S}$, $\bar{\mathbf{S}} \neq 0$. Let*

$$\Phi(i\omega, \alpha) = (i\omega \mathbf{I} - \mathbf{A})^{-1} \mathbf{R} (i\omega \mathbf{I} - \mathbf{A})^{-*}.$$

The following hold:

1. $\exists \bar{\omega} > 0$ s.t. $\text{tr} [\Phi(i\omega, 0)] > \text{tr} [\Phi(i\omega, \infty)]$, $\forall \omega$, $|\omega| \leq \bar{\omega}$;

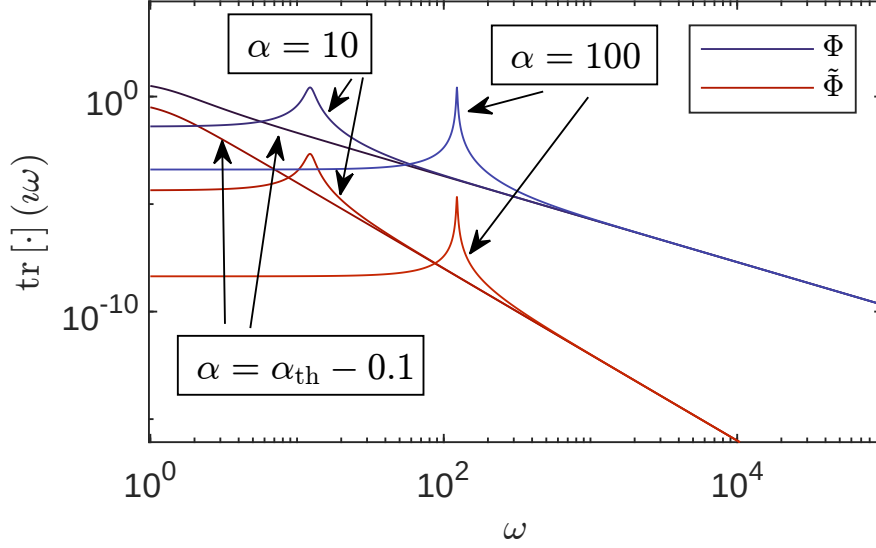


Figure 10.5: Plot in a logarithmic scale of the trace of the frequency response for the $\text{tr} [\Phi(j\omega)]$ and its approximate version $\text{tr} [\tilde{\Phi}(j\omega)]$ for different values of α , in particular $\alpha = \alpha_{\text{th}} - 0.1, 10, 100$, and $d_1 > d_2 > 0$ have been generated random between 0 and 10. α_{th} denotes the threshold at which the system exhibits resonance.

2. If $\tilde{\mathbf{S}}$ is full rank, $\text{tr} [\Phi(j\omega, \alpha)] \rightarrow 0$ as $\alpha \rightarrow \infty, \forall \omega \in \mathbb{R}$.

Proof. Let us define $\mathbf{F}(j\omega, \alpha) := (j\omega \mathbf{I} - \mathbf{A}_\alpha)^* \mathbf{R}^{-1} (j\omega \mathbf{I} - \mathbf{A}_\alpha)$ so that $\Phi(j\omega, \alpha) = \mathbf{F}(j\omega, \alpha)^{-1}$. Then, it holds

$$\mathbf{F}(j\omega, \alpha) = \frac{\mathbf{P}^{-1} \mathbf{R} \mathbf{P}^{-1}}{4} + \tilde{\mathbf{F}}(j\omega, \alpha), \quad (10.15)$$

where

$$\tilde{\mathbf{F}}(j\omega, \alpha) := \omega^2 \mathbf{R}^{-1} + \omega \alpha (\mathbf{R}^{-1} \tilde{\mathbf{S}} \mathbf{P}^{-1} + \mathbf{P}^{-1} \tilde{\mathbf{S}} \mathbf{R}^{-1}) + \alpha^2 \mathbf{P}^{-1} \tilde{\mathbf{S}}^\top \mathbf{R}^{-1} \tilde{\mathbf{S}} \mathbf{P}^{-1}. \quad (10.16)$$

By setting $\omega = 0$, (10.15) becomes

$$\mathbf{F}(0, \alpha) = \frac{\mathbf{P}^{-1} \mathbf{R} \mathbf{P}^{-1}}{4} + \alpha^2 \mathbf{P}^{-1} \tilde{\mathbf{S}}^\top \mathbf{R}^{-1} \tilde{\mathbf{S}} \mathbf{P}^{-1}. \quad (10.17)$$

Note that $\mathbf{P}^{-1} \tilde{\mathbf{S}}^\top \mathbf{R}^{-1} \tilde{\mathbf{S}} \mathbf{P}^{-1} \geq 0$ since $\mathbf{R}, \mathbf{P} > 0$. This implies that, for all $\alpha > 0$, $\mathbf{F}(0, \alpha) \geq \mathbf{F}(0, 0) > 0$, or, equivalently, $\mathbf{F}(0, 0)^{-1} \geq \mathbf{F}(0, \alpha)^{-1} > 0$. In addition, if $\alpha > 0$, there is at

least one eigenvalue of $\mathbf{F}(0, 0)^{-1}$ that is not an eigenvalue of $\mathbf{F}(0, \alpha)^{-1}$.^{*} Therefore, for all $\alpha > 0$, $\text{tr}[\mathbf{F}(0, 0)^{-1}] > \text{tr}[\mathbf{F}(0, \alpha)^{-1}]$. Point 1) now follows from the latter inequality and the continuity of $\Phi(\omega, \alpha)$ with respect to ω .

To prove point 2), we introduce the following notation:

$$\begin{aligned}\tilde{a} &:= \lambda_{\min}(\mathbf{P}^{-1}\mathbf{R}\mathbf{P}^{-1}), \tilde{b} := \lambda_{\min}(\mathbf{R}^{-1}), \\ \tilde{c} &:= \lambda_{\min}(\mathbf{P}^{-1}\tilde{\mathbf{S}}^{\top}\mathbf{R}^{-1}\tilde{\mathbf{S}}\mathbf{P}^{-1}), \tilde{d} := \|\mathbf{P}^{-1}\tilde{\mathbf{S}}\mathbf{R}^{-1}\|,\end{aligned}$$

where $\tilde{a}, \tilde{b}, \tilde{c}, \tilde{d} > 0$ from the positive definiteness of $\mathbf{P}, \mathbf{R}$, and the full rank condition of $\tilde{\mathbf{S}}$. The following hold:

$$\mathbf{P}^{-1}\mathbf{R}\mathbf{P}^{-1} \succeq \tilde{a}\mathbf{I}, \quad (10.18)$$

$$\mathbf{R}^{-1} \succeq \tilde{b}\mathbf{I}, \quad (10.19)$$

$$\mathbf{P}^{-1}\tilde{\mathbf{S}}^{\top}\mathbf{R}^{-1}\tilde{\mathbf{S}}\mathbf{P}^{-1} \succeq \tilde{c}\mathbf{I}. \quad (10.20)$$

It remains to bound the term $\omega\iota(\mathbf{R}^{-1}\tilde{\mathbf{S}}\mathbf{P}^{-1} + \mathbf{P}^{-1}\tilde{\mathbf{S}}\mathbf{R}^{-1})$ in (10.16). First, we see that $\mathbf{R}^{-1}\tilde{\mathbf{S}}\mathbf{P}^{-1} + \mathbf{P}^{-1}\tilde{\mathbf{S}}\mathbf{R}^{-1}$ is skew-symmetric so that $\iota(\mathbf{R}^{-1}\tilde{\mathbf{S}}\mathbf{P}^{-1} + \mathbf{P}^{-1}\tilde{\mathbf{S}}\mathbf{R}^{-1})$ is an Hermitian matrix. So it holds

$$\iota(\mathbf{R}^{-1}\tilde{\mathbf{S}}\mathbf{P}^{-1} + \mathbf{P}^{-1}\tilde{\mathbf{S}}\mathbf{R}^{-1}) \preceq 2\tilde{d}\mathbf{I}. \quad (10.21)$$

We also know that the eigenvalues of $\iota(\mathbf{R}^{-1}\tilde{\mathbf{S}}\mathbf{P}^{-1} + \mathbf{P}^{-1}\tilde{\mathbf{S}}\mathbf{R}^{-1})$ are symmetric with respect to zero. Thus, (10.21) yields the bounds $-2\tilde{d}\mathbf{I} \preceq \iota(\mathbf{R}^{-1}\tilde{\mathbf{S}}\mathbf{P}^{-1} + \mathbf{P}^{-1}\tilde{\mathbf{S}}\mathbf{R}^{-1}) \preceq 2\tilde{d}\mathbf{I}$, which in turn implies

$$\omega\iota(\mathbf{R}^{-1}\tilde{\mathbf{S}}\mathbf{P}^{-1} + \mathbf{P}^{-1}\tilde{\mathbf{S}}\mathbf{R}^{-1}) \succeq -2|\omega|\tilde{d}\mathbf{I}. \quad (10.22)$$

By applying (10.19), (10.20) and (10.22) to $\tilde{\mathbf{F}}(\omega, \alpha)$ in (10.15), we obtain

$$\begin{aligned}\tilde{\mathbf{F}}(\omega, S) &\succeq \left[\omega^2\tilde{b} - 2|\omega|\alpha\tilde{d} + \alpha^2\tilde{c} \right] \mathbf{I} \\ &= \underbrace{\tilde{b} \left[\omega^2 - 2|\omega|\alpha\frac{\tilde{d}}{\tilde{b}} + \alpha^2\frac{\tilde{c}}{\tilde{b}} \right]}_{=: \tilde{f}(\omega, \alpha)} \mathbf{I}.\end{aligned} \quad (10.23)$$

^{*}Indeed, assume by contradiction that $\mathbf{F}(0, 0)^{-1}$ and $\mathbf{F}(0, \alpha)^{-1}$ have the same eigenvalues. This implies that $\mathbf{F}(0, 0)$ and $\mathbf{F}(0, \alpha)$ have the same eigenvalues, and therefore the same trace. However, from (10.17), if $\alpha > 0$, $\text{tr}[\mathbf{F}(0, \alpha)] > \text{tr}[\mathbf{F}(0, 0)]$ since $\tilde{\mathbf{S}} \neq 0$.

Let $c := \tilde{c}/\tilde{b}$, $d := \tilde{d}/\tilde{b}$ and fix any $\bar{\omega} > 0$. Then, for all $|\omega| \leq \bar{\omega}$ and $\alpha \geq \bar{\alpha} := 4d\bar{\omega}/c$ we have that:

$$\begin{aligned} f(\omega, \alpha) &= \omega^2 - 2|\omega|\alpha d + \alpha^2 c \\ &\geq -2|\omega|\alpha d + \alpha^2 c = \alpha^2 \left[c - 2\frac{|\omega|}{\alpha}d \right] \geq \alpha^2 \left[c - \frac{c}{2} \right] = \alpha^2 \frac{c}{2}, \end{aligned}$$

where the last inequality follows from $|\omega|/\alpha \leq \bar{\omega}/\bar{\alpha} = c/(4d)$. From the latter bound and (10.23),

$$\tilde{\mathbf{F}}(\omega, \alpha) \geq \tilde{b} \alpha^2 \frac{c}{2} \mathbf{I}. \quad (10.24)$$

By substituting (10.24) and (10.18) in (10.15) we get

$$\mathbf{F}(\omega, \alpha) \geq \left[\frac{\tilde{a}}{4} + \tilde{b} \alpha^2 \frac{c}{2} \right] \mathbf{I} \quad \text{if } |\omega| \leq \bar{\omega}.$$

Point 2) now follows from the latter inequality and the fact that $\bar{\omega}$ can be chosen arbitrarily. $\square$

Loosely speaking, Theorem 6 says that when $\mathbf{S}$ is large the system attenuates inputs at low frequencies. In particular, if $\mathbf{S}$ has full rank and sufficiently large α , low-frequency inputs are completely blocked by the system. Moreover, since the overall spectral energy is preserved as $\mathbf{S}$ varies (cf. Eq. (10.12)), Theorem 6 implies that as $\mathbf{S}$ grows the spectral energy of the system moves from low to high frequencies.

10.5 DISCUSSION

In this chapter, we analyze the properties of stochastic linear systems with a prescribed steady-state covariance. These systems are used to model dynamic brain networks that generate a fixed rsFC. We show that the state matrix of these systems can be parametrized by a skew-symmetric matrix $\mathbf{S}$, and we study the role of $\mathbf{S}$ in the stability, excitability, and frequency-domain behavior of the system. Through theoretical and numerical results, we find that, as the entries of $\mathbf{S}$ grow, the system becomes more stable, more excitable, and more similar to a high-pass filter.

There are several promising directions for future research. In particular, it would be interesting to examine in depth the statistical interpretation of $\mathbf{S}$ in Remark 12 and to analyze how $\mathbf{S}$ affects other system properties (e.g., controllability). In addition, there is a strong

connection with modern ideas from the machine-learning community, which we discuss next.

CONNECTIONS TO MACHINE LEARNING AND SSMs. We highlight a particularly relevant link to modern deep-learning architectures, namely SSMs. SSMs are sequence models that operate with a hidden state and are widely used for language modeling, time-series forecasting, and other sequential tasks. A convenient continuous-time formulation is the Recurrent Neural Network (RNN)

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{f}_{\mathbf{w}}(\mathbf{x}(t), \mathbf{u}(t), t), \\ \mathbf{y}(t) &= \mathbf{g}_{\mathbf{w}'}(\mathbf{x}(t), \mathbf{u}(t), t),\end{aligned}\tag{10.25}$$

where $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^n$ and $\mathbf{g} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^p$ are parametric mappings, $\mathbf{w} \in \mathbb{R}^\ell$ and $\mathbf{w}' \in \mathbb{R}^\ell$ denote trainable weights, $\mathbf{u}(t) \in \mathbb{R}^m$ is the input, $\mathbf{x}(t) \in \mathbb{R}^n$ is the hidden state, and $\mathbf{y}(t) \in \mathbb{R}^p$ is the output signal. In deep learning, one seeks $\mathbf{w}, \mathbf{w}'$ so that the induced input-output map $\mathbf{u} \mapsto \mathbf{y}$ models the target sequence distribution. Training such models is challenging, especially when long-range dependencies are present.

Notably, SSMs, that are special cases of RNNs, are in principle less expressive than fully nonlinear recurrences, yet they substantially simplify the training pipeline and often yield models that outperform standard RNNs and, in some settings, even Transformer-based architectures [116]. This underscores a key property of these models: they are *simple yet sufficiently expressive*, and, crucially for our purposes, the SSM formalism aligns exactly with our mathematical model of brain dynamics (10.1), making our analysis directly applicable to these architectures. Although RNNs are typically implemented in discrete time, several modern SSMs are first developed in continuous time and subsequently discretized; this discretization is not merely a technicality but an explicit mechanism to embed *selectivity* (e.g., via input-dependent or time-varying updates, as in Mamba [59]). This perspective is closely aligned with the framework proposed in this chapter.

Beyond their neuroscientific interest, our parametrization (10.4) provides principled initializations for the SSM state-transition matrix. Prior work (e.g., [60]) showed that SSM performance is sensitive to careful initialization (High-order Polynomial Projection Operators (HiPPO)), motivated by an online function-approximation objective that preserves past information efficiently. Complementarily, brain connectivity suggests three qualitatively different parameterizations for the transition operator: structural connectivity (SC), functional connectivity (FC), and effective connectivity (EC). SC is typically asymmetric and anatomically grounded but does not encode mutual influence; FC captures correlations yet

loses causal directionality due to symmetry; EC is sensitive to causal effects and, under our parametrization (see (10.4)), inherits the desirable dynamical properties established in this chapter.

These observations point to initialization schemes that emphasize larger $\mathcal{S}$ in (10.1), thereby promoting system properties associated with strong (and possibly nonnormal) transient amplification. In particular, initializations inducing high nonnormality [111], close to Jordan-block structures, support *eidetic* (long-lived) memory [122], whereas smaller values of $\mathcal{S}$ bias the dynamics toward fading memory with predominantly real modes and trajectories nearer the stability boundary. Such connectivity-informed initializations offer a neuroscience-grounded way to tune SSM training: since these models are exactly described by (10.1), the resulting initialization is natural within our framework and integrates easily with modern SSM implementations.

11

Multi-task architectures

This chapter concerns the study of neuro-inspired models for *multi-task* approximation, a relevant problem in machine learning. Recent advances have highlighted the benefits of transferring knowledge across tasks to accelerate training and reduce computational resources [80, 107, 121, 42], while also enhancing the ability of neural networks to solve broad, multidisciplinary problems. We investigate how to embed neuroscience concepts to improve adaptability. Whereas natural systems seamlessly adapt across tasks and transfer learned skills to novel contexts, machine-learning pipelines are often trained for a single dataset and objective and typically require costly retraining to accommodate changes in data or task specifications [14, 31]. In neuroscience, for instance, astrocytes are believed to bias neuronal function, enabling contextual modulation without altering synaptic wiring [57]. Motivated by this discrepancy between biological and artificial intelligence, multi-task learning methods have demonstrated that a single artificial architecture can learn to solve multiple tasks [80, 121]; nevertheless, broadly applicable methods with theoretical guarantees remain limited.

We propose a nonlinear, dynamic, brain-inspired *multi-task* approximator tailored to dynamical tasks that integrates these principles from neuroscience and machine learning. Specifically, we introduce a recurrent neural architecture, inspired by [119], equipped with a (smooth) ReLU activation and a *shared* state-connectivity matrix trained once at design time. Task adaptation is achieved via a finite collection of bias vectors (astrocyte-like modulation), each trained offline and selected online according to the task or context (possibly

integrating an online optimization procedure presented in Part I). This design mirrors biological contextual modulation and enables rapid task switching without retraining the core architecture. Accordingly, we formulate task approximation for the unified architecture as a *dynamical-systems response-approximation* problem: each linearized dynamics of the RNN encodes a task’s behavior. We expose the control-theoretic structure and ground the analysis in established control and dynamical-systems metrics.

The literature on multi-task learning in the context of control is relatively scarce. Classical design methods, such as Linear Quadratic Regulation (LQR) and Model Predictive Control (MPC) [2, 71], are tailored to a single dynamical system and require a complete redesign when the system changes. Adaptive control techniques [7] allow a controller to adjust to a changing plant; however, the controller is typically tuned for one plant at a time, and plant variations are usually assumed to occur smoothly and gradually. Similarly, robust control methods [126] guarantee applicability over a family of systems, but these families are generally restricted to sufficiently similar dynamics; moreover, broader applicability often comes at the expense of optimality for any particular system [1]. Recent studies on multi-task control include [120, 36], which focus on system identification with multiple datasets, and [123, 62], which address transfer and imitation control. By contrast, this chapter considers a setting in which the target systems may be unrelated and can span arbitrarily different domains; to the best of our knowledge, no existing results address this case.

In machine learning, multi-task methods enable a single Artificial Neural Network (ANN) to solve different problems. For example, [80] employs masking to select the subnetwork that solves a desired task. A well-known challenge is *catastrophic forgetting*, whereby the network forgets previously learned tasks when exposed to new data [94, 88]. While the literature on multi-task learning is extensive [80, 107, 121, 42], most approaches emphasize mechanisms that are effectively *static* from a control viewpoint and do not leverage control- and systems-theoretic tools for *dynamic* task adaptation.

The work closest to ours is [67], which studies the simultaneous approximation of multiple systems using a single approximating model. In contrast, this chapter allows for multiple approximating systems that are coupled through the nonlinear RNN structure and may have dimensions different from those of the target systems. Because the implemented approximations cannot be selected independently, the LMI methods developed in [67] do not apply directly to our setting.

Finally, the neuroscience inspiration behind this chapter is closely related to the integration of astrocytic and neuronal functions in the human brain [61, 110, 84]. Although the

functional roles of astrocytes are still being elucidated, compelling theories suggest that astrocytes bias neuronal activity and enable the seamless adaptation of a mostly static neuronal network (on relevant time scales) to different contexts and tasks [57]. In our architecture, the bias vector plays an analogous role by providing contextual adaptation capabilities.

11.1 PROBLEM FORMULATION & ARCHITECTURE STRUCTURE

Consider the following multi-task neural architecture:

$$\Sigma : \begin{cases} \dot{\mathbf{x}}(t) = -\mathbf{x}(t) + \boldsymbol{\Phi}(\mathbf{W}\mathbf{x}(t) + \mathbf{d}) + \mathbf{B}\mathbf{u}(t), \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t), \end{cases} \quad (11.1)$$

where $\mathbf{x} \in \mathbb{R}^N$, $\mathbf{d} \in \mathbb{R}^N$, $\mathbf{u} \in \mathbb{R}^m$, and $\mathbf{y} \in \mathbb{R}^p$ are the state, a free parameter, the input, and the output of the network, respectively; $\mathbf{W} \in \mathbb{R}^{N \times N}$, $\mathbf{B} \in \mathbb{R}^{N \times m}$, $\mathbf{C} \in \mathbb{R}^{p \times N}$, and the vector $\mathbf{d}$ are the trainable parameters.

The activation function $\boldsymbol{\Phi} : \mathbb{R}^N \rightarrow \mathbb{R}_{>0}^N$ applies, elementwise, the scalar nonlinearity $\phi : \mathbb{R} \rightarrow \mathbb{R}_{>0}$, here chosen as the softplus

$$\phi(x) = \ln(1 + e^x),$$

a smooth approximation of ReLU.*

When $\mathbf{u} = 0$, the equilibria of (11.1) satisfy

$$\mathbf{x}_{\text{eq}} = \boldsymbol{\Phi}(\mathbf{W}\mathbf{x}_{\text{eq}} + \mathbf{d}), \quad (11.2)$$

and, locally, obey the linearized dynamics

$$\Sigma^{\text{L}} : \left(-\mathbf{I} + \underbrace{\text{diag}(\boldsymbol{\Phi}_{\text{d}}(\mathbf{W}\mathbf{x}_{\text{eq}} + \mathbf{d}))}_{\mathbf{D}} \mathbf{W}, \mathbf{B}, \mathbf{C} \right), \quad (11.3)$$

where $\boldsymbol{\Phi}_{\text{d}} : \mathbb{R}^N \rightarrow (0, 1)^N$ applies elementwise the derivative

$$\phi_{\text{d}}(x) = \frac{d}{dx}\phi(x) = \frac{1}{1 + e^{-x}}.$$

*Alternative activation functions could be used; here we select the softplus activation for its connections to computational neuroscience [46].

Theorem 7 (Parametrization via $\mathbf{d}$). *For any matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$ and any vector $\bar{\mathbf{d}} \in (0, 1)^N$, there exist $\mathbf{x}_{\text{eq}}$ and $\mathbf{d}$ that satisfy (11.2) and $\Phi_{\mathbf{d}}(\mathbf{W}\mathbf{x}_{\text{eq}} + \mathbf{d}) = \bar{\mathbf{d}}$.*

Proof. The function $\phi_{\mathbf{d}}$ is strictly increasing and hence injective, so its inverse is well defined. Let

$$\mathbf{x}_{\text{eq}} = \Phi(\Phi_{\mathbf{d}}^{-1}(\bar{\mathbf{d}})), \quad \mathbf{d} = \Phi_{\mathbf{d}}^{-1}(\bar{\mathbf{d}}) - \mathbf{W}\Phi(\Phi_{\mathbf{d}}^{-1}(\bar{\mathbf{d}})).$$

Then, using (11.2),

$$\mathbf{x}_{\text{eq}} = \Phi(\mathbf{W}\Phi(\Phi_{\mathbf{d}}^{-1}(\bar{\mathbf{d}})) + \Phi_{\mathbf{d}}^{-1}(\bar{\mathbf{d}}) - \mathbf{W}\Phi(\Phi_{\mathbf{d}}^{-1}(\bar{\mathbf{d}}))) = \Phi(\Phi_{\mathbf{d}}^{-1}(\bar{\mathbf{d}})) = \mathbf{x}_{\text{eq}},$$

and

$$\Phi_{\mathbf{d}}(\mathbf{W}\Phi(\Phi_{\mathbf{d}}^{-1}(\bar{\mathbf{d}})) + \Phi_{\mathbf{d}}^{-1}(\bar{\mathbf{d}}) - \mathbf{W}\Phi(\Phi_{\mathbf{d}}^{-1}(\bar{\mathbf{d}}))) = \bar{\mathbf{d}}.$$

□

Theorem 7 shows that there exist a vector $\mathbf{d}$ and an equilibrium $\mathbf{x}_{\text{eq}}$ that realize any desired matrix $\mathbf{D}$ in (11.3). Hence, in what follows, we derive conditions and algorithms for the matrix $\mathbf{D}$, with the understanding that such a matrix can ultimately be realized by choosing the vector $\mathbf{d}$.

We are now ready to formalize the structure of the proposed multi-task neural architecture. In particular, assume a set of M distinct, stable, controllable, and observable LTI systems (the desired task solvers), considered given,

$$\Sigma_i^{\text{D}} : (\mathbf{A}_i, \mathbf{B}_i, \mathbf{C}_i), \quad i = 1, \dots, M, \quad (11.4)$$

with $\mathbf{A}_i \in \mathbb{R}^{n \times n}$, $\mathbf{B}_i \in \mathbb{R}^{n \times m}$, and $\mathbf{C}_i \in \mathbb{R}^{p \times n}$. The goal of the multi-task neural architecture is to combine the knowledge of different task solvers within a single efficient approximating model. We formalize this objective via

$$\underset{\mathbf{W}, \{\mathbf{D}_i\}_{i=1}^M, \mathbf{B}, \mathbf{C}}{\text{minimize}} \quad \underbrace{\sum_{i=1}^M \|\Sigma_i^{\text{D}} - \Sigma_i^{\text{L}}\|_2^2}_{\mathcal{J}}, \quad (11.5)$$

where Σ_i^{L} denotes the i -th linearized dynamics in (11.3) with diagonal matrix $\mathbf{D}_i$, and $\|\cdot\|_2$ is the $\mathcal{H}_2$ norm. In (11.5), the optimization variables determine how the multi-task neural architecture (11.1) approximates the desired task solvers (11.4) locally around its equilibrium points.

Remark 13. *A point worth emphasizing is that the multi-task neural architecture is intended, in principle, as a memory-efficient mechanism to store and combine the capabilities of the task solvers (11.4). Storing M distinct task solvers (11.4) requires $M(n^2 + n(m + p))$ parameters to specify the matrices $\mathbf{A}_i$, $\mathbf{B}_i$, and $\mathbf{C}_i$. In contrast, the multi-task neural architecture (11.1) uses only $MN + N^2 + N(m + p)$ parameters, making it increasingly efficient as M grows, provided $N < n(n + m + p)$.*

Remark 14. *Although Problem (11.5) is presented under the simplifying assumption that all systems in (11.4) share the same state dimension n , this choice is primarily for notational convenience. The theoretical developments, including the gradient expressions in Section 11.2, extend straightforwardly to heterogeneous collections with state dimensions n_i , after adjusting block sizes accordingly.*

The approximation error in (11.5) depends in a nontrivial way on several factors, including the dimension N of the multi-task architecture, the number M of task solvers, and the diversity among the systems to be merged. In the next sections we derive a closed-form expression for the gradient required to implement an efficient gradient-descent solver for (11.5) (Section 11.2). Because the loss in (11.5) is nonstandard, the analytical gradient substantially accelerates training. We also develop a theoretical analysis of fundamental limitations and achievable performance for the resulting multi-task neural architecture.

11.2 NUMERICAL ANALYSIS

This section derives analytical expressions for the gradient of the objective (11.5) with respect to the parameters of the multi-task neural architecture, and evaluates the approximation capabilities of the proposed model through numerical experiments.

The resulting gradients are used for efficient numerical optimization. Direct (closed-form) gradients significantly improve accuracy and speed compared to gradient approximation methods. To this end, define the error system and its block matrices as

$$\Sigma_{\text{err}}^i : (\mathbf{A}_{\text{err}}^i, \mathbf{B}_{\text{err}}^i, \mathbf{C}_{\text{err}}^i), \quad i = 1, \dots, M, \quad (11.6)$$

with

$$\begin{aligned}\mathbf{A}_{\text{err}}^i &= \begin{bmatrix} \mathbf{A}_i & 0 \\ 0 & -\mathbf{I} + \mathbf{D}_i \mathbf{W} \end{bmatrix}, \quad \mathbf{B}_{\text{err}}^i = \begin{bmatrix} \mathbf{B}_i \\ \mathbf{B} \end{bmatrix}, \\ \mathbf{C}_{\text{err}}^i &= [\mathbf{C}_i \quad -\mathbf{C}],\end{aligned}$$

and let $\mathbf{Q}^i$ and $\mathbf{P}^i$ denote the observability and controllability Gramians of Σ_{err}^i , partitioned as

$$\mathbf{Q}^i = \begin{bmatrix} \mathbf{Q}_{11}^i & \mathbf{Q}_{12}^i \\ \mathbf{Q}_{12}^{i\top} & \mathbf{Q}_{22}^i \end{bmatrix}, \quad \mathbf{P}^i = \begin{bmatrix} \mathbf{P}_{11}^i & \mathbf{P}_{12}^i \\ \mathbf{P}_{12}^{i\top} & \mathbf{P}_{22}^i \end{bmatrix}. \quad (11.7)$$

Theorem 8 (Analytical gradient of (11.5)). *The gradients of the objective $\mathcal{J}$ defined in (11.5) with respect to the parameters of (11.1) are*

$$\begin{aligned}\frac{\partial \mathcal{J}}{\partial \mathbf{W}} &= 2 \sum_{i=1}^M \mathbf{D}_i \left(\mathbf{Q}_{12}^{i\top} \mathbf{P}_{12}^i + \mathbf{Q}_{22}^i \mathbf{P}_{22}^i \right), \\ \frac{\partial \mathcal{J}}{\partial \mathbf{B}} &= 2 \sum_{i=1}^M \left(\mathbf{Q}_{12}^{i\top} \mathbf{B}_i + \mathbf{Q}_{22}^i \mathbf{B} \right), \\ \frac{\partial \mathcal{J}}{\partial \mathbf{C}} &= 2 \sum_{i=1}^M \left(-\mathbf{C}_i \mathbf{P}_{12}^i + \mathbf{C} \mathbf{P}_{22}^i \right), \\ \frac{\partial \mathcal{J}}{\partial (\mathbf{D}_i)_{bb}} &= 2 \left(\left(\mathbf{Q}_{12}^{i\top} \mathbf{P}_{12}^i + \mathbf{Q}_{22}^i \mathbf{P}_{22}^i \right) \mathbf{W}^\top \right)_{bb}, \quad b = 1, \dots, N.\end{aligned} \quad (11.8)$$

Proof. See Appendix B. □

The analytical gradients in Theorem 8 enable first-order methods (e.g., gradient descent) to solve the minimization problem (11.5) efficiently.

Example 6. *We illustrate the approximation performance on a control-oriented test. Consider $M = 4$ plants, and let the tasks be infinite-horizon (I/O) LQR regulation problems with iden-*

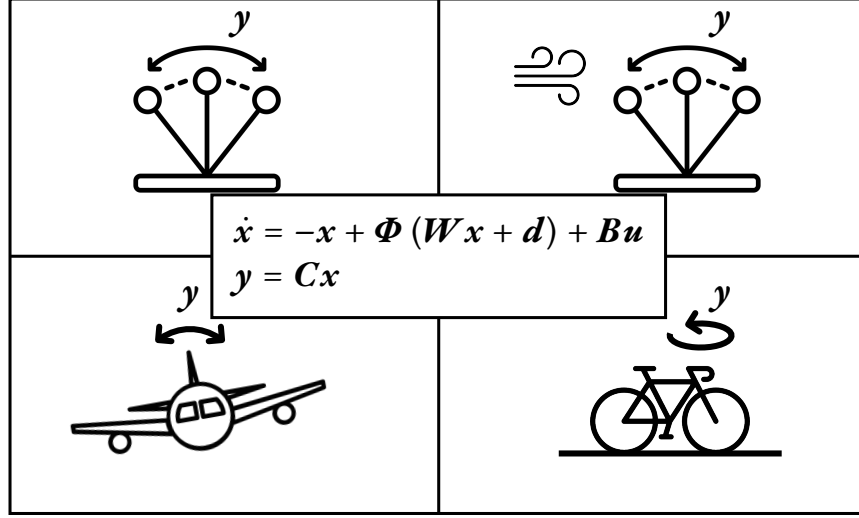


Figure 11.1: Schematic of the multi-task approximator considered in this chapter. A library of LTI systems $\{\Sigma_i^D\}_{i=1}^M$ is integrated by a single RNN architecture Σ in (11.1).

tity state and input weights for the plants in Figure 11.1, whose simplified dynamics are

$$\begin{aligned}
 \Sigma_{Aircraft} &: \left(\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 \\ r/J \end{bmatrix}, [0 \quad 1] \right), \\
 \Sigma_{Pendulum} &: \left(\begin{bmatrix} 0 & 1 \\ \frac{mgl}{J_t} & 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \frac{1}{J_t} \end{bmatrix}, [1 \quad 0] \right), \\
 \Sigma_{Pendulum + frict.} &: \left(\begin{bmatrix} 0 & 1 \\ \frac{mgl}{J_t} & \frac{\gamma}{J_t} \end{bmatrix}, \begin{bmatrix} 0 \\ \frac{1}{J_t} \end{bmatrix}, [1 \quad 0] \right), \\
 \Sigma_{Bicycle} &: \left(\begin{bmatrix} 0 & 1 \\ \frac{mgb}{J} & 0 \end{bmatrix}, \begin{bmatrix} \frac{Dv_0}{bJ} \\ \frac{mv_0^2 b}{bJ} \end{bmatrix}, [1 \quad 0] \right).
 \end{aligned} \tag{11.9}$$

Let $g = 9.8$ and use the parameters (cf. [8]):

J_t	m	l	γ	J	r	D	v_0	M	b	b	J_b
0.006	0.2	0.3	0.01	0.0475	25	4.8	2.5	8	1	1.2	8

Given the plants, we compute the LQRs (the task solvers) and embed them in the multi-task neural architecture (11.1). The task solvers Σ_i^D ($i = 1, \dots, 4$) have order $n = 2$. Figure 11.2 shows the closed-loop impulse responses of the desired task solvers Σ_i^D and of the multi-task neural approximations Σ_i^L obtained by training with the gradient in Theorem 8. While this gradient-

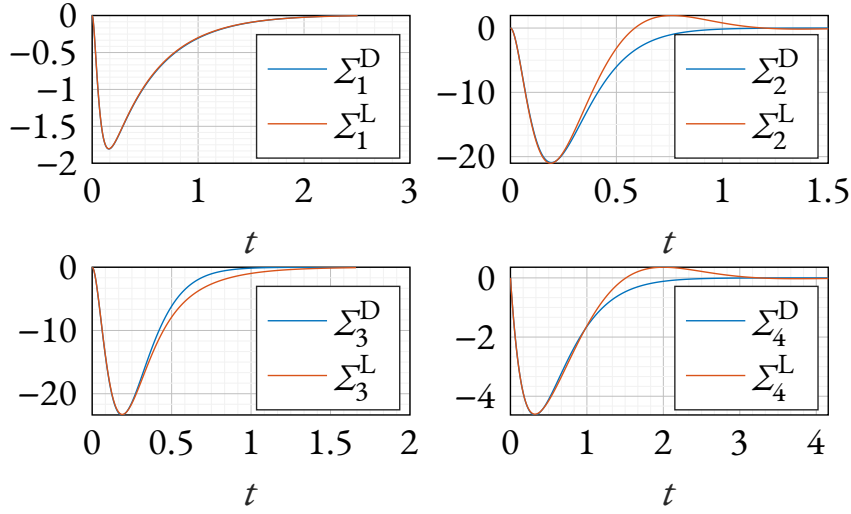


Figure 11.2: Impulse responses of the feedback interconnections between the desired controllers Σ_i^D (blue) and their implementations via the multi-task neural architecture (11.1) (red), for the plants in Figure 11.1. The “approximating controllers” are obtained by linearizing the backbone around the selected equilibria.

based procedure does not a priori guarantee stability or performance (see Section 11.3 for theoretical limitations), the results are promising and illustrate the viability of the proposed approach.

Thanks to the explicit gradient computation, we can conduct a series of experiments that highlight the performance of the proposed architecture. For example, in Figure 11.3 we consider $M = 5$ randomly generated SISO task solvers ($n = 2$) and report the value of the cost (11.5) as the state dimension N of the multi-task approximator increases. Notably, even for small N , the architecture exhibits strong approximation capability, with the median error decreasing approximately linearly (on a logarithmic scale) as N grows.

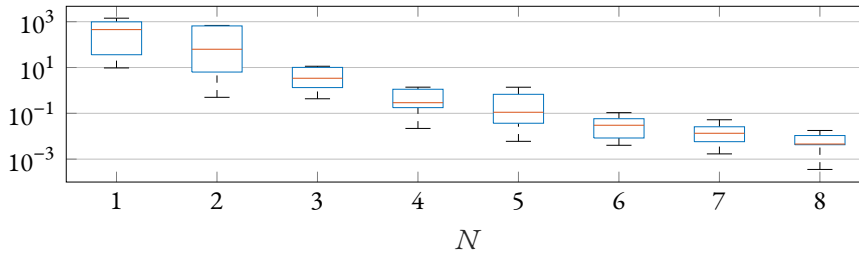


Figure 11.3: Box plot of the final cost (11.5) across 17 simulations with distinct sets ($M = 5$) of randomly generated task solvers Σ_i^D , as the architecture dimension N increases from 1 to 8. The red line denotes the median; box edges mark the 25th and 75th percentiles; whiskers cover approximately 99.3% of the data.

Similarly, in Figure 11.4 we show how the cost (11.5) varies as the number of tasks M

increases, using randomly generated SISO task solvers ($n = 2$) and a multi-task neural architecture with fixed dimension $N = 4$.

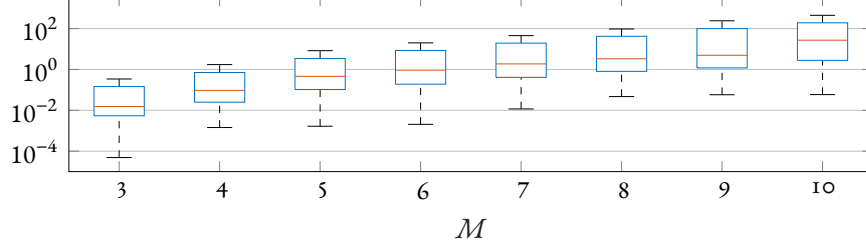


Figure 11.4: Box plot (formatted as in Figure 11.3) of the final cost (11.5) across 100 simulations for 100 randomly selected sets of task solvers Σ_i^D . We evaluate the cost while increasing the number of systems M from 3 to 10.

11.3 THEORETICAL ANALYSIS

In this section we analyze the approximation capabilities of the multi-task neural architecture proposed in (11.1). Our aim is twofold: (i) to identify classes of task solvers that can (or cannot) be approximated by the architecture, and (ii) to quantify the achievable performance through upper and lower bounds as a function of the architecture dimension.

11.3.1 FUNDAMENTAL LIMITATIONS

We begin by clarifying when a single network of the form (11.1) can exactly reproduce a collection of task solvers. Although, in some instances, perfect approximation is possible, this is not guaranteed in general and increasing the dimension of the multi-task architecture does not always guarantee perfect approximation.

Theorem 9 (Fundamental limitations of (11.5)). *Using the notation in (11.3) and (11.4), if $C_i B_i \neq C_j B_j$ for some $i, j = 1, \dots, M$ with $i \neq j$ then $\exists i^*$ such that $\|\Sigma_{i^*}^L - \Sigma_{i^*}^D\|_2^2 \neq 0$.*

Proof. By contradiction. Suppose $\|\Sigma_w^L - \Sigma_w^D\|_2^2 = 0$ for every $w = 1, \dots, M$. Then Σ_w^L and Σ_w^D have the same I/O behavior and hence the same impulse response:

$$C_i e^{A_i t} B_i = C e^{(-I + D_i W) t} B \quad \forall t \geq 0, \quad (11.10)$$

and also

$$C_j e^{A_j t} B_j = C e^{(-I + D_j W) t} B \quad \forall t \geq 0. \quad (11.11)$$

Evaluating (11.10)–(11.11) at $t = 0$ yields $\mathbf{C}_i \mathbf{B}_i = \mathbf{C} \mathbf{B} = \mathbf{C}_j \mathbf{B}_j$, a contradiction. $\square$

This result holds because the impulse response at $t = 0$ depends only on $\mathbf{B}$ and $\mathbf{C}$, not on the state-transition matrix. Consequently, the degrees of freedom provided by $\mathbf{D}_i$ cannot eliminate this error simultaneously for all $i = 1, \dots, M$. Notably, this limitation does not depend on the dimension N of the multi-task neural architecture.

Therefore, a perfect multi-task approximation is typically impossible. However, if we relax the objective and seek only arbitrarily small error, perfect approximation is achievable under mild assumptions. In particular, when $N \geq nM$, the architecture can approach the desired behaviors arbitrarily well.

11.3.2 CASE $N \geq nM$

We next show that, in a specific setting, the approximation error can be made arbitrarily small. We focus on the boundary case $N = nM$, where it is already possible to design a multi-task approximator that drives the cost $\mathcal{J}$ in (11.5) as low as desired. Specifically:

Theorem 10 (Arbitrarily accurate approximation). *Using the notation in (11.3) and (11.4), and assuming each Σ_i^D ($i = 1, \dots, M$) has $\mathbf{A}_i$ with eigenvalues strictly less than -1 , consider a multi-task neural architecture Σ^L of dimension $N = nM$ with*

$$\begin{aligned} \mathbf{W} &= \frac{\gamma}{\epsilon} \begin{bmatrix} \mathbf{I} + \mathbf{A}_1 & & \\ & \ddots & \\ & & \mathbf{I} + \mathbf{A}_M \end{bmatrix} & \mathbf{B} &= \begin{bmatrix} \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_M \end{bmatrix} \\ \mathbf{C} &= [\mathbf{C}_1 \quad \dots \quad \mathbf{C}_M] & & \\ \mathbf{D}_i &= \epsilon \operatorname{diag} \left\{ \mathbf{I}, \dots, \underbrace{\frac{1}{\gamma} \mathbf{I}}_{i\text{-th position}}, \dots, \mathbf{I} \right\} \forall i = 1, \dots, M \end{aligned} \quad (11.12)$$

and $\gamma > 1$, $\epsilon \in (0, 1)$. Then, for fixed γ ,

$$\sum_{i=1}^M \|\Sigma_i^D - \Sigma_i^L\|_2^2 \leq \frac{K}{\gamma}, \quad (11.13)$$

where $K > 0$ depends on the desired systems Σ_i^D .

Proof. See Appendix B. $\square$

The choice (11.12) realizes (up to the $-\mathbf{I}$ drift and the diagonal scaling via $\mathbf{D}_i$) a parallel connection of the desired systems. The matrix $\mathbf{D}_i$ acts as a selector for the active subsystem. The assumption that the eigenvalues of $\mathbf{A}_i$ are strictly less than -1 accounts for the presence of $-\mathbf{I}$ in Σ_i^L and ensures Hurwitz stability of the relevant blocks of $\mathbf{W}$. The term $-\mathbf{I}$ is inherited from the original nonlinear architecture (11.1); from a neuroscience perspective, it enforces a leakage term. If, for instance, one premultiplies the drift by a parameter $\tau < 1$ (i.e., $-\tau\mathbf{x}(t)$), a minor modification of Theorem 10 yields the same conclusion for task solvers with $\mathbf{A}_i < -\tau\mathbf{I}$ in place of $\mathbf{A}_i < -\mathbf{I}$.

11.3.3 CASE $N < nM$

The architecture is intended to *economically* store and reuse task solvers. The almost-perfect regime with $N = nM$ essentially memorizes all tasks “in parallel,” without exploiting similarities among them. This is visible in (11.12): the result in Theorem 10 achieves recovery from a fully stored block-diagonal model rather than from a compressed representation.

Hence, we focus on theoretical performance limits for the best *compressed* (i.e. $N < nM$) approximator obtained by solving (11.5). The optimization is highly nonconvex and solved numerically using the gradients in (11.8). Since the optimizer depends on the numerical procedure and is unknown *a priori*, we derive bounds that (i) provide interpretable performance guarantees, (ii) suggest informative initializations, and (iii) guide the design choice of N by indicating when the induced approximation is acceptable.

Remark 15. *Although the focus is on the case $N < nM$, the bounds we will establish in the next Theorems 11, 12, and 13 are applicable also when $N \geq nM$.*

UPPER BOUNDS

We present two upper bounds on (11.5). The first is broadly applicable but potentially conservative; the second applies to SISO problems with real-diagonalizable $\mathbf{A}_i$ and exploits additional structure.

The first bound rests on the observation that the error in (11.5) with *distinct* $\mathbf{D}_i$ is no larger than the error obtained when all $\mathbf{D}_i$ are constrained to be *equal*. Equivalently, an upper bound arises from approximating all task solvers with a single “average” solver. Consider

$$\underset{\mathbf{W}, \mathbf{D}, \mathbf{B}, \mathbf{C}}{\text{minimize}} \quad \sum_{i=1}^M \|\Sigma_i^D - \Sigma^L\|_2^2 \quad (11.14)$$

which yields an upper bound on the optimal value of (11.5). Problem (11.14) resembles model reduction for a family of systems.

Introduce the stacked system

$$\Sigma_{\text{ext}} : (\mathbf{A}_{\text{ext}}, \mathbf{B}_{\text{ext}}, \mathbf{C}_{\text{ext}}), \quad (11.15)$$

with

$$\mathbf{A}_{\text{ext}} = \begin{bmatrix} \mathbf{A}_1 & & \\ & \ddots & \\ & & \mathbf{A}_M \end{bmatrix}, \quad \mathbf{B}_{\text{ext}} = \begin{bmatrix} \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_M \end{bmatrix},$$

$$\mathbf{C}_{\text{ext}} = \begin{bmatrix} \mathbf{C}_1 & & \\ & \ddots & \\ & & \mathbf{C}_M \end{bmatrix},$$

and let

$$\Sigma_{\text{ext}}^{\text{B}} = (\mathbf{A}_{\text{ext}}^{\text{B}}, \mathbf{B}_{\text{ext}}^{\text{B}}, \mathbf{C}_{\text{ext}}^{\text{B}}) \quad (11.16)$$

be a balanced minimal realization [109], partitioned as

$$\mathbf{A}_{\text{ext}}^{\text{B}} = \begin{bmatrix} \mathbf{A}_{\text{ext}11}^{\text{B}} & \mathbf{A}_{\text{ext}12}^{\text{B}} \\ \mathbf{A}_{\text{ext}21}^{\text{B}} & \mathbf{A}_{\text{ext}22}^{\text{B}} \end{bmatrix}, \quad \mathbf{B}_{\text{ext}}^{\text{B}} = \begin{bmatrix} \mathbf{B}_{\text{ext}1}^{\text{B}} \\ \mathbf{B}_{\text{ext}2}^{\text{B}} \end{bmatrix},$$

$$\mathbf{C}_{\text{ext}}^{\text{B}} = \begin{bmatrix} \mathbf{C}_{\text{ext}1}^{\text{B}} & \mathbf{C}_{\text{ext}2}^{\text{B}} \end{bmatrix}. \quad (11.17)$$

Let R be the state dimension of $\Sigma_{\text{ext}}^{\text{B}}$ and let N denote the dimension of $\mathbf{A}_{\text{ext}11}^{\text{B}}$ when $R > N$.^{*} The controllability and observability Gramians of (11.16) satisfy

$$\mathbf{P}_{\text{ext}}^{\text{B}} = \mathbf{Q}_{\text{ext}}^{\text{B}} = \begin{bmatrix} \mathbf{S}_1 & \\ & \mathbf{S}_2 \end{bmatrix} \succ 0, \quad (11.18)$$

with $\mathbf{S}_1 = \text{diag}(\sigma_1, \dots, \sigma_N)$, $\mathbf{S}_2 = \text{diag}(\sigma_{N+1}, \dots, \sigma_R)$, and $\{\sigma_i\}_{i=1}^R$ the Hankel singular values of (11.16) [4].

Theorem 11 (Upper bound for (11.5)). *Let $\{\Sigma_i^{\text{L}}\}_{i=1}^M$ and $\{\Sigma_i^{\text{D}}\}_{i=1}^M$ be the LTI systems defined*

^{*}If $R \leq N$ we set $\mathbf{A}_{\text{ext}}^{\text{B}} = \mathbf{A}_{\text{ext}11}^{\text{B}}$, $\mathbf{B}_{\text{ext}}^{\text{B}} = \mathbf{B}_{\text{ext}1}^{\text{B}}$, and $\mathbf{C}_{\text{ext}}^{\text{B}} = \mathbf{C}_{\text{ext}1}^{\text{B}}$.

in (11.3) and (11.4), respectively. If $R > N$, then

$$\min_{\substack{\mathbf{W}, \mathbf{D}_1, \dots, \mathbf{D}_M \\ \mathbf{B}, \mathbf{C}}} \sum_{i=1}^M \|\Sigma_i^{\text{D}} - \Sigma_i^{\text{L}}\|_2^2 \leq \left(\mathcal{J}^{\text{B}} + \sqrt{\sigma_1} \|\mathbf{A}\mathbf{C}\|_F \right)^2. \quad (11.19)$$

If $R \leq N$, then

$$\min_{\substack{\mathbf{W}, \mathbf{D}_1, \dots, \mathbf{D}_M \\ \mathbf{B}, \mathbf{C}}} \sum_{i=1}^M \|\Sigma_i^{\text{D}} - \Sigma_i^{\text{L}}\|_2^2 \leq \sigma_1 \|\mathbf{A}\mathbf{C}\|_F^2, \quad (11.20)$$

where, using the notation in (11.17),*

$$\mathcal{J}^{\text{B}} = \sqrt{\text{tr}(\mathbf{B}_{\text{ext}2}^{\text{B} \top} \mathbf{S}_2 \mathbf{B}_{\text{ext}2}^{\text{B}}) + 2N \|\Sigma_{\text{aux}}\|_{\infty}},$$

with $\Sigma_{\text{aux}} = (\mathbf{A}_{\text{ext}}^{\text{B}}, \mathbf{B}_{\text{aux}}, \mathbf{C}_{\text{aux}})$,

$$\mathbf{B}_{\text{aux}} = \begin{bmatrix} 0 \\ \mathbf{S}_2 \mathbf{A}_{\text{ext}21}^{\text{B}} \end{bmatrix}, \quad \mathbf{C}_{\text{aux}} = \begin{bmatrix} 0 & \mathbf{A}_{\text{ext}12}^{\text{B}} \mathbf{S}_2 \end{bmatrix},$$

and

$$\mathbf{A}\mathbf{C} = \left(\mathbf{I}_{Mp} - \frac{1}{M} \begin{bmatrix} \mathbf{I}_p \\ \vdots \\ \mathbf{I}_p \end{bmatrix} \begin{bmatrix} \mathbf{I}_p & \dots & \mathbf{I}_p \end{bmatrix} \right) \mathbf{C}_{\text{ext}1}^{\text{B}}.$$

Proof. We consider the cases $R > N$ and $R \leq N$ separately.

Case $R > N$. Use (11.14) as an upper bound for (11.5), and choose $\mathbf{W}$, $\mathbf{D}$, and $\mathbf{B}$ in (11.14) so that $-\mathbf{I} + \mathbf{D}\mathbf{W} = \mathbf{A}_{\text{ext}11}^{\text{B}}$, $\mathbf{B} = \mathbf{B}_{\text{ext}1}^{\text{B}}$, and $\mathbf{C} = \frac{1}{M} \begin{bmatrix} \mathbf{I} & \dots & \mathbf{I} \end{bmatrix} \mathbf{C}_{\text{ext}1}^{\text{B}}$.[†] Then

$$\sum_{i=1}^M \|\Sigma_i^{\text{D}} - \Sigma_i^{\text{L}}\|_2^2 = \|\Sigma_{\text{ext}} - \Sigma_{\text{ext}}^{\text{L}}\|_2^2,$$

where

$$\Sigma_{\text{ext}}^{\text{L}} = \left(\mathbf{A}_{\text{ext}11}^{\text{B}}, \mathbf{B}_{\text{ext}1}^{\text{B}}, \underbrace{\begin{bmatrix} \mathbf{C} \\ \vdots \\ \mathbf{C} \end{bmatrix}}_{\mathbf{C}_{\text{ext}}^{\text{L}}} \right).$$

*We use $\|\cdot\|_F$ and $\|\cdot\|_{\infty}$ to denote the Frobenius and the $\mathcal{H}_{\infty}$ norms.

[†]The dimensions of $\mathbf{C}$ and $\mathbf{C}_{\text{ext}1}^{\text{B}}$ differ, which prevents the direct use of balanced truncation to solve (11.14); we therefore employ the averaged output as a best Frobenius norm surrogate.

Let $\Sigma_{\text{ext}1} = (\mathbf{A}_{\text{ext}1}^{\text{B}}, \mathbf{B}_{\text{ext}1}^{\text{B}}, \mathbf{C}_{\text{ext}1}^{\text{B}})$ and $\Sigma_{\text{err}} = (\mathbf{A}_{\text{ext}1}^{\text{B}}, \mathbf{B}_{\text{ext}1}^{\text{B}}, \mathcal{A}\mathbf{C})$ with $\mathcal{A}\mathbf{C} = \mathbf{C}_{\text{ext}1}^{\text{B}} - \mathbf{C}_{\text{ext}}^{\text{L}}$. By the triangle inequality,

$$\|\Sigma_{\text{ext}} - \Sigma_{\text{ext}}^{\text{L}}\|_2 \leq \|\Sigma_{\text{ext}} - \Sigma_{\text{ext}1}\|_2 + \|\Sigma_{\text{err}}\|_2.$$

Using the standard balanced truncation bound [4] gives $\|\Sigma_{\text{ext}} - \Sigma_{\text{ext}1}\|_2 \leq \mathcal{J}^{\text{B}}$, while

$$\|\Sigma_{\text{err}}\|_2^2 = \text{tr}(\mathbf{S}_1 \mathcal{A}\mathbf{C}^{\top} \mathcal{A}\mathbf{C}) \leq \sigma_1 \|\mathcal{A}\mathbf{C}\|_F^2.$$

Combining these yields (11.19).

Case $R \leq N$. The balanced realization already has order at most N . Choose

$$-\mathbf{I} + \mathbf{D}\mathbf{W} = \begin{bmatrix} \mathbf{A}_{\text{ext}}^{\text{B}} & 0 \\ 0 & -\mathbf{I} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{B}_{\text{ext}}^{\text{B}} \\ 0 \end{bmatrix}, \quad \mathbf{C} = \left[\frac{1}{M} [\mathbf{I} \ \dots \ \mathbf{I}] \mathbf{C}_{\text{ext}}^{\text{B}} \ 0 \right].$$

Then the first term in the previous triangle inequality vanishes, leading to (11.20). $\square$

Some comments are in order: (i) The case $R > N$ arises when the multi-task neural architecture state dimension is smaller than the number of distinct balanced modes of the stacked system (11.15); conversely, $R \leq N$ when the multi-task neural architecture is sufficiently large to capture all modes.* (ii) The term $\mathcal{J}^{\text{B}}$ depends on the “discarded” Hankel singular values (entries of $\mathbf{S}_2$ in (11.18)) and vanishes when $R \leq N$ (cf. (11.20)); this is the same phenomenon underpinning balanced truncation error bounds [4]. (iii) The term $\mathcal{A}\mathbf{C}$ captures the mismatch between the stacked output matrix and the averaged output matrix implementable by a single averaged task solver; taking the row-wise average minimizes this discrepancy in Frobenius norm. If the task solvers share identical (or very similar) balanced outputs, this term is small. Overall, Theorem 11 shows that the merging error depends on (i) how many modes can be captured through $\mathcal{J}^{\text{B}}$ (the differences among the systems) and (ii) the similarity of the task solvers’ output matrices through $\mathcal{A}\mathbf{C}$.

As noted at the start of this subsection, the bound in Theorem 11 is broadly applicable but does not exploit the degrees of freedom afforded by *distinct* $\mathbf{D}_i$. To reduce conservatism in the SISO setting, we derive a second upper bound that leverages diagonal structure.

Let $\mathbf{W}$ and all $\mathbf{D}_i$ be diagonal. Then each $-\mathbf{I} + \mathbf{D}_i\mathbf{W}$ is diagonal, and for each i the positive diagonal scalings can assign the real eigenvalues of $\mathbf{D}_i\mathbf{W}$ while preserving the inertia[†] of $\mathbf{W}$.

*If $R > N$, then necessarily $N < nM$.

[†]Number of negative, zero and positive eigenvalues of a symmetric matrix.

Theorem 12 (Upper bound, diagonal $\mathbf{W}$). *Let $\{\Sigma_i^L\}_{i=1}^M$ and $\{\Sigma_i^D\}_{i=1}^M$ be controllable and observable SISO systems as in (11.3)–(11.4), with each $\mathbf{A}_i$ diagonalizable with real eigenvalues. Fix $\mathbf{W}$ and $\{\mathbf{D}_i\}_{i=1}^M$ in (11.3) to be diagonal. Then*

$$\min_{\substack{\mathbf{W}, \mathbf{D}_1, \dots, \mathbf{D}_M, \\ \mathbf{B}, \mathbf{C}}} \sum_{i=1}^M \|\Sigma_i^D - \Sigma_i^L\|_2^2 \leq R/R_{22}, \quad (11.21)$$

where R/R_{22} denotes the Schur complement of

$$\mathbf{R} = \begin{bmatrix} R_{11} & \mathbf{R}_{12} \\ \mathbf{R}_{12}^\top & R_{22} \end{bmatrix},$$

with

$$R_{11} := \sum_{i=1}^M \mathbf{B}_i^\top \mathbf{Q}_{11}^i \mathbf{B}_i, \quad \mathbf{R}_{12} := \sum_{i=1}^M \mathbf{B}_i^\top \mathbf{Q}_{12}^i, \quad R_{22} := \sum_{i=1}^M \mathbf{P}_{22}^i,$$

and where the blocks $\mathbf{Q}_{11}^i, \mathbf{Q}_{12}^i, \mathbf{Q}_{22}^i$ are as in (11.7).

Proof. We compute the optimal value on the right-hand side while $\mathbf{W}$ and $\{\mathbf{D}_i\}$ are fixed and diagonal. Since restricting $\mathbf{W}$ to be diagonal reduces the feasible set of (11.5), the resulting optimum provides an upper bound, i.e.,

$$\min_{\substack{\mathbf{W}, \mathbf{D}_1, \dots, \mathbf{D}_M, \\ \mathbf{B}, \mathbf{C}}} \mathcal{J} \leq \min_{\mathbf{B}, \mathbf{C}} \sum_{i=1}^M \|\Sigma_i^D - \Sigma_i^L\|_2^2. \quad (11.22)$$

Because each $\mathbf{A}_i$ is diagonalizable with real spectrum, we may work in diagonal canonical form (see, e.g., [108]), so that $\mathbf{A}_i$ is diagonal with entries $\{a_k^i\}_{k=1}^n$ and (without loss of generality) $\mathbf{C}_i$ can be chosen so that $\text{diag}(\mathbf{C}_i) = \mathbf{I}$. In this case, the Gramians of the error systems (11.6) take a Cauchy-like form [49]:

$$\mathbf{Q}_{11}^i = -\mathbf{T}_{11}^i, \quad \mathbf{Q}_{12}^i = \mathbf{T}_{12}^i \text{diag}(\mathbf{C}), \quad \mathbf{Q}_{22}^i = -\text{diag}(\mathbf{C}) \mathbf{T}_{22}^i \text{diag}(\mathbf{C}),$$

where the Cauchy matrices are

$$\begin{aligned} [T_{11}^i]_{k,l} &:= \frac{1}{a_k^i + a_l^i}, & 1 \leq k, l \leq n, \\ [T_{12}^i]_{k,l} &:= \frac{1}{a_k^i - 1 + d_l^i w_l}, & 1 \leq k \leq n, \ 1 \leq l \leq N, \\ [T_{22}^i]_{k,l} &:= \frac{1}{e_k^i - 1 + d_l^i w_l}, & 1 \leq k, l \leq N, \end{aligned}$$

and d_l^i, w_l are the diagonal entries of $\mathbf{D}_i$ and $\mathbf{W}$, respectively. Hence the cost can be written as, leveraging the relations (11.6), (11.7) and the known relationship between the $\mathcal{H}_2$ norm of a system and its observability grammian,

$$\mathcal{J} = \sum_{i=1}^M (\mathbf{B}_i^\top \mathbf{Q}_{11}^i \mathbf{B}_i + 2 \mathbf{B}_i^\top \text{diag}(\mathbf{C}_i) \mathbf{T}_{12}^i \mathbf{X} - \mathbf{X}^\top \mathbf{T}_{22}^i \mathbf{X}). \quad (11.23)$$

where $\mathbf{X} := \text{diag}(\mathbf{C}) \mathbf{B}$ and setting $\mathbf{C}$ to the all-ones vector (w.l.o.g.), so that the right-hand side of (11.22) becomes $\min_{\mathbf{X}} \mathcal{J}$ with

$$\frac{\partial \mathcal{J}}{\partial \mathbf{X}} = 2 \sum_{i=1}^M (\mathbf{T}_{12}^{i\top} \text{diag}(\mathbf{C}_i) \mathbf{B}_i - \mathbf{T}_{22}^i \mathbf{X}).$$

Setting the derivative to zero yields

$$\mathbf{X}^* = - \left(\sum_{i=1}^M \mathbf{Q}_{22}^i \right)^{-1} \left(\sum_{i=1}^M \mathbf{Q}_{12}^{i\top} \mathbf{B}_i \right). \quad (11.24)$$

Substituting $\mathbf{X}^*$ into (11.23) gives

$$\mathcal{J}^* = R_{11} - \mathbf{R}_{12} \mathbf{R}_{22}^{-1} \mathbf{R}_{12}^\top = R/R_{22},$$

which proves (11.21). $\square$

Remark 16. *Theorem 12 is valid for any fixed choices of $\mathbf{W}$ and $\{\mathbf{D}_i\}_{i=1}^M$, and it works as an upper bound thanks to the inequality (11.22): minimizing only over $\mathbf{B}$ and $\mathbf{C}$ yields an objective no smaller than minimizing jointly over $\mathbf{W}$, $\{\mathbf{D}_i\}_{i=1}^M$, $\mathbf{B}$, and $\mathbf{C}$. While the theorem does not prescribe how to pick $\mathbf{W}$ and $\{\mathbf{D}_i\}_{i=1}^M$, a useful heuristic is to choose them so that the eigenvalues*

of $-\mathbf{I} + \mathbf{D}_i \mathbf{W}$ match those of $\mathbf{A}_i$. In Figure 11.5 we plot the bounds from Theorems 11.1 and 12; for the latter, we set the eigenvalues of $-\mathbf{I} + \mathbf{D}_i \mathbf{W}$ to those of $\mathbf{A}_i$.

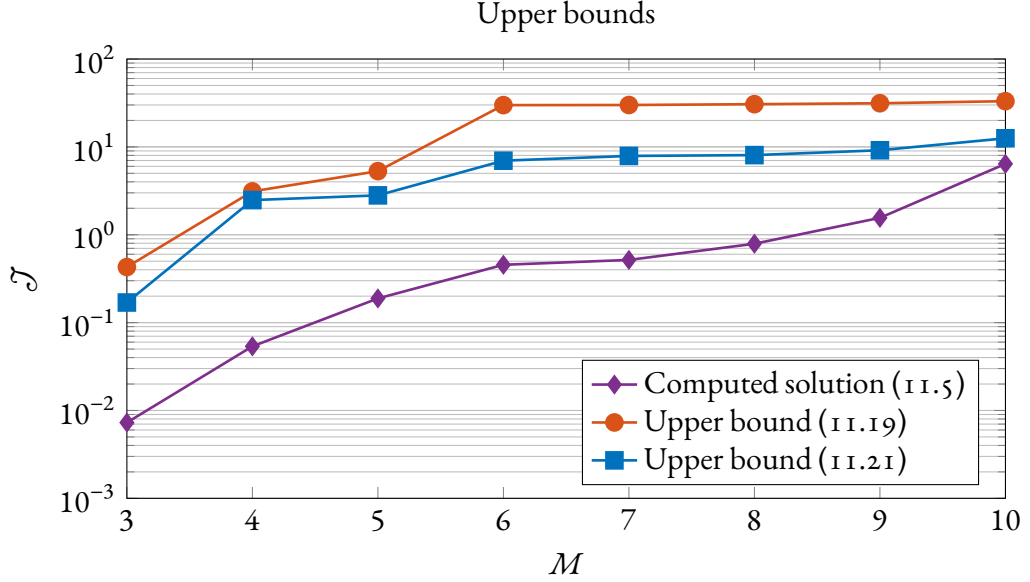


Figure 11.5: Comparison between the minimum of the cost (11.5) obtained via gradient descent using the explicit gradient of Section 11.2, the upper bound (11.19) and the upper bound (11.21). Results are generated random with a set of task solvers $\{\Sigma_i^D\}_{i=1}^M$ of dimension $n = 2$ and a multi-task neural architecture of dimension $N = 2$.

The two bounds above correspond to specific structural choices for the trainable matrices $\mathbf{W}$, $\mathbf{B}$, $\mathbf{C}$, and $\{\mathbf{D}_i\}_{i=1}^M$. They provide complementary performance guarantees for different classes of task solvers.

LOWER BOUNDS

Deriving lower bounds for optimization problems in system norms is notoriously challenging; see, e.g., [4]. Even for classical $\mathcal{H}_2$ model reduction, globally optimal solutions remain elusive. Inspired by that literature, we consider alternative norms that admit analytical treatment, while retaining the gradient-based $\mathcal{H}_2$ numerical procedure derived in Section 11.2. However, from an analytical standpoint the approximation problem in (11.5) remains difficult.

We first present an example that illustrates the challenges and then provide a lower bound tightly connected to the fundamental limitation in Section 11.3.1.

Example 7. Let us consider the scalar case, and consider $\Sigma_i^D = (a_i, b_i, c_i)$ satisfying $a_i < 0$ and $r_i = b_i c_i > 0$ for all $i = 1, \dots, M$. Assume also that $r_1 \leq r_2 \leq \dots \leq r_M$. In this case if Σ_i^L in (11.3) has $N = 1$ and defining

$$A(j, l) = \frac{r_j}{a_j} - \frac{\eta}{a_j} \left(\frac{\ln(\eta/r_j)}{W_{-1}(-1/2e)} + 1 \right),$$

for any $1 \leq j < l \leq M$, where $W_{-1}(\cdot)$ denotes the Lambert W function on the negative branch [38], we can lower bound the approximation cost

$$\underbrace{\inf_{\substack{W, D_1, \dots, D_M \\ B, C}} \sum_{i=1}^M \|\Sigma_i^D - \Sigma_i^L\|_1}_{\mathcal{J}_1}, \quad (11.25)$$

where $\|\cdot\|_1$ denotes the 1-norm, with the following bound:

$$\inf_{\substack{W, D_1, \dots, D_M \\ B, C}} \mathcal{J}_1 \geq \min \left\{ A(j, l), -A(l, j), -\frac{r_j}{a_j} \right\}. \quad (11.26)$$

In particular when $M = 2$, the bound in (11.26) holds with equality, providing the optimal merging cost for (11.25). In this case, the optimum requires the multi-task neural architecture to satisfy $\|\Sigma_1^D - \Sigma_1^L\|_1 = 0$ or $\|\Sigma_2^D - \Sigma_2^L\|_1 = 0$, i.e., match one of the two systems exactly. For $M > 2$, the bound is obtained by restricting to two systems; it can be sharpened by maximizing over the pair (j, l) .

For any $1 \leq j < l \leq M$,

$$\inf_{\substack{W, D_1, \dots, D_M \\ B, C}} \mathcal{J}_1 \geq \inf_{\substack{W, D_j, D_l \\ B, C}} \underbrace{\overbrace{\|\Sigma_j^D - \Sigma_j^L\|_1}^{\tilde{A}_j} + \overbrace{\|\Sigma_l^D - \Sigma_l^L\|_1}^{\tilde{A}^l}}_{\mathcal{J}_1}. \quad (11.27)$$

Since $N = 1$, define the scalars $r = BC$, $x_l = -1 + D_l W$, and $x_j = -1 + D_j W$. Noting that (11.27) is equivalent to minimizing over r , x_j , and x_l , we show that the optimal value on

the right-hand side is realized by:

	Case 1	Case 2	Case 3	
r	r_j	η	η	(11.28)
x_l	$x_l^*(r_j)$	a_l	a_l	
x_j	a_j	$x_j^*(\eta)$	$-\infty$	

where each case corresponds to one term in the minimum in (11.26), with

$$x_j^*(r) = \begin{cases} a_j \frac{y_j^*(r)}{y_j^*(r)+1} & \text{if } r > r_j \\ a_j & \text{if } r = r_j \end{cases} \quad (11.29)$$

$$x_l^*(r) = \begin{cases} a_l \frac{y_l^*(r)}{y_l^*(r)-1} & \text{if } r < \eta \\ a_l & \text{if } r = \eta \end{cases}, \quad (11.30)$$

and

$$y_j^*(r) = \ln(r/r_j)^{-1} [1 + W_{-1}(-1/2e)] \quad (11.31)$$

$$y_l^*(r) = \ln(\eta/r)^{-1} [1 + W_{-1}(-1/2e)]. \quad (11.32)$$

The detailed derivation of (11.28) is deferred to Appendix B.

Example 7 highlights the intrinsic difficulty of lower-bounding approximation problems defined via system norms. Consequently, a lower bound for the original objective remains out of reach. Nevertheless, we provide a lower bound aligned with the limitation in Theorem 9, quantifying the irreducible mismatch that arises even when the approximator's state-transition matrix is left unconstrained while keeping the input/output matrices fixed for all $i = 1, \dots, M$. Denote these relaxed approximants by

$$\widehat{\Sigma}_i^L : (\mathbf{X}_i, \mathbf{B}, \mathbf{C}).$$

Theorem 13 (Lower bound on impulse responses). *Using the notation in (11.3) and (11.4),*

we have

$$\min_{\substack{\mathbf{W}, \mathbf{D}_1, \dots, \mathbf{D}_M, \\ \mathbf{B}, \mathbf{C}}} \underbrace{\sum_{i=1}^M \sup_{t \geq 0} \|\mathbf{C}_i e^{\mathbf{A}_i t} \mathbf{B}_i - \mathbf{C} e^{(-\mathbf{I} + \mathbf{D}_i \mathbf{W})t} \mathbf{B}\|_2}_{\mathcal{J}_2} \geq \sqrt{\frac{\sum_{i=1}^M \|\mathbf{C}_i \mathbf{B}_i - (\sum_{i=1}^M \mathbf{C}_i \mathbf{B}_i) / M\|_F^2}{\min\{p, m\}}}. \quad (11.33)$$

Proof. We first observe that

$$\min_{\substack{\mathbf{W}, \mathbf{D}_1, \dots, \mathbf{D}_M, \\ \mathbf{B}, \mathbf{C}}} \mathcal{J}_2 \geq \min_{\mathbf{X}_1, \dots, \mathbf{X}_M, \mathbf{B}, \mathbf{C}} \underbrace{\sum_{i=1}^M \sup_{t \geq 0} \|\mathbf{C}_i e^{\mathbf{A}_i t} \mathbf{B}_i - \mathbf{C} e^{\mathbf{X}_i t} \mathbf{B}\|_2}_{\hat{\mathcal{J}}_2}, \quad (11.34)$$

with $\mathbf{X}_i \in \mathbb{R}^{N \times N}$ for $i = 1, \dots, M$. This relaxation yields a lower bound because the variables in (11.34) include those of (11.5). Setting $\mathbf{CB} = (\sum_{i=1}^M \mathbf{C}_i \mathbf{B}_i) / M$, we obtain (11.33). Indeed,

$$\begin{aligned} \hat{\mathcal{J}}_2 &= \sum_{i=1}^M \sup_{t \geq 0} \|\mathbf{C}_i e^{\mathbf{A}_i t} \mathbf{B}_i - \mathbf{C} e^{\mathbf{X}_i t} \mathbf{B}\|_2 \\ &\geq \sum_{i=1}^M \|\mathbf{C}_i \mathbf{B}_i - \mathbf{CB}\|_2 \geq \sqrt{\sum_{i=1}^M \|\mathbf{C}_i \mathbf{B}_i - \mathbf{CB}\|_2^2} \\ &\geq \sqrt{\frac{\sum_{i=1}^M \|\mathbf{C}_i \mathbf{B}_i - \mathbf{CB}\|_F^2}{\min\{p, m\}}}, \end{aligned} \quad (11.35)$$

where the last inequality uses $\|\mathbf{X}\|_2 \geq \|\mathbf{X}\|_F / \sqrt{d}$ with $d \leq \min\{p, m\}$ (see [91]). Minimizing the rightmost term in (11.35) over $\mathbf{CB}$ yields (11.33). $\square$

Theorem 13 bounds the objective in (11.33) by replacing the supremum over time with the error at time zero, as in Theorem 9, so the bound depends only on the input and output matrices. The minimum is attained by choosing the multi-task neural architecture input/output product $\mathbf{CB}$ as the average of the task solvers' input/output products.

11.4 DISCUSSION

This chapter developed a control-theoretic framework for multi-task learning via a multi-task neural architecture inspired by the coordination of neurons and astrocytes in the brain.

The backbone is a neuroscience-inspired recurrent module with a leaky drift and softplus nonlinearity; task selection is effected by a context-driven vector $\mathbf{d}$ that, through equilibria and linearization, parameterizes diagonal gains $\mathbf{D}_i$ and thereby modulates the local state transition $-\mathbf{I} + \mathbf{D}_i\mathbf{W}$. This viewpoint recasts multi-task learning as a *dynamical response approximation* problem: a single nonlinear backbone is trained so that its task-conditioned linearizations match a collection of target LTI task solvers $\{\Sigma_i^{\mathbf{D}}\}_{i=1}^M$.

On the algorithmic side, we derived closed-form expressions for the exact gradients of the nonstandard loss, enabling efficient first-order optimization without numerical differentiation. On the analytical side, we established fundamental limitations, provided an almost-perfect approximation design in the regime $N \geq nM$ (via a block-structured construction), and obtained performance guarantees for compressed regimes $N < nM$ through two complementary upper bounds: one general and one tailored to diagonalizable SISO settings. We also presented lower bounds that quantify irreducible error grounded in input/output incompatibilities, thereby clarifying how approximation quality depends on architectural dimension, task diversity, and the alignment of $\mathbf{C}_i\mathbf{B}_i$.

CONNECTIONS TO MACHINE LEARNING AND SSMs. This chapter interfaces directly with modern *multi-task* learning, where transferring knowledge across tasks accelerates training and reduces computation time [80, 107, 121, 42]. A prominent line of work aggregates specialized components within a single model via Mixtures-of-Experts (MoE) [92, 3]. For comparison, a classical (static) MoE [83] comprises experts $\{f_i\}_{i=1}^M$ and a gating rule $\{w_i(\mathbf{u}(t))\}_{i=1}^M$ acting pointwise on the input at time t , and mixes outputs as

$$f(\mathbf{u}(t)) = \sum_{i=1}^M w_i(\mathbf{u}(t))f_i(\mathbf{u}(t)), \quad (11.36)$$

often with pretrained experts (upcycling/fine-tuning) able to solve specific tasks [69, 28, 99, 72]. In contrast, many application domains (language, time series, and control) are inherently dynamical, suggesting that expert selection and mixing should themselves be *dynamic*.

Our construction makes this shift explicit. The experts are M pretrained SSMs $\{\Sigma_i^{\mathbf{D}}\}_{i=1}^M$ that process the *entire* input signal, while a single nonlinear RNN (11.1) implements *dynamic* routing and mixing. The context-driven vector $\mathbf{d}$ selects equilibria that induce diagonal gains $\mathbf{D}_i$, thereby modulating the local state transition $-\mathbf{I} + \mathbf{D}_i\mathbf{W}$; training of the RNN proceeds by mixing already trained SSMs while minimizing the objective (11.5) in the $\mathcal{H}_2$

sense, rather than performing a per-expert, task-specific optimization. Thus, mixing occurs through the backbone dynamics rather than through a per-time-step combination (11.36). Theorem 7 provides the bridge between the nonlinear backbone and the linear SSM experts, ensuring that any desired diagonal matrix $\mathbf{D}$ can be realized by a suitable choice of $\mathbf{d}$, so that local linearizations of (11.1) faithfully emulate the desired systems (11.4).

From an implementation standpoint, adopting continuous-time notation is standard and not restrictive: discrete-time realizations are obtained by standard discretization steps that typically preserve the intended behavior and are integral to recent SSM architectures [59, 60]. Finally, the mapping from context to $\mathbf{d}$ (the gating mechanism) is intentionally modular and separate from the mixing architecture: it can be instantiated, for example, via online optimization or input/output-dependent switching (see Part I). Viewed through this lens, (11.1) furnishes a principled, control-grounded *dynamic* MoE: experts are SSMs, routing is effected by $\mathbf{d}$, and mixing is achieved by the shared dynamics, with (11.5) quantifying multi-task approximation quality.

12

Conclusions

This chapter concludes the thesis. The work is organized around two main themes: Part I, which focuses on mathematical aspects of online optimization, and Part II, which addresses modern neural architectures from a control-theoretic perspective.

Although distinct, both parts share a common goal: exploiting control and systems theory to analyze, design, and investigate modern scenarios in a variety of research domains.

The first part introduces a control-theoretic framework for studying *time-varying* optimization problems. Here, control emerges as a natural analytical tool to establish guarantees and to design algorithms. The thesis presents a unified and schematic methodology to address such problems, highlighting that the results obtained so far cover only a subset of the potential applications. Future research directions include extending these techniques to new settings, such as primal-dual algorithms under stochastic models, establishing performance guarantees for nonlinear problems directly in discrete time, and exploring higher-order optimization methods (e.g., zero-, second-, or even higher-order approaches). Another promising line of work lies in bridging theory with practice by validating these algorithms on real-world problems, where the structure of the oracle and the available information may strongly influence the choice of the method.

The Part II, investigates neural architecture design from the perspective of systems and control theory. The thesis provides theoretical insights that reveal how principles of control can naturally motivate and support modern architectures design. This work is primarily theoretical in nature, aiming to establish the mathematical foundations of such designs. Nev-

ertheless, an important future step is to translate these principles into concrete architectures, with a stronger emphasis on empirical validation and experimental performance.

Finally, an especially compelling future direction is to connect the two parts of this thesis: the development of brain-inspired, control-based neural architectures that embed online optimization mechanisms as a core tool for in-context learning. This integration, which lies at the intersection of optimization, control, and machine learning, remains in part unexplored but holds great promise for advancing both theoretical understanding and practical applications.



Online optimization

Theorem 14 ([41, Theorem 2]). *Consider the convex polytope*

$$\mathcal{F} := \left\{ \mathbf{F}(\phi) : \mathbf{F}(\phi) = \sum_{i=1}^N \phi_i \mathbf{F}_i, \sum_{i=1}^N \phi_i = 1, \phi_i \geq 0, \mathbf{F}_i \in \mathbb{R}^{m \times m} \right\}. \quad (\text{A.1})$$

If there exist symmetric matrices $\mathbf{P}_i$ and a matrix $\mathbf{L}$ (of compatible dimensions) such that

$$\begin{bmatrix} \mathbf{P}_i & \mathbf{F}_i^\top \mathbf{L}^\top \\ \star & \mathbf{L} + \mathbf{L}^\top - \mathbf{P}_i \end{bmatrix} \succ 0, \quad i = 1, \dots, N, \quad (\text{A.2})$$

then all $\mathbf{F}(\phi) \in \mathcal{F}$ are Schur-stable.

Theorem 15 ([41, Theorem 3]). *Let $\mathbf{F} \in \mathbb{R}^{m \times m}$ and consider the convex polytope*

$$\mathcal{C} := \left\{ \mathbf{C}(\chi) : \mathbf{C}(\chi) = \sum_{i=1}^N \chi_i \mathbf{C}_i, \sum_{i=1}^N \chi_i = 1, \chi_i \geq 0, \mathbf{C}_i \in \mathbb{R}^m \right\}. \quad (\text{A.3})$$

If there exist symmetric matrices $\mathbf{P}_i$ and matrices $\mathbf{Q}$ and $\mathbf{R}$ (of compatible dimensions) such that

$$\begin{bmatrix} \mathbf{P}_i & \mathbf{F}\mathbf{Q} + \mathbf{C}_i\mathbf{R} \\ \star & \mathbf{Q} + \mathbf{Q}^\top - \mathbf{P}_i \end{bmatrix} \succ 0, \quad i = 1, \dots, N, \quad (\text{A.4})$$

then, for all $\mathbf{C}(\chi) \in \mathcal{C}$, the matrix $\mathbf{F} + \mathbf{C}(\chi)\mathbf{K}$ is Schur-stable. Moreover, a stabilizing gain

is $\mathbf{K} := \mathbf{R}\mathbf{Q}^{-1}$.

Lemma 1 (Eigenvalues of $\tilde{\mathbf{B}}(\tau)$). *Given $\tilde{\mathbf{B}}(\tau)$ as in (7.15), if τ is such that*

$$0 < \tau \leq \frac{\lambda_{\min}(\mathbf{A})}{4\lambda_{\max}(\mathbf{H}\mathbf{A}^{-1}\mathbf{H}^\top)}, \quad (\text{A.5})$$

then the eigenvalues of $\tilde{\mathbf{B}}(\tau)$ are real and belong to the following interval

$$[\tau\lambda_{\min}(\mathbf{H}\mathbf{A}^{-1}\mathbf{H}^\top), \lambda_{\max}(\mathbf{A})]. \quad (\text{A.6})$$

Proof. As proved in [17, Corollary 2.7] if $\lambda_{\min}(\mathbf{A}) \geq 4\tau\lambda_{\max}(\mathbf{H}\mathbf{A}^{-1}\mathbf{H}^\top)$, then $\tilde{\mathbf{B}}(\tau)$ has all real eigenvalues, which implies inequality (A.5) and this implies that $\tilde{\mathbf{B}}(\tau)$ has all real eigenvalues. Moreover, thanks to the fact that $\mathbf{H}$ is full row rank and using [17, Proposition 2.12] and [101, Theorem 2.1] we know that the eigenvalues of $\tilde{\mathbf{B}}(\tau)$ belongs to the interval $[\min\{\lambda_{\min}(\mathbf{A}), \tau\lambda_{\min}(\mathbf{H}\mathbf{A}^{-1}\mathbf{H}^\top)\}, \lambda_{\max}(\mathbf{A})]$. Observe finally that, using (A.5) we can argue that

$$\begin{aligned} \tau\lambda_{\min}(\mathbf{H}\mathbf{A}^{-1}\mathbf{H}^\top) &\leq \frac{\lambda_{\min}(\mathbf{A})}{4\lambda_{\max}(\mathbf{H}\mathbf{A}^{-1}\mathbf{H}^\top)}\lambda_{\min}(\mathbf{H}\mathbf{A}^{-1}\mathbf{H}^\top) \\ &= \frac{\lambda_{\min}(\mathbf{A})}{4\kappa(\mathbf{H}\mathbf{A}^{-1}\mathbf{H}^\top)} \leq \lambda_{\min}(\mathbf{A}), \end{aligned} \quad (\text{A.7})$$

which yields (A.6). $\square$

Proposition 5. *Let $\mathbf{F}_1, \mathbf{F}_2$, and $\mathbf{F}_3$ be vectors in a vector space, and define*

$$\mathcal{S} := \{\mathbf{F}_1 + x\mathbf{F}_2 + xy\mathbf{F}_3 : x \in [x_1, x_2], y \in [\gamma_1, \gamma_2]\}.$$

Then $\mathcal{S}$ is a convex polytope. In particular,

$$\mathcal{S} = \left\{ \sum_{i,j=1,2} \alpha_{ij} \mathbf{V}_{ij} : \alpha_{ij} \geq 0, \sum_{i,j=1,2} \alpha_{ij} = 1 \right\},$$

where

$$\mathbf{V}_{ij} := \mathbf{F}_1 + x_i\mathbf{F}_2 + x_i\gamma_j\mathbf{F}_3, \quad i, j \in \{1, 2\},$$

denote the four vertices of $\mathcal{S}$.

Proof. Define

$$\bar{\mathcal{S}} := \left\{ \sum_{i,j=1,2} \alpha_{ij} \mathbf{V}_{ij} : \alpha_{ij} \geq 0, \sum_{i,j=1,2} \alpha_{ij} = 1 \right\}.$$

We prove the claim by showing both inclusions: $\mathcal{S} \subseteq \bar{\mathcal{S}}$ and $\mathcal{S} \supseteq \bar{\mathcal{S}}$.

For the first inclusion, take any $s \in \mathcal{S}$. Then there exist $x \in [x_1, x_2]$ and $y \in [y_1, y_2]$ such that

$$s = \mathbf{F}_1 + x\mathbf{F}_2 + xy\mathbf{F}_3.$$

Let $x = (1 - \alpha)x_1 + \alpha x_2$ and $y = (1 - \beta)y_1 + \beta y_2$ with $\alpha, \beta \in [0, 1]$. Then

$$s = \mathbf{F}_1 + ((1 - \alpha)x_1 + \alpha x_2)\mathbf{F}_2 + ((1 - \alpha)x_1 + \alpha x_2)((1 - \beta)y_1 + \beta y_2)\mathbf{F}_3.$$

Using the identities

$$\begin{aligned} \mathbf{F}_1 &= ((1 - \alpha)(1 - \beta) + (1 - \alpha)\beta + (1 - \beta)\alpha + \alpha\beta)\mathbf{F}_1, \\ \mathbf{F}_2 &= ((1 - \beta) + \beta)\mathbf{F}_2, \end{aligned}$$

we obtain

$$s = \alpha_{11}\mathbf{V}_{11} + \alpha_{12}\mathbf{V}_{12} + \alpha_{21}\mathbf{V}_{21} + \alpha_{22}\mathbf{V}_{22},$$

where $\alpha_{ij} \geq 0$ and $\sum_{i,j} \alpha_{ij} = 1$. Thus $s \in \bar{\mathcal{S}}$, and hence $\mathcal{S} \subseteq \bar{\mathcal{S}}$.

For the reverse inclusion, note that each vertex $\mathbf{V}_{ij}$ belongs to $\mathcal{S}$ by definition. Therefore, it suffices to show that $\mathcal{S}$ is convex. Take $s' = \mathbf{F}_1 + x'\mathbf{F}_2 + x'y'\mathbf{F}_3$ and $s'' = \mathbf{F}_1 + x''\mathbf{F}_2 + x''y''\mathbf{F}_3$ in $\mathcal{S}$. For $\alpha \in [0, 1]$, consider

$$s = (1 - \alpha)s' + \alpha s'' = \mathbf{F}_1 + x\mathbf{F}_2 + xy\mathbf{F}_3,$$

where

$$x = (1 - \alpha)x' + \alpha x'', \quad y = \frac{(1 - \alpha)x'y' + \alpha x''y''}{x}.$$

By construction, $x \in [x_1, x_2]$. Moreover, $f(\alpha) := y$ satisfies $f(0) = y'$ and $f(1) = y''$. A direct analysis of the second derivative of $f(\alpha)$ shows that $f(\alpha)$ is monotone on $[0, 1]$, and thus $y \in [y_1, y_2]$. Therefore $s \in \mathcal{S}$, proving that $\mathcal{S}$ is convex. Hence $\bar{\mathcal{S}} \subseteq \mathcal{S}$. $\square$

B

Neural architectures

Proof of Theorem 8. Let $\mathcal{J}_i = \|\Sigma_i^D - \Sigma_i^L\|_2^2 = \|\Sigma_{\text{err}}^i\|_2^2$. From [115],

$$\frac{\partial \mathcal{J}_i}{\partial \mathbf{A}_{\text{err}}^i} = \frac{\partial \|\Sigma_{\text{err}}^i\|_2^2}{\partial \mathbf{A}_{\text{err}}^i} = 2 \mathbf{Q}^i \mathbf{P}^i.$$

Using the chain rule [91],

$$\begin{aligned} \frac{\partial \mathcal{J}_i}{\partial \mathbf{W}_{bl}} &= \sum_{k=1}^{N+n} \sum_{j=1}^{N+n} \frac{\partial \mathcal{J}_i}{\partial (\mathbf{A}_{\text{err}}^i)_{kj}} \frac{\partial (\mathbf{A}_{\text{err}}^i)_{kj}}{\partial \mathbf{W}_{bl}} \\ &= \frac{\partial \mathcal{J}_i}{\partial (\mathbf{A}_{\text{err}}^i)_{(n+b)(n+l)}} \frac{\partial (\mathbf{A}_{\text{err}}^i)_{(n+b)(n+l)}}{\partial \mathbf{W}_{bl}} = 2(\mathbf{D}_i)_{bb} (\mathbf{Q}^i \mathbf{P}^i)_{(n+b)(n+l)}, \end{aligned} \tag{B.1}$$

where, for $1 \leq b, l \leq N$,

$$\frac{\partial (\mathbf{A}_{\text{err}}^i)_{kj}}{\partial \mathbf{W}_{bl}} = \begin{cases} (\mathbf{D}_i)_{bb}, & \text{if } (k, j) = (n+b, n+l), \\ 0, & \text{otherwise.} \end{cases}$$

Writing (B.1) in matrix form and summing over i yields the first identity in (11.8). Similarly,

$$\frac{\partial(\mathbf{A}_{\text{err}}^i)_{kj}}{\partial(\mathbf{D}_i)_{bb}} = \begin{cases} \mathbf{W}_{b,(j-n)}, & \text{if } k = n + b \text{ and } n < j \leq n + N, \\ 0, & \text{otherwise,} \end{cases}$$

and hence, for each i ,

$$\frac{\partial \mathcal{J}_i}{\partial(\mathbf{D}_i)_{bb}} = 2 \left(\mathbf{Q}^i \mathbf{P}^i \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{W}^\top \end{bmatrix} \right)_{(n+b)(n+b)},$$

which gives the last identity in (11.8).

For $\mathbf{B}$ and $\mathbf{C}$, recall

$$\|\Sigma_{\text{err}}^i\|_2^2 = \text{tr}(\mathbf{C}_{\text{err}}^i \mathbf{P}^i \mathbf{C}_{\text{err}}^{i\top}) = \text{tr}(\mathbf{B}_{\text{err}}^{i\top} \mathbf{Q}^i \mathbf{B}_{\text{err}}^i).$$

Since $\frac{\partial \mathcal{J}_i}{\partial \mathbf{B}_{\text{err}}^i} = 2 \mathbf{Q}^i \mathbf{B}_{\text{err}}^i$, we have

$$\frac{\partial \mathcal{J}_i}{\partial \mathbf{B}_{bl}} = \sum_{k=1}^{N+n} \sum_{j=1}^m \frac{\partial \mathcal{J}_i}{\partial(\mathbf{B}_{\text{err}}^i)_{kj}} \frac{\partial(\mathbf{B}_{\text{err}}^i)_{kj}}{\partial \mathbf{B}_{bl}} = \frac{\partial \mathcal{J}_i}{\partial(\mathbf{B}_{\text{err}}^i)_{(n+b)l}} = 2 (\mathbf{Q}^i \mathbf{B}_{\text{err}}^i)_{(n+b)l},$$

or, in compact form,

$$\frac{\partial \mathcal{J}_i}{\partial \mathbf{B}} = 2 (\mathbf{Q}_{12}^{i\top} \mathbf{B}_i + \mathbf{Q}_{22}^i \mathbf{B}). \quad (\text{B.2})$$

Similarly,

$$\frac{\partial \mathcal{J}_i}{\partial \mathbf{C}} = 2 (-\mathbf{C}_i \mathbf{P}_{12}^i + \mathbf{C} \mathbf{P}_{22}^i). \quad (\text{B.3})$$

Summing (B.2)–(B.3) over $i = 1, \dots, M$ yields the remaining two identities in (11.8). $\square$

Proof of Theorem 10. With the previous parametrization (11.12) of Σ_i^{L} , the error term (11.5) becomes

$$\sum_{i=1}^M \left\| \sum_{j=1, j \neq i}^M \mathbf{C}_j \left(s\mathbf{I} - (-\mathbf{I} + \gamma(\mathbf{I} + \mathbf{A}_j)) \right)^{-1} \mathbf{B}_j \right\|_2^2, \quad (\text{B.4})$$

which, using the triangular inequality and the properties of the $\mathcal{H}_2$ norm, leads to:

$$\sum_{i=1}^M \|\Sigma_i^D - \Sigma_i^L\|_2^2 \leq \sum_{i=1}^M \sum_{j=1, j \neq i}^M \sqrt{\text{tr}(\mathbf{B}_j^\top \mathbf{Q}_j \mathbf{B}_j)} \left(\sqrt{\text{tr}(\mathbf{B}_j^\top \mathbf{Q}_j \mathbf{B}_j)} + \sum_{k=1, k \neq j}^M \sqrt{\text{tr}(\mathbf{B}_k^\top \mathbf{Q}_k \mathbf{B}_k)} \right), \quad (\text{B.5})$$

where $\mathbf{Q}_j$ is the observability grammian of $(-\mathbf{I} + \gamma(\mathbf{I} + \mathbf{A}_j), \mathbf{C}_j)$. We will show that the right-hand side of (B.5) can be bounded by a sum of nonnegative terms multiplied by the scalar $1/\gamma$, leading to (11.13). First, we show that $\mathbf{Q}_j > 0$. This can be deduced directly from the observability of $(-\mathbf{I} + \gamma(\mathbf{I} + \mathbf{A}_j), \mathbf{C}_j)$. Indeed, given a state matrix $\mathbf{F}$ and an output matrix $\mathbf{H}$:

1. if $(\mathbf{F}, \mathbf{H})$ is observable then $(\pm \mathbf{I} + \mathbf{F}, \mathbf{H})$ is observable;
2. if $(\mathbf{F}, \mathbf{H})$ is observable then $(\gamma \mathbf{F}, \mathbf{H})$ is observable.

The first point 1 is a direct consequence of the PBH test for observability. For the second point, if we denote by $\widehat{\mathbf{Q}}$ the observability grammian of $(\mathbf{F}, \mathbf{H})$ and by $\mathbf{Q}^*$ the observability grammian of $(\gamma \mathbf{F}, \mathbf{H})$ the solution of the Lyapunov equation:

$$(\gamma \mathbf{F})^\top \mathbf{Q}^* + \mathbf{Q}^* (\gamma \mathbf{F}) = -\mathbf{H}^\top \mathbf{H}$$

for $\gamma > 1$ is $\mathbf{Q}^* = \widehat{\mathbf{Q}}/\gamma$ and nonsingular if $\widehat{\mathbf{Q}}$ is nonsingular, thus proving that the pair $(\gamma \mathbf{F}, \mathbf{H})$ is observable. Combining the two points 1, 2 with the stability of $\mathbf{I} + \mathbf{A}_j$, it is possible to prove that $\mathbf{Q}_j > 0$. Denoting by $\mathbf{Q}_j^*$ the observability grammian of $(\gamma(\mathbf{I} + \mathbf{A}_j), \mathbf{C}_j)$, we can write the following two Lyapunov equations:

$$\begin{aligned} (-\mathbf{I} + \gamma(\mathbf{I} + \mathbf{A}_j))^\top \mathbf{Q}_j + \mathbf{Q}_j (-\mathbf{I} + \gamma(\mathbf{I} + \mathbf{A}_j)) &= -\mathbf{C}_j^\top \mathbf{C}_j \\ \gamma(\mathbf{I} + \mathbf{A}_j)^\top \mathbf{Q}_j^* + \mathbf{Q}_j^* \gamma(\mathbf{I} + \mathbf{A}_j) &= -\mathbf{C}_j^\top \mathbf{C}_j. \end{aligned}$$

Subtracting these two equations, we obtain:

$$\gamma(\mathbf{I} + \mathbf{A}_j)^\top (\mathbf{Q}_j^* - \mathbf{Q}_j) + (\mathbf{Q}_j^* - \mathbf{Q}_j) \gamma(\mathbf{I} + \mathbf{A}_j) = -2\mathbf{Q}_j.$$

Since $\mathbf{Q}_j > 0$ and $\gamma(\mathbf{I} + \mathbf{A}_j)$ is stable, by the Lyapunov stability criteria, there exists a unique positive definite matrix that solves this equation, implying that $\mathbf{Q}_j < \mathbf{Q}_j^*$ and, in particular

$\text{tr}(\mathbf{Q}_j) < \text{tr}(\mathbf{Q}_j^*)$. Therefore, from (B.5), it is possible to obtain:

$$\begin{aligned}
(\text{B.5}) &= \sum_{i=1}^M \sum_{j=1, j \neq i}^M \sqrt{\text{tr}(\mathbf{Q}_j \mathbf{B}_j \mathbf{B}_j^\top)} \left(\sqrt{\text{tr}(\mathbf{Q}_j \mathbf{B}_j \mathbf{B}_j^\top)} + \sum_{k=1, k \neq j}^M \sqrt{\text{tr}(\mathbf{Q}_k \mathbf{B}_k \mathbf{B}_k^\top)} \right) \\
&\leq \sum_{i=1}^M \sum_{j=1, j \neq i}^M \sqrt{\text{tr}(\mathbf{Q}_j \mathbf{B}_j \mathbf{B}_j^\top)} \left(\sqrt{\text{tr}(\mathbf{Q}_j \mathbf{B}_j \mathbf{B}_j^\top)} + \sum_{k=1, k \neq j}^M \sqrt{\text{tr}(\mathbf{Q}_k \mathbf{B}_k \mathbf{B}_k^\top)} \right) \\
&\leq \sum_{i=1}^M \sum_{j=1, j \neq i}^M \sqrt{\text{tr}(\mathbf{Q}_j) \text{tr}(\mathbf{B}_j \mathbf{B}_j^\top)} \left(\sqrt{\text{tr}(\mathbf{Q}_j) \text{tr}(\mathbf{B}_j \mathbf{B}_j^\top)} + \sum_{k=1, k \neq j}^M \sqrt{\text{tr}(\mathbf{Q}_k) \text{tr}(\mathbf{B}_k \mathbf{B}_k^\top)} \right)
\end{aligned}$$

At this point, we use the fact that $\mathbf{Q}_j^* = \widehat{\mathbf{Q}}_j / \gamma$ where $\widehat{\mathbf{Q}}_j$ is the observability grammian associated to the pair $(\mathbf{I} + \mathbf{A}_j, \mathbf{C}_j)$ and is constant with respect to γ . We then conclude our chain of inequalities:

$$\sum_{i=1}^M \sum_{j=1, j \neq i}^M \sqrt{\text{tr}(\mathbf{Q}_j^*) \text{tr}(\mathbf{B}_j \mathbf{B}_j^\top)} \left(\sqrt{\text{tr}(\mathbf{Q}_j^*) \text{tr}(\mathbf{B}_j \mathbf{B}_j^\top)} + \sum_{k=1, k \neq j}^M \sqrt{\text{tr}(\mathbf{Q}_k^*) \text{tr}(\mathbf{B}_k \mathbf{B}_k^\top)} \right) = \frac{K}{\gamma},$$

where

$$K = \sum_{i=1}^M \sum_{j=1, j \neq i}^M \sqrt{\text{tr}(\widehat{\mathbf{Q}}_j) \text{tr}(\mathbf{B}_j \mathbf{B}_j^\top)} \left(\sqrt{\text{tr}(\widehat{\mathbf{Q}}_j) \text{tr}(\mathbf{B}_j \mathbf{B}_j^\top)} + \sum_{k=1, k \neq j}^M \sqrt{\text{tr}(\widehat{\mathbf{Q}}_k) \text{tr}(\mathbf{B}_k \mathbf{B}_k^\top)} \right).$$

From this, we directly have (11.13). $\square$

Computation of Example 7. To prove (11.28), first fix $r_j \leq r \leq \eta$ and minimize over $x_j(r)$ and $x_l(r)$; then optimize over r . (The cases $r \leq r_j$ and $r \geq \eta$ follow by analogous arguments.) If $x_j > a_j$ and $x_l < a_l$, one shows the minimum of $\tilde{\mathcal{J}}_1$ (for fixed r) is approached as $x_j \rightarrow a_j^+$ and $x_l \rightarrow a_l^-$. Hence assume $x_j \leq a_j$ and $x_l \geq a_l$. When $x_j < a_j$ and $x_l > a_l$, the impulse responses of Σ_j^L and Σ_j^D (and of Σ_l^L and Σ_l^D) intersect once. With the changes of variables

$$\begin{aligned}
y_j: (-\infty, a_j) &\rightarrow (-\infty, -1), & x_j &\mapsto \frac{x_j}{a_j - x_j}, \\
y_l: (a_l, 0) &\rightarrow (-\infty, 0), & x_l &\mapsto \frac{x_l}{x_l - a_l},
\end{aligned}$$

we obtain $\tilde{\mathcal{J}}_1 = \tilde{A}_j + \tilde{A}^l$ with

$$\begin{aligned}\tilde{A}_j(\gamma_j, r) &= \frac{r \left(2e^{\gamma_j \ln(r/r_j)} - 1 \right)}{a_j \gamma_j} - \frac{r}{a_j} + \frac{r_j}{a_j} \\ \tilde{A}^l(\gamma_l, r) &= \frac{r \left(2e^{\gamma_l \ln(r/r_l)} - 1 \right)}{a_l \gamma_l} + \frac{r}{a_l} - \frac{r_l}{a_l}.\end{aligned}\tag{B.6}$$

Setting $\frac{\partial}{\partial \gamma_j} \tilde{A}_j(\gamma_j, r) = 0$ yields

$$2 \ln(r/r_j) e^{\gamma_j \ln(r/r_j)} - 2e^{\gamma_j \ln(r/r_j)} + 1 = 0,\tag{B.7}$$

whose unique negative solution on the relevant branch is given by (11.31) via the Lambert W function $W_{-1}(\cdot)$; similarly for (11.32). These are indeed minima for $\gamma_j, \gamma_l \in (-\infty, 0)$. For (B.6), distinguish:

1. $\gamma_j^* \in (-\infty, -1)$ if $r < r_j e^{-[1+W_{-1}(-1/2e)]}$;
2. $\gamma_j^* \notin (-\infty, -1)$ if $r \geq r_j e^{-[1+W_{-1}(-1/2e)]}$.

If $\eta < r_j e^{-[1+W_{-1}(-1/2e)]}$, we are in Scenario 1; one shows that $\tilde{\mathcal{J}}_1$ (for $r_j < r < \eta$ with $x_j = x_j^*(r)$ and $x_l = x_l^*(r)$ as in (11.29)–(11.30)) has no interior minimum in r . By the Weierstrass extreme value theorem, the minimum over $r \in [r_j, \eta]$ is attained at the endpoints, i.e.,

$$\min \left\{ \tilde{A}^l(x_l^*(r_j), r_j), \tilde{A}_j(x_j^*(\eta), \eta) \right\} = \inf_{x_j, x_l, r} \tilde{\mathcal{J}}_1,\tag{B.8}$$

which corresponds to Cases 1 and 2 in (11.28).

In Scenario 2, for $r_j e^{-[1+W_{-1}(-1/2e)]} \leq r \leq \eta$, monotonicity gives

$$\tilde{A}_j(x_j, r) \xrightarrow{x_j \rightarrow -\infty} -\frac{r_j}{a_j} = \inf_{x_j} \tilde{A}_j(x_j, r).$$

Since this infimum is independent of x_l and r ,

$$\inf_{x_j, x_l, r} \tilde{\mathcal{J}}_1 \geq -\frac{r_j}{a_j} + \underbrace{\inf_{x_l, r} \tilde{A}^l(x_l, r)}_{=0},$$

which yields Case 3 in (11.28).

For $r_j \leq r < r_j e^{-[1+W_{-1}(-1/2e)]}$, the same reasoning gives

$$\inf_{x_j, x_l, r} \tilde{\mathcal{J}}_1 = \inf_{\substack{W, D_j, D_l \\ B, C}} \tilde{\mathcal{J}}_1 \geq \min \left\{ \tilde{A}^l(x_l^*(r_j), r_j), -\frac{r_j}{a_j} \right\}. \quad (\text{B.9})$$

Combining (B.8) and (B.9) yields (11.26), with $\tilde{A}^l(x_l^*(r_j), r_j) = -A(l, j)$ and $\tilde{A}_j(x_j^*(\eta), \eta) = A(j, l)$.

Bibliography

- [1] Al Makdah, A. A., Katewa, V., & Pasqualetti, F. (2020). Accuracy prevents robustness in perception-based control. In *American Control Conference* Denver, CO, USA.
- [2] Anderson, B. D. & Moore, J. B. (2007). *Optimal control: linear quadratic methods*. Courier Corporation.
- [3] Anthony, Q., Tokpanov, Y., Glorioso, P., & Millidge, B. (2024). Blackmamba: Mixture of experts for state-space models.
- [4] Antoulas, A. C. & Sorensen, D. C. (2001). Approximation of large-scale dynamical systems: an overview. *Int. J. Applied Mathematics and Computer Science*, 11(5), 1093–1121.
- [5] Astrom, K. & Hagglund, T. (2006). *Advanced PID control*. ISA-The Instrumentation, Systems, and Automation Society; Research Triangle Park, NC 27709.
- [6] Astrom, K. J. (1970). *Introduction to stochastic control theory*. Academic Press.
- [7] Aström, K. J. (1991). Adaptive control. In A. C. Antoulas (Ed.), *Mathematical System Theory: The Influence of R. E. Kalman* (pp. 437–450). Springer Berlin Heidelberg.
- [8] Astrom, K. J. & Murray, R. M. (2008). *Feedback Systems*. Princeton University Press.
- [9] Baggio, G., Pasqualetti, F., & Zampieri, S. (2022). Energy-aware controllability of complex networks. *Annual Review of Control, Robotics, and Autonomous Systems*, 5, 465–489.
- [10] Baggio, G. & Zampieri, S. (2021). Non-normality improves information transmission performance of network systems. *IEEE Transactions on Control of Network Systems*, 8(4), 1846–1858.
- [11] Bai, Z. & Silverstein, J. (2009). *Spectral Analysis of Large Dimensional Random Matrices*. Springer Series in Statistics. Springer New York.

- [12] Bastianello, N., Carli, R., & Simonetto, A. (2023a). Extrapolation-Based Prediction-Correction Methods for Time-varying Convex Optimization. *Signal Processing*, 210, 109089.
- [13] Bastianello, N., Carli, R., & Zampieri, S. (2023b). Internal Model-Based Online Optimization. *IEEE Transactions on Automatic Control*, (pp. 1–8).
- [14] Baxter, J. (2000). A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12, 149–198.
- [15] Benozzo, D., Baron, G., Coletta, L., Chiuso, A., Gozzi, A., & Bertoldo, A. (2023). Macroscale coupling between structural and effective connectivity in the mouse brain. *bioRxiv*, (pp. 2023–02).
- [16] Benzi, M., Golub, G. H., & Liesen, J. (2005). Numerical solution of saddle point problems. *Acta Numerica*, 14, 1–137.
- [17] Benzi, M. & Simoncini, V. (2006). On the eigenvalues of a class of saddle point matrices. *Numerische Mathematik*, 103(2), 173–196.
- [18] Bernstein, A., Dall’Anese, E., & Simonetto, A. (2019). Online Primal-Dual Methods With Measurement Feedback for Time-Varying Convex Optimization. *IEEE Transactions on Signal Processing*, 67(8), 1978–1991.
- [19] Bertsekas, D. P. (2014). *Constrained optimization and Lagrange multiplier methods*. Academic press.
- [20] Bianchin, G. & Van Scoy, B. (2024). The internal model principle of time-varying optimization. *arXiv preprint arXiv:2407.08037*.
- [21] Bianchin, G. & Van Scoy, B. (2025). Feedback optimization of dynamical systems in time-varying environments: An internal model principle approach. *arXiv preprint arXiv:2508.03503*.
- [22] Bijsterbosch, J., Smith, S. M., & Beckmann, C. (2017). *An introduction to resting state fMRI functional connectivity*. Oxford University Press.
- [23] Bin, M., Notarnicola, I., & Parisini, T. (2022). Stability, linear convergence, and robustness of the wang-elia algorithm for distributed consensus optimization. In *2022 IEEE 61st Conference on Decision and Control (CDC)* (pp. 1610–1615).

- [24] Bowman, F. D. (2014). Brain imaging analysis. *Annual review of statistics and its application*, 1, 61–85.
- [25] Boyd, S. & Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.
- [26] Bullmore, E. & Sporns, O. (2009). Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature reviews neuroscience*, 10(3), 186–198.
- [27] Byrnes, C. I., Priscoli, F. D., & Isidori, A. (1997). *Output Regulation of Uncertain Nonlinear Systems*. Systems & Control: Foundations & Applications. Boston, MA: Birkhäuser Boston.
- [28] Cai, R., Ro, Y., Kim, G.-W., Wang, P., Bejnordi, B. E., Akella, A., & Wang, Z. (2024). $\text{\textit{Read-ME}}$: Refactorizing LLMs as router-decoupled mixture of experts with system co-design. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- [29] Cao, X. & Liu, K. J. R. (2019). Online Convex Optimization With Time-Varying Constraints and Bandit Feedback. *IEEE Transactions on Automatic Control*, 64(7), 2665–2680.
- [30] Cao, X., Zhang, J., & Poor, H. V. (2018). A Virtual-Queue-Based Algorithm for Constrained Online Convex Optimization With Applications to Data Center Resource Allocation. *IEEE Journal of Selected Topics in Signal Processing*, 12(4), 703–716.
- [31] Caruana, R. (1997). Multitask learning. *Machine Learning*, 28(1), 41.
- [32] Casti, U., Bastianello, N., Carli, R., & Zampieri, S. (2025). A control theoretical approach to online constrained optimization. *Automatica*, 176, 112107.
- [33] Casti, U. & Zampieri, S. (2024). Stochastic models for online optimization.
- [34] Caverly, R. J. & Forbes, J. R. (2019). Lmi properties and applications in systems, stability, and control theory. *ArXiv*, abs/1903.08599.
- [35] Chang, T.-H., Hong, M., Wai, H.-T., Zhang, X., & Lu, S. (2020). Distributed Learning in the Nonconvex World: From batch data to streaming and beyond. *IEEE Signal Processing Magazine*, 37(3), 26–38.

- [36] Chen, Y., Ospina, A. M., Pasqualetti, F., & Dall’Anese, E. (2023). Multi-task system identification of similar linear time-invariant dynamical systems. In *IEEE Conf. on Decision and Control* Marina Bay Sands, Singapore. To appear. arXiv preprint arXiv:2301.01430.
- [37] Chen, Y., Rosen, B. Q., & Sejnowski, T. J. (2022). Dynamical differential covariance recovers directional network structure in multiscale neural systems. *Proceedings of the National Academy of Sciences*, 119(24), e2117234119.
- [38] Corless, R. M., Gonnet, G. H., Hare, D. E. G., Jeffrey, D. J., & Knuth, D. E. (1996). On the lambert w function. *Advances in Computational Mathematics*, 5(1), 329–359.
- [39] Davydov, A., Centorrino, V., Gokhale, A., Russo, G., & Bullo, F. (2023). Contracting Dynamics for Time-Varying Convex Optimization. arXiv:2305.15595 [cs, eess, math].
- [40] De Bruyne, F., Anderson, B., & Gevers, M. (1995). Relating $\mathcal{H}_2$ and $\mathcal{H}_\infty$ bounds for finite-dimensional systems. *Systems & Control Letters*, 24(3), 173–181.
- [41] de Oliveira, M., Bernussou, J., & Geromel, J. (1999). A new discrete-time robust stability condition. *Systems & Control Letters*, 37(4), 261–265.
- [42] Deshmukh, A. A., Dogan, U., & Scott, C. (2017). Multi-task learning for contextual bandits. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems*, volume 30: Curran Associates, Inc.
- [43] Dixit, R., Bedi, A. S., Tripathi, R., & Rajawat, K. (2019). Online Learning with Inexact Proximal Online Gradient Descent Algorithms. *IEEE Transactions on Signal Processing*, 67(5), 1338 – 1352.
- [44] Dong, Q., Li, L., Dai, D., Zheng, C., Ma, J., Li, R., Xia, H., Xu, J., Wu, Z., Chang, B., Sun, X., Li, L., & Sui, Z. (2024). A survey on in-context learning. In Y. Al-Onaizan, M. Bansal, & Y.-N. Chen (Eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing* (pp. 1107–1128). Miami, Florida, USA: Association for Computational Linguistics.

- [45] Dubey, S. R., Singh, S. K., & Chaudhuri, B. B. (2022a). Activation functions in deep learning: A comprehensive survey and benchmark. *Neurocomputing*, 503, 92–108.
- [46] Dubey, S. R., Singh, S. K., & Chaudhuri, B. B. (2022b). Activation functions in deep learning: A comprehensive survey and benchmark. *Neurocomputing*, 503, 92–108.
- [47] Eriksson, K., Estep, D., & Johnson, C. (2010). *Applied Mathematics: Body and Soul. Derivatives and geometry in IR_3* , volume 1. Springer Berlin Heidelberg.
- [48] Fazlyab, M., Paternain, S., Preciado, V. M., & Ribeiro, A. (2018). Prediction-Correction Interior-Point Method for Time-Varying Convex Optimization. *IEEE Transactions on Automatic Control*, 63(7), 1973–1986.
- [49] Fiedler, M. (2010). Notes on hilbert and cauchy matrices. *Linear Algebra and Its Applications - LINEAR ALGEBRA APPL*, 432, 351–356.
- [50] Fosson, S. M. (2021). Centralized and Distributed Online Learning for Sparse Time-Varying Optimization. *IEEE Transactions on Automatic Control*, 66(6), 2542–2557.
- [51] Frässle, S., Lomakina, E. I., Kasper, L., Manjaly, Z. M., Leff, A., Pruessmann, K. P., Buhmann, J. M., & Stephan, K. E. (2018). A generative model of whole-brain effective connectivity. *Neuroimage*, 179, 505–529.
- [52] Friston, K. J. (2011). Functional and effective connectivity: a review. *Brain connectivity*, 1(1), 13–36.
- [53] Friston, K. J., Harrison, L., & Penny, W. (2003). Dynamic causal modelling. *Neuroimage*, 19(4), 1273–1302.
- [54] Friston, K. J., Li, B., Daunizeau, J., & Stephan, K. E. (2011). Network discovery with dcm. *Neuroimage*, 56(3), 1202–1221.
- [55] Gao, X.-B. (2003). Exponential stability of globally projected dynamic systems. *IEEE Transactions on Neural Networks*, 14(2), 426–431.
- [56] Gindullina, E., Zorzi, M., Bertoldo, A., & Chiuso, A. (2021). Estimating effective connectivity using brain partitioning. In *2021 60th IEEE Conference on Decision and Control (CDC)* (pp. 1574–1579).

- [57] Gong, L., Pasqualetti, F., Papouin, T., & Ching, S. (2023). Astrocytes as a mechanism for meta-plasticity and contextually-guided network function. *Nature Communications*. Submitted. arXiv preprint arXiv:2311.03508.
- [58] Grimm, G., Teel, A. R., & Zaccarian, L. (2004). Linear lmi-based external anti-windup augmentation for stable linear systems. *Automatica*, 40(11), 1987–1996.
- [59] Gu, A. & Dao, T. (2023). Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.
- [60] Gu, A., Dao, T., Ermon, S., Rudra, A., & Ré, C. (2020). Hippo: Recurrent memory with optimal polynomial projections. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*.
- [61] Guerra-Gomes, S., Sousa, N., Pinto, L., & Oliveira, J. F. (2018). Functional roles of astrocyte calcium elevations: From synapses to behavior. *Frontiers in Cellular Neuroscience*, 11, 427.
- [62] Guo, T., Al Makdah, A. A., Krishnan, V., & Pasqualetti, F. (2023). Imitation and transfer learning for LQG control. *IEEE Control Systems Letters*, 7, 2149–2154.
- [63] Hall, E. C. & Willett, R. M. (2015). Online Convex Optimization in Dynamic Environments. *IEEE Journal of Selected Topics in Signal Processing*, 9(4), 647–662.
- [64] Hespanha, J. P. (2018). *Linear systems theory*. Princeton university press.
- [65] Horn, R. A. & Johnson, C. R. (2012). *Matrix analysis*. Cambridge university press.
- [66] Karami, M. & Mirrokni, V. (2025). Lattice: Learning to efficiently compress the memory.
- [67] Kavranoglu, D., Bettayeb, M., & Anjum, M. F. (1996). $\mathcal{L}_\infty$ norm simultaneous system approximation. *International Journal on Robust and Nonlinear Control*, 6(9-10), 999–1014.
- [68] Khalil, H. K. (2002). *Nonlinear Systems*. Prentice-Hall.

- [69] Komatsuzaki, A., Puigcerver, J., Lee-Thorp, J., Ruiz, C. R., Mustafa, B., Ainslie, J., Tay, Y., Dehghani, M., & Houlsby, N. (2023). Sparse upcycling: Training mixture-of-experts from dense checkpoints. In *The Eleventh International Conference on Learning Representations*.
- [70] Kothare, M. V., Campo, P. J., Morari, M., & Nett, C. N. (1994). A unified framework for the study of anti-windup designs. *Automatica*, 30(12), 1869–1883.
- [71] Kouvaritakis, B. & Cannon, M. (2016). Model predictive control. *Switzerland: Springer International Publishing*, 38, 13–56.
- [72] Lee, H., Soldaini, L., Cohan, A., Seo, M., & Lo, K. (2025). Routerretriever: Routing over a mixture of expert embedding models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(11), 11995–12003.
- [73] Lessard, L., Recht, B., & Packard, A. (2016). Analysis and Design of Optimization Algorithms via Integral Quadratic Constraints. *SIAM Journal on Optimization*, 26(1), 57–95.
- [74] Li, J., Xie, W., & Seco, L. (2025). *Dynamic Investment Strategies Through Market Classification and Volatility: A Machine Learning Approach*. Papers 2504.02841, arXiv.org.
- [75] Liao-McPherson, D., Nicotra, M., & Kolmanovsky, I. (2018). A Semismooth Predictor Corrector Method for Real-Time Constrained Parametric Optimization with Applications in Model Predictive Control. In *2018 IEEE Conference on Decision and Control (CDC)* (pp. 3600–3607).
- [76] Lin, T. W., Das, A., Krishnan, G. P., Bazhenov, M., & Sejnowski, T. J. (2017). Differential covariance: A new class of methods to estimate sparse connectivity from neural recordings. *Neural computation*, 29(10), 2581–2632.
- [77] Lindquist, A. & Picci, G. (2015). Linear stochastic systems. *Series in Contemporary Mathematics*, 1.
- [78] Lupien, J.-L. & Lesage-Landry, A. (2023). An Online Newton’s Method for Time-Varying Linear Equality Constraints. *IEEE Control Systems Letters*, 7, 1423–1428.

- [79] Magnus, J. R. (1985). On differentiating eigenvalues and eigenvectors. *Econometric theory*, 1(2), 179–191.
- [80] Mallya, A. & Lazebnik, S. (2018). Packnet: Adding multiple tasks to a single network by iterative pruning. In *IEEE Conf. on Computer Vision and Pattern Recognition* (pp. 7765–7773). Los Alamitos, CA, USA.
- [81] Marcenko, V. A. & Pastur, L. A. (1967). Distribution of eigenvalues for some sets of random matrices. *Mathematics of the USSR-Sbornik*, 1(4), 457.
- [82] Martensson, K. (2012). *Gradient Methods for Large-Scale and Distributed Linear Quadratic Control*. Doctoral thesis (compilation), Department of Automatic Control. Defence details Date: 2012-04-27 Time: 10:15 Place: Room M:B, M-building, Ole Römers väg 1, Lund University Faculty of Engineering External reviewer(s) Name: Bamieh, Bassam Title: Prof. Affiliation: University of California at Santa Barbara, USA —.
- [83] Mu, S. & Lin, S. (2025). A comprehensive survey of mixture-of-experts: Algorithms, theory, and applications.
- [84] Murphy-Royal, C., Ching, S., & Papouin, T. (2023). A conceptual framework for astrocyte function. *Nature Neuroscience*, 26(11), 1848–1856.
- [85] Natali, A., Coutino, M., Isufi, E., & Leus, G. (2021). Online Time-Varying Topology Identification Via Prediction-Correction Algorithms. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 5400–5404). Toronto, ON, Canada: IEEE.
- [86] Nocedal, J. & Wright, S. (2006). *Numerical optimization*. Springer series in operations research and financial engineering. New York, NY: Springer, 2. ed. edition.
- [87] Olsson, C., Elhage, N., Nanda, N., Joseph, N., DasSarma, N., Henighan, T., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., Drain, D., Ganguli, D., Hatfield-Dodds, Z., Hernandez, D., Johnston, S., Jones, A., Kernion, J., Lovitt, L., Ndousse, K., Amodei, D., Brown, T., Clark, J., Kaplan, J., McCandlish, S., & Olah, C. (2022). In-context learning and induction heads.

- [88] Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., & Wermter, S. (2019). Continual lifelong learning with neural networks: A review. *Neural Networks*, 113, 54–71.
- [89] Paskota, M., Sreeram, V., Teo, K. L., & Mees, A. I. (1994). Optimal simultaneous stabilization of linear single-input systems via linear state feedback control. *International Journal of Control*, 60(4), 483–498.
- [90] Paternain, S., Morari, M., & Ribeiro, A. (2019). Real-Time Model Predictive Control Based on Prediction-Correction Algorithms. In *2019 IEEE 58th Conference on Decision and Control (CDC)* (pp. 5285–5291).
- [91] Petersen, K. B. & Pedersen, M. S. (2012). *The Matrix Cookbook*. Technical University of Denmark.
- [92] Pióro, M., Ciebiera, K., Król, K., Ludziejewski, J., Krutul, M., Krajewski, J., Antoniuk, S., Miłoś, P., Cygan, M., & Jaszczur, S. (2024). Moe-mamba: Efficient selective state space models with mixture of experts.
- [93] Prando, G., Zorzi, M., Bertoldo, A., Corbetta, M., Zorzi, M., & Chiuso, A. (2020). Sparse DCM for whole-brain effective connectivity from resting-state fMRI data. *NeuroImage*, 208, 116367.
- [94] Richards, B., Tsao, D., & Zador, A. (2022). The application of artificial intelligence to biology and neuroscience. *Cell*, 185(15), 2640–2643.
- [95] Rodrigues, L. A., Oliveira, R. C. L. F., & Camino, J. F. (2015). New extended lmi characterization for state feedback control of continuous-time uncertain linear systems. In *2015 European Control Conference (ECC)* (pp. 1992–1997).
- [96] Scherer, C. (2001). Theory of robust control. *Delft University of Technology*, (pp. 1–160).
- [97] Scherer, C. W., Ebenbauer, C., & Holicki, T. (2023). Optimization Algorithm Synthesis based on Integral Quadratic Constraints: A Tutorial.
- [98] Selvaratnam, D., Shames, I., Manton, J. H., & Zamani, M. (2018). Numerical Optimisation of Time-Varying Strongly Convex Functions Subject to Time-Varying Constraints. In *2018 IEEE Conference on Decision and Control (CDC)* (pp. 849–854).

- [99] Seo, J., Kim, J., & Shin, H. (2025). MoFE: Mixture of frozen experts architecture. In W. Chen, Y. Yang, M. Kachuee, & X.-Y. Fu (Eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: Industry Track)* (pp. 340–348). Albuquerque, New Mexico: Association for Computational Linguistics.
- [100] Shalev-Shwartz, S. (2011). Online Learning and Online Convex Optimization. *Foundations and Trends® in Machine Learning*, 4(2), 107–194.
- [101] Shen, S.-Q., Huang, T.-Z., & Yu, J. (2010). Eigenvalue Estimates for Preconditioned Nonsymmetric Saddle Point Matrices. *SIAM Journal on Matrix Analysis and Applications*, 31(5), 2453–2476.
- [102] Simonetto, A. (2019). Dual Prediction–Correction Methods for Linearly Constrained Time-Varying Convex Programs. *IEEE Transactions on Automatic Control*, 64(8), 3355–3361.
- [103] Simonetto, A., Dall’Anese, E., Paternain, S., Leus, G., & Giannakis, G. B. (2020). Time-Varying Convex Optimization: Time-Structured Algorithms and Applications. *Proceedings of the IEEE*, 108(11), 2032–2048.
- [104] Simonetto, A. & Massioni, P. (2023). Optimization Filters for Stochastic Time-Varying Convex Optimization. In *2023 European Control Conference (ECC)* (pp. 1–6).
- [105] Simonetto, A. & Massioni, P. (2024). Nonlinear optimization filters for stochastic time-varying convex optimization. *International Journal of Robust and Nonlinear Control*, 34(12), 8065–8089.
- [106] Simonetto, A., Mokhtari, A., Koppel, A., Leus, G., & Ribeiro, A. (2016). A Class of Prediction-Correction Methods for Time-Varying Convex Optimization. *IEEE Transactions on Signal Processing*, 64(17), 4576–4591.
- [107] Soare, M., Alsharif, O., Lazaric, A., & Pineau, J. (2014). Multi-task linear bandits. In *NIPS2014 Workshop on Transfer and Multi-task Learning: Theory meets Practice*.
- [108] Sun, S. & Mo, Y. (2022). Fundamental identification limit on diagonal canonical form for siso systems. In *2022 IEEE 17th International Conference on Control & Automation (ICCA)* (pp. 728–733).

- [109] Tombs, M. S. & Postlethwaite, I. (1987). Truncated balanced realization of a stable non-minimal state-space system. *International Journal of Control*, 46, 1319–1330.
- [110] Tran, C. H. T., Peringod, G., & Gordon, G. R. (2018). Astrocytes integrate behavioral state and vascular signals during functional hyperemia. *Neuron*, 100(5), 1133–1148.e3.
- [111] Trefethen, L. N. & Embree, M. (1999). *Spectra and Pseudospectra*, (pp. 217–250). Springer Berlin Heidelberg: Berlin, Heidelberg.
- [112] Trefethen, L. N. & Embree, M. (2005). *Spectra and Pseudospectra: the Behavior of Nonnormal Matrices and Operators*. Princeton University Press.
- [113] van Schuppen, J. H. (2021). *Control and system theory of discrete-time stochastic systems*. Springer.
- [114] Van Scoy, B. & Lessard, L. (2022). A universal decomposition for distributed optimization algorithms. *IEEE Control Systems Letters*, 6, 3044–3049.
- [115] Vanbiervliet, J., Vandereycken, B., Michiels, W., Vandewalle, S., & Diehl, M. (2009). The smoothed spectral abscissa for robust stability optimization. *SIAM Journal on Optimization*, 20(1), 156–171.
- [116] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.
- [117] Wang, J. & Elia, N. (2010). Control approach to distributed optimization. In *2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)* (pp. 557–561).
- [118] Wang, K. A., Shi, J., & Fox, E. B. (2025). Test-time regression: a unifying framework for designing sequence models with associative memory.
- [119] Werbos, P. J. (1989). Neural networks for control and system identification. In *IEEE Conf. on Decision and Control* (pp. 260–265). Tampa, FL, USA.
- [120] Xin, L., Ye, L., Chiu, G., & Sundaram, S. (2022). Identifying the dynamics of a system by leveraging data from similar systems. In *American Control Conference* (pp. 818–824). Atlanta, GA, USA.

- [121] Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A review of recurrent neural networks: Lstm cells and network architectures. *Neural computation*, 31(7), 1235–1270.
- [122] Zancato, L., Seshadri, A., Dukler, Y., Golatkar, A., Shen, Y., Bowman, B., Trager, M., Achille, A., & Soatto, S. (2024). B'MOJO: Hybrid state space realizations of foundation models with eidetic and fading memory. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- [123] Zhang, T. T., Kang, K., Lee, B. D., Tomlin, C., Levine, S., Tu, S., & Matni, N. (2022). Multi-task imitation learning for linear dynamical systems. *arXiv preprint arXiv:2212.00186*.
- [124] Zhang, Y., Dall'Anese, E., & Hong, M. (2021). Online Proximal-ADMM for Time-Varying Constrained Convex Optimization. *IEEE Transactions on Signal and Information Processing over Networks*, 7, 144–155.
- [125] Zhou, K. & Doyle, J. C. (1998a). *Essentials of Robust Control*. Prentice-Hall.
- [126] Zhou, K. & Doyle, J. C. (1998b). *Essentials of robust control*, volume 104. Prentice Hall Upper Saddle River, NJ.