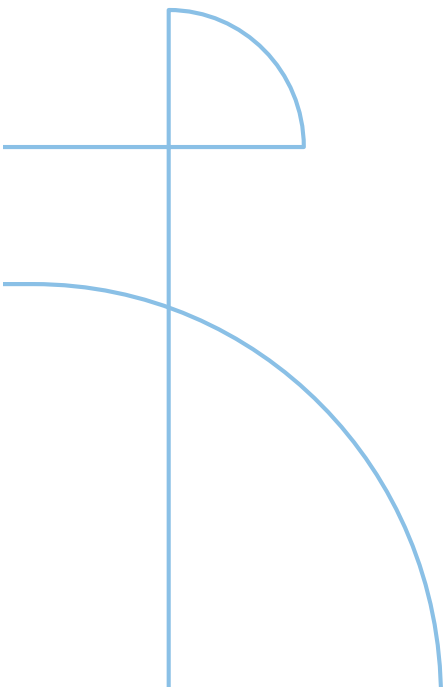
Licentiate Thesis in Electrical Engineering

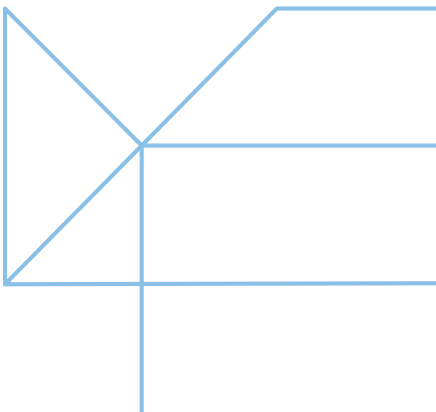
Data-Driven Methods for Learning and Control of Dynamical Systems:

Robustness and Explainability Aspects

MARGARITA ANDREA GUERRERO SALAZAR

KTH ROYAL INSTITUTE OF TECHNOLOGY





Data-Driven Methods for Learning and Control of Dynamical Systems:

Robustness and Explainability Aspects

MARGARITA ANDREA GUERRERO SALAZAR

Academic Dissertation which, with due permission of the KTH Royal Institute of Technology, is submitted for public defence for the Degree of Licentiate of Engineering on Friday, the 6th November 2026, at 09:00 in Harry Nyquist, Malvinas väg 10, Stockholm.

Licentiate Thesis in Electrical Engineering
KTH Royal Institute of Technology
Stockholm, Sweden 2026

© Margarita Andrea Guerrero Salazar

TRITA-EECS-AVL-2026:88
ISBN 978-91-8106-719-4

Printed by Universitetsservice US-AB, Sweden 2026

Abstract

Engineering systems are often analyzed and operated using models that are incomplete, uncertain, or, in some cases, unavailable. In control engineering, the analysis of important system properties commonly relies on approximate parametric models, typically obtained through a prior system identification step. This thesis develops data-driven methods for characterizing behaviorally significant perturbations in dynamical systems and machine learning models using experimental observations rather than complete parametric descriptions.

The first part of the thesis concerns robustness analysis under model uncertainty. Given an unknown dynamical system, the objective is to estimate control-relevant quantities directly from input–output data. In particular, we develop methods based on sequential input design to characterize structured perturbations that may destabilize a system and to quantify the mismatch between a nominal model and the true plant from a closed-loop perspective. This information makes it possible to assess whether stability and performance conclusions obtained from the nominal model remain valid for the actual system.

In the second part of the thesis, we study the complementary problem of identifying minimal perturbations capable of producing a desired change in the behavior of a black-box model. Inspired by explainable artificial intelligence, we develop an adaptive algorithm for identifying feasible and actionable modifications using a limited number of model evaluations. We further apply this method to controller fine-tuning, where the objective is to identify small changes in controller parameters that lead to satisfactory closed-loop behavior while respecting operational constraints.

Together, these contributions show how task-relevant information about system sensitivity can be extracted directly from data, without reconstructing a complete model of the underlying system.

Sammanfattning

Tekniska system analyseras och drivs ofta med hjälp av modeller som är ofullständiga, osäkra eller i vissa fall inte tillgängliga. Inom reglertekniken bygger analysen av viktiga systemegenskaper vanligtvis på approximativa parametriska modeller, vilka typiskt erhålls genom ett föregående systemidentifieringssteg. Denna avhandling utvecklar datadrivna metoder för att karakterisera beteendemässigt betydelsefulla perturbationer i dynamiska system och maskininlärningsmodeller, med hjälp av experimentella observationer snarare än fullständiga parametriska beskrivningar.

Avhandlingens första del behandlar robusthetsanalys under modellosäkerhet. För ett okänt dynamiskt system är målet att uppskatta reglerrelevanta storheter direkt från in- och utdata. I synnerhet utvecklar vi metoder baserade på sekventiell insignaldesign för att karakterisera strukturerade perturbationer som kan destabilisera ett system samt för att kvantifiera avvikelsen mellan en nominell modell och den verkliga processen ur ett återkopplingsperspektiv. Denna information gör det möjligt att bedöma huruvida slutsatser om stabilitet och prestanda som erhållits från den nominella modellen även gäller för det verkliga systemet.

I avhandlingens andra del studerar vi det kompletterande problemet att identifiera minimala perturbationer som kan åstadkomma en önskad förändring i beteendet hos en svartlådemodell. Inspirerade av förklarbar artificiell intelligens utvecklar vi en adaptiv algoritm för att identifiera genomförbara och handlingsbara förändringar med hjälp av ett begränsat antal modellutvärderingar. Vi tillämpar vidare metoden på finjustering av regulatorer, där målet är att identifiera små förändringar i regulatorparametrarna som leder till ett tillfredsställande beteende i sluten slinga samtidigt som driftbegränsningar respekteras.

Sammantaget visar dessa bidrag hur uppgiftsrelevant information om systems känslighet kan utvinnas direkt ur data, utan att en fullständig modell av det underliggande systemet behöver rekonstrueras.

Different roads sometimes lead to the same castle.
George R.R. Martin

To my father, Ricardo Alexis Guerrero Álvarez.

Acknowledgment

Writing acknowledgements is difficult, as many people have shaped these years in ways they may not even realize. I have therefore chosen to omit most names, but every sentence below is written with particular people and memories in mind.

First, I would like to express my deepest gratitude to my supervisor, Cristian R. Rojas. Thank you for trusting me, and for sharing your knowledge so generously throughout these years. I am especially grateful for the freedom you have given me to develop my own ideas and find my own path as a researcher, while always being there when I needed guidance.

I am particularly grateful to those who accompanied me during my first year of the PhD. Some of you are still here, while others have already moved on to new chapters elsewhere. I look back on that first year with a certain melancholy, knowing that those days cannot quite be repeated, but also with enormous gratitude for having shared them with you.

Outside the PhD, I have been incredibly fortunate to meet people who have made my life in Sweden much warmer and more joyful. To my Swedish friends, and especially to *The convo*, I appreciate you all from the bottom of my heart. To my *Latin American* friends, thank you for bringing a little piece of home to a country so far away from where many of us began.

I wish to thank my friends in Chile, especially Sebastián José Muñoz Córdova. I honestly do not know what these last two years would have been like without your advice and your company. You have listened to me laugh, get angry, and cry, and I owe you more than I could ever properly express. Thank you to the *Amigurrumis* and to *501* for still being present in my life after so many years.

Finally, I want to thank my family. To my Papá, you taught me the meaning of hard work, sacrifice, perseverance, and responsibility long before I understood how important those values would become in my own life. You have supported me and believed in me through every stage of this journey, and I would need several lifetimes to repay even a fraction of everything you have given me. Thank you, Mamá, for encouraging me when I needed it, for your advice, and for your company, which I have always treasured with so much love. I still listen to your music while I work on my research, and in those moments, I always feel that you are with me. Thank you to my brothers, Ricardo and Cristian, for the laughter and companionship throughout all these years. To my niece, Magdalena, and my sister-in-law, Natalia, I hold you both very close to my heart. And to the entire Guerrero and Salazar family, I love you all.

MAGS
Stockholm, November 2026.

Contents

Abstract . . . iii

Sammanfattning . . . v

Acknowledgment . . . ix

Contents . . . xi

List of Figures . . . xiii

List of Tables . . . xv

1 Introduction . . . 1
- 1.1 Motivation . . . 4
- 1.2 Literature Review . . . 5
- 1.3 Thesis Outline and Contributions . . . 7
- 1.4 Declaration of Use of Generative AI . . . 9

2 Preliminaries . . . 11
- 2.1 Robustness Analysis of Uncertain Systems . . . 12
- 2.2 Iterative Methods . . . 15
- 2.3 Explainable Artificial Intelligence . . . 19

I Data-Driven Estimation of Robustness Measures 23

3 Structured Robustness: Data-Driven Estimation of the Structured Singular Value . . . 25
- 3.1 Introduction . . . 25
- 3.2 Problem Statement . . . 26
- 3.3 Preliminaries . . . 28
- 3.4 Proposed Approach . . . 30
- 3.5 Experiments . . . 32
- 3.6 Summary and Future Work . . . 37

4 Model Uncertainty: Data-Driven Estimation of the SISO Vinnicombe Metric 39
- 4.1 Introduction . . . 39
- 4.2 Problem Statement . . . 41
- 4.3 Proposed Approach . . . 41
- 4.4 Experiments . . . 45

4.5 Summary and Conclusions . . . 49

5 MIMO Robustness: Krylov-Based Estimation of the Vinnicombe Metric . . . 51
5.1 Introduction . . . 51
5.2 Problem Statement . . . 52
5.3 Preliminaries . . . 53
5.4 Proposed Approach . . . 57
5.5 Experiments . . . 66
5.6 Summary and Future Work . . . 70

II Counterfactual Explanations and Safe Perturbation Search 71

6 Counterfactual Explanations: Adaptive Search in Black-Box Models . . . 73
6.1 Introduction . . . 74
6.2 Problem Statement . . . 75
6.3 ACE Algorithm . . . 75
6.4 Quantitative Evaluation . . . 79
6.5 Adaptive Control Tuning . . . 82
6.6 Qualitative Analysis . . . 84
6.7 Conclusions and Future Work . . . 85
6.A Desiderata for Counterfactual Explanations . . . 87
6.B Extended Cost Function . . . 87
6.C Class-1 Posterior Calculations . . . 88
6.D Delta Method for cross-covariance term . . . 89
6.E Branch and Bound Example . . . 90
6.F Datasets . . . 90
6.G System Identification . . . 93

7 Data-Driven Controller Tuning: Counterfactual Search for an Axial Compressor . . . 97
7.1 Introduction . . . 97
7.2 Process Description and Problem Formulation . . . 99
7.3 CFE Approach to Sample-Efficient Control Tuning . . . 102
7.4 Results and Discussions . . . 106
7.5 Conclusion and Future Work . . . 111

8 Summary and Future Research Directions . . . 113
8.1 Summary . . . 113
8.2 Future Research Directions . . . 114

Bibliography . . . 117

List of Figures

2.1.1 General uncertain-system interconnection. 12
2.1.2 Standard feedback configuration. 14
2.2.3 Relation between the signals u, e and y associated with the system G. . 16

3.2.1 Interconnection of the nominal system $\mathbf{G}_0$ and the uncertainty block $\mathbf{\Delta}$. 27
3.5.2 Comparison between $\mu_{\Delta}(\mathbf{G}_0)$ and N. Solid line: $\tilde{\mu}$, dashed line: $\bar{\mu}$, dotted line: `mussv`. From top to bottom, the plots correspond to systems with 1 block, 2 blocks, and 3 blocks, respectively. 35
3.5.3 Convergence of the lower bound over iterations (l). Solid line: $\tilde{\mu}$, dashed line: $\bar{\mu}$, dotted line: last iteration. 36
3.5.4 Frequency response of the structured singular value. 36
3.5.5 Percentage of converging cases of the data-driven power method. . . . 37

4.4.1 Estimate of the ν-gap $\hat{\delta}_\nu$ (black, dashed), average of 100 Monte Carlo (MC) simulations (blue, solid), MATLAB reference δ_ν (red, dashed), and 95% MC bounds (gray shaded area). 46
4.4.2 Estimate of the ν-gap $\hat{\delta}_\nu$ (black, dashed), average of 100 MC simulations (blue, solid), MATLAB reference δ_ν (red, dashed), and 95% MC bounds (gray shaded area). 47
4.4.3 Estimate of the ν-gap $\hat{\delta}_\nu$ (black, dashed), average of 100 MC simulations (blue, solid), MATLAB reference δ_ν (red, dashed), and 95% MC bounds (gray shaded area). 48

5.5.1 Estimation of $\delta_\nu(\mathbf{G}_0, \mathbf{G}_1)$. Approach #1 estimate (blue, solid), approach #2 estimate (dark-green, solid), and MATLAB reference δ_ν (red, dashed). 69
5.5.2 Estimation of $\delta_\nu(\mathbf{G}_0, \mathbf{G}_2)$. Approach #1 estimate (blue, solid), approach #2 estimate (dark-green, solid), and MATLAB reference δ_ν (red, dashed). 69

6.6.1 Box plot of component-wise shifts $\mathbf{\Delta} = \boldsymbol{\theta}_{\text{cand}} - \boldsymbol{\theta}_0$ across 100 ACE runs. 84
6.6.2 Step responses: baseline $\boldsymbol{\theta}_0$ (red) vs. one representative CFE (blue). . . 84
6.6.3 First: ACE on Make Moons. Second: GS on Make Moons. Third-Fifth: MNIST results (original, ACE CFE, and OmniXAI CFE). 85
6.E.4 Branch and Bound method with L-BFGS-B Initialization. 91
6.G.5 Closed-loop system setting, noiseless input, noisy output case. 94

7.2.1 Schematic of the compressor system geometry. 100
7.2.2 Cascaded system description for IGV control, with $C_1(q)$, $C_2(q)$, $G_1^{\mathsf{d}}(q)$ and $G_2^{\mathsf{d}}(q)$ being the outer controller, inner controller, compressor-plenum system, and IGV actuator, respectively. 101
7.4.3 P&ID of the compressor system. 107
7.4.4 Case 1 - Box plot of component-wise shifts $\mathbf{\Delta} = \boldsymbol{\theta}_{\text{cand}} - \boldsymbol{\theta}_0$ across 100 ACE runs. 109

7.4.5 Case 1 - Step responses: baseline $\boldsymbol{\theta}_0$ (red) vs. one representative CFE (blue). 109
7.4.6 Case 2 - Box plot of component-wise shifts $\boldsymbol{\Delta} = \boldsymbol{\theta}_{\text{cand}} - \boldsymbol{\theta}_0$ across 100 ACE runs. 110
7.4.7 Case 2 - Step responses: baseline $\boldsymbol{\theta}_0$ (red) vs. one representative CFE (blue). 111

List of Tables

3.5.1 Uncertainty structures $\mathbf{\Delta} \in \mathbb{C}^{3\times 3}$. 34
3.5.2 Structured singular values, and peak frequencies, for different noise levels. 36

6.4.1 Summary of real-world datasets. 79
6.4.2 Fixed test for continuous datasets with 100 tested points per experiment. Best CFE score are highlighted. 81
6.4.3 Fixed test for heterogeneous datasets with 100 tested points per experiment. Best CFE score are highlighted. 81
6.4.4 Mixed test for continuous datasets. Best CFE scores are highlighted. . . 82
6.4.5 Mixed test for heterogeneous datasets. Best CFE scores are highlighted. 82
6.F.6 Hyperparameter ranges explored. 92
6.F.7 Summary of Hyperparameters. 93
6.F.8 Fixed test for continuous datasets with 100 tested points per experiment. The best Score CFE values are highlighted. 94
6.F.9 Fixed test for heterogeneous datasets with 100 tested points per experiment. The best Score CFE values are highlighted. 94
6.F.10 Mixed test for continuous datasets with 100 tested points. The best Score CFE values are highlighted. 95
6.F.11 Mixed test for heterogeneous datasets with 100 tested points. The best Score CFE values are highlighted. 96

7.4.1 Case 1 (single loop): summary over 100 ACE runs per instance. 108
7.4.2 Case 2 (cascade): summary over 100 ACE runs per instance. 111

1 Introduction

Throughout the history of science, empirical observations have been transformed into knowledge through the construction of models, which provide useful representations of the phenomena from which those observations arise [1]. Models play related but distinct roles across scientific disciplines. In the natural sciences, they are primarily used to explain phenomena, test hypotheses, and improve our understanding of the physical world. In engineering and other applied sciences, models are additionally used to guide intervention: they support prediction, design, optimization, and the manipulation of physical processes toward desired outcomes. This predictive and operational role is particularly important when the object under study is viewed as a *system* [2].

Broadly speaking, a system is an entity composed of interacting components whose collective behavior gives rise to observable phenomena. From an input–output perspective, a system receives external stimuli, referred to as *inputs*, and produces measurable responses, referred to as *outputs* [3]. Some external variables can be manipulated by an experimenter or decision maker, whereas others act as uncontrolled disturbances. When the response of the system depends not only on its current inputs but also on its past evolution, the system is said to be *dynamical* [4].

Mathematical models provide formal descriptions of such systems by expressing relations among their inputs, internal variables, disturbances, and outputs. Depending on the application and the available knowledge, these descriptions may take the form of differential or difference equations, state-space representations, transfer functions, statistical models, or game-theoretic models [4, 5, 6]. A model is not the system itself, but an abstraction whose usefulness depends on its assumptions, the information employed in its construction, and the task for which it is intended.

In favorable cases, a mathematical model of a dynamical system can be derived from first principles by combining established physical laws, such as conservation

relations, Newton's laws, or Kirchhoff's laws. Models obtained in this way often possess variables and parameters with a direct physical interpretation. However, for many modern systems, the underlying mechanisms may be only partially understood, too complex to describe accurately, or influenced by interactions that are difficult to capture from first principles.

An alternative is to learn a model from experimental observations. In the context of dynamical systems, the field concerned with constructing mathematical models from measured input and output data is known as *system identification* [5, 7]. Rather than deriving the complete behavior of a process solely from physical principles, system identification uses experimental data to estimate a model within a prescribed model class. Prediction-error, maximum-likelihood, and subspace methods are prominent examples of data-driven techniques that have become standard tools for dynamical modeling [5, 8]. This data-driven modeling paradigm has become a standard component of systems and control engineering.

In control engineering, however, constructing a model is rarely the final objective. Rather, models are used to analyze the behavior of a process and to design mechanisms that drive the system toward a desired outcome. Such objectives may include stability, reference tracking, disturbance rejection, constraint satisfaction, or the optimization of a prescribed performance criterion. In the classical model-based paradigm, experimental observations are first used to obtain a mathematical model, which is subsequently employed for system analysis or controller design. This gives rise to an indirect data-driven approach [9],

$$\textit{experimental data} \longrightarrow \textit{identified model} \longrightarrow \textit{analysis or control design}.$$

More recently, increasing attention has been devoted to direct data-driven methods, which seek to infer system properties or determine decisions directly from measured data, without first identifying a complete parametric representation of the underlying dynamics [10, 11]. The motivation for bypassing explicit identification is not merely computational: recovering a complete model may be unnecessary when only a specific task-relevant property or decision is required. Moreover, identification may be difficult when experiments are costly, measurements are noisy, the available data are insufficiently informative, or the underlying dynamics cannot be represented reliably within the selected model class.

Despite the significant progress made in data-driven analysis and control, uncertainty remains unavoidable. Experimental data are necessarily finite, may be corrupted by noise, and generally represent only a limited range of operating conditions. Consequently, any nominal, identified, or implicitly data-based description of a dynamical system may differ from the actual process. In control applications, this discrepancy raises a fundamental question: *whether conclusions about stability and performance obtained from the available information will remain valid when the true system departs from its assumed representation.*

Robust control addresses this problem by studying whether the stability and performance of a closed-loop system (that is, the feedback interconnection between a dynamical system and its controller) are preserved in the presence of modeling errors, unmodeled dynamics, parameter variations, and other sources of uncertainty [12, 13].

In this context, robustness characterizes the *sensitivity* of the closed-loop behavior to discrepancies between the nominal model, i.e., the mathematical approximation used for analysis or controller design, and the actual system. Different robustness quantities capture complementary aspects of this sensitivity: induced system gains, such as the H_∞-norm, characterize worst-case amplification; the structured singular value quantifies the size of structured perturbations that may compromise robust stability; and plant-to-plant distances such as the ν-gap characterize model mismatch from a closed-loop perspective. Even though these quantities characterize the perturbations (together with their relevant magnitudes or structures) that may degrade performance or threaten stability, their conventional computation generally relies on explicit mathematical models of the system and its uncertainty. Constructing such models may be unnecessarily demanding when only a specific robustness-related quantity is sought. This motivates the development of data-driven robustness methods that estimate control-relevant measures of uncertainty and model mismatch directly from input–output experiments, similar to direct data-driven approaches.

This robustness perspective provides a natural conceptual connection to the interpretation of models learned from data. Robustness analysis asks how changes in an uncertain dynamical system can alter stability or performance. A related question arises when considering the *explainability* of machine learning predictors [14, 15]: how changes in the input, model parameters, or surrounding conditions affect the prediction produced by a data-driven model. Although the two problems have different objectives and mathematical formulations, both concern the sensitivity of an observable outcome to modifications of a system whose internal description is unavailable, incomplete, or too complex to analyze directly. In robust control, the outcome of interest may be closed-loop stability or performance; in explainability, it is the decision produced by a predictive model.

This interpretability challenge becomes particularly relevant for modern machine learning models, which can exhibit highly complex input–output relations while remaining difficult to interpret or too complicated to provide a transparent account of individual predictions. Thus, even when a model achieves high predictive accuracy, observing its output does not necessarily explain why a particular decision was made or how it could be changed. Explainable artificial intelligence seeks to address this limitation by providing interpretable information about the behavior of learned models.

Among the available approaches, *counterfactual explanations* provide an especially actionable form of explanation [16]. Rather than only attributing importance to individual input features, a counterfactual explanation identifies a small and feasible modification of an input that changes the model prediction to a desired outcome. In this sense, counterfactual explanations characterize prediction sensitivity by determining which changes are sufficient to alter an observed decision. This perspective is also relevant for dynamical-system applications, where the input features may represent operating conditions or controller parameters, and the desired outcome may correspond to satisfying prescribed performance requirements.

However, generating counterfactual explanations can require a large number of evaluations of the underlying model. This becomes particularly challenging when an

unknown system can only be evaluated through costly input–output experiments, either because individual evaluations are computationally expensive or because they involve a physical process, where interruptions are costly and aggressive excitation may compromise normal operation. Moreover, the resulting modifications must remain feasible, actionable, and consistent with the domain in which the model is deployed. A central challenge is therefore to obtain meaningful counterfactual explanations from a limited number of adaptively selected observations.

The challenges described above are addressed in this thesis. The overarching objective is to develop data-driven methods for extracting task-relevant information from dynamical systems and machine learning models whose internal descriptions are unavailable, incomplete, or unnecessarily expensive to reconstruct. The first research direction concerns the direct estimation of control-relevant robustness quantities from experimental input–output data, whereas the second research direction concerns how to identify minimal, feasible, and actionable changes capable of modifying a model prediction, thus explaining how a desired outcome can be achieved. Although the two research directions involve different mathematical objects and objectives, they share a common methodological principle: rather than reconstructing a complete description of the underlying system, the available data are used to recover the specific information required to understand, analyze, or modify its behavior.

1.1 Motivation

In our setting, we consider a critical process or a model of particular interest to its user, where we are interested in the *characterization of behaviorally significant perturbations*: small changes in the system, its model, operating conditions, inputs, or controller parameters that may lead to significant variations in its behavior. Such variations may include loss of stability, performance degradation, changes in the output signals, or transitions between operating regimes. For example, in an industrial facility—such as a compressor in a natural-gas transmission process or a gas turbine in a power plant—it may be important to determine how modeling errors or unmodeled dynamics affect the performance of a controller operating in feedback, or how the real system can be excited without exposing it to unacceptable operating conditions, while still obtaining informative data and satisfying prescribed control specifications. The common objective is therefore to understand which changes are behaviorally significant while respecting the physical and operational limitations of the process.

When the object under study is unknown and accessible only through input–output measurements or black-box queries, behaviorally significant perturbations cannot generally be characterized from an explicit description. Instead, they must be inferred by interacting with the system. In the setting considered in this thesis, data are collected sequentially: at each iteration, an input signal, query, or candidate modification is selected using the information obtained from previous observations. Moreover, the observations can be used to update the available knowledge of the black-box behavior and guide the following interaction. The objective is not necessarily to identify a complete model, but to recover the specific property or modification required by the problem at hand. This model-free perspective is

motivated by the fact that the system under study is unknown and that experiments on critical processes may be costly or operationally constrained.

To make this setting concrete, let $\mathbf{G}_0$ denote an available nominal model and let $\mathbf{G}$ denote the unknown true system. Both may be interconnected with a controller $\mathbf{C}(\boldsymbol{\theta})$, parameterized by $\boldsymbol{\theta} \in \Theta$. We use $\delta(\mathbf{G}_0, \mathbf{G})$ to denote a task-relevant characterization of the discrepancy between the nominal and actual systems; depending on the problem, this may represent a plant-to-plant distance, a measure associated with structured uncertainty, or another control-relevant quantity. For a fixed controller, the interest lies in determining whether conclusions obtained from the nominal interconnection $[\mathbf{G}_0, \mathbf{C}(\boldsymbol{\theta})]$ remain valid for the actual interconnection $[\mathbf{G}, \mathbf{C}(\boldsymbol{\theta})]$. Conversely, for a fixed system, one may investigate how an initial decision or controller configuration $\boldsymbol{\theta}_0$ should be modified in order to produce a desired change in the observed behavior. In both cases, the required information is inferred from sequentially collected observations rather than from a complete model of $\mathbf{G}$.

The preceding discussion motivates the following research questions.

> **RQ1.** How can the robustness of an unknown dynamical system against potentially destabilizing perturbations be assessed directly from input–output experiments?
>
> **RQ2.** Can an intuitive data-driven measure quantify how much mismatch between a nominal model and the true plant can be tolerated by a controller designed for the nominal model?
>
> **RQ3.** Can combinations of controller parameters associated with safe and unsafe closed-loop behavior be identified using only a limited number of simulations or queries to a black-box model?
>
> **RQ4.** How can these methods be applied when experiments on the real system are costly, access is limited, and operation is subject to stability and physical constraints?

These research questions guide the contributions developed in this thesis. The relevant literature is reviewed next, while Section 1.3 summarizes how each question is addressed and identifies the corresponding chapters.

1.2 Literature Review

Over the last decades, the increasing availability of experimental and computational data has stimulated a substantial shift toward data-driven methods in systems and control, where models, system properties, or control decisions can be inferred directly from observations without necessarily relying on a complete parametric description [11, 17]. This development has been accompanied by a broad family of iterative approaches, including iterative learning control, iterative feedback tuning, and iterative identification and control [18, 19, 20]. A common principle behind these methods is that information gathered from previous experiments is used to refine subsequent inputs, models, or control decisions. Such sequential use of data also establishes a close connection between iterative methods and experiment design, where new experiments can be adapted toward the particular system property or

control objective of interest [21]. Importantly, this iterative viewpoint also provides a unifying methodological thread throughout this thesis: the robustness problems considered here rely on iterative numerical ideas such as power methods and Krylov-subspace methods, whereas the explainability problems are approached through sequential acquisition strategies that iteratively select informative queries and refine the solution.

This perspective is particularly attractive in robust control, where many fundamental quantities originate from model-based formulations. Classical examples include the structured singular value, which explicitly accounts for the structure of model uncertainty [22], and the ν-gap, which provides a control-relevant measure of discrepancy between dynamical systems [23]. Interestingly, iterative numerical ideas already play an important role in the computation of such quantities: Packard, Fan, and Doyle proposed a power method for computing lower bounds on the structured singular value in [24]. At the same time, several approaches address the quantification of model uncertainty from experimental data, including parametric model-error descriptions, nonparametric frequency-domain models, and model-validation methods [25, 26, 27]. These approaches, however, may require the selection of an intermediate model structure, rely on asymptotic assumptions, suffer from finite-frequency-grid effects, or require specifically designed validation experiments [28]. The data-driven perspective takes a different route by seeking the robustness quantity itself directly from experiments. In particular, Wahlberg et al. developed nonparametric gain estimation through iterative experiments [29], while Rojas et al. analyzed an iterative procedure for estimating the H_∞ norm, establishing its stochastic properties and interpreting the successive experiments from an experiment-design perspective [30]. Oomen et al. subsequently extended this framework to multivariable systems, where the iteration also identifies the worst-case input and output directions [31]. Related developments include data-driven computation of the gap metric and the optimal stability margin [32], as well as more general formulations of stabilization, optimality, and robustness directly in terms of measured data [11].

A similar iterative perspective emerges in explainable artificial intelligence (XAI), where increasingly complex predictive models motivate methods for interpreting individual decisions and understanding how those decisions change under perturbations. Widely used approaches such as LIME and SHAP provide local explanations of model predictions [33, 34], and related ideas increasingly appear in control to interpret control actions, data-driven policies, and closed-loop decisions [35, 36, 37]. Within this broader landscape, counterfactual explanations provide an action-oriented perspective by identifying minimal and feasible changes that produce a desired model outcome [16, 38]. Their computation naturally gives rise to iterative search procedures, where candidate solutions are successively generated, evaluated, and refined. Existing approaches include evolutionary and constraint-guided methods [39, 40], as well as surrogate-based strategies such as BayCon [41], which uses Bayesian optimization and previously evaluated counterfactuals to guide subsequent queries. This sequential use of information becomes particularly important when black-box evaluations are costly, motivating adaptive acquisition methods that concentrate queries on informative regions of the decision boundary [42]. In control applications, the same iterative principle enables counterfactual

search to move beyond prediction explanation toward controller adjustment, where successive experiments seek the smallest parameter modification that satisfies prescribed closed-loop specifications [43].

1.3 Thesis Outline and Contributions

The material presented in this thesis is based on articles that are published, submitted for publication, or in preparation; journal articles and conference papers are denoted by [J] and [C], respectively. These contributions are organized into several chapters, whose connections and individual contributions are outlined below in relation to the relevant literature and the research questions introduced earlier.

Chapter 2

In this chapter, the notation and preliminaries required to follow the developments of the thesis are introduced. The chapter first reviews fundamental concepts in robust control [12, 23], and then introduces iterative numerical methods underlying the data-driven procedures developed in subsequent chapters, with particular emphasis on power iterations and Krylov-subspace methods [44, 45]. Finally, the main concepts of Explainable Artificial Intelligence required for the second part of the thesis are presented [46].

Part I. Data-Driven Estimation of Robustness Measures

Chapter 3

Chapter 3 addresses **RQ1** by introducing a fully data-driven power iteration method for estimating a lower bound on the structured singular value directly from input–output experiments. This enables the vulnerability of an unknown dynamical system to structured uncertainty to be assessed without an explicit parametric model. The results presented in this chapter are based on:

[J1] M. A. Guerrero, B. Lakshminarayanan and C. R. Rojas. Data-Driven Estimation of Structured Singular Values. In IEEE Control Systems Letters, vol. 9, pp. 1976-1981, 2025.

Chapter 4

Chapter 4 addresses **RQ2** in the single-input single-output (SISO) setting by developing an identification-free iterative method for estimating the ν-gap directly from time-domain input–output data. The method provides a control-relevant quantification of the mismatch between a nominal model and an unknown plant, and includes a data-driven verification of the associated admissibility conditions. The results presented in this chapter are based on:

[J2] M. A. Guerrero, H. Sandberg and C. R. Rojas. Data-Driven Estimation of Vinnicombe Metric. In IEEE Control Systems Letters, vol. 10, pp. 1105-1110, 2026.

Chapter 5

Chapter 5 extends the study of **RQ2** to multi-input multi-output (MIMO) systems. The ν-gap estimation problem is formulated through matrix-free finite-horizon operator actions, and two iterative estimators based on power iterations and Golub–Kahan–Lanczos bidiagonalization are developed. The results presented in this chapter are based on the following manuscript:

> M. A. Guerrero and C. R. Rojas. On Data-Driven Estimation of the ν-gap for MIMO Systems. In preparation.

Part II. Counterfactual Explanations and Safe Perturbation Search

Chapter 6

Chapter 6 addresses **RQ3** by introducing Adaptive Sampling for Counterfactual Explanations (ACE), a query-efficient method that combines Gaussian-process surrogate modeling and Bayesian optimization to sequentially select informative queries. The method seeks minimal, feasible, and actionable counterfactual explanations using limited access to a black-box model. The results presented in this chapter are based on:

[C1] M. A. Guerrero and C. R. Rojas. ACE: Adapting sampling for Counterfactual Explanations. In Proceedings of The 8th Annual Learning for Dynamics and Control Conference, PMLR 331:242-264, 2026.

Chapter 7

Chapter 7 addresses **RQ4** by applying counterfactual search to the retuning of an axial-compressor pressure-control loop under limited and constrained experimentation. Building on ACE, the chapter seeks minimal and operator-actionable controller modifications through a small number of closed-loop evaluations while restricting the search to an operationally relevant region. The results presented in this chapter are based on:

[J3] M. A. Guerrero, R. A. González, and C. R. Rojas. Sample-efficient counterfactual tuning for compressor pressure control. In Journal of Process Control, vol. 162, p. 103723, 2026.

Chapter 8

Chapter 8 summarizes the main conclusions of the thesis and provides suggestions for future research directions.

Contributions not included in this thesis

The following contributions have not been included in the thesis:

[J4] M. A. Guerrero, B. Lakshminarayanan and C. R. Rojas. Gain-Scheduled Data-Enabled Predictive Control: A DeePC Approach for Nonlinear Systems. In IEEE Control Systems Letters, vol. 9, pp. 3041-3046, 2025.

[J5] B. Lakshminarayanan, M. A. Guerrero and C. R. Rojas. Fine-Tuning a Simulation-Driven Estimator. In IEEE Control Systems Letters, vol. 9, pp. 2975-2980, 2025.

Contribution by the author

The order of authors reflects their relative contribution to each work, with the first author having the leading role. Accordingly, the thesis author is the primary contributor to the works for which she appears as first author. In all listed publications, all authors are actively involved in problem formulation, methodological development, evaluation and interpretation of the results, and manuscript preparation.

1.4 Declaration of Use of Generative AI

In the preparation of this thesis, generative artificial intelligence (AI) tools based on large language models (LLMs) are used as a supporting tool for language refinement, clarity, and the exploration of relevant background material. All AI-assisted suggestions are critically reviewed, verified, and revised by the author. Full responsibility for the content and conclusions of the thesis remains with the author.

2 Preliminaries

This chapter presents the theoretical foundations required for the developments that follow and places the main contributions of the thesis within their broader methodological context. Additional chapter-specific preliminaries are introduced when needed. Section 2.1 first reviews fundamental concepts from robust control, including uncertain-system representations and quantities used to characterize robustness. In Section 2.2, the chapter then introduces selected iterative numerical methods that underpin several of the data-driven procedures developed later in the thesis. Finally, Section 2.3 presents key concepts from machine learning (ML), with particular emphasis on explainable artificial intelligence and counterfactual explanations, thereby providing the conceptual foundation for the second part of the thesis.

Notation

The imaginary number $\sqrt{-1}$ is written as j. The real and imaginary parts of a complex number z are denoted as $\mathrm{Re}\{z\}$ and $\mathrm{Im}\{z\}$, respectively. All matrices and vectors are written in bold, and vectors are column vectors, unless transposed. We write the n-th dimensional (column) vector $\mathbf{x}$ with entries $x_1, x_2, \dots, x_n$ as $\mathbf{x} = [x_1,\ x_2, \dots,\ x_n]^T$. We stack sequences column-wise by default: for any signal $\mathbf{x}$, $\mathbf{x}_{a:b} := \mathrm{col}(\mathbf{x}_{k+a}, \dots, \mathbf{x}_{k+b}) := [\,\mathbf{x}_{k+a}^\top \ \cdots\ \mathbf{x}_{k+b}^\top\,]^\top$. If $\mathbf{A}$ is a matrix, then $\mathbf{A}^\mathsf{T}$ and $\mathbf{A}^\mathsf{H}$ denote its transpose and Hermitian (complex conjugate transpose), respectively. The identity matrix of size n is denoted $\mathbf{I}_n$, where the subscript can be absent if no ambiguity exists. The symbol $\mathrm{diag}(\cdots)$ denotes a block-diagonal matrix with blocks given by its arguments. The symbol $\perp$ denotes orthogonality; in particular, $\mathbf{x} \perp \mathcal{S}$ means that $\mathbf{x}$ is orthogonal to every vector in the subspace $\mathcal{S}$. The vector $\mathbf{e}_p$ represents the p-th canonical basis vector in $\mathbb{R}^n$. The space ℓ_2^n denotes the space of square-summable discrete-time sequences $\mathbf{x} = \{\mathbf{x}[k]\}_{k=0}^{\infty}$, with $\mathbf{x}[k] \in \mathbb{C}^n$,

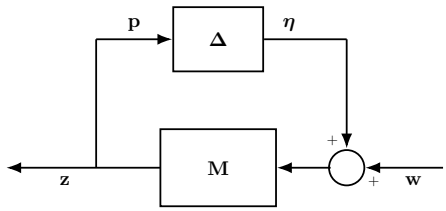


Figure 2.1.1: General uncertain-system interconnection.

satisfying

$$\|\mathbf{x}\|_{\ell_2}^2 := \sum_{k=0}^{\infty} \|\mathbf{x}[k]\|_2^2 < \infty.$$

For two normed spaces $\mathcal{X}$ and $\mathcal{Y}$, $\mathcal{L}(\mathcal{X}, \mathcal{Y})$ denotes the space of bounded linear operators from $\mathcal{X}$ to $\mathcal{Y}$. The set $\mathbb{S}^m$ consists of all Hermitian positive definite matrices of size $m \times m$ and the set $\overline{\mathbb{C}}_+ := \{s \in \mathbb{C} : \ \mathrm{Re}(s) \geq 0\}$ denotes the closed right-half complex plane. The symbol $\det(\cdot)$ denotes the determinant of a square matrix. The open unit disk in the complex plane is denoted by $\mathbb{D} := \{z \in \mathbb{C} : |z| < 1\}$. The symbols q and z denote the forward shift operator and the complex frequency variable in the z-domain, respectively. $[n]$ represents the set of integers from 1 to n and $[a]_+ := \max\{a, 0\}$.

If $g(\mathbf{x})$ is a scalar function of the vector $\mathbf{x}$, then $\partial g(\mathbf{x})/\partial \mathbf{x}$ denotes the gradient of the function g with respect to $\mathbf{x}$, and it is a column vector. If $\mathbf{M} \in \mathbb{C}^{n \times n}$ is a complex matrix, $\bar{\sigma}(\mathbf{M})$ denotes its largest singular value, and $\rho(\mathbf{M})$ its spectral radius. If $\mathbf{x} \in \mathbb{R}^n$, and $\mathbf{Q} \in \mathbb{R}^{n \times n}$ is a positive definite matrix, then the weighted 2-norm is defined as $\|\mathbf{x}\|_{\mathbf{Q}} = \sqrt{\mathbf{x}^\mathsf{T}\mathbf{Q}\mathbf{x}}$.

2.1 Robustness Analysis of Uncertain Systems

This section introduces the uncertain-system framework and the fundamental robustness concepts underlying the developments in Chapters 3, 4, and 5. Throughout this section, all dynamical systems and signals are considered in discrete time.

2.1.1 Uncertain-System Setup

In this thesis, uncertain systems are represented by the feedback interconnection shown in Figure 2.1.1, where $\mathbf{M}$ denotes a known linear time-invariant system and Δ represents modeling uncertainty or unmodeled dynamics. The signals $\mathbf{w} \in \ell_2^{m_2}$ and $\mathbf{z} \in \ell_2^{p_2}$ denote the external input and output, respectively, while $\mathbf{p} \in \ell_2^{p_1}$ and $\boldsymbol{\eta} \in \ell_2^{m_1}$ describe the internal interaction between the nominal system and the uncertainty block.

Here, $\mathbf{M}$ and $\boldsymbol{\Delta}$ are bounded operators on ℓ_2, and the input $\mathbf{w}$ belongs to ℓ_2. Figure 2.1.1 represents the map $\mathbf{w} \mapsto \mathbf{z}$, which is formally defined by the

equations

$$\begin{bmatrix} \mathbf{p} \\ \mathbf{z} \end{bmatrix} = \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta} \\ \mathbf{w} \end{bmatrix}, \qquad \boldsymbol{\eta} = \boldsymbol{\Delta}\mathbf{p}, \tag{2.1.1}$$

where $\mathbf{M}$ is compatibly partitioned with respect to its inputs and outputs. More precisely, $\mathbf{M}$ is an operator between Cartesian products of ℓ_2 spaces, $\mathbf{M}\colon \ell_2^{m_1} \times \ell_2^{m_2} \to \ell_2^{p_1} \times \ell_2^{p_2}$, and each $\mathbf{M}_{ik}$ is a bounded linear operator of appropriate dimensions [47].

Solving for the internal signals $\mathbf{p}$ and $\boldsymbol{\eta}$, the resulting input–output relation between $\mathbf{w}$ and $\mathbf{z}$ is

$$\mathbf{z} = \left[\mathbf{M}_{22} + \mathbf{M}_{21}\boldsymbol{\Delta}\left(\mathbf{I} - \mathbf{M}_{11}\boldsymbol{\Delta}\right)^{-1}\mathbf{M}_{12}\right]\mathbf{w}, \tag{2.1.2}$$

provided that $\mathbf{I} - \mathbf{M}_{11}\boldsymbol{\Delta}$ is invertible.

The block $\boldsymbol{\Delta}$ may represent uncertain parameters, neglected dynamics, or discrepancies between a nominal model and the true system. Depending on the available knowledge, $\boldsymbol{\Delta}$ may be treated as either unstructured or structured. In the unstructured case, only a bound on its induced norm is assumed. In the structured case, additional information about its block configuration, repeated parameters, or real and complex components is retained.

2.1.2 The Small Gain Theorem

For robustness analysis, we focus on the internal feedback interconnection between $\mathbf{M}_{11}$ and $\boldsymbol{\Delta}$ in Figure 2.1.1. Setting the external input $\mathbf{w}$ to zero gives

$$\mathbf{p} = \mathbf{M}_{11}\boldsymbol{\eta}, \qquad \boldsymbol{\eta} = \boldsymbol{\Delta}\mathbf{p}. \tag{2.1.3}$$

For notational simplicity, throughout this subsection we write $\mathbf{M} \coloneqq \mathbf{M}_{11}$ and assume that both $\mathbf{M}$ and $\boldsymbol{\Delta}$ are stable. An important question is how large $\boldsymbol{\Delta}$, in the sense of $\|\boldsymbol{\Delta}\|_\infty$, can be without destabilizing the feedback interconnection.

This question can be addressed using the small gain theorem [12, Section 11.2.1], which states that the feedback interconnection is stable if

$$\|\mathbf{M}\|_\infty \|\boldsymbol{\Delta}\|_\infty < 1, \tag{2.1.4}$$

and this bound is tight for unstructured uncertainty: for every $\alpha > 1/\|\mathbf{M}\|_\infty$, there exists a stable unstructured perturbation $\boldsymbol{\Delta}$ satisfying $\|\boldsymbol{\Delta}\|_\infty = \alpha$ that destabilizes the feedback interconnection.

Consider now the set of stable unstructured perturbations satisfying $\|\boldsymbol{\Delta}\|_\infty < \alpha$. Starting from a sufficiently small value of α, the feedback interconnection remains stable for every perturbation in this set. The largest such value, denoted by $\alpha_{\max}$, is referred to as the *robust stability margin*. For unstructured uncertainty, the small gain theorem gives

$$\frac{1}{\alpha_{\max}} = \|\mathbf{M}\|_\infty \coloneqq \sup_{s \in \overline{\mathbb{C}}_+} \bar{\sigma}(\mathbf{M}(s)) = \sup_{\omega \in \mathbb{R}} \bar{\sigma}(\mathbf{M}(j\omega))\,, \tag{2.1.5}$$

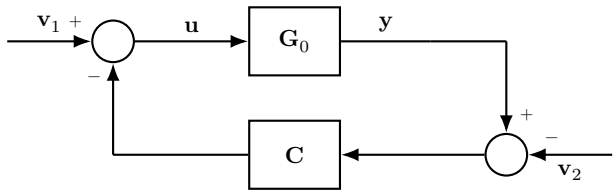


Figure 2.1.2: Standard feedback configuration.

where $\bar{\sigma}(\cdot)$ denotes the largest singular value.

At any fixed $s \in \overline{\mathbb{C}}_+$, the largest singular value can equivalently be characterized as

$$\bar{\sigma}(\mathbf{M}(s)) = \frac{1}{\min_{\mathbf{\Delta}} \{\bar{\sigma}(\mathbf{\Delta}) : \ \det(\mathbf{I} - \mathbf{M}(s)\mathbf{\Delta}) = 0\}}. \tag{2.1.6}$$

Thus, the reciprocal of $\bar{\sigma}(\mathbf{M}(s))$ characterizes the magnitude of the smallest unstructured perturbation that makes $\mathbf{I} - \mathbf{M}(s)\mathbf{\Delta}$ singular. In the dynamical setting, the peak value of $\bar{\sigma}(\mathbf{M}(j\omega))$ over frequency therefore determines the size of unstructured uncertainty that the feedback interconnection can tolerate.

The small gain theorem does not account for any prior knowledge about the form of $\mathbf{\Delta}$. When the uncertainty has a prescribed structure, the smallest destabilizing perturbation may be characterized less conservatively by restricting the minimization in (2.1.6) to perturbations having that structure. This observation motivates the structured singular value studied in Chapter 3.

2.1.3 The Vinnicombe Metric

Consider the standard negative-feedback interconnection of a nominal discrete-time LTI plant $\mathbf{G}_0(z)$ and a compensator $\mathbf{C}(z)$ as shown in Figure 2.1.2. Let $\mathbf{v}_1$ and $\mathbf{v}_2$ denote additive disturbances at the controller input and output, respectively, and let $(\mathbf{y}, \mathbf{u})$ be the plant output and input. The closed-loop transfer matrix from $[\mathbf{v}_1 \ \mathbf{v}_2]^\top$ to $[\mathbf{y} \ \mathbf{u}]^\top$ is

$$T(\mathbf{G}_0, \mathbf{C}) \ := \ \begin{bmatrix} \mathbf{G}_0 \\ \mathbf{I} \end{bmatrix} (\mathbf{I} + \mathbf{C}\mathbf{G}_0)^{-1} \begin{bmatrix} -\mathbf{C} & \mathbf{I} \end{bmatrix}, \tag{2.1.7}$$

where $I = 1$ is the SISO case.

We define the *stability margin* of the feedback interconnection $[\mathbf{G}_0, \mathbf{C}]$ as

$$b_{\mathbf{G}_0,\mathbf{C}} := \begin{cases} \|T(\mathbf{G}_0, \mathbf{C})\|_\infty^{-1}, & \text{if } [\mathbf{G}_0, \mathbf{C}] \text{ is stable,} \\ 0, & \text{otherwise.} \end{cases} \tag{2.1.8}$$

The number $b_{\mathbf{G}_0,\mathbf{C}}$ can be regarded as a measure of the performance of the feedback system comprising $\mathbf{G}_0$ and $\mathbf{C}$, with larger values of (2.1.8) corresponding to better performance. Note that (2.1.8) lies in the range $[0, 1]$ [48].

Definition 2.1. *Let $\mathbf{G}_0(z)$ and $\mathbf{G}(z)$ be discrete-time transfer functions. We denote by $\mathcal{C}$ the set of admissible pairs $(\mathbf{G}_0, \mathbf{G})$ for which $\det(\mathbf{I} + \mathbf{G}^*\mathbf{G}_0)(e^{j\omega}) \neq 0 \ \forall \omega$ and wno $\det(\mathbf{I} + \mathbf{G}^*\mathbf{G}_0) + \eta(\mathbf{G}_0) - \eta(\mathbf{G}) = 0$, where $\eta(\mathbf{G})$ gives the number of poles of $\mathbf{G}$ in the complement of the closed unit disk, counted with multiplicity. The symbol* wno$(\cdot)$ *refers to the winding number about the origin evaluated along the standard discrete-time Nyquist contour,* i.e.*, the unit circle indented into the exterior of the unit disk around any poles and zeros on the unit circle [49]. The* Vinnicombe metric *(or ν-gap) [50] between $\mathbf{G}_0$ and $\mathbf{G}$ is defined as*

$$\delta_\nu(\mathbf{G}_0, \mathbf{G}) := \begin{cases} \left\| (\mathbf{I} + \mathbf{G}\mathbf{G}^*)^{-1/2}(\mathbf{G} - \mathbf{G}_0)(\mathbf{I} + \mathbf{G}_0^*\mathbf{G}_0)^{-1/2} \right\|_\infty, & \text{if } (\mathbf{G}_0, \mathbf{G}) \in \mathcal{C}, \\ 1, & \text{otherwise.} \end{cases} \tag{2.1.9}$$

The condition $(\mathbf{G}_0, \mathbf{G}) \in \mathcal{C}$ is an index, or winding-number, admissibility requirement that ensures a meaningful notion of continuity: along a continuous perturbation path from $\mathbf{G}_0$ to $\mathbf{G}$, it rules out situations in which an intermediate plant $\tilde{\mathbf{G}}$ would yield $\delta_\nu(\tilde{\mathbf{G}}, \mathbf{G}_0) = 1$, even though the endpoint $\mathbf{G}$ may appear close to $\mathbf{G}_0$ in the frequency-domain expression [4, Section 12.1]. In the SISO case, this condition can be checked via a Nyquist/winding-number test; in particular, the Nyquist plot of $G(e^{j\omega})G_0(e^{-j\omega})$ should not encircle the critical point -1 for $\omega \in (-\pi, \pi]$. Additionally, in this same SISO setting, the pointwise quantity underlying the ν-gap admits a geometric interpretation as the chordal distance between the projected Nyquist diagrams of two plants onto the Riemann sphere [50, Sec. 3]. Therefore, δ_ν may be viewed as the maximum normalized separation over frequency, which explains its relevance as a closed-loop similarity metric.

Proposition 2.1 ([50, Prop. 1.2])**.** *Given a nominal plant $\mathbf{G}_0$ and a perturbed plant $\mathbf{G}$, the interconnection $[\mathbf{G}, \mathbf{C}]$ is stable for all compensators $\mathbf{C}$ satisfying $b_{\mathbf{G}_0,\mathbf{C}} > \beta$ if and only if $\delta_\nu(\mathbf{G}_0, \mathbf{G}) \leq \beta$.*

The ν-gap metric $\delta_\nu(\mathbf{G}_0, \mathbf{G})$ quantifies the distance between two plants from a closed-loop perspective. In particular, $\delta_\nu(\mathbf{G}_0, \mathbf{G}) \in [0, 1]$, with $\delta_\nu(\mathbf{G}_0, \mathbf{G}) = 0$ if and only if $\mathbf{G}_0 = \mathbf{G}$. Hence, when $\mathbf{G}$ represents the (unknown) true plant, a small value of $\delta_\nu(\mathbf{G}_0, \mathbf{G})$ indicates that the mismatch between $\mathbf{G}_0$ and $\mathbf{G}$ is negligible in terms of closed-loop stability.

2.2 Iterative Methods

The goal of this section is to introduce the iterative numerical methods underlying the data-driven developments in Chapters 3, 4, and 5. Power iterations provide a common computational foundation for all three chapters, while Krylov-subspace methods support the more computationally demanding MIMO analysis developed in Chapter 5.

2.2.1 Power Iterations

Consider a linear discrete-time, stable, causal, and proper SISO system of the form

$$y_k = G(q)u_k + e_k \qquad k = 0, 1, \ldots$$

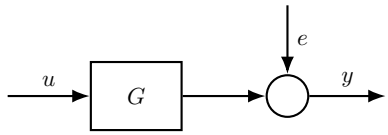


Figure 2.2.3: Relation between the signals u, e and y associated with the system G.

where $\{u_k\}$ is an input signal, G is a transfer function, $\{y_k\}$ is the output of the system, and $\{e_k\}$ is Gaussian, not necessarily white, noise. q denotes the shift operator, $qx_k = x_{k+1}$, and G is a transfer function which can be described as

$$G(q) = g_0 + g_1 q^{-1} + g_2 q^{-2} + \cdots = \sum_{i=0}^{\infty} g_i q^{-i}.$$

The relation between the signal u, e and y is shown graphically in Figure 2.2.3. Now, we collect batches of input–output data: at round or iteration t, a finite input sequence $\mathbf{u}^t := [u_0^t, \dots, u_{N-1}^t]^T$ of length N is applied to the systen, and its output $\mathbf{y}^t := [y_0^t, \dots, y_{N-1}^t]^T$ is recorded. Assume that at iteration t, the system starts from zero initial conditions. If there is no noise affecting the output of the system, the mapping from $\mathbf{u}^t$ to $\mathbf{y}^t$ can be written as

$$\mathbf{y}^t = \mathbf{G}_N \mathbf{u}^t,$$

where

$$\mathbf{G}_N = \begin{bmatrix} g_0 & 0 & 0 & \cdots & 0 \\ g_1 & g_0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ g_{N-1} & g_{N-2} & \cdots & g_1 & g_0 \end{bmatrix} \tag{2.2.10}$$

is a triangular Toeplitz matrix.

The matrix $\mathbf{G}_N$ provides a finite-horizon representation of the input–output map of G. A natural measure of the amplification induced by this map is the induced ℓ_2-gain of G, defined as

$$\|G\|_{i2,N} = \sup_{\mathbf{u}^t \in \mathbb{R}^{N\times 1} \setminus \{\mathbf{0}\}} \frac{\|\mathbf{y}^t\|_2}{\|\mathbf{u}^t\|_2},$$

Using the finite-horizon representation from (2.2.10), this quantity can be approximated, for N sufficiently large, as

$$\|G\|_{i2,N}^2 = \sup_{\mathbf{u}^t \neq 0} \frac{(\mathbf{u}^t)^T \mathbf{G}_N^T \mathbf{G}_N \mathbf{u}^t}{(\mathbf{u}^t)^T \mathbf{u}^t} = \lambda^{\max}(\mathbf{G}_N^T \mathbf{G}_N)\,. \tag{2.2.11}$$

The fraction inside the supremum in (2.2.11) is the Rayleigh quotient associated with $\mathbf{G}_N^T \mathbf{G}_N$; its maximization over all nonzero inputs gives the largest eigenvalue, which can be approximated using the power iteration method.

Algorithm 2.1 presents the classical power iteration method for estimating the eigenvector associated with the dominant eigenvalue of a square matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, *i.e.*, the eigenvalue of largest magnitude [51, Section 8.2].

Algorithm 2.1 Power iteration method

1: Choose $\mathbf{x}_0 \in \mathbb{R}^n$, $\mathbf{x}_0 \neq 0$, randomly
2: Set $i \leftarrow 1$
3: **while** the stopping criterion is not satisfied **do**
4: Compute $\mathbf{x}_i \leftarrow \frac{\mathbf{A}\mathbf{x}_{i-1}}{\|\mathbf{A}\mathbf{x}_{i-1}\|_2}$
5: $i \leftarrow i + 1$
6: **end while**

For normalized iterates, a natural estimate of the magnitude of the dominant eigenvalue is $\|\mathbf{A}\mathbf{x}_{i-1}\|_2$.

The key observation behind the data-driven use of Algorithm 2.1 is that the iteration only requires the action of $\mathbf{A}$ on a vector, not entries of $\mathbf{A}$ itself. When $\mathbf{A}$ represents an unknown dynamical system, this action can be obtained via input–output experiments, without forming $\mathbf{A}$ or identifying a model.

2.2.2 Krylov Methods

There exists an important amount of methods for solving large linear systems in an iterative scheme. Among these, the Krylov Subspace Methods, which are based on projection processes, present an advantage over other methods as these algorithms give a partial result after a small number of iterations.

Krylov Subspace

Given a matrix $\mathbf{A}$ and a vector $\mathbf{b}$, Krylov subspaces are the linear spaces spanned by successive matrix-vector products involving $\mathbf{A}$, i.e.,

$$\mathcal{K}_m(\mathbf{A}, \mathbf{b}) = \text{span}\{\mathbf{b}, \mathbf{A}\mathbf{b}, \mathbf{A}^2\mathbf{b}, \dots, \mathbf{A}^{m-1}\mathbf{b}\}, \tag{2.2.12}$$

which will be denoted simply by $\mathcal{K}_m$ if there is no ambiguity. Viewed from the angle of approximation theory, the approximations obtained from a Krylov subspace method are of the form

$$\mathbf{A}^{-1}\mathbf{b} \approx q_{m-1}(\mathbf{A})\mathbf{b}$$

in which q_{m-1} is a certain polynomial of degree $m-1$ [44].

Krylov subspaces are used in algorithms for finding approximate solutions to high-dimensional linear algebra problems, being extremely useful in situations when there is no explicit representation of $\mathbf{A}$. In control theory, Krylov subspaces use (2.2.12) to approximate large-scale, complex linear systems, treating $\mathbf{A}$ as a dynamical system and the matrix-vector product as its output.

Among Krylov methods, the Arnoldi iteration [52]-[53, Ch. VI, Sec. 2] provides an efficient way to approximate selected eigenvalues of a general non-Hermitian matrix $\mathbf{A}$ through orthogonal projections onto $\mathcal{K}_m$. When $\mathbf{A}$ is Hermitian, Arnoldi's method reduces to the Lanczos algorithm, leading to significant computational savings. This property is particularly useful in Chapter 5, where Lanczos' method is employed to

approximate the matrix-function–vector product $\mathbf{C}^{-1/2}\mathbf{b}$ without explicitly forming $\mathbf{C}^{-1/2}$ [54].

Lanczos Algorithm

For $\mathbf{A} \in \mathbb{C}^{n\times n}$, we want to compute $\mathbf{A} = \mathbf{QHQ}^*$, where $\mathbf{H}$ is upper Hessenberg, *i.e.*, $h_{ij} = 0$ for $i > j+1$, and $\mathbf{Q}$ is unitary. When n is large, we do not compute the full factorization; instead, we consider only the first $m \ll n$ columns of $\mathbf{AQ} = \mathbf{QH}$. Therefore, on the left-hand side, we only need the matrix $\mathbf{Q}_m \in \mathbb{C}^{n\times m}$:

$$\mathbf{Q}_m = \left[\begin{array}{c|c|c|c} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_m \end{array}\right], \tag{2.2.13}$$

where the columns $\mathbf{q}_1, \ldots, \mathbf{q}_m$ form an orthonormal basis of the Krylov subspace $\mathcal{K}_m(\mathbf{A}, \mathbf{b})$ [44, Prop. 6.4]. On the right-hand side, only the first m columns of $\mathbf{H}$ are required. More specifically, due to the upper-Hessenberg structure, we only need $\tilde{\mathbf{H}}_m$ which is the $(m+1)\times m$ upper-left section of $\mathbf{H}$. Hence,

$$\mathbf{AQ}_m = \mathbf{Q}_{m+1}\tilde{\mathbf{H}}_m.$$

Removing the last row of $\tilde{\mathbf{H}}_m$ yields the projected matrix $\mathbf{H}_m = \mathbf{Q}_m^*\mathbf{AQ}_m \in \mathbb{C}^{m\times m}$.

Proposition 2.2. *Let $\mathbf{A}$ be symmetric with real entries, and hence it has real eigenvalues. Then $\mathbf{H}_m = \mathbf{Q}_m^T\mathbf{AQ}_m$ is also symmetric with real eigenvalues and tridiagonal.*

Proof. First, since $\mathbf{A}$ is symmetric,

$$\mathbf{H}_m^T = \left(\mathbf{Q}_m^T\mathbf{AQ}_m\right)^T = \mathbf{Q}_m^T\mathbf{A}^T\mathbf{Q}_m = \mathbf{Q}_m^T\mathbf{AQ}_m = \mathbf{H}_m,$$

and hence $\mathbf{H}_m$ is symmetric. To establish the tridiagonal structure, consider the ij entry of $\mathbf{H}_m$, $h_{ij} = \mathbf{q}_i^T\mathbf{Aq}_j$. Then we have $\mathbf{Aq}_j \in \mathcal{K}_{j+1}(\mathbf{A}, \mathbf{b}) = \operatorname{span}\{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_{j+1}\}$, and hence $h_{ij} = \mathbf{q}_i^T(\mathbf{Aq}_j) = 0$ for $i > j+1$ since

$$\mathbf{q}_i \perp \operatorname{span}\{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_{j+1}\}, \quad \text{for } i > j+1.$$

Also, since $\mathbf{H}_m$ is symmetric, we have $h_{ij} = h_{ji} = \mathbf{q}_j^T(\mathbf{Aq}_i)$ which implies that $h_{ij} = 0$ for $j > i+1$, by the same reasoning as above. □

In view of Proposition 2.2, we can write $\mathbf{H}_m$ as

$$\mathbf{T}_m = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \beta_{m-1} \\ & & & \beta_{m-1} & \alpha_m \end{bmatrix}. \tag{2.2.14}$$

Finally, given the structure of (2.2.14), a three-term recurrence at step m can be written as

$$\mathbf{A}\mathbf{q}_m = \beta_{m-1}\mathbf{q}_{m-1} + \alpha_m\mathbf{q}_m + \beta_m\mathbf{q}_{m+1}.$$

We rearrange this to give

$$\mathbf{q}_{m+1} = (\mathbf{A}\mathbf{q}_m - \beta_{m-1}\mathbf{q}_{m-1} - \alpha_m\mathbf{q}_m)/\beta_m,$$

which leads to the Lanczos iteration in Algorithm 2.2.

Algorithm 2.2 Lanczos iteration

$\beta_0 = 0,\ \mathbf{q}_0 = 0$
Choose $\mathbf{b}$ arbitrarily and set $\mathbf{q}_1 = \mathbf{b}/\|\mathbf{b}\|_2$
for $m = 1, 2, 3, \ldots$ **do**
    $\mathbf{v} = \mathbf{A}\mathbf{q}_m$
    $\alpha_m = \mathbf{q}_m^\top \mathbf{v}$
    $\mathbf{v} = \mathbf{v} - \beta_{m-1}\mathbf{q}_{m-1} - \alpha_m\mathbf{q}_m$
    $\beta_m = \|\mathbf{v}\|_2$
    $\mathbf{q}_{m+1} = \mathbf{v}/\beta_m$
end for

The Lanczos process therefore reduces the large Hermitian eigenvalue problem to the eigenanalysis of the much smaller tridiagonal matrix $\mathbf{T}_m$. In particular, the extreme eigenvalues of A can be approximated by those of $\mathbf{T}_m$ [51, Sec. 10.1.4–10.1.5].

2.3 Explainable Artificial Intelligence

The goal of this section is to introduce the reader to the fundamental concepts of XAI underlying the developments in Chapters 6 and 7. These concepts provide the foundation for interpreting black-box models, even when their internal structure, parameters, or training procedure are inaccessible.

2.3.1 Explainability in Machine Learning

Machine learning models have become increasingly complex as more flexible modeling techniques have been developed. From the perspective of interpretability, predictive models can be broadly distinguished as *white-box* or *black-box* models [55]. White-box models possess an internal structure that can be inspected and understood by a domain expert, *e.g.*, small decision trees and sparse linear models. In contrast, the internal mechanisms of black-box models are difficult to understand directly from their parameters, even when their predictions exhibit high accuracy [56].

Explainable artificial intelligence studies models and methods that make the behavior and predictions of machine learning systems understandable to humans. Explanations may be generated directly from an intrinsically interpretable model or through a separate *explanation method* applied after the predictive model has been

trained. The latter is particularly relevant when the model is treated as a black box and only its inputs and predictions are available.

Let

$$h \colon \mathcal{X} \to \mathcal{Y} \tag{2.3.15}$$

denote a predictive model, where $\mathcal{X}$ is the input space and $\mathcal{Y}$ is the output space. Given an instance $x \in \mathcal{X}$, an explanation method aims to relate the feature values of x to the prediction $h(x)$ in a form that can be understood by a human. Different explanation methods characterize a model's behavior from different perspectives, for example by identifying influential features, extracting decision rules, constructing local surrogate models, highlighting representative instances, or finding alternative inputs that would change the prediction.

Explanation methods can also be distinguished according to their scope. *Global* explanations attempt to characterize the overall behavior of a model, whereas *local* explanations interpret the prediction associated with a particular instance. Moreover, a method is called *model-agnostic* when it does not require access to the internal structure or parameters of the predictive model and instead relies only on input–output evaluations. The methods considered in the second part of this thesis are local and model-agnostic, since they seek to explain individual predictions by querying a black-box model.

Among local explanation methods, *counterfactual explanations* are particularly relevant to this thesis. Rather than describing which features contributed to a prediction, they identify how the input would need to change for the model to produce a desired alternative outcome. In this way, counterfactual explanations connect the task of explaining a prediction with the problem of identifying meaningful perturbations of the input.

2.3.2 Counterfactual Explanations

Counterfactual reasoning consists of imagining an alternative situation in which some observed circumstances are different and asking whether the outcome would consequently change. In the context of machine learning, a *counterfactual explanation* identifies how an input instance could be modified so that a predictive model produces a desired alternative outcome. For example, a person whose loan application was rejected may ask whether the decision would have been different had their income been higher or their debt been lower. Such an explanation describes the behavior of the predictive model under a hypothetical modification of its input, but does not necessarily imply a causal relationship in the real world.

From a mathematical perspective, let (2.3.15) denote a predictive model, and let $\tilde{x} \in \mathcal{X}$ be the instance whose prediction is to be explained. Given a desired outcome y^*, a counterfactual explanation is an alternative instance x^* that achieves this outcome while remaining close to $\tilde{x}$. It can be formulated as

$$x^* \in \operatorname{argmin}_{x \in \mathcal{X}_{\text{feas}}} d(x, \tilde{x}) \quad \text{subject to} \quad h(x) = y^*, \tag{2.3.16}$$

where $d(\cdot, \cdot)$ measures the magnitude of the modification and $\mathcal{X}_{\text{feas}}$ denotes the set of admissible inputs. In binary classification, validity may alternatively be expressed as $h(x^*) \neq h(\tilde{x})$.

Satisfying the desired prediction alone does not guarantee that the counterfactual is meaningful. High-quality counterfactual explanations should also satisfy several desirable properties. They should be *valid*, by achieving the desired outcome; *proximal*, by remaining close to the original instance; and *sparse*, by modifying as few features as possible. The proposed changes should also be *actionable*, involving only variables that can be modified in practice, and *plausible*, respecting the domain constraints and the distribution of realistic observations. When multiple explanations are provided, they should be *diverse*, offering meaningfully different alternatives. Finally, the generation procedure should be *scalable*, particularly when the input dimension is large or model evaluations are costly [57, 58].

In the applications considered in this thesis, the input components may represent parameters of a dynamical system or a feedback controller rather than conventional data features. Consequently, the feasible set may also encode physical, operational, and closed-loop stability constraints.

Part I

Data-Driven Estimation of Robustness Measures

3 Structured Robustness: Data-Driven Estimation of the Structured Singular Value

Building on the concepts introduced in Chapter 2, this chapter marks the beginning of Part I of the thesis, devoted to the data-driven characterization of robustness in unknown dynamical systems. In particular, we study the *design vulnerability* of an unknown system to structured modeling errors: namely, how small a perturbation, constrained to a prescribed structure, can be before stability is lost.

Addressing Research Question 1 posed in Section 1.1, we develop a fully data-driven method for estimating a lower bound on the structured singular value, whose reciprocal characterizes the smallest admissible structured perturbation capable of destabilizing the system. In contrast with the unstructured uncertainty description underlying the small gain theorem introduced in Section 2.1.2, the structured singular value explicitly exploits prior knowledge about the form and interconnection of the uncertainty. It therefore provides a richer and potentially less conservative characterization of the system's vulnerability to modeling errors.

Extensive numerical simulations demonstrate the effectiveness of the proposed method in assessing vulnerability to structured uncertainty.

3.1 Introduction

Modeling plays a crucial role in designing feedback controllers. As noted in Chapter 1, numerous techniques for modeling dynamical systems have been developed in the literature [5]. Controllers that are based on these models aim to ensure stability of the closed loop, assuming that the model is a faithful representation of the system. However, models often fail to capture every nuance of real-world dynamical systems, as they are inevitably subject to *modeling errors*, which need to be accounted for in control design.

Many robust control theories have been developed to address modeling errors and quantify uncertainties [12]. One popular approach involves treating the modeling error as a linear operator of bounded $\mathcal{H}_\infty$ norm, which is defined as the supremum over all frequencies of the largest singular value of its frequency response. For linear systems, the $\mathcal{H}_\infty$ norm of the modeling error can be directly estimated from experiments on the system by employing data-driven methods, such as power iterations [29, 30] and Thompson sampling [59], without the need for a parametric model.

In the case of structured modeling error, an appropriate metric for the design of robust controllers is the *structured singular value* [22]. For systems with unstructured uncertainty, the structured singular value coincides with the $\mathcal{H}_\infty$ norm. In the presence of structured uncertainty, however, the structured singular value provides a tighter measure of how large the modeling error can be (in terms of its $\mathcal{H}_\infty$ norm) without leading to instability. Nonetheless, in spite of its usefulness for the establishment of necessary and sufficient conditions for the stability of feedback controllers under structured modeling errors, the structured singular value cannot be easily computed, as its calculation is in general NP-hard [22], so in practice only lower and upper bounds can be determined. Recent works propose tighter structured singular value lower and upper bounds for special block structures [60, 61, 62]; however they also require a model for the system and impose additional constraints on structural uncertainties.

Motivated by these limitations, this chapter introduces a fully data-driven power iteration scheme, based on [30], for estimating a lower bound on the structured singular value of a linear dynamical system. The proposed method avoids the need for an explicit parametric model and instead relies exclusively on input–output experimental data. In particular,

- We propose a data-driven approach to numerically compute a lower bound on the structured singular value for dynamical systems.
- We demonstrate our method's effectiveness through numerical simulations, where the estimated lower bounds agree with MATLAB© `mussv` in most cases.

The remainder of this chapter is organized as follows: In Section 3.2, we state our problem setup. Section 3.3 provides some preliminaries on the power method. In Section 3.4, we outline our proposed approach, and demonstrate its efficacy in Section 3.5. Finally, the chapter is concluded in Section 3.6.

3.2 Problem Statement

Consider a linear time-invariant square multivariable discrete-time dynamical system defined by its transfer function $\mathbf{G}(z)$. We assume that it is composed of a "nominal" stable and strictly proper model $\mathbf{G}_0(z)$ and a stable block $\mathbf{\Delta}(z)$ denoting a multiplicative uncertainty, as shown in Figure 3.2.1. Let $\mathbf{U}(z) \in \mathbb{C}^n$ and $\mathbf{Y}(z) \in \mathbb{C}^n$ denote the Z-transforms of the input and output signals of $\mathbf{G}$, respectively.

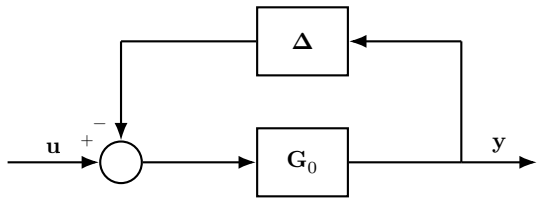


Figure 3.2.1: Interconnection of the nominal system $\mathbf{G}_0$ and the uncertainty block $\boldsymbol{\Delta}$.

Then,

$$\mathbf{Y}(z) = \underbrace{[\mathbf{I} + \mathbf{G}_0(z)\boldsymbol{\Delta}(z)]^{-1}\mathbf{G}_0(z)}_{=:\mathbf{G}(z)}\,\mathbf{U}(z).$$

Since $\mathbf{G}_0$ is stable and strictly proper (*i.e.*, the poles of $\det[\mathbf{G}_0(z)]$ are in the open unit disk $\mathbb{D} := \{z \in \mathbb{C} \colon |z| < 1\}$), the full system $\mathbf{G}$ is stable if and only if all the zeros of $\det[\mathbf{I} + \mathbf{G}_0(z)\boldsymbol{\Delta}(z)]$ are in the open unit disk. Furthermore, since $\boldsymbol{\Delta}$ is also stable, the poles of $\det[\mathbf{I} + \mathbf{G}_0(z)\boldsymbol{\Delta}(z)]$ are all in $\mathbb{D}$, so $\mathbf{G}$ is stable if and only if, by the Nyquist criterion, the curve that $z \mapsto \det[\mathbf{I} + \mathbf{G}_0(z)\boldsymbol{\Delta}(z)]$ draws on the complex plane, as z runs over $\mathbb{T} := \partial\mathbb{D}$ counter-clockwise, does *not* encircle (clockwise) the point 0.

Suppose that $\boldsymbol{\Delta}$ is known to be of the form $\mathrm{diag}(\delta_1\mathbf{I}_{r_1}, \dots, \delta_s\mathbf{I}_{r_s}, \boldsymbol{\Delta}_1, \dots, \boldsymbol{\Delta}_f)$, where $\delta_1, \dots, \delta_s \in \mathbb{C}$ and $\boldsymbol{\Delta}_1, \dots, \boldsymbol{\Delta}_f$ are stable dynamical systems of sizes $m_1 \times m_1$, ..., $m_f \times m_f$, respectively. We say that $\boldsymbol{\Delta} \in \Delta$ if it possesses this structure. Then, the *structured singular value* of $\mathbf{G}_0$ is defined as

$$\mu_\Delta(\mathbf{G}_0) := \frac{1}{\min\left\{\|\boldsymbol{\Delta}\|_\infty \colon \boldsymbol{\Delta} \in \Delta, \det\Big[\mathbf{I} + \mathbf{G}_0(z)\,\boldsymbol{\Delta}(z)\Big] = 0 \text{ for some } z \in \mathbb{T}\right\}}. \quad (3.2.1)$$

The interest in $\mu_\Delta(\mathbf{G}_0)$ lies in its use to characterize the stability of $\mathbf{G}$, as stated in the following theorem.

Theorem 3.1 ([12, Theorem 11.8])**.** *The system $\mathbf{G} = [\mathbf{I} + \mathbf{G}_0\boldsymbol{\Delta}]^{-1}\mathbf{G}_0$ is stable for all $\boldsymbol{\Delta} \in \Delta$ such that $\|\boldsymbol{\Delta}\|_\infty \leq \alpha$ if and only if $\alpha\,\mu_\Delta(\mathbf{G}_0) < 1$.*

This result is an extension of the *small gain theorem* (cf. Section 2.1.2) to structured uncertainty, and it is fundamental for the study of the stability of uncertain systems in robust control [12]. The quantity $\mu_\Delta(\mathbf{G}_0)$ can be evaluated frequency-wisely as

$$\mu_\Delta(\mathbf{G}_0) := \sup_{\omega \in (-\pi, \pi]} \frac{1}{\min\{\bar{\sigma}(\boldsymbol{\Delta}) \colon \boldsymbol{\Delta} \in \Delta, \det[\mathbf{I} + \mathbf{G}_0(e^{i\omega})\boldsymbol{\Delta}] = 0\}}, \quad (3.2.2)$$

where now $\boldsymbol{\Delta}$ is a (static) complex matrix having the structure given by Δ. Unfortunately, computing $\mu_\Delta(\mathbf{G}_0)$ is hard, hence practical approaches typically rely on lower and upper bounds [22]. An appealing approach that exists in the literature [24] is to compute a lower bound on $\mu_\Delta(\mathbf{G}_0)$ based on the power iterations method [51] and a model for $\mathbf{G}_0$. Inspired by that method, our main contribution in

this chapter is to propose a fully data-driven approach to compute a lower bound on $\mu_\Delta(\mathbf{G}_0)$ that does not require knowledge of $\mathbf{G}_0$. To that end, we consider a discrete-time LTI system described by

$$\mathbf{y}_k = \mathbf{G}_0(q)\mathbf{u}_k + \mathbf{e}_k, \tag{3.2.3}$$

where $\mathbf{G}_0(q)$ is an unknown but stable system, $\mathbf{u}_k$ is the input at time k, $\mathbf{y}_k$ is the corresponding output, and $\mathbf{e}_k$ is a zero-mean additive disturbance. Then, the underlying data-generation process follows an iterative redesign of the input signal $\{\mathbf{u}_k\}_{k=1}^N$, based on the power iteration method, to focus sampling on frequencies that contribute most to $\mu_\Delta(\mathbf{G}_0)$. This redesign uses information extracted from the output $\{\mathbf{y}_k\}_{k=1}^N$ obtained in previous iterations. We assume $\mathbf{e}_k = \mathbf{0}$ throughout the analysis, while noise is included in the simulations of Section 3.5 to evaluate robustness. To ensure consistency with the power method in [24], we have adopted the notation used in that paper.

Before we propose our approach, we shall present several technical preliminaries that are required for this chapter.

3.3 Preliminaries

This section introduces the definitions and results on the structured singular value that are specifically required in this chapter. The presentation follows [24, 22].

Structured Singular Value and the power method

Let Δ denote the set of admissible block-diagonal uncertainty matrices, and let $\mathrm{B}\Delta$ denote its corresponding unit ball, defined as

$$\Delta := \{\mathrm{diag}(\delta_1\mathbf{I}_{r_1}, \dots, \delta_s\mathbf{I}_{r_s}, \mathbf{\Delta}_1, \dots, \mathbf{\Delta}_f) \colon \delta_1, \dots, \delta_s \in \mathbb{C},\ \mathbf{\Delta}_1 \in \mathbb{C}^{m_1\times m_1}, \dots, \mathbf{\Delta}_f \in \mathbb{C}^{m_f\times m_f}\},$$

$$\mathrm{B}\Delta := \{\mathbf{\Delta} \in \Delta \colon \bar{\sigma}(\mathbf{\Delta}) \le 1\},$$

where $r_1, \dots, r_s, m_1, \dots, m_f$ are fixed positive integers such that $r_1 + \cdots + r_s + m_1 + \cdots + m_f = n$. Observe that Δ is a (complex) linear subspace of $\mathbb{C}^{n\times n}$. Based on these sets, the structured singular value can be defined at a given frequency $\omega \in [-\pi, \pi)$ (by letting $\mathbf{M} = \mathbf{G}_0(e^{i\omega})$) as

Definition 3.1 (Structured singular value)**.** *For $\mathbf{M} \in \mathbb{C}^{n\times n}$, let*

$$\mu'_\Delta(\mathbf{M}) := \frac{1}{\min\{\bar{\sigma}(\mathbf{\Delta}) \colon \mathbf{\Delta} \in \Delta,\ \det(\mathbf{I} + \mathbf{M}\mathbf{\Delta}) = 0\}}.$$

In case there is no $\mathbf{\Delta} \in \Delta$ for which $\det(\mathbf{I} + \mathbf{M}\mathbf{\Delta}) = 0$, we define $\mu'_\Delta(\mathbf{M}) = 0$. Note that $\mu_\Delta(G_0) = \sup_\omega \mu'_\Delta(M)$.

Let us define the sets of unitary and structured Hermitian positive definite matrices $\mathcal{Q}$ and $\mathcal{D}$, respectively, as:

$$\mathcal{Q} := \{\mathbf{\Delta} \in \Delta \colon \mathbf{\Delta}^H\mathbf{\Delta} = \mathbf{I}\},$$

$$\mathcal{D} := \{\text{diag}(\mathbf{D}_1, \dots, \mathbf{D}_s, d_1\mathbf{I}_{m_1}, \dots, d_f\mathbf{I}_{m_f}) : \mathbf{D}_1 \in \mathbb{S}^{r_1}, \dots, \mathbf{D}_s \in \mathbb{S}^{r_s}, d_1, \dots, d_f \in \mathbb{R}_+\}.$$

These sets satisfy the following properties:

Lemma 3.1 ([24, p.1]). *Let $\mathbf{\Delta} \in \Delta$, $\mathbf{Q} \in \mathcal{Q}$ and $\mathbf{D} \in \mathcal{D}$. Then, $\mathbf{Q}^H \in \mathcal{Q}$, $\mathbf{Q\Delta} \in \Delta$, $\mathbf{\Delta Q} \in \Delta$, $\bar{\sigma}(\mathbf{Q\Delta}) = \bar{\sigma}(\mathbf{\Delta Q}) = \bar{\sigma}(\mathbf{\Delta})$, and $\mathbf{D\Delta} = \mathbf{\Delta D}$.*

The following theorem establishes bounds on μ'_{Δ}.

Theorem 3.2 ([22, Theorem 1]). *For all $\mathbf{M} \in \mathbb{C}^{n\times n}$,*

$$\max_{\mathbf{Q}\in\mathcal{Q}} \rho(\mathbf{QM}) = \max_{\mathbf{\Delta}\in B\Delta} \rho(\mathbf{\Delta M}) = \mu'_{\Delta}(\mathbf{M}) \leq \inf_{\mathbf{D}\in\mathcal{D}} \bar{\sigma}(\mathbf{DMD}^{-1}).$$

As established in Theorem 3.2, our goal is to derive a method for finding a local maximum of the function $\mathbf{\Delta} \mapsto \rho(\mathbf{\Delta M})$ over all $\mathbf{\Delta} \in \text{B}\Delta$; every such local maximum provides a lower bound to $\mu'_{\Delta}(\mathbf{M})$. Packard et al. derived in [24] the following necessary characterization of $\mu'_{\Delta}(\mathbf{M})$, based on Theorem 3.2.

Theorem 3.3 ([24, Theorem 6.3]). *Given a matrix $\mathbf{M} \in \mathbb{C}^{n\times n}$, let $\mathbf{Q}_0 \in \mathcal{Q}$ achieve the global maximum in $\max_{\mathbf{Q}\in\mathcal{Q}} \rho(\mathbf{QM})$, and assume that the maximum eigenvalue of $\mathbf{Q}_0\mathbf{M}$ is simple, real and positive; call it μ. If $\mathbf{x} = [\mathbf{x}_{r_1}^T, \dots, \mathbf{x}_{r_s}^T, \mathbf{x}_{m_1}^T, \dots, \mathbf{x}_{m_f}^T]^T \in \mathbb{C}^n$ and $\mathbf{y} = [\mathbf{y}_{r_1}^T, \dots, \mathbf{y}_{r_s}^T, \mathbf{y}_{m_1}^T, \dots, \mathbf{y}_{m_f}^T]^T \in \mathbb{C}^n$ are right and left eigenvectors of $\mathbf{Q}_0\mathbf{M}$ associated with this eigenvalue (where the partitions of $\mathbf{x}$ and $\mathbf{y}$ are compatible with the block structure of Δ) such that $\mathbf{y}^H\mathbf{x} = 1$, $\mathbf{y}_{r_j}^H\mathbf{x}_{r_j} \neq 0$ for all $j = 1, \dots, s$, and $\mathbf{x}_{m_k}, \mathbf{y}_{m_k} \neq 0$ for all $k = 1, \dots, f$, then there exists a $\mathbf{D} \in \mathcal{D}$ such that*

$$\mathbf{Q}_0\mathbf{DMD}^{-1}(\mathbf{Dx}) = \mu\mathbf{Dx},\ \mathbf{x}^H\mathbf{DQ}_0\mathbf{DMD}^{-1} = \mu\mathbf{x}^H\mathbf{D}.$$

According to Theorem 3.3, and via the change of variable $\bar{\mathbf{x}} := \mathbf{Dx}$, given a matrix $\mathbf{M} \in \mathbb{C}^{n\times n}$, to find $\mu = \max_{\mathbf{Q}\in\mathcal{Q}} \rho(\mathbf{QM})$ we can try to find matrices $\mathbf{Q} \in \mathcal{Q}$ and $\mathbf{D} \in \mathcal{D}$, and a vector $\bar{\mathbf{x}} \in \mathbb{C}^n$ with $\|\bar{\mathbf{x}}\| = 1$ such that

$$\mathbf{QDMD}^{-1}\bar{\mathbf{x}} = \mu\bar{\mathbf{x}},\ \mathbf{D}^{-1}\mathbf{M}^H\mathbf{DQ}^H\bar{\mathbf{x}} = \mu\bar{\mathbf{x}},$$

which can, in turn, be re-written as

$$\mathbf{M}(\mathbf{D}^{-1}\bar{\mathbf{x}}) = \mu(\mathbf{D}^{-1}\mathbf{Q}^H\bar{\mathbf{x}}),\ \mathbf{M}^H(\mathbf{DQ}^H\bar{\mathbf{x}}) = \mu(\mathbf{D}\bar{\mathbf{x}}).$$

For fixed $\mathbf{Q} \in \mathcal{Q}$ and $\mathbf{D} \in \mathcal{D}$, let us define the vectors

$$\mathbf{a} := \mathbf{D}^{-1}\mathbf{Q}^H\bar{\mathbf{x}}, \mathbf{b} := \mathbf{D}^{-1}\bar{\mathbf{x}}, \mathbf{w} := \mathbf{D}\bar{\mathbf{x}}, \text{ and } \mathbf{z} := \mathbf{DQ}^H\bar{\mathbf{x}}.$$

With these definitions, we have that $\mathbf{Mb} = \mu\mathbf{a}$ and $\mathbf{M}^H\mathbf{z} = \mu\mathbf{w}$. We can eliminate $\bar{\mathbf{x}}$ from these definitions, using Lemma 3.1 and replacing $\mathbf{D}^2$ with $\mathbf{D}$, obtaining

$$\mathbf{b} = \mathbf{Qa},\ \mathbf{b} = \mathbf{D}^{-1}\mathbf{w},\ \mathbf{z} = \mathbf{Da},\ \mathbf{z} = \mathbf{Q}^H\mathbf{w}.$$

After eliminating $\bar{\mathbf{x}}$, we want to remove $\mathbf{Q}$ and $\mathbf{D}$ from these definitions. To this end, let $\mathbf{a} = [\mathbf{a}_{r_1}^T, \dots, \mathbf{a}_{r_s}^T, \mathbf{a}_{m_1}^T, \dots, \mathbf{a}_{m_f}^T]^T \in \mathbb{C}^n$ according to the structure of Δ, and similarly for the other vectors. Then, we have the following lemma.

Lemma 3.2 ([24, Lemma 7.1])**.** *Given non-zero vectors* $\mathbf{a}, \mathbf{b}, \mathbf{w}, \mathbf{z} \in \mathbb{C}^n$*, there are matrices* $\mathbf{Q} \in \mathcal{Q}$ *and* $\mathbf{D} \in \mathcal{D}$ *such that* $\mathbf{b} = \mathbf{Q}\mathbf{a}$*,* $\mathbf{b} = \mathbf{D}^{-1}\mathbf{w}$*,* $\mathbf{z} = \mathbf{D}\mathbf{a}$*, and* $\mathbf{z} = \mathbf{Q}^H\mathbf{w}$*, if and only if*

$$\mathbf{z}_{r_j} = \frac{\mathbf{w}_{r_j}^H \mathbf{a}_{r_j}}{|\mathbf{w}_{r_j}^H \mathbf{a}_{r_j}|}\mathbf{w}_{r_j};\ \mathbf{b}_{r_j} = \frac{\mathbf{a}_{r_j}^H \mathbf{w}_{r_j}}{|\mathbf{a}_{r_j}^H \mathbf{w}_{r_j}|}\mathbf{a}_{r_j}, \quad j = 1, \dots, s,$$
$$\mathbf{z}_{m_k} = \frac{\|\mathbf{w}_{m_k}\|}{\|\mathbf{a}_{m_k}\|}\mathbf{a}_{m_k};\ \mathbf{b}_{m_k} = \frac{\|\mathbf{a}_{m_k}\|}{\|\mathbf{w}_{m_k}\|}\mathbf{w}_{m_k}, \quad k = 1, \dots, f.$$

Combining the conditions in Lemma 3.2 with the conditions that $\mathbf{M}\mathbf{b} = \mu\mathbf{a}$ and $\mathbf{M}^H\mathbf{z} = \mu\mathbf{w}$ suggests the following power method for determining a lower bound on μ:

$$\begin{aligned}
\tilde{\mu}(l+1) &\leftarrow \|\mathbf{M}\mathbf{b}(l)\| \\
\mathbf{a}(l+1) &\leftarrow \frac{1}{\tilde{\mu}(l+1)}\mathbf{M}\mathbf{b}(l) \\
\mathbf{z}_{r_j}(l+1) &\leftarrow \frac{\mathbf{w}_{r_j}^H(l)\mathbf{a}_{r_j}(l+1)}{|\mathbf{w}_{r_j}^H(l)\mathbf{a}_{r_j}(l+1)|}\mathbf{w}_{r_j}(l), \quad j = 1, \dots, s \\
\mathbf{z}_{m_k}(l+1) &\leftarrow \frac{\|\mathbf{w}_{m_k}(l)\|}{\|\mathbf{a}_{m_k}(l+1)\|}\mathbf{a}_{m_k}(l+1), \quad k = 1, \dots, f \\
\bar{\mu}(l+1) &\leftarrow \|\mathbf{M}^H\mathbf{z}(l+1)\| \\
\mathbf{w}(l+1) &\leftarrow \frac{1}{\bar{\mu}(l+1)}\mathbf{M}^H\mathbf{z}(l+1) \\
\mathbf{b}_{r_j}(l+1) &\leftarrow \frac{\mathbf{a}_{r_j}^H(l+1)\mathbf{w}_{r_j}(l+1)}{|\mathbf{a}_{r_j}^H(l+1)\mathbf{w}_{r_j}(l+1)|}\mathbf{a}_{r_j}(l+1), \quad j = 1, \dots, s \\
\mathbf{b}_{m_k}(l+1) &\leftarrow \frac{\|\mathbf{a}_{m_k}(l+1)\|}{\|\mathbf{w}_{m_k}(l+1)\|}\mathbf{w}_{m_k}(l+1), \quad k = 1, \dots, f.
\end{aligned}$$

Here the entire set of assignments is iterated over $l = 0, 1, \dots$ until convergence, where the lower bound on μ is obtained as an equilibrium point $\tilde{\mu}(l) = \bar{\mu}(l)$. Here $\mathbf{b}(0)$ and $\mathbf{w}(0)$ are chosen as unit vectors.

3.4 Proposed Approach

In this section, we propose a data-driven approach to compute a lower bound on μ_Δ. Assuming that a model of the plant is unavailable and only time-domain experiments can be conducted, we adapt the frequency-domain power iteration scheme of [24] to operate purely on input–output data.

Firstly, note that the power iteration of Section 3.3 is applied separately at each frequency. Since experiments are conducted in the time domain, it is necessary to map signals to the frequency domain, where the updates of $\mathbf{b}$, $\mathbf{a}$, $\mathbf{w}$, and

$\mathbf{z}$—cf. Lemma 3.2—are computed over the relevant frequency range. To that end, we associate with each time-domain signal its discrete Fourier transform (DFT), and use its inverse to reconstruct the time-domain signals when needed.

In the sequel, we add the frequency argument m to vectors $\mathbf{b}$, $\mathbf{a}$, $\mathbf{w}$, and $\mathbf{z}$ and write them in uppercase to denote their frequency-domain versions, with lowercase denoting their time-domain counterparts. This leads to the following frequency-domain power iteration scheme (for each $m = 0, \dots, N-1$; $j = 1, \dots, s$; and $k = 1, \dots, f$), where we have replaced $\mathbf{M}$ by $\mathbf{G}_0(e^{i2\pi m/N})$, which will be reinterpreted through experimental data in the next step:

$$\tilde{\mu}[l+1,m] \leftarrow \left\|\mathbf{G}_0(e^{i2\pi m/N})\mathbf{B}[l,m]\right\| \tag{3.4.4}$$

$$\mathbf{A}[l+1,m] \leftarrow \frac{1}{\tilde{\mu}[l+1,m]}\mathbf{G}_0(e^{i2\pi m/N})\mathbf{B}[l,m] \tag{3.4.5}$$

$$\mathbf{Z}_{r_j}[l+1,m] \leftarrow \frac{\mathbf{W}_{r_j}^H[l,m]\mathbf{A}_{r_j}[l+1,m]}{\left|\mathbf{W}_{r_j}^H[l,m]\mathbf{A}_{r_j}[l+1,m]\right|}\mathbf{W}_{r_j}[l,m] \tag{3.4.6}$$

$$\mathbf{Z}_{m_k}[l+1,m] \leftarrow \frac{\left\|\mathbf{W}_{m_k}[l,m]\right\|}{\left\|\mathbf{A}_{m_k}[l+1,m]\right\|}\mathbf{A}_{m_k}[l+1,m] \tag{3.4.7}$$

$$\bar{\mu}[l+1,m] \leftarrow \left\|\mathbf{G}_0^T(e^{-i2\pi m/N})\mathbf{Z}[l+1,m]\right\| \tag{3.4.8}$$

$$\mathbf{W}[l+1,m] \leftarrow \frac{1}{\bar{\mu}[l+1,m]}\mathbf{G}_0^T(e^{-i2\pi m/N})\mathbf{Z}[l+1,m] \tag{3.4.9}$$

$$\mathbf{B}_{r_j}[l+1,m] \leftarrow \frac{\mathbf{A}_{r_j}^H[l+1,m]\mathbf{W}_{r_j}[l+1,m]}{\left|\mathbf{A}_{r_j}^H[l+1,m]\mathbf{W}_{r_j}[l+1,m]\right|}\mathbf{A}_{r_j}[l+1,m] \tag{3.4.10}$$

$$\mathbf{B}_{m_k}[l+1,m] \leftarrow \frac{\left\|\mathbf{A}_{m_k}[l+1,m]\right\|}{\left\|\mathbf{W}_{m_k}[l+1,m]\right\|}\mathbf{W}_{m_k}[l+1,m] \tag{3.4.11}$$

To implement this algorithm in a data-driven setting, we must re-write the operations involving $\mathbf{G}_0(e^{i2\pi m/N})$ and $\mathbf{G}_0^T(e^{-i2\pi m/N})$ in terms of time-domain experiments. Specifically, we define $\mathbf{P}[l,m] := \mathbf{G}_0(e^{i2\pi m/N})\mathbf{B}[l,m]$ and $\mathbf{R}[l+1,m] := \mathbf{G}_0^T(e^{-i2\pi m/N})\mathbf{Z}[l+1,m]$, which must now be approximated from data. The computation of $\mathbf{P}$ can be carried out as follows:

$$\mathbf{b}[l,t] \leftarrow \frac{1}{N}\sum_{m=0}^{N-1}\mathbf{B}[l,m]e^{i2\pi mt/N}, \quad t = 1, \dots, N \tag{3.4.12}$$

$$\mathbf{p}[l,:] \leftarrow \mathbf{G}_0(q)\mathbf{b}[l,:] \tag{3.4.13}$$

$$\mathbf{P}[l,m] \leftarrow \sum_{t=1}^{N}\mathbf{p}[l,t]e^{-i2\pi mt/N}, \quad m = 0, 1, \dots, N-1 \tag{3.4.14}$$

Here, $\mathbf{p}[l,:]$ is the output of $\mathbf{G}_0$ when excited with $\mathbf{b}[l,t]$, a time-domain signal derived from the frequency-domain vector $\mathbf{B}[l,:]$ in Lemma 3.2, chosen to span the desired frequencies. Similarly, $\mathbf{r}[l+1,:]$ is the response of $\mathbf{G}_0^T$ to $\mathbf{z}[l+1,:]$, also from Lemma 3.2 and mapped to time, since all experiments on $\mathbf{G}_0$ are conducted in the time domain.

The computation of $\mathbf{R}$ is a bit trickier, due to the transpose and complex conjugate operations. The complex conjugate operation in the frequency domain corresponds to applying $\mathbf{G}_0$ "backwards in time", or by replacing t with $N+1-t$ in the computation of the discrete Fourier transform and its inverse. To account for the transpose of $\mathbf{G}_0$, we can appeal to the trick in [31, Eq. (21)], according to which

$$\mathbf{G}_0^T(q) = \sum_{\alpha=1}^{n}\sum_{\beta=1}^{n} \mathbf{e}_\alpha \mathbf{e}_\beta^T \mathbf{G}_0(q) \mathbf{e}_\alpha \mathbf{e}_\beta^T. \tag{3.4.15}$$

Combining these ideas, we obtain the following pseudo code for computing $\mathbf{R}$:

$$\mathbf{z}[l+1,t] \leftarrow \frac{1}{N}\sum_{m=0}^{N-1} \mathbf{Z}[l+1,m] e^{i2\pi m(1-t)/N}, \qquad t = 1,\dots,N \tag{3.4.16}$$

$$\mathbf{r}[l+1,t] \leftarrow \mathbf{0} \tag{3.4.17}$$

for $\alpha = 1,\dots,n$**:**

  for $\beta = 1,\dots,n$**:**

$$\mathbf{r}[l+1,:] \leftarrow \mathbf{r}[l+1,:] + \mathbf{e}_\alpha \mathbf{e}_\beta^T \mathbf{G}_0(q) \mathbf{e}_\alpha \mathbf{e}_\beta^T \mathbf{z}[l+1,:] \tag{3.4.18}$$

$$\mathbf{R}[l+1,m] \leftarrow \sum_{t=1}^{N} \mathbf{r}[l+1,t] e^{i2\pi m(t-1)/N}, \qquad m = 0,\dots,N-1 \tag{3.4.19}$$

Note that the first and last lines do *not* correspond to the standard discrete Fourier transform and its inverse, but to their "time-reversed" versions. Additionally, to prevent signals such as $\mathbf{b}[l,t]$ from growing unbounded as l increases, we normalize them after each iteration by their 2-norms.

The previous discussions finally lead to the pseudo-code for the power method shown in Algorithm 3.1. The algorithm terminates when $\bar{\mu} \approx \tilde{\mu}$ and their values remain unchanged across iterations for each frequency, i.e., $\tilde{\mu}(l+1,m) = \tilde{\mu}(l,m)$ and $\bar{\mu}(l+1,m) = \bar{\mu}(l,m)$. Finally, $\mu_\Delta(\mathbf{G}_0)$ is obtained by selecting the maximum μ across all frequencies.

3.5 Experiments

This section presents a comprehensive set of simulations to assess the performance of Algorithm 3.1 against the lower bound provided by the `mussv` command from the MATLAB© Robust Control Toolbox [63, Ch. 10]. For clarity, we refer to this lower bound as μ_M throughout.

3.5.1 Experimental Setting

As described in Algorithm 3.1, the computation of the structured singular value requires two experiments on $\mathbf{G}_0$ and $\mathbf{G}_0^H$ (the latter being based on (3.4.15)). However, to properly account for the real-valued nature of the pulse response of $\mathbf{G}_0$, and as indicated in [31, Procedure 2], the input signals used in these experiments must be real. This constraint imposes a frequency symmetry condition on the initial

vectors $\mathbf{B}$ and $\mathbf{W}$, which are otherwise randomly chosen. Due to this randomness, the initial input is a white noise signal, exciting all frequencies uniformly, while subsequent iterations redesign it to focus sampling effort on frequencies where $\mu_{\Delta}(\mathbf{G}_0)$ is expected to be maximal. Moreover, to avoid degeneracy, the initial vectors $\mathbf{b}[l,t]$ and $\mathbf{z}[l+1,t]$ must have nonzero components along the dominant eigenvector directions; otherwise, the algorithm is heuristically restarted with a different initial condition, i.e., by reinitializing $\mathbf{B}[0,:]$ and $\mathbf{W}[0,:]$. Therefore, under these necessary conditions—namely, non-degenerate initialization, spectral symmetry, and a real and simple dominant eigenvalue of $\mathbf{Q}_0\mathbf{M}$, as discussed in Theorem 3.3—the iterative scheme in Algorithm 3.1 is expected to converge reliably in many practical scenarios.

Another critical aspect concerns transient effects in physical systems or finite-time simulations. Since the procedure is carried out in the frequency domain, a sufficiently large number of time samples or frequency points must be considered, to reduce the effects of transients. This requirement aligns with the condition $N \to \infty$ stated in [31], which, in turn, ensures consistency with the requirements of the *power method* [51], namely, $\|\mathbf{\Delta}\|_{i2,N} \to \|\mathbf{\Delta}\|_{\infty}$ as $N \to \infty$.

To clarify the results presented in the next subsection, we denote the uncertainty structure by $\mathbf{r} = [\, s,\, r_1, \dots, r_s \,]$, $\mathbf{m} = [\, f,\, m_1, \dots, m_f \,]$, where s and f are the numbers

Algorithm 3.1 Computation of a lower bound on μ

Require: $\mathbf{B}[0,:]$, $\mathbf{W}[0,:]$
for $l = 0, 1, \dots$ **do**
  for $t = 1, \dots, N$, **compute** $\mathbf{b}[l,t]$ **via** (3.4.12), **end for**
  apply $\mathbf{b}[l,:]$ **to** $\mathbf{G}_0(q)$ **and observe** $\mathbf{p}[l,:]$ **via** (3.4.13)
  for $m = 0, \dots, N-1$, **compute** $\mathbf{P}[l,m]$ **via** (3.4.14), **end for**
  for $m = 0, \dots, N-1$ **do**
    compute $\tilde{\mu}[l+1,m]$ **via** (3.4.4)
    compute $\mathbf{A}[l+1,m]$ **via** (3.4.5)
    for $j = 1, \dots, s$, **compute** $\mathbf{Z}_{r_j}[l+1,m]$ **via** (3.4.6), **end for**
    for $k = 1, \dots, f$, **compute** $\mathbf{Z}_{m_k}[l+1,m]$ **via** (3.4.7), **end for**
  end for
  for $t = 1, \dots, N$, **compute** $\mathbf{z}[l+1,t]$ **via** (3.4.16), **end for**
  initialize $\mathbf{r}[l+1,t]$ **via** (3.4.17)
  for $\alpha = 1, \dots, n$ **do**
    for $\beta = 1, \dots, n$ **apply** $\mathbf{z}[l+1,:]$ **to** (3.4.15) **and observe** $\mathbf{r}[l+1,:]$ **via** (3.4.18), **end for**
  end for
  for $m = 0, \dots, N-1$ **compute** $\mathbf{R}[l+1,m]$ via (3.4.19), **end for**
  for $m = 0, \dots, N-1$ **do**
    compute $\bar{\mu}[l+1,m]$ **via** (3.4.8)
    compute $\mathbf{W}[l+1,m]$ **via** (3.4.9)
    for $j = 1, \dots, s$, **compute** $\mathbf{B}_{r_j}[l+1,m]$ **via** (3.4.10), **end for**
    for $k = 1, \dots, f$, **compute** $\mathbf{B}_{m_k}[l+1,m]$ **via** (3.4.11), **end for**
  end for
  $\mathbf{B}[l+1,:] \leftarrow \dfrac{1}{\sqrt{\sum_{i=0}^{N-1} \|\mathbf{B}[l+1,i]\|_2^2}} \mathbf{B}[l+1,:]$
end for

of scalar and full blocks, of sizes r_j and m_k, respectively. For example, $\mathbf{r} = [2,\ 1,\ 1]$, $\mathbf{m} = [1,\ 2]$ describes two 1×1 scalar blocks and one 2×2 full block.

3.5.2 Experimental Results

The experimental evaluation consists of two test cases aimed at assessing the accuracy of the data-driven estimation of the lower bound on the structured singular value and the influence of the uncertainty structure $\mathbf{\Delta}$ on its performance.

Test #1

In [24], it was shown that if $2s + f \leq 3$, then the right-hand side of Theorem 3.2 becomes an equality. To construct a fair and meaningful benchmark for our method based on this insight, we have constructed a specific random 3×3 MIMO system $\mathbf{G}_0(z)$ with $\mathbf{\Delta} \subset \mathbb{C}^{3\times 3}$ such that the lower and upper bounds of $\mu_{\mathbf{\Delta}}(\mathbf{G}_0)$ coincide. This guarantees that the true value of $\mu_{\mathbf{\Delta}}$ is known, enabling a direct and reliable comparison between our estimate and the output of `mussv`. Eight distinct block structures are considered, as described in Table 3.5.1, and evaluated in terms of (i) the convergence of $\tilde{\mu}$ and $\bar{\mu}$, and (ii) the number of frequency samples required to reach this convergence. Although not all block configurations satisfy the condition $2s + f \leq 3$ (e.g., when $s \geq 2$), we have identified a specific system $\mathbf{G}_0(z)$ for which this condition holds across all uncertainty structures considered.

Extensive simulations have demonstrated that, in most cases where $s = 0$ and $f = 1$ (i.e., a single full block for a given n, with $\mathbf{M} \in \mathbb{C}^{n\times n}$), both $\tilde{\mu}$ and $\bar{\mu}$ converge, and their values match the lower bound provided by `mussv`. Similarly, when the number of full blocks exceeds the number of scalar blocks ($f > s$), the algorithm generally exhibits good performance (and often when $m_k > r_j$ for all j, k). An exception to these cases occurs when the uncertainty structure satisfies $s = n$ (a single $n \times n$ scalar block) and $f = 0$, in which case $\tilde{\mu} \neq \bar{\mu}$ for most instances, indicating poorer convergence.

Table 3.5.1: Uncertainty structures $\mathbf{\Delta} \in \mathbb{C}^{3\times 3}$.

Case	$\mathbf{r}$ (Scalar)	$\mathbf{m}$ (Full)
1	$[1, 3]$	$[0, 0]$
2	$[0, 0]$	$[1, 3]$
3	$[1, 2]$	$[1, 1]$
4	$[1, 1]$	$[1, 2]$
5	$[2, 1, 1]$	$[1, 1]$
6	$[1, 1]$	$[2, 1, 1]$
7	$[3, 1, 1, 1]$	$[0, 0]$
8	$[0, 0]$	$[3, 1, 1, 1]$

Figure 3.5.2 presents the results for the configurations described in Table 3.5.1. Note that, as N increases, the performance of Algorithm 3.1 improves, as previously discussed. Cases 1 ($f = 0$) and 3 ($r_j > m_k$) do not converge to μ_M; this

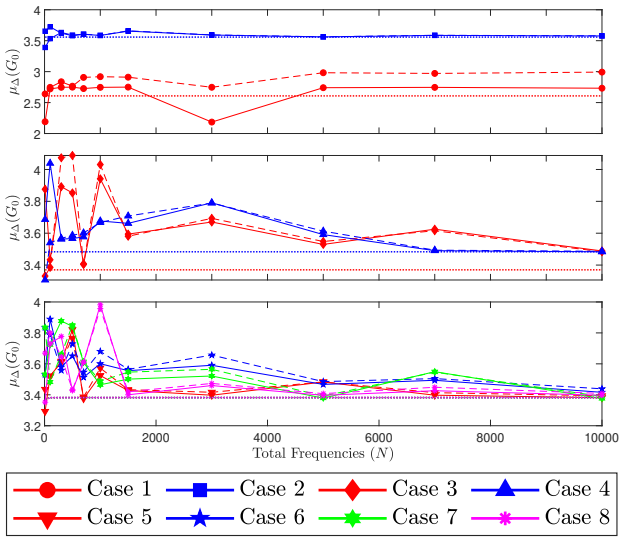


Figure 3.5.2: Comparison between $\mu_{\Delta}(\mathbf{G}_0)$ and N. Solid line: $\tilde{\mu}$, dashed line: $\bar{\mu}$, dotted line: `mussv`. From top to bottom, the plots correspond to systems with 1 block, 2 blocks, and 3 blocks, respectively.

behavior—involving large repeated scalar blocks—aligns with prior observations in [24, 64], which suggest that power iteration methods may fail to converge or enter limit cycles under such structures. This is likely due to the limited directional excitation offered by scalar blocks, in contrast to full blocks that span richer subspaces and more effectively align with directions of maximal growth. Additionally, Cases 2 and 8 exhibit the best convergence across all cases, which is both encouraging and relevant for practical applications, as it suggests that the proposed algorithm is especially effective in computing reliable lower bounds in realistic robustness scenarios, where full complex blocks are commonly used to model input–output interactions and coupled perturbations.

In practice, when the number of frequencies N is large, the number of iterations l required for the algorithm to converge typically ranges between 15 and 30. For the test under consideration, where $N = 10000$ was used to ensure convergence, Figure 3.5.3 illustrates the iterative behavior for cases where the lower bound successfully reaches `mussv`, highlighting the effect of the number of blocks on the stopping criterion, where a higher number of blocks requires more iterations to meet the termination condition. Only the converging cases are shown, thus excluding Cases 1 and 3. Cases 5–8, which all involve three blocks, required the same number of iterations.

Figure 3.5.4 illustrates the structured singular value as a function of frequency for Case 4. The observed behavior, where $\tilde{\mu}$ and $\bar{\mu}$ align well with `mussv` at the dominant frequency but deviate at others, is expected based on the properties of the power method [30]. Indeed, the power method generates input signals whose energy

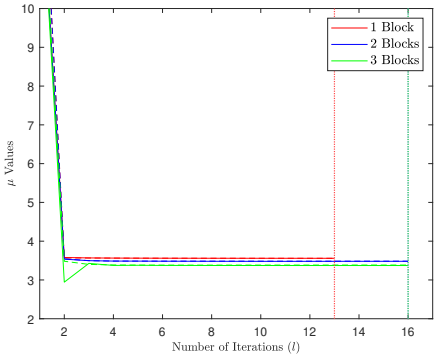


Figure 3.5.3: Convergence of the lower bound over iterations (l). Solid line: $\tilde{\mu}$, dashed line: $\bar{\mu}$, dotted line: last iteration.

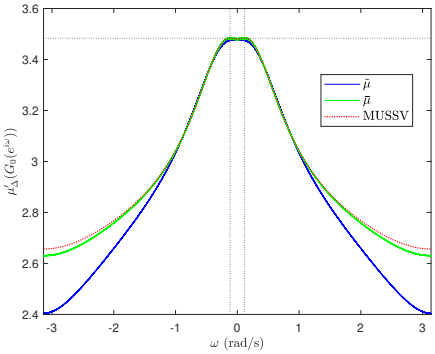


Figure 3.5.4: Frequency response of the structured singular value.

iteratively concentrates around the frequency ω at which the lower bound on $\mu'_{\Delta}(\mathbf{G}_0(e^{i\omega}))$ is largest, which leads to better estimates at that frequency at the expense of poorer estimates for other frequencies.

Table 3.5.2 illustrates how Gaussian noise in the output, with variance σ^2, affects the convergence of $\tilde{\mu}$ and $\bar{\mu}$, which show that the algorithm can be sensitive to noise. The algorithm's robustness to noise can be improved, using, *e.g.*, instrumental variables methods, but this is left for future research.

Table 3.5.2: Structured singular values, and peak frequencies, for different noise levels.

σ^2	$\tilde{\mu}$		$\bar{\mu}$	
	Value	Frequency	Value	Frequency
0	3.4832	0.0000	3.4864	0.1137
10^{-6}	3.9034	0.0415	4.1362	0.0258
10^{-5}	4.7400	0.1005	6.0842	0.4876
10^{-4}	8.2426	2.9016	11.6509	0.3914
10^{-3}	22.4976	2.3989	33.6600	0.4115
0.01	65.4947	1.6757	125.8755	2.7307
MUSSV	3.4833	0.1181	–	–

In summary, this benchmark confirms that Algorithm 3.1 effectively estimates the structured singular value under favorable structural conditions, while also highlighting directions for improvement in noisy or scalar-dominated scenarios.

Test #2

To obtain more general insights into the convergence behavior of Algorithm 3.1, we have conducted a large set of simulations using randomly generated $n \times n$ complex systems, with $n = 2, 3, \ldots, 8$. For each n, we tested the algorithm under the following

three block structure configurations: (i) $\mathbf{r} = [0, 0]$ and $\mathbf{m} = [1, n]$, (ii) $\mathbf{r} = [n, 1, \cdots, 1]$ and $\mathbf{m} = [0, 0]$, (iii) for even n, $\mathbf{r} = [n/2, 1, \dots, 1]$ and $\mathbf{m} = [1, n/2]$; for odd n, $\mathbf{r} = [(n-1)/2, 1, \dots, 1]$ and $\mathbf{m} = [1, (n+1)/2]$. We ran 700 experiments per configuration, totaling 2100 simulations. Figure 3.5.5 illustrates the percentage of

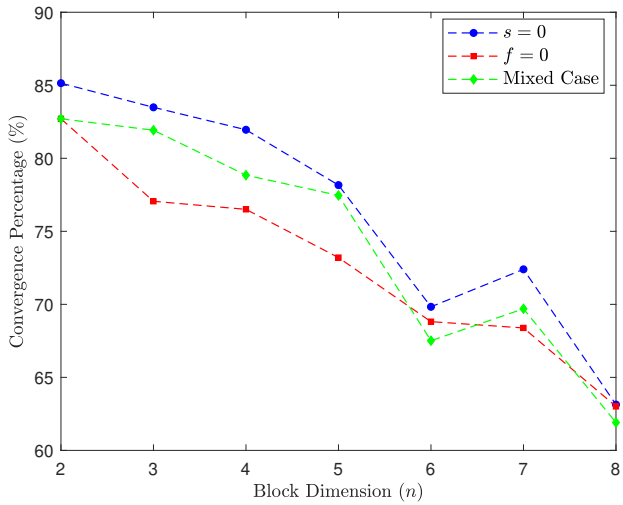


Figure 3.5.5: Percentage of converging cases of the data-driven power method.

simulations where the average of $\tilde{\mu}$ and $\bar{\mu}$ converge to μ_M, which demonstrates better performance for $s = 0$ and, across all three cases, a deterioration of the algorithm when $n > 5$. It should be noted that, for $s + 2f > 3$, the exact value of $\mu_{\Delta}(\mathbf{G}_0)$ is unknown, potentially introducing bias when comparing the lower bounds to `mussv`. Nevertheless, using randomly generated synthetic systems provides a first step in evaluating algorithm performance, and future work could explore its applicability to real-world system dynamics.

Overall, the results show that Algorithm 3.1 yields reliable lower bound estimates for $\mu_{\Delta}(\mathbf{G}_0)$, especially with full blocks and moderate system order. Additionally, it shows improved convergence in systems with faster dynamics, likely because shorter transients reduce the impact of non-steady-state data.

3.6 Summary and Future Work

In this chapter, we have introduced a fully data-driven method for estimating a lower bound on the structured singular value of an unknown dynamical system directly from input–output experiments. In contrast to model-based approaches, the proposed method does not require an explicit parametric representation of the plant, thereby extending the data-driven robustness perspective beyond the unstructured uncertainty setting associated with the small gain theorem. While induced-gain quantities can be efficiently estimated through power iterations, the proposed

framework incorporates information about the structure of the uncertainty, providing a richer characterization of the system's vulnerability to modeling errors.

Extensive numerical experiments over different uncertainty structures and noise levels demonstrate the effectiveness of the proposed approach, with the estimated lower bounds closely matching those obtained with `mussv` in the noise-free case. The results also reveal the sensitivity of the estimation procedure to measurement noise, which motivates the development of more robust noise-handling and denoising strategies. Further directions include the data-driven estimation of corresponding upper bounds on the structured singular value and the development of formulations operating entirely in the time domain, avoiding the need for intermediate frequency-domain representations. Together, these developments would move toward a more complete data-driven characterization of structured robustness using only experimental observations of the underlying system.

4 Model Uncertainty: Data-Driven Estimation of the SISO Vinnicombe Metric

Chapter 3 considered the vulnerability of an unknown dynamical system to structured modeling uncertainty. This chapter turns to a complementary question: how much mismatch between a nominal model and the true plant can be tolerated by a controller designed for the nominal model? Focusing on single-input single-output systems, we develop a data-driven method for estimating the Vinnicombe metric, which quantifies this mismatch from a closed-loop perspective, and evaluate it through extensive numerical simulations.

4.1 Introduction

Modeling errors have been a central concern throughout this thesis. Since no model can reproduce every aspect of a real dynamical system, some degree of mismatch between the model and the underlying plant is inevitable. Nevertheless, achieving reliable and satisfactory closed-loop behavior despite such imperfections lies at the heart of automatic control. A fundamental objective is therefore to design robust controllers whose stability and performance are preserved when the true system deviates from the model used for controller synthesis. Several frameworks have been developed to address this problem, including small-gain arguments [65], integral quadratic constraints [66], and $\mathcal{H}_\infty$ control [12]. Broadly speaking, these approaches provide stability or performance guarantees for all plants whose uncertainty satisfies a prescribed description, such as norm bounds, structural assumptions, or multiplier-based conditions.

A control-relevant way to quantify model mismatch is provided by the Vinnicombe metric [48, 50], also known as the ν-gap metric, defined for linear time-invariant (LTI) systems. In contrast with the robust-control frameworks mentioned above, which typically require a prescribed uncertainty description, the advantage of the ν-gap is that it provides a scalar plant-to-plant distance directly tied to closed-loop

robustness. In particular, it induces the weakest topology in which closed-loop stability is a robust property, and it quantifies how much mismatch between a nominal model and the true plant can be tolerated by a controller designed for the nominal model. More precisely, the ν-gap is directly related to the generalized stability margin: if the ν-gap between a nominal model and the true plant is sufficiently small, then a controller that stabilizes the nominal model with adequate margin also stabilizes the true plant, under suitable admissibility conditions. Moreover, although the ν-gap is topologically equivalent to the classical gap metric [67, 68], in the SISO case it is particularly useful because it admits a direct frequency-response interpretation as the supremum of a normalized chordal distance between the two plants, making the notion of plant distance more transparent.

In practice, however, the usefulness of both robust-control guarantees and ν-gap-based reasoning depends on how accurately the plant uncertainty or model mismatch can be characterized. When the true plant is unknown and prior knowledge is inaccurate, constructing a meaningful uncertainty description may be difficult, conservative, or infeasible.

Motivated by this, in this chapter we propose an identification-free, iterative approach that estimates the ν-gap between two plants directly from time-domain input–output data. The proposed method builds on the core idea of power iterations to compute a numerical estimate of the distance between two dynamical systems, while an iterative winding-number calculation is used to verify the admissibility conditions under which the ν-gap is well defined. To complement the methodological validation, we also include a compact industrially motivated benchmark based on heavy-duty gas-turbine models, illustrating the relevance of the proposed estimator in a practically motivated setting.

In summary, in this chapter,

- We propose a novel identification-free, data-driven method for estimating the Vinnicombe metric directly from time-domain input–output experiments.
- We introduce an identification-free procedure for verifying the admissibility conditions of the ν-gap through an iterative winding-number computation based on Welch-inspired frequency-domain estimates.
- We demonstrate the effectiveness of the proposed method through numerical simulations, showing that the resulting estimates closely agree with MATLAB® `gapmetric` command.
- We include a compact heavy-duty gas-turbine benchmark to illustrate the practical relevance of the proposed estimator in an industrially motivated setting.

The remainder of this chapter is organized as follows: In Section 4.2, we state our problem setup and in Section 4.3 we outline our proposed approach. We demonstrate our method efficacy in Section 4.4 while remaining questions and next steps for research are specified in Section 4.5.

4.2 Problem Statement

Consider a stable unity-feedback interconnection, as shown in Figure 2.1.2 of a nominal plant G_0 with a controller C, denoted by $[G_0, C]$. Suppose that the nominal plant differs from an unknown true system G. Our objective is to guarantee that the controller C will perform as intended on the true process regardless of the unmodeled dynamics. Throughout this chapter, G_0 and G are assumed to be stable, and (G_0, G, C) are discrete-time, linear time-invariant, single-input single-output systems; hence, G_0, G, and C denote scalar transfer functions and the corresponding signals are scalar-valued.

The aforementioned problem can be handled with a well-known distance metric, namely the ν-gap, also called the Vinnicombe metric [48, 50]. As reviewed in Section 2.1.3, this metric, denoted by δ_ν, is an appropriate measure for assessing how two systems differ in their closed-loop behavior, satisfying $0 \leq \delta_\nu(G_1, G_2) \leq 1$.

This quantity admits a closed-loop interpretation: it measures, in a normalized worst-case sense, the discrepancy between the complementary sensitivity behaviors induced by unity feedback around G and G_0. In particular, taking a sufficiently small value of δ_ν implies that any controller that stabilizes G_0 with adequate margin also stabilizes G (under additional conditions summarized in Section 2.1.3).

When the real plant is unknown, computing $\delta_\nu(G, G_0)$ becomes untenable, as it requires knowledge of the true process. A classical workaround is to postulate that G lies in a neighborhood of the nominal model, i.e., $\mathcal{B}_\tau(G_0) := \{\tilde{G} : \delta_\nu(\tilde{G}, G_0) \leq \tau\}$ [50, Proposition 1.2]. However, specifying such a set typically relies on prior modeling knowledge about unmodeled dynamics and may be conservative or inaccurate.

Motivated by these limitations, we propose in Section 4.3 an identification-free, data-driven procedure to compute the Vinnicombe metric for the single-input single-output (SISO) case using a power iteration principle [51], requiring only input–output experiments on the black-box system G and an iterative redesign of the excitation signal. While we primarily assume that G_0 is an available nominal model, the same methodology can be used as a data-driven similarity test when both systems are unknown, e.g., to compare the opening/closing dynamics of two valves from closed-loop measurements.

4.3 Proposed Approach

In this section, we propose a data-driven procedure to estimate the Vinnicombe ν-gap metric from time-domain experiments. Assuming that a nominal model G_0 is available and that only input–output data can be collected, we adapt the model-based construction in [48] to operate directly on frequency-domain quantities estimated from time-domain data.

4.3.1 ν-gap estimation

We focus on the practically relevant setting where G denotes the (unknown) plant and G_0 is a nominal model. When $(G, G_0) \in \mathcal{C}$, the ν-gap in (2.1.9) admits the equivalent frequency-domain expression

$$\delta_\nu(G, G_0) = \sup_{\omega \in (-\pi, \pi]} \frac{|G(e^{j\omega}) - G_0(e^{j\omega})|}{\sqrt{|1 + |G(e^{j\omega})|^2}\,\sqrt{1 + |G_0(e^{j\omega})|^2}}. \tag{4.3.1}$$

For a given finite-length input vector $\mathbf{u} \in \mathbb{R}^N$, let $U(e^{j\omega})$ be its discrete-time Fourier transform (DFT), and let $Y(e^{j\omega})$ and $Y_0(e^{j\omega})$ denote the corresponding output spectra obtained from experiments on G and G_0, respectively, so that $Y(e^{j\omega}) = G(e^{j\omega})U(e^{j\omega})$ and $Y_0(e^{j\omega}) = G_0(e^{j\omega})U(e^{j\omega})$. Assuming $U(e^{j\omega}) \neq 0$ on the considered frequency grid, this expression can be rewritten in terms of input–output spectra as

$$\delta_\nu(G, G_0) = \sup_{\omega \in (-\pi, \pi]} \frac{|Y(e^{j\omega}) - Y_0(e^{j\omega})|\,|U(e^{j\omega})|}{\sqrt{|U(e^{j\omega})|^2 + |Y(e^{j\omega})|^2}\,\sqrt{|U(e^{j\omega})|^2 + |Y_0(e^{j\omega})|^2}}.$$

As in $\mathcal{H}_\infty$ norm estimation, the ν-gap is the supremum of the modulus of a frequency function. In the noiseless case, this motivates a power-iteration-type redesign in which the current input spectrum is multiplied by the pointwise normalized discrepancy between G and G_0. Specifically, given spectra $(\mathbf{U}^n, \mathbf{Y}^n, \mathbf{Y}_0^n)$ at iteration n, where each vector collects the DFT coefficients of the corresponding finite-length input or output signal over the considered frequency grid, we update the next input in the frequency domain according to

$$\tilde{U}^{n+1}(e^{j\omega}) = \frac{\left|Y^n(e^{j\omega}) - Y_0^n(e^{j\omega})\right|\left|U^n(e^{j\omega})\right|^2}{\sqrt{\left|U^n(e^{j\omega})\right|^2 + \left|Y^n(e^{j\omega})\right|^2}\,\sqrt{\left|U^n(e^{j\omega})\right|^2 + \left|Y_0^n(e^{j\omega})\right|^2}}. \tag{4.3.2}$$

The role of (4.3.2) is analogous to the matrix-vector product in the classical power iteration. At iteration n, the frequency-domain vector $\mathbf{U}^n$ is mapped to its time-domain counterpart $\mathbf{u}^n$ and applied to both G and G_0. The resulting time-domain outputs are then transformed to the frequency domain to obtain $\mathbf{Y}^n$ and $\mathbf{Y}_0^n$; the unknown system G enters only through the measured output spectrum $Y^n(e^{j\omega})$. The update weights the frequency components according to their contribution to the normalized discrepancy between G and G_0, maps the result back to the time domain via an inverse DFT, rescales the resulting time-domain signal to have fixed ℓ_2 norm as in Algorithm 2.1, and uses it as $\mathbf{u}^{n+1}$ in the next power iteration experiment. As in the classical power method, the initial input $\mathbf{u}^0$ should not be orthogonal to the dominant direction; in practice, this motivates a sufficiently rich initial excitation. Hence, the excitation is progressively steered toward the frequencies where the discrepancy is largest, without evaluating $G(e^{j\omega})$, identifying a transfer function, or constructing $\mathbf{G}_N$. The nominal model G_0 is used only as a reference model, while G is accessed solely through input–output measurements; in this sense, the method is identification-free with respect to G.

More generally, the same principle can be used to compare two black-box systems from input–output measurements, as illustrated in Section 4.4.1; here, we present the case where G_0 is available as the nominal reference plant.

4.3.2 wno **estimation**

The update rule in (4.3.2) suggests a straightforward implementation; however, the index condition $(G_0, G) \in \mathcal{C}$ (c.f. (2.1.9)) cannot be overlooked. This condition rules out pathological situations where, despite an apparently small closed-loop mismatch between G_0 and G, along a continuous perturbation path from G_0 to G an intermediate transfer function $\widetilde{G}$ may yield a ν-gap equal to one.

To understand the role of this condition, it is useful to specialize the general definition of Section 2.1.3 to the SISO setting considered in this chapter. Following the scalar interpretation of the Vinnicombe metric in [4], let $d(G_0, \widetilde{G})$ denote the normalized pointwise frequency-domain distance underlying (4.3.1), *i.e.*, the quantity inside the supremum. Even if the distance between the endpoint plants G_0 and G is small, a continuous perturbation from G_0 to G may contain an intermediate plant $\widetilde{G}$ for which the pointwise chordal distance becomes one. Indeed, for any such intermediate plant,

$$1 - d^2(G_0, \widetilde{G}) = \frac{(1 + \widetilde{G}(e^{j\omega})G_0(e^{-j\omega}))(1 + \widetilde{G}(e^{-j\omega})G_0(e^{j\omega}))}{(1 + |G_0(e^{j\omega})|^2)(1 + |\widetilde{G}(e^{j\omega})|^2)}. \tag{4.3.3}$$

Hence, the pointwise distance reaches one whenever

$$1 + \widetilde{G}(e^{j\omega})G_0(e^{-j\omega}) = 0 \tag{4.3.4}$$

for some frequency ω.

To determine whether such a crossing can occur along the perturbation path, *i.e.*, as $\widetilde{G}$ is continuously perturbed from G_0 to G, we define the scalar index functions

$$f_1(z) := 1 + G_0(z)G_0(z^{-1}),$$
$$f_2(z) := 1 + G(z)G_0(z^{-1}).$$

If f_1 and f_2 do not have the same number of zeros outside the unit circle, continuity implies that an intermediate transfer function $\widetilde{G}$ exists for which (4.3.4) holds. Consequently, the SISO admissibility condition requires f_1 and f_2 to have the same number of zeros in the unstable region, *i.e.*, outside the unit disk, $|z| > 1$.

Accordingly, we verify that $(G_0, G) \in \mathcal{C}$ in a data-driven fashion by computing winding numbers of the index functions f_1 and f_2, i.e., by checking that the Nyquist curve of $G(e^{j\omega})G_0(e^{-j\omega})$ does not encircle the critical point -1 for $\omega \in (-\pi, \pi]$.

Using the same information from the Fourier transform of the input and outputs of (4.3.2), and inspired by Welch's method [69] we compute f_2 as follows:

$$f_2(e^{j\omega_k}) \approx 1 + \frac{\left(\sum_{n=1}^{N_{\text{acc}}} Y_n[k]\, U_n[k]^*\right)\left(\sum_{n=1}^{N_{\text{acc}}} Y_{0,n}[k]\, U_n[k]^*\right)^*}{\left(\max\left(\sum_{n=1}^{N_{\text{acc}}} |U_n[k]|^2,\, \varepsilon_0\right)\right)^2}, \tag{4.3.5}$$

where n is the current iteration, $\omega_k := 2\pi k/N$, $k \in \{0, 1, \dots, N-1\}$ is the DFT grid on the unit circle, N_{acc} is the number of iterations used to approximate f_2, and ϵ_0 is a

small number to avoid division by 0. An analogous expression can be written for f_1 by replacing $(Y_n[k], Y_{0,n}[k])$ with $(Y_{0,n}[k], Y_{0,n}[k])$ in (4.3.5). Following Welch's averaged periodogram estimator [5, Ch. 6, Eq. (6.70)], we estimate the index functions by averaging across power iteration experiments; this reduces variance in the estimated index functions, following the standard spectral-analysis principle that averaging/smoothing empirical transfer-function estimates (ETFEs) trades variance reduction against finite-record bias [5, Secs. 6.3–6.4]. Unlike the classical Welch method, which partitions a single record into multiple sub-batches and recomputes the DFT for each batch, our approach reuses the DFTs that are already computed at each power iteration step. We therefore average over the first $N_{\text{acc}} \ll M$ iterations: as the power method progresses, the redesigned input becomes increasingly narrowband around the peak frequency, which reduces spectral coverage and makes later iterations less informative for index verification [30].

Finally, for $i \in \{1, 2\}$, let $f_i[k] := f_i(e^{j\omega_k})$ denote the samples of f_i on the DFT grid. We compute the winding numbers $\text{wno}(f_2)$ and $\text{wno}(f_1)$ by summing discrete phase increments of f_i around the unit circle, where we set $f_i[N] := f_i[0]$ to close the sampled path. Specifically, we define

$$\Theta_i = \sum_{k=0}^{N-1} \arg\left(\frac{f_i[k+1]}{f_i[k]}\right), \ \text{wno}(f_i) = \text{round}\left(\frac{\Theta_i}{2\pi}\right). \tag{4.3.6}$$

The data-driven winding-number computation is summarized in Algorithm 4.2.

These considerations lead to the pseudo-code for the data-driven computation of the Vinnicombe metric in Algorithm 4.1. The procedure terminates early if $\text{wno}(f_1) \neq \text{wno}(f_2)$ within the first N_{acc} iterations (indicating $(G_0, G) \notin \mathcal{C}$); otherwise, it runs for the full M iterations.

Algorithm 4.1 Data-driven estimation of the ν-gap

Require: A random input vector $\mathbf{u}^0 \in \mathbb{R}^N$
for $n = 0, 1, 2, \dots, M-1$ **do**
  Apply $\mathbf{u}^n$ to G and G_0, and collect outputs $\mathbf{y}^n$ and $\mathbf{y}_0^n$, respectively.
  Compute the DFT of $\mathbf{u}^n$, $\mathbf{y}^n$ and $\mathbf{y}_0^n$, and calculate (4.3.2), for each frequency ω.
  if $n < N_{\text{acc}}$, update accumulators via

$$S_{yu} \mathrel{+}= Y_n[k]U_n[k]^*, \ S_{y_0u} \mathrel{+}= Y_{0,n}[k]U_n[k]^*, \ S_u \mathrel{+}= |U_n[k]|^2$$

  end if
  if $n = N_{\text{acc}}$ **then**
    Run Algorithm 4.2
    if `inC` is false, **Terminate** and set $\delta_\nu \leftarrow 1$ **end if**
  end if
  Compute $\tilde{\mathbf{u}}^{n+1}$ as the IDFT of $\tilde{\mathbf{U}}^{n+1}$
  Normalize $\tilde{\mathbf{u}}^{n+1}$ by its 2-norm to obtain $\mathbf{u}^{n+1}$
end for
Estimate δ_ν as the 2-norm of $\tilde{\mathbf{u}}^M$

Algorithm 4.2 Data-driven index condition $(G_1, G_2) \in \mathcal{C}$

Require: Accumulators $(S_{yu}, S_{y_0 u}, S_u)$, N_{acc} batches; tolerance $\varepsilon_0 > 0$; threshold $\texttt{tol}_f > 0$.

1: **for** $n = 0, 1, \dots, N_{\text{acc}}$, Construct f_1 and f_2 via (4.3.5), **end for**
2: Compute minimum distances to the origin: $m_1 = \min_k |f_1[k]|$ and $m_2 = \min_k |f_2[k]|$.
3: Compute winding numbers via (4.3.6), with $f_i[N] = f_i[0],\ i \in \{1, 2\}$
4: Set $\texttt{inC} \leftarrow (m_1 > \texttt{tol}_f) \wedge (m_2 > \texttt{tol}_f) \wedge (\text{wno}(f_1) = \text{wno}(f_2))$

4.4 Experiments

This section presents simulation studies to assess Algorithms 4.1–4.2. The goal is twofold: first, to validate the proposed estimator against the `nugap` output of MATLAB's `gapmetric` routine; and second, to illustrate how the method behaves in both an industrially motivated benchmark and a synthetic example where the winding-number condition can be checked analytically. The study includes: (i) a compact heavy-duty gas-turbine benchmark, used only to illustrate practical relevance; and (ii) a textbook example for which the admissibility condition fails.

4.4.1 Heavy-Duty Gas-Turbine Benchmark

We use a heavy-duty gas-turbine (HDGT) benchmark, representative of large industrial power-generation units, to illustrate the proposed estimator in a practically motivated setting. Following Rowen's simplified representation [70], we consider the small-signal input–output map from the gas control valve (GCV) position u_{GCV} to generated power P. After linearization around an operating point, the incremental plant is given by

$$G(s) := \frac{\Delta P(s)}{\Delta u_{\text{GCV}}(s)} = \frac{B e^{-sT_{\text{CR}}}}{(T_f s + 1 + K_F)(T_{\text{cd}} s + 1)}. \tag{4.4.7}$$

Here, T_f, K_F, T_{CR}, T_{cd}, and B are the corresponding Rowen model parameters. The HDGT model is used only to generate the benchmark data and the MATLAB reference values; in contrast, the proposed algorithm receives only time-domain input–output signals from the systems under comparison and does not use the internal gas-turbine model or its parameters.

A.1. Nominal Model

For the nominal-model experiment, we use two Rowen-parameterized HDGT models from the literature: the parameter set in [71, Table VII] is treated as the closest available description of the true plant G, while the identified model in [72, Table II] is used as the nominal model G_0 from u_{GCV} to generated power P.

Assume that we have derived a controller C that stabilizes G_0 on a high-fidelity industrial simulator. The objective is to assess whether deploying C on the true operating power station is justified from a closed-loop robustness viewpoint, by estimating $\delta_\nu(G_0, G)$ from data. To emulate realistic operating constraints, we

consider that only small variations in generated power are permitted during testing. Under these constraints, we apply a bounded excitation signal $\mathbf{u}^0 \in \mathbb{R}^N$ (small-signal variations around the operating point) to both G and G_0, collect the corresponding input–output data, and run Algorithm 4.1 to obtain a data-driven estimate of the ν-gap and the associated index-condition check.

As shown in Fig. 4.4.1, the Monte Carlo (MC) average of the proposed estimate $\hat{\delta}_\nu$, computed over 100 runs, becomes nearly settled after roughly 60 iterations and approaches MATLAB's reference value $\delta_\nu = 0.2172$, indicating convergence of the proposed power iteration scheme. In each MC run, the initial input $\mathbf{u}^0$ is regenerated using a different random seed as a normalized broadband Gaussian signal, while independent measurement-noise realizations are added to the outputs; the shaded region in Fig. 4.4.1 shows the empirical 95% confidence bounds computed over the 100 MC runs. The estimate is computed using $N_{\text{acc}} = 10$, $N = 9000$, $T_s = 0.05$, and measurement-noise variance $\sigma_y^2 = 0.01$ on both plants.

To further interpret this result, we inspect the dominant frequency in the last iteration and estimate the nominal local gain as $p_0(\omega_{\text{peak}}) \approx \frac{Y_0(\omega_{\text{peak}})}{U(\omega_{\text{peak}})}$, where $\mathbf{Y}_0$ and $\mathbf{U}$ are the DFTs of the nominal output and input, respectively. In this case, the dominant peak occurs at $\omega_{\text{peak}} \approx 0.136$ rad/sample, with $|p_0(\omega_{\text{peak}})| \approx 0.54$. According to Vinnicombe's Riemann-sphere interpretation [50, Section 3], a ν-gap around 0.2 corresponds to a relatively small discrepancy near unity gain, namely, at frequencies where $|p_0(\omega)| \approx 1$. In the present case, however, the maximizing frequency lies below unity gain, since $|p_0(\omega_{\text{peak}})| \approx 0.54$. Still, the value $\hat{\delta}_\nu = 0.2172$ suggests that the nominal and real plants are reasonably close from a closed-loop viewpoint. In particular, if the deployed controller C satisfies $b_{G_0,C} > 0.2172$, then Proposition 2.1 guarantees that it also stabilizes the true plant.

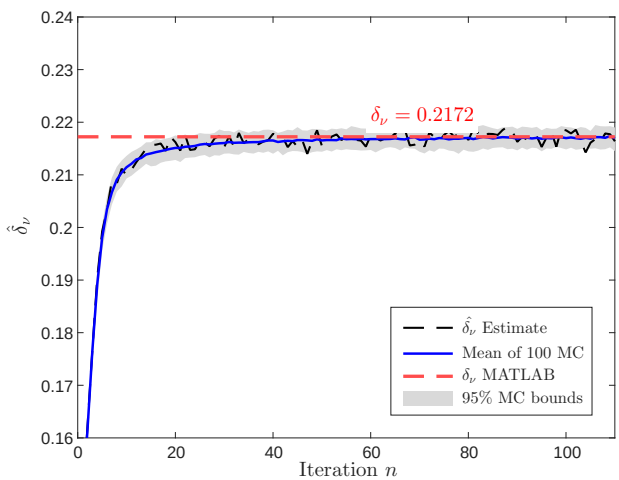


Figure 4.4.1: Estimate of the ν-gap $\hat{\delta}_\nu$ (black, dashed), average of 100 Monte Carlo (MC) simulations (blue, solid), MATLAB reference δ_ν (red, dashed), and 95% MC bounds (gray shaded area).

A.2. Unknown Models

We next consider the case where both systems are treated as black boxes. Let $G_1(z)$ and $G_2(z)$ denote two unknown discrete-time transfer functions from the GCV input u_{GCV} to generated power P, obtained from Rowen parameter sets corresponding to GE Frame 9F and Frame 6F units, respectively [70, Notes 4–7]. By applying Algorithm 4.1 to short, bounded input–output tests on each unit, we estimate $\delta_\nu(G_1, G_2)$. This enables an assessment of whether a GCV controller that stabilizes G_1 could deliver comparable closed-loop behavior on G_2, i.e., whether the plant-to-plant mismatch is sufficiently small from a closed-loop viewpoint.

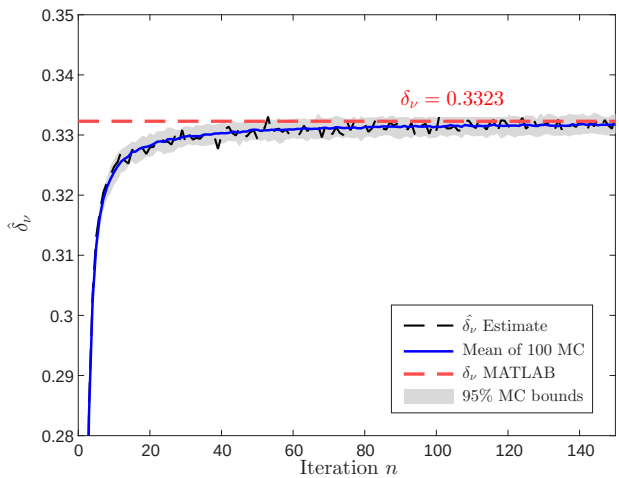


Figure 4.4.2: Estimate of the ν-gap $\hat{\delta}_\nu$ (black, dashed), average of 100 MC simulations (blue, solid), MATLAB reference δ_ν (red, dashed), and 95% MC bounds (gray shaded area).

Using the same parameters as before ($N_{\text{acc}} = 10$, $N = 9000$, $T_s = 0.05$, and $\sigma_y^2 = 0.01$), the MC mean of the proposed estimate over 100 runs stabilizes after about 100 iterations and converges to MATLAB's value $\delta_\nu = 0.3323$, as shown in Fig. 4.4.2. The dominant peak is attained at $\omega_{\text{peak}} \approx 0.0126$ rad/sample, with $|p_1(\omega_{\text{peak}})| \approx 0.644$ for the reference system G_1, which places the maximizing frequency closer to unity gain. From Vinnicombe's Riemann-sphere interpretation, this makes the geometric intuition behind the ν-gap more directly relevant here; nevertheless, the larger value $\hat{\delta}_\nu = 0.3323$ indicates a moderate but non-negligible closed-loop mismatch. From a practical viewpoint, this suggests that a controller tuned for the GE 9F unit, represented by G_1, may also be applicable to the GE 6F unit, represented by G_2. A formal guarantee, however, would require estimating the corresponding stability margin $b_{G_1,C}$, for example through a power-iteration-based estimate of $\|T(G_1, C)\|_\infty$. In that case, Proposition 2.1 would apply whenever $b_{G_1,C} > 0.3323$.

4.4.2 Synthetic Example: Poles in RHP

We evaluate the ν-gap metric on a discrete-time SISO case study with sampling time $T_s = 1\,\mathrm{s}$. Consider the stable plant

$$G_1(z) = \frac{1 - z^{-1}}{1 - 0.8z^{-1}}, \qquad G_2(z) = 1.8\, z^{-1}\, G_1(z). \tag{4.4.8}$$

Plant G_2 differs from G_1 by a gain of 1.8 and an additional one-sample delay z^{-1}, which introduces extra phase lag. We can obtain explicit expressions for the index functions on the unit circle $z = e^{j\omega}$. Since the plants are SISO with real coefficients, $G_1(e^{-j\omega}) = \overline{G_1(e^{j\omega})}$, and thus

$$f_1(e^{j\omega}) = 1 + G_1(e^{j\omega})G_1(e^{-j\omega}) = 1 + \left|G_1(e^{j\omega})\right|^2,$$

$$f_2(e^{j\omega}) = 1 + G_2(e^{j\omega})G_1(e^{-j\omega}) = 1 + G_2(e^{j\omega})\,\overline{G_1(e^{j\omega})}.$$

By (4.4.8), we can simplify $f_2(e^{j\omega})$ as

$$f_2(e^{j\omega}) = 1 + 1.8\, e^{-j\omega}\left|G_1(e^{j\omega})\right|^2.$$

Note that $f_1(e^{j\omega})$ is real and strictly positive for all ω, hence $\mathrm{wno}(f_1) = 0$. In contrast, the factor $e^{-j\omega}$ induced by the one-sample delay rotates the complex term in $f_2(e^{j\omega})$ as ω varies, so the curve $f_2(e^{j\omega})$ may encircle the origin, yielding $\mathrm{wno}(f_2) \neq 0$.

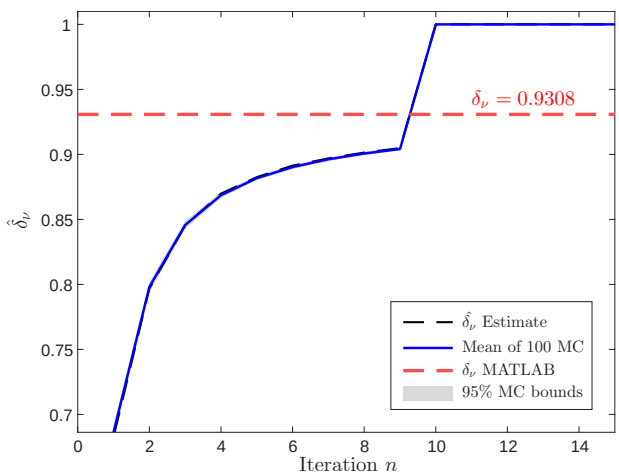


Figure 4.4.3: Estimate of the ν-gap $\hat{\delta}_\nu$ (black, dashed), average of 100 MC simulations (blue, solid), MATLAB reference δ_ν (red, dashed), and 95% MC bounds (gray shaded area).

Using $N_{\mathrm{acc}} = 10$, $N = 1000$, $T_s = 1$, and $\sigma_y^2 = 0.01$, Fig. 4.4.3 shows that the power iteration estimate initially approaches MATLAB's value $\delta_\nu = 0.9308$, with narrow, barely visible 95% MC confidence bounds. However, at $n = N_{\mathrm{acc}}$, Algorithm 4.2 detects $\mathrm{wno}(f_1) \neq \mathrm{wno}(f_2)$, indicating that the pair fails the set $\mathcal{C}$ condition and triggering a hard stop.

4.5 Summary and Conclusions

This chapter has presented an identification-free, data-driven procedure for estimating the Vinnicombe ν-gap between two discrete-time SISO systems directly from time-domain input–output experiments. The proposed method combines an iterative estimation scheme with a data-driven admissibility test, thereby providing not only an estimate of the metric itself, but also a mechanism to detect violations of the index condition required for the ν-gap to be well defined. In this sense, the approach delivers a control-relevant characterization of model mismatch without requiring the explicit identification of parametric models. The numerical studies, including an industrially motivated example, show close agreement with MATLAB's `gapmetric` implementation and demonstrate the ability of the method to correctly identify cases in which the admissibility condition fails.

Despite these encouraging results, the development in this chapter also highlights two natural directions for improvement. First, as in Chapter 3, part of the computational construction is still formulated through frequency-domain quantities that must ultimately be related to time-domain experiments. Although this already enables a data-driven implementation, it also suggests the importance of deriving methods that operate entirely in the time domain, avoiding the need to explicitly construct frequency-domain approximations of the relevant operators. Such a formulation would be more directly aligned with the experimental setting, where only finite-length input–output trajectories are available.

Second, the present chapter focuses on the SISO case. While this setting is essential for introducing the main ideas and validating the data-driven admissibility mechanism, many control applications of practical interest are inherently multivariable. Extending the method to MIMO systems is therefore a crucial step. This extension is not merely a matter of notation: in the MIMO case, the ν-gap involves operator-valued quantities, singular values, and a nontrivial index condition that must be verified to avoid incorrect approximations of the metric.

These two directions are addressed in the following chapter, where the Vinnicombe metric is reformulated from a time-domain operator perspective and extended to the MIMO setting. In this way, the SISO methodology established here serves as the foundation for a time-domain operator-based, multivariable framework.

5 MIMO Robustness: Krylov-Based Estimation of the Vinnicombe Metric

Advancing the methodology introduced in Chapter 4, this chapter considers the data-driven quantification of plant mismatch for multi-input multi-output (MIMO) systems. A novel approach based on Krylov subspace methods is developed to quantify the extent of the discrepancy between a nominal model and the true plant that can be tolerated by a controller designed for the nominal model.

5.1 Introduction

Chapter 4 introduced a data-driven procedure for estimating the Vinnicombe metric directly from input–output experiments in the SISO setting. In that case, the scalar structure of the problem permits the normalized plant mismatch underlying the ν-gap to be evaluated through an iterative power method, while the corresponding admissibility condition can be verified through a winding-number calculation. Many systems of practical interest, however, are inherently multivariable, and extending this framework to MIMO systems is not merely a matter of replacing scalar transfer functions by transfer matrices.

The main difficulty arises from the structure of the MIMO ν-gap itself [49]. For an admissible pair of stable LTI plants $\mathbf{G}_0$ and $\mathbf{G}$, the metric depends on the induced norm of the normalized difference operator

$$\mathbf{M} = \left(\mathbf{I} + \mathbf{G}\mathbf{G}^*\right)^{-1/2} \left(\mathbf{G}_0 - \mathbf{G}\right) \left(\mathbf{I} + \mathbf{G}_0^*\mathbf{G}_0\right)^{-1/2}, \tag{5.1.1}$$

whose $\mathcal{H}_\infty$ norm appears in the definition of the Vinnicombe metric in Section 2.1.3, cf. (2.1.9).

Unlike in the SISO case, its computation therefore involves matrix-valued normalization factors and the largest singular value of a multivariable operator.

Directly constructing the corresponding finite-horizon lifted matrices rapidly becomes computationally expensive as the number of input–output channels and the experimental horizon increase, while also departing from the data-driven philosophy pursued in this thesis.

Data-driven methods for multivariable systems have received increasing attention in recent years, particularly in problems of controller synthesis and predictive control. Behavioral approaches such as Data-enabled Predictive Control (DeePC) exploit measured input–output trajectories to construct control policies without explicitly identifying a parametric plant model [73], while related developments have addressed direct data-driven output-feedback control for general MIMO LTI systems [74] and model-free adaptive predictive control for nonlinear MIMO systems [75]. From a robust-control perspective, frequency-response data have also been used directly for multivariable controller design. In particular, robust MIMO synthesis with $\mathcal{H}_2$, $\mathcal{H}_\infty$, and loop-shaping specifications has been formulated through convex optimization [76], while related frequency-domain approaches have considered the design of structured MIMO controllers, including PID controllers, through iterative LMI restrictions [77].

While these developments demonstrate the potential of data and experimental frequency-response information for multivariable control without relying on a complete parametric description of the plant, the problem considered in this chapter is complementary. Rather than designing a controller directly from data, we seek to quantify the control-relevant discrepancy between a nominal MIMO model and an unknown plant directly from input–output experiments. Accordingly, building on the MIMO ν-gap definition introduced in Section 2.1.3, we develop a matrix-free formulation for estimating the metric in which only the action of the relevant operators on finite-length signals is required. In particular:

- We formulate the estimation of the MIMO Vinnicombe metric through finite-horizon operator actions, avoiding the explicit construction of large lifted system matrices and the parametric identification of the unknown plant.
- We develop two iterative estimators of the dominant singular value underlying the metric: a power iteration method applied to $\mathbf{M}^*\mathbf{M}$ and a Golub–Kahan–Lanczos approach operating directly on $\mathbf{M}$ and its adjoint.
- We extend the data-driven admissibility check of Chapter 4 to stable MIMO systems and compare the resulting estimates against explicit finite-horizon computations and standard model-based evaluations of the ν-gap.

5.2 Problem Statement

Consider a nominal discrete-time LTI MIMO plant $\mathbf{G}_0$ and an unknown true plant $\mathbf{G}$ of compatible dimensions. As in Chapter 4, our objective is to quantify the control-relevant discrepancy between the nominal and true plants directly from input–output experiments, without identifying a parametric model of $\mathbf{G}$. In the present chapter, however, both plants are multivariable. The corresponding Vinnicombe metric is given by Definition 2.1 in Section 2.1.3, together with its associated admissibility condition.

Assuming that $(\mathbf{G}_0, \mathbf{G}) \in \mathcal{C}$, the computation of the ν-gap requires evaluating the induced norm of the normalized difference operator $\mathbf{M}$ introduced in (5.1.1). For the subsequent developments, it is convenient to write

$$\mathbf{M} = \mathbf{A}^{-1/2}\mathbf{B}\mathbf{C}^{-1/2}, \tag{5.2.2}$$

where

$$\mathbf{A} = \mathbf{I} + \mathbf{G}\mathbf{G}^*, \qquad \mathbf{B} = \mathbf{G}_0 - \mathbf{G}, \qquad \mathbf{C} = \mathbf{I} + \mathbf{G}_0^*\mathbf{G}_0.$$

Hence,

$$\delta_\nu(\mathbf{G}_0, \mathbf{G}) = \|\mathbf{M}\|_\infty.$$

Although Section 2.2.1 provides an iterative mechanism for estimating an induced norm, its application here requires the action of $\mathbf{M}$ on arbitrary input signals. For a given input $\mathbf{u}$, the product $\mathbf{B}\mathbf{u}$ can be obtained directly from the difference between the responses of $\mathbf{G}_0$ and $\mathbf{G}$ to the same input. The main difficulty lies in the normalization operators $\mathbf{A}^{-1/2}$ and $\mathbf{C}^{-1/2}$, whose explicit construction would require forming and manipulating large finite-horizon lifted matrices.

The problem addressed in this chapter is therefore to approximate the action of these inverse-square-root operators, and consequently that of $\mathbf{M}$, using only matrix-free operator evaluations. To this end, we exploit the Krylov-subspace methods introduced in Section 2.2.2, which allow matrix-function–vector products of the form $\mathbf{A}^{-1/2}\mathbf{b}$ and $\mathbf{C}^{-1/2}\mathbf{b}$, for a vector $\mathbf{b}$ of compatible dimension, to be approximated from successive operator actions. In addition to the power iteration approach, we consider a direct singular-value estimation strategy based on Golub–Kahan–Lanczos bidiagonalization [78, Sec. 6.3.3]. Since this latter method requires additional machinery beyond the Krylov methods reviewed in Chapter 2, we briefly introduce it in the following section before presenting the proposed data-driven estimators.

5.3 Preliminaries

This section introduces the additional numerical methods required in this chapter. In particular, we review the Golub–Kahan–Lanczos (GKL) bidiagonalization procedure for singular-value estimation and the multivariable extension of the power iteration method introduced in Section 2.2.1. The presentation follows [78, 31].

5.3.1 Golub-Kahan-Lanczos Algorithm

While the Lanczos algorithm, introduced in Section 2.2.2, is naturally associated with Hermitian eigenvalue problems, the Golub–Kahan–Lanczos (GKL) bidiagonalization is particularly suited to singular-value computations for a general linear operator $\mathbf{A} \in \mathbb{C}^{p\times m}$, where m and p denote the dimensions of its input and output spaces, respectively. This is particularly relevant here, since the induced 2-norm of the operator in (5.2.2) is determined by its largest singular value.

The GKL bidiagonalization constructs orthonormal bases collected in matrices $\mathbf{U}$ and $\mathbf{V}$ such that the projection of $\mathbf{A}$ onto these bases takes bidiagonal form,

$$\mathbf{U}^*\mathbf{A}\mathbf{V} = \mathbf{B} = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ & \alpha_2 & \beta_2 & & \\ & & \alpha_3 & \beta_3 & \\ & & & \ddots & \ddots \end{bmatrix}. \tag{5.3.3}$$

Let $\mathbf{u}_k \in \mathbb{C}^p$ and $\mathbf{v}_k \in \mathbb{C}^m$ denote the kth columns of $\mathbf{U}$ and $\mathbf{V}$, respectively. The coefficients α_k and β_k correspond to the diagonal and superdiagonal entries of $\mathbf{B}$ and are given by

$$\alpha_k = \mathbf{u}_k^*\mathbf{A}\mathbf{v}_k, \qquad \beta_k = \mathbf{u}_k^*\mathbf{A}\mathbf{v}_{k+1}.$$

These coefficients are real and nonnegative as a consequence of the normalization employed in the bidiagonalization, even when $\mathbf{A}$ is complex. The first column $\mathbf{v}_1$ of $\mathbf{V}$ can be chosen as an arbitrary unit vector, after which the remaining columns of $\mathbf{U}$ and $\mathbf{V}$ are generated recursively.

From (5.3.3), we may derive a double recursion for the columns $\mathbf{u}_k$ and $\mathbf{v}_k$ of $\mathbf{U}$ and $\mathbf{V}$, respectively. Left multiplying (5.3.3) by $\mathbf{U}$ gives

$$\mathbf{A}\begin{bmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_n \end{bmatrix} = \begin{bmatrix} \mathbf{u}_1 & \cdots & \mathbf{u}_n \end{bmatrix} \begin{bmatrix} \alpha_1 & \beta_1 & & & & \\ & \alpha_2 & \beta_2 & & & \\ & & \alpha_3 & \beta_3 & & \\ & & & \ddots & \ddots & \\ & & & & & \alpha_n \end{bmatrix}.$$

Equating the kth columns on both sides yields

$$\alpha_k\mathbf{u}_k = \mathbf{A}\mathbf{v}_k - \beta_{k-1}\mathbf{u}_{k-1}, \qquad \beta_0 = 0.$$

Similarly, taking the adjoint relation gives $\mathbf{A}^*\mathbf{U} = \mathbf{V}\mathbf{B}^*$, from which

$$\beta_k\mathbf{v}_{k+1} = \mathbf{A}^*\mathbf{u}_k - \alpha_k\mathbf{v}_k$$

is obtained. Since the vectors $\mathbf{u}_k$ and $\mathbf{v}_k$ are normalized at each iteration, the recurrence coefficients can be equivalently computed as

$$\alpha_k = \left\|\mathbf{A}\mathbf{v}_k - \beta_{k-1}\mathbf{u}_{k-1}\right\|_2, \qquad \beta_k = \left\|\mathbf{A}^*\mathbf{u}_k - \alpha_k\mathbf{v}_k\right\|_2.$$

The resulting bidiagonalization procedure is summarized in Algorithm 5.1.

After k bidiagonalization steps, the projection of $\mathbf{A}$ is represented by the $k \times k$ bidiagonal matrix $\mathbf{B}_k$. The dominant singular value of the original operator can then be approximated from the considerably smaller matrix $\mathbf{B}_k$. In particular,

$$\bar{\sigma}(\mathbf{B}_k)^2 = \lambda_{\max}\left(\mathbf{B}_k^*\mathbf{B}_k\right). \tag{5.3.4}$$

Hence, as the Krylov subspace is expanded, $\bar{\sigma}(\mathbf{B}_k)$ provides an iterative approximation of the dominant singular value of $\mathbf{A}$. Taking $\mathbf{A} = \mathbf{M}$ will therefore provide the approximation required for the induced norm appearing in the MIMO ν-gap.

Algorithm 5.1 Golub–Kahan–Lanczos Algorithm

Choose $\mathbf{v}_1$ such that $\|\mathbf{v}_1\|_2 = 1$ and set $\beta_0 = 0$
for $k = 1, 2, \ldots$ **do**
  $\mathbf{u}_k \leftarrow \mathbf{A}\mathbf{v}_k - \beta_{k-1}\mathbf{u}_{k-1}$
  $\alpha_k \leftarrow \|\mathbf{u}_k\|_2$
  $\mathbf{u}_k \leftarrow \mathbf{u}_k/\alpha_k$
  $\mathbf{v}_{k+1} \leftarrow \mathbf{A}^*\mathbf{u}_k - \alpha_k\mathbf{v}_k$
  $\beta_k \leftarrow \|\mathbf{v}_{k+1}\|_2$
  $\mathbf{v}_{k+1} \leftarrow \mathbf{v}_{k+1}/\beta_k$
end for

5.3.2 Data-Driven Estimation of the MIMO Induced 2-Norm

Section 2.2.1 introduced the power iteration method for estimating the induced 2-norm of a linear operator. In the MIMO setting, however, a direct data-driven implementation requires particular attention to the action of the adjoint operator, since the power iteration involves successive applications of $\mathbf{G}\mathbf{G}^*$ (or $\mathbf{G}_0^*\mathbf{G}_0$). We therefore briefly review the experimental procedure of [31], which enables the action of the adjoint of a multivariable finite-horizon operator to be recovered from input–output experiments.

To this end, consider a linear discrete-time stable causal and proper system of the form

$$\mathbf{y}_k = \mathbf{G}(q)\mathbf{u}_k + \mathbf{e}_k$$

where $\{\mathbf{u}_k\} \in \mathbb{R}^{m\times 1}$ is an input signal, $\mathbf{G} \in \mathcal{RH}_\infty^{p\times m}$ is a transfer function, $\{\mathbf{y}_k\} \in \mathbb{R}^{p\times 1}$ is the output of the system, and $\{\mathbf{e}_k\} \in \mathbb{R}^{p\times 1}$ is Gaussian, not necessarily white noise. Next, recall that the $\mathcal{H}_\infty$ norm is an induced norm, i.e., $\|\mathbf{G}\|_\infty = \|\mathbf{G}\|_{i2}$; see [79, App. A.5.7] for a proof. Assume that the signals have a finite length $N \in \mathbb{N}$ with no measurement noise, and denote by underlined symbols the corresponding finite-time (lifted) representations over the horizon N; for example, $\underline{\mathbf{u}_{gj}} = [u_j[0],\, u_j[1],\, \ldots,\, u_j[N-1]]^T$ denotes the finite-length trajectory of input channel j, with an analogous definition for $\underline{\mathbf{y}_{gi}}$. Then, the mapping from input to output can be written as

$$\underbrace{\begin{bmatrix}\underline{\mathbf{y}_{g1}}\\ \vdots \\ \underline{\mathbf{y}_{gp}}\end{bmatrix}}_{:=\underline{\mathbf{y}_g}} = \underbrace{\begin{bmatrix}\underline{\mathbf{g}_{11}} & \cdots & \underline{\mathbf{g}_{1m}}\\ \vdots & \ddots & \vdots \\ \underline{\mathbf{g}_{p1}} & \cdots & \underline{\mathbf{g}_{pm}}\end{bmatrix}}_{:=\underline{\mathbf{G}}} \underbrace{\begin{bmatrix}\underline{\mathbf{u}_{g1}}\\ \vdots \\ \underline{\mathbf{u}_{gm}}\end{bmatrix}}_{:=\underline{\mathbf{u}_g}} \tag{5.3.5}$$

where $\underline{\mathbf{u}_{gj}} \in \mathbb{R}^{N\times 1}$, $j = 1, \ldots, m$, $\underline{\mathbf{y}_{gi}} \in \mathbb{R}^{N\times 1}$, $i = 1, \ldots, p$ and $\underline{\mathbf{G}} \in \mathbb{R}^{pN\times mN}$. Each block $\underline{\mathbf{g}_{ij}} \in \mathbb{R}^{N\times N}$ denotes the finite-horizon lifted representation of the SISO channel from input j to output i. Hence,

$$\|\underline{\mathbf{G}}\|_{i2} = \sup_{\underline{\mathbf{u}_g}\in\mathbb{R}^{mN\times 1}\setminus\{0\}} \frac{\|\underline{\mathbf{y}_g}\|}{\|\underline{\mathbf{u}_g}\|} \tag{5.3.6}$$

where $\|\underline{\mathbf{G}}\|_{i2} \to \|\mathbf{G}\|_{\infty}$ for $N \to \infty$, see [30, Th. 3]. In addition, (5.3.6) is equivalent to

$$\|\underline{\mathbf{G}}\|_{i2}^2 = \sup_{\underline{\mathbf{u}}_g \neq 0} \frac{\underline{\mathbf{u}}_g{}^T \underline{\mathbf{G}}^T \underline{\mathbf{G}}\, \underline{\mathbf{u}}_g}{\underline{\mathbf{u}}_g{}^T \underline{\mathbf{u}}_g} = \lambda^{\max}\left(\underline{\mathbf{G}}^T \underline{\mathbf{G}}\right). \tag{5.3.7}$$

The quotient in (5.3.7) is the Rayleigh quotient of $\underline{\mathbf{G}}^T \underline{\mathbf{G}}$. Its maximum corresponds to the largest eigenvalue of $\underline{\mathbf{G}}^T \underline{\mathbf{G}}$ and can be estimated through the power iteration method introduced in Section 2.2.1, without requiring an explicit parametric model of $\mathbf{G}$.

To proceed, observe that (5.3.7) requires the transpose of $\underline{\mathbf{G}}$ which for the multivariate case, as shown in [31], is given by

$$\underline{\mathbf{G}}^T = \underbrace{\begin{bmatrix} \mathcal{T}_N & & 0 \\ & \ddots & \\ 0 & & \mathcal{T}_N \end{bmatrix}}_{:=\mathcal{T}_{mN}} \underbrace{\begin{bmatrix} \underline{g}_{11} & \cdots & \underline{g}_{p1} \\ \vdots & \ddots & \vdots \\ \underline{g}_{1m} & \cdots & \underline{g}_{pm} \end{bmatrix}}_{:=\underline{\widetilde{\mathbf{G}}}} \underbrace{\begin{bmatrix} \mathcal{T}_N & & 0 \\ & \ddots & \\ 0 & & \mathcal{T}_N \end{bmatrix}}_{:=\mathcal{T}_{pN}} \tag{5.3.8}$$

where $\underline{\widetilde{\mathbf{G}}}$ is the finite time representation of the transfer function matrix $\widetilde{\mathbf{G}}$ and $\mathcal{T}_N$ is a time-reversal operator or permutation matrix of size $N \times N$, i.e.,

$$\widetilde{\mathbf{G}} = \begin{bmatrix} g_{11} & \cdots & g_{p1} \\ \vdots & \ddots & \vdots \\ g_{1m} & \cdots & g_{pm} \end{bmatrix}, \qquad \mathcal{T}_N = \begin{bmatrix} 0 & \cdots & 0 & 1 \\ 0 & \cdots & 1 & 0 \\ \vdots & & & \vdots \\ 1 & \cdots & 0 & 0 \end{bmatrix}.$$

The key step in the forthcoming developments is the observation that

$$\widetilde{\mathbf{G}} = \sum_{\alpha=1}^{m} \sum_{\beta=1}^{p} \mathcal{J}_{\alpha\beta} \mathbf{G} \mathcal{J}_{\alpha\beta} \tag{5.3.9}$$

where

$$\mathcal{J}_{\alpha\beta} = \begin{bmatrix} 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & & \ddots & 0 \\ 0 & \cdots & 1 & \cdots & 0 \\ \vdots & \ddots & & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{m \times p}, \tag{5.3.10}$$

and the 1 is on the (α, β)th location. To derive the finite time equivalent of (5.3.9), note that the finite time representation of $\mathcal{J}_{\alpha\beta}$ is given by

$$\underline{\mathcal{J}}_{\alpha\beta} = \begin{bmatrix} 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & & \ddots & 0 \\ 0 & \cdots & \mathbf{I}_N & \cdots & 0 \\ \vdots & \ddots & & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{mN \times pN}. \tag{5.3.11}$$

where $\underline{\mathcal{I}}_{\alpha\beta}$ is an $m \times p$ block matrix, with blocks of dimension $N \times N$, whose only nonzero block is $\mathbf{I}_N$ at the (α, β)th position. The corresponding system $\mathcal{I}_{\alpha\beta}$ can therefore be interpreted as a static channel-selection operator containing only a direct feedthrough term. Using these finite-horizon selection operators, the representation in (5.3.9) becomes

$$\widetilde{\underline{\mathbf{G}}} = \sum_{\alpha=1}^{m} \sum_{\beta=1}^{p} \underline{\mathcal{I}}_{\alpha\beta} \underline{\mathbf{G}} \underline{\mathcal{I}}_{\alpha\beta}. \tag{5.3.12}$$

The results in (5.3.8), (5.3.11) and (5.3.12) are the basis for the $\mathcal{H}_\infty$ norm estimation of a MIMO system required to compute the Vinnicombe metric in (2.1.9), as summarized in Algorithm 5.2.

Algorithm 5.2 MIMO $\|\underline{\mathbf{G}}\|_{i2}$ estimation

Set $i = 1$ and choose $\underline{\mathbf{u}_g}^{(1)} \in \mathbb{R}^{mN \times 1}$, $\underline{\mathbf{u}_g}^{(1)} \neq 0$, randomly
while the stopping criterion is not satisfied **do**
  Determine $\underline{\mathbf{y}_g}^{(i)} = \underline{\mathbf{G}}\, \underline{\mathbf{u}_g}^{(i)}$ by performing an experiment on the system
  Time-reverse: $\underline{\mathbf{x}_g}^{(i)} = \mathcal{T}_{pN} \underline{\mathbf{y}_g}^{(i)}$
  Set $\underline{\mathbf{z}_g}^{(i)} = 0$
  for $\alpha = 1, \dots, m$ **do**
    for $\beta = 1, \dots, p$ **do**
      $\underline{\mathbf{z}_g}^{(i)} \leftarrow \underline{\mathbf{z}_g}^{(i)} + \underline{\mathcal{I}}_{\alpha\beta} \underline{\mathbf{G}} \underline{\mathcal{I}}_{\alpha\beta} \underline{\mathbf{x}_g}^{(i)}$
    end for
  end for
  Time-reverse: $\underline{\mathbf{w}_g}^{(i)} = \mathcal{T}_{mN} \underline{\mathbf{z}_g}^{(i)}$
  Set $\underline{\mathbf{u}_g}^{(i+1)} = \underline{\mathbf{w}_g}^{(i)}$
  $i \leftarrow i + 1$
end while

From the iterates generated by Algorithm 5.2, the multivariable $\mathcal{H}_\infty$ norm can be estimated as

$$\hat{\gamma}_2^{(n)} = \sqrt{\frac{\left(\underline{\mathbf{u}_g}^{(n)}\right)^T \underline{\mathbf{w}_g}^{(n)}}{\left(\underline{\mathbf{u}_g}^{(n)}\right)^T \underline{\mathbf{u}_g}^{(n)}}}.$$

5.4 Proposed Approach

Analogously to the SISO case, the Vinnicombe metric provides a robustness measure for quantifying model mismatch in multi-input, multi-output systems. In this section, two data-driven approaches are proposed for estimating the Vinnicombe metric in the MIMO setting. The first one applies power iteration to the self-adjoint positive semidefinite operator $\mathbf{M}^*\mathbf{M}$, while Lanczos iterations are used internally to approximate the inverse-square-root normalization operators. The second approach, developed subsequently, applies Golub–Kahan–Lanczos bidiagonalization directly to $\mathbf{M}$ and $\mathbf{M}^*$.

5.4.1 Approach 1 (PI): Power Iteration on $\mathbf{M}^*\mathbf{M}$

Recall from Section 5.3.2 that, over a finite experimental horizon N, the induced 2-norm of the lifted operator $\underline{\mathbf{M}}$ satisfies

$$\|\underline{\mathbf{M}}\|_{i2}^2 = \lambda_{\max}\left(\underline{\mathbf{M}}^T\underline{\mathbf{M}}\right). \tag{5.4.13}$$

As N increases, this finite-horizon induced norm approaches the $\mathcal{H}_\infty$ norm appearing in the definition of the Vinnicombe metric. Hence, assuming that the admissibility condition is satisfied, the problem reduces to estimating the dominant eigenvalue of $\underline{\mathbf{M}}^T\underline{\mathbf{M}}$.

From (5.2.2), and using the fact that $\mathbf{A}$ and $\mathbf{C}$ are Hermitian positive definite, we have

$$\mathbf{M}^*\mathbf{M} = \mathbf{C}^{-1/2}\mathbf{B}^*\mathbf{A}^{-1}\mathbf{B}\mathbf{C}^{-1/2}. \tag{5.4.14}$$

Therefore, power iteration can be applied without explicitly forming $\mathbf{M}$ or $\mathbf{M}^*$, provided that the action of each operator in (5.4.14) can be evaluated from input–output experiments. We next describe these operator actions.

Inverse-square-root normalization via Lanczos

Consider first the computation of

$$\mathbf{x} = \mathbf{C}^{-1/2}\mathbf{b}, \qquad \mathbf{C} = \mathbf{I} + \mathbf{G}_0^*\mathbf{G}_0,$$

for a given finite-length vector $\mathbf{b}$. Since $\mathbf{C}$ is Hermitian positive definite, its action on an arbitrary vector $\mathbf{u}$ can be evaluated without constructing the corresponding lifted matrix. Specifically, an application of $\mathbf{G}_0$ gives

$$\mathbf{y}_0 = \mathbf{G}_0\mathbf{u},$$

while the experimental adjoint procedure of Section 5.3.2 provides the action $\mathbf{G}_0^*\mathbf{y}_0$. Hence,

$$\mathbf{C}\mathbf{u} = \mathbf{u} + \mathbf{G}_0^*\mathbf{G}_0\mathbf{u}. \tag{5.4.15}$$

Starting from $\mathbf{q}_1 = \mathbf{b}/\|\mathbf{b}\|_2$, the Lanczos iteration introduced in Section 2.2.2 constructs an orthonormal basis

$$\mathbf{Q}_n = \begin{bmatrix}\mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_n\end{bmatrix}$$

for the Krylov subspace $\mathcal{K}_n(\mathbf{C}, \mathbf{b})$, together with the tridiagonal projection $\mathbf{T}_n = \mathbf{Q}_n^*\mathbf{C}\mathbf{Q}_n$.

To motivate the inverse-square-root computation, consider first the linear system $\mathbf{C}\mathbf{x} = \mathbf{b}$. Seeking an approximation of the form $\mathbf{x}_n = \mathbf{Q}_n\mathbf{y}_n$ and projecting the system onto the Krylov subspace gives

$$\mathbf{Q}_n^*\mathbf{C}\mathbf{Q}_n\mathbf{y}_n = \mathbf{Q}_n^*\mathbf{b}.$$

Since $\mathbf{q}_1 = \mathbf{b}/\|\mathbf{b}\|_2$ and $\mathbf{e}_1 = [1, 0, \dots, 0]^T$ denotes the first canonical basis vector in the reduced Krylov space, we have $\mathbf{Q}_n \mathbf{e}_1 = \mathbf{q}_1$, or equivalently, $\mathbf{Q}_n^* \mathbf{q}_1 = \mathbf{e}_1$.
Therefore,

$$\mathbf{Q}_n^* \mathbf{b} = \|\mathbf{b}\|_2 \mathbf{e}_1,$$

and it follows that

$$\mathbf{x}_n = \|\mathbf{b}\|_2 \mathbf{Q}_n \mathbf{T}_n^{-1} \mathbf{e}_1.$$

The same Lanczos basis can be used to approximate the action of the inverse square root. Following [54], the inverse square root of a Hermitian positive-definite operator admits the integral representation

$$\mathbf{C}^{-1/2} = \frac{2}{\pi} \int_0^\infty \left(t^2 \mathbf{I} + \mathbf{C}\right)^{-1} \mathrm{d}t. \tag{5.4.16}$$

A key property of the Lanczos process is its invariance under scalar shifts. Thus, adding $t^2 \mathbf{I}$ to $\mathbf{C}$ leaves the Krylov basis unchanged and shifts only the projected tridiagonal matrix. Consequently,

$$\left(t^2 \mathbf{I} + \mathbf{C}\right)^{-1} \mathbf{b} \approx \|\mathbf{b}\|_2 \mathbf{Q}_n \left(t^2 \mathbf{I} + \mathbf{T}_n\right)^{-1} \mathbf{e}_1. \tag{5.4.17}$$

Substituting 5.4.17 into 5.4.16 gives

$$\mathbf{C}^{-1/2} \mathbf{b} \approx \|\mathbf{b}\|_2 \mathbf{Q}_n \mathbf{T}_n^{-1/2} \mathbf{e}_1.$$

Since $\mathbf{T}_n$ is a small Hermitian matrix, its inverse square root can be computed explicitly from the eigendecomposition $\mathbf{T}_n = \mathbf{V}_n \mathbf{\Lambda}_n \mathbf{V}_n^*$, which yields

$$\mathbf{T}_n^{-1/2} = \mathbf{V}_n \mathbf{\Lambda}_n^{-1/2} \mathbf{V}_n^*.$$

Thus, the large lifted matrix associated with $\mathbf{C}$ is never explicitly constructed or factorized. Instead, the Lanczos iteration requires only successive evaluations of (5.4.15), while the inverse-square-root operation is carried out on the considerably smaller tridiagonal matrix $\mathbf{T}_n$.

Evaluation of $\mathbf{A}^{-1/2}$ and $\mathbf{B}$

The treatment of the second normalization operator, $\mathbf{A}^{-1/2}$, is entirely analogous to that of $\mathbf{C}^{-1/2}$. The main difference lies in the order in which the forward and adjoint plant operators are applied. Since $\mathbf{A} = \mathbf{I} + \mathbf{G}\mathbf{G}^*$, its action on an output-space vector $\mathbf{y}$ is evaluated as

$$\mathbf{A}\mathbf{y} = \mathbf{y} + \mathbf{G}\left(\mathbf{G}^* \mathbf{y}\right).$$

Hence, the adjoint action $\mathbf{G}^* \mathbf{y}$ is first obtained using the experimental adjoint construction of Section 5.3.2, after which the resulting signal is applied as an input to $\mathbf{G}$.

Applying the same Lanczos construction used for $\mathbf{C}^{-1/2}$ therefore gives

$$\mathbf{A}^{-1/2}\mathbf{b} \approx \|\mathbf{b}\|_2 \widetilde{\mathbf{Q}}_n \widetilde{\mathbf{T}}_n^{-1/2} \mathbf{e}_1,$$

where $\widetilde{\mathbf{Q}}_n$ and $\widetilde{\mathbf{T}}_n$ denote the Krylov basis and projected matrix associated with $\mathbf{A}$, respectively, with $\widetilde{\mathbf{T}}_n = \widetilde{\mathbf{Q}}_n^* \mathbf{A} \widetilde{\mathbf{Q}}_n$. Thus, both normalization factors in (5.4.14) are evaluated using the same Lanczos matrix-function approximation, differing only in the operator action used to generate the corresponding Krylov basis.

In contrast, the action of the difference operator $\mathbf{B} = \mathbf{G}_0 - \mathbf{G}$ does not require a Krylov approximation. Given a common input $\mathbf{u}$, both systems are excited with the same signal and their responses are subtracted, giving

$$\mathbf{B}\mathbf{u} = \mathbf{G}_0\mathbf{u} - \mathbf{G}\mathbf{u}. \tag{5.4.18}$$

Likewise, its adjoint is obtained as

$$\mathbf{B}^*\mathbf{y} = \mathbf{G}_0^*\mathbf{y} - \mathbf{G}^*\mathbf{y},$$

where each adjoint action is evaluated using the experimental construction of Section 5.3.2.

Notice that the evaluation of (5.4.18) naturally provides more information than the difference signal alone. In particular, the common input $\mathbf{u}$ and the individual responses $\mathbf{G}_0\mathbf{u}$ and $\mathbf{G}\mathbf{u}$ are available at every forward evaluation of $\mathbf{B}$. This information is reused below to verify the admissibility condition without requiring an additional set of experiments.

Data-driven admissibility verification

The computation of the induced norm in (5.4.13) corresponds to the Vinnicombe metric only when $(\mathbf{G}_0, \mathbf{G}) \in \mathcal{C}$. We therefore verify the admissibility condition concurrently with the iterative norm estimation.

At the ith power iteration step, let $\mathbf{u}^{(i)}$ denote the signal applied to both plants when evaluating (5.4.18), and let

$$\mathbf{y}_0^{(i)} = \mathbf{G}_0\mathbf{u}^{(i)}, \qquad \mathbf{y}^{(i)} = \mathbf{G}\mathbf{u}^{(i)}$$

denote the corresponding nominal and true-system outputs. Rather than performing separate frequency-response experiments, we retain these signals during the first N_{acc} iterations and use them to estimate the frequency responses required by the index condition.

Let $\mathbf{U}_i[k]$, $\mathbf{Y}_{0,i}[k]$, and $\mathbf{Y}_i[k]$ denote their DFT coefficients at frequency ω_k. For each

frequency bin, we accumulate

$$\begin{aligned}\mathbf{S}_{uu}[k] &= \sum_{i=1}^{N_{\text{acc}}} \mathbf{U}_i[k]\mathbf{U}_i[k]^*,\\ \mathbf{S}_{y_0u}[k] &= \sum_{i=1}^{N_{\text{acc}}} \mathbf{Y}_{0,i}[k]\mathbf{U}_i[k]^*,\\ \mathbf{S}_{yu}[k] &= \sum_{i=1}^{N_{\text{acc}}} \mathbf{Y}_i[k]\mathbf{U}_i[k]^*.\end{aligned} \tag{5.4.19}$$

The corresponding frequency-response matrices are estimated as

$$\widehat{\mathbf{G}}_0[k] = \mathbf{S}_{y_0u}[k]\mathbf{S}_{uu}[k]^{-1}, \qquad \widehat{\mathbf{G}}[k] = \mathbf{S}_{yu}[k]\mathbf{S}_{uu}[k]^{-1}. \tag{5.4.20}$$

When $\mathbf{S}_{uu}[k]$ is poorly conditioned, a small diagonal regularization is introduced following the diagonal-loading approach in [80, Eq. (6)] by replacing it with

$$\tilde{\mathbf{S}}_{uu}[k] = \mathbf{S}_{uu}[k] + \mu_k \mathbf{I}_m, \qquad \mu_k = \varepsilon_{\text{reg}} \frac{\operatorname{tr}(\mathbf{S}_{uu}[k])}{m},$$

where $\varepsilon_{\text{reg}} > 0$ is a small regularization parameter, and the resulting matrix $\tilde{\mathbf{S}}_{uu}[k]$ is used in (5.4.20).

For the stable MIMO systems considered in this chapter, the pole-count term in the general index condition of Section 2.1.3 vanishes. Hence, define

$$\mathbf{H}[k] := \mathbf{I} + \widehat{\mathbf{G}}[k]^*\widehat{\mathbf{G}}_0[k]. \tag{5.4.21}$$

The admissibility condition requires $\mathbf{H}[k]$ to remain nonsingular over the frequency grid and the winding number of its determinant to be zero. Numerically, nonsingularity is verified through

$$\min_k \sigma_{\min}(\mathbf{H}[k]) > \varepsilon_{\text{sv}}. \tag{5.4.22}$$

where $\varepsilon_{\text{sv}} > 0$ is a small numerical tolerance.

To compute the winding number, let

$$d[k] := \det(\mathbf{H}[k])\,.$$

After closing the sampled contour, the winding number is estimated from the accumulated phase increments as

$$\operatorname{wno}(\det \mathbf{H}) = \operatorname{round}\left(\frac{1}{2\pi}\sum_{k=0}^{N-1} \arg\left(\frac{d[k+1]}{d[k]}\right)\right), \qquad d[N] := d[0]. \tag{5.4.23}$$

The pair $(\mathbf{G}_0, \mathbf{G})$ is regarded as admissible when the nonsingularity test is satisfied and (5.4.23) is zero. Otherwise, the Vinnicombe metric is set to one according to Definition 2.1. The complete data-driven admissibility test is summarized in Algorithm 5.3. Since this procedure only requires the input and the individual responses of $\mathbf{G}_0$ and $\mathbf{G}$, it is independent of the particular iterative method used to estimate the induced norm and will therefore be reused in both proposed approaches.

Algorithm 5.3 Data-driven MIMO admissibility test

Require: Accumulated spectral matrices $\mathbf{S}_{uu}$, $\mathbf{S}_{y_0u}$, and $\mathbf{S}_{yu}$; tolerance $\varepsilon_{\text{sv}} > 0$
1: Estimate $\widehat{\mathbf{G}}_0[k]$ and $\widehat{\mathbf{G}}[k]$ for $k = 0, \dots, N-1$ via (5.4.20)
2: Construct $\mathbf{H}[k]$ via (5.4.21)
3: Verify the nonsingularity condition via (5.4.22)
4: Compute $\text{wno}(\det \mathbf{H})$ via (5.4.23)
5: Set inC $\leftarrow$ true if (5.4.22) is satisfied and $\text{wno}(\det \mathbf{H}) = 0$; otherwise set inC $\leftarrow$ false
6: **return** inC

Iterative MIMO ν-gap estimation

With all required operator actions available, the dominant eigenvalue in (5.4.13) can now be estimated using the power iteration method introduced in Section 2.2.1. Given a normalized vector $\mathbf{v}^{(i)}$, the action of $\mathbf{M}^*\mathbf{M}$ is evaluated according to the factorization in (5.4.14). Starting from the rightmost factor, we successively apply $\mathbf{C}^{-1/2}$, $\mathbf{B}$, $\mathbf{A}^{-1}$, $\mathbf{B}^*$, and finally $\mathbf{C}^{-1/2}$. As discussed above, the action of $\mathbf{A}^{-1}$ is obtained through two consecutive applications of $\mathbf{A}^{-1/2}$. Hence,

$$\begin{aligned}
\mathbf{t}^{(i)} &= \mathbf{C}^{-1/2}\mathbf{v}^{(i)}, \\
\mathbf{r}^{(i)} &= \mathbf{B}\mathbf{t}^{(i)}, \\
\mathbf{z}^{(i)} &= \mathbf{A}^{-1}\mathbf{r}^{(i)}, \\
\mathbf{s}^{(i)} &= \mathbf{B}^*\mathbf{z}^{(i)}, \\
\mathbf{w}^{(i)} &= \mathbf{C}^{-1/2}\mathbf{s}^{(i)} = \mathbf{M}^*\mathbf{M}\mathbf{v}^{(i)}.
\end{aligned} \tag{5.4.24}$$

The resulting vector $\mathbf{w}^{(i)}$ is normalized to obtain the next power iteration vector,

$$\mathbf{v}^{(i+1)} = \frac{\mathbf{w}^{(i)}}{\|\mathbf{w}^{(i)}\|_2}. \tag{5.4.25}$$

A further application of $\mathbf{M}^*\mathbf{M}$ to $\mathbf{v}^{(i+1)}$ is then used to evaluate the corresponding Rayleigh quotient,

$$\hat{\lambda}^{(i+1)} = \left(\mathbf{v}^{(i+1)}\right)^* \mathbf{M}^*\mathbf{M}\mathbf{v}^{(i+1)}, \tag{5.4.26}$$

which approaches the dominant eigenvalue of $\mathbf{M}^*\mathbf{M}$ under the usual power iteration conditions. Convergence is assessed by measuring how closely the current vector satisfies the corresponding eigenvalue equation. Specifically, the eigenpair residual is defined as

$$\mathbf{r}^{(i+1)} = \mathbf{M}^*\mathbf{M}\mathbf{v}^{(i+1)} - \hat{\lambda}^{(i+1)}\mathbf{v}^{(i+1)},$$

and the iteration is terminated when

$$\left\|\mathbf{r}^{(i+1)}\right\|_2 < \varepsilon_{\text{PI}}, \tag{5.4.27}$$

where $\varepsilon_{\text{PI}} > 0$ denotes the prescribed power iteration tolerance.

During the evaluation of $\mathbf{B}\mathbf{t}^{(i)}$, the common input $\mathbf{t}^{(i)}$ and the individual responses $\mathbf{G}_0\mathbf{t}^{(i)}$ and $\mathbf{G}\mathbf{t}^{(i)}$ are retained during the first N_{acc} iterations. Once these data have

been accumulated, Algorithm 5.3 is invoked to verify that $(\mathbf{G}_0, \mathbf{G}) \in \mathcal{C}$. If the condition fails, the iteration is terminated and $\hat{\delta}_\nu$ is set to one. Otherwise, the power iteration continues until convergence, after which

$$\hat{\delta}_\nu = \sqrt{\hat{\lambda}_{\max}}$$

provides the finite-horizon estimate of the Vinnicombe metric.

The complete approach is summarized in Algorithm 5.4.

Algorithm 5.4 Data-driven MIMO ν-gap estimation via power iteration

Require: A normalized initial vector $\mathbf{v}^{(1)}$; maximum number of iterations M_{PI}; tolerance ε_{PI}; number of accumulation experiments N_{acc}
Initialize the spectral accumulators $\mathbf{S}_{uu}$, $\mathbf{S}_{y_0u}$, and $\mathbf{S}_{yu}$
for $i = 1, 2, \dots, M_{\text{PI}}$ **do**
  Evaluate $\mathbf{w}^{(i)} = \mathbf{M}^*\mathbf{M}\mathbf{v}^{(i)}$ via (5.4.24)
  if $i \leq N_{\text{acc}}$ **then**
    Compute the DFTs of $\mathbf{t}^{(i)}$, $\mathbf{y}_0^{(i)}$, and $\mathbf{y}^{(i)}$ obtained during the evaluation of $\mathbf{B}\mathbf{t}^{(i)}$
    Update the spectral accumulators via (5.4.19)
  end if
  if $i = N_{\text{acc}}$ **then**
    Run Algorithm 5.3
    if inC is false **then**
      return $\hat{\delta}_\nu \leftarrow 1$
    end if
  end if
  Compute $\widehat{\lambda}^{(i)}$ via (5.4.26)
  if (5.4.27) is satisfied **then**
    return $\hat{\delta}_\nu \leftarrow \sqrt{\widehat{\lambda}^{(i)}}$
  end if
  Compute $\mathbf{v}^{(i+1)}$ via (5.4.25)
end for
return $\hat{\delta}_\nu \leftarrow \sqrt{\widehat{\lambda}^{(M_{\text{PI}})}}$

5.4.2 Approach 2 (GKL): Golub–Kahan–Lanczos Bidiagonalization

The first approach estimates the dominant eigenvalue of $\mathbf{M}^*\mathbf{M}$ by repeatedly applying the complete composite operator. An alternative is to avoid applying power iteration directly in the finite-horizon signal space and instead construct progressively expanding low-dimensional projections of $\mathbf{M}$ using the Golub–Kahan–Lanczos (GKL) bidiagonalization introduced in Section 5.3.1.

In particular, GKL alternates applications of $\mathbf{M}$ and $\mathbf{M}^*$ to construct orthonormal bases for their input and output spaces. After k iterations, the action of $\mathbf{M}$ is represented by the considerably smaller bidiagonal matrix $\mathbf{B}_k$. The dominant singular value of $\mathbf{M}$ can then be estimated from this reduced representation, while the actions of $\mathbf{M}$ and $\mathbf{M}^*$ themselves remain entirely matrix-free.

Matrix-free actions of $\mathbf{M}$ and $\mathbf{M}^*$

The evaluation of $\mathbf{M}$ and $\mathbf{M}^*$ reuses the operator actions developed for Approach 1. From (5.2.2), the action of $\mathbf{M}$ on a vector $\mathbf{v}$ is evaluated sequentially as

$$\begin{aligned} \mathbf{t} &= \mathbf{C}^{-1/2}\mathbf{v}, \\ \mathbf{r} &= \mathbf{B}\mathbf{t}, \\ \mathbf{y} &= \mathbf{A}^{-1/2}\mathbf{r} = \mathbf{M}\mathbf{v}. \end{aligned} \tag{5.4.28}$$

Here, the inverse-square-root actions are computed using the Lanczos procedure described in Section 5.4.1, while $\mathbf{B}\mathbf{t}$ is obtained directly from the difference between the responses of $\mathbf{G}_0$ and $\mathbf{G}$ to the common input $\mathbf{t}$.

Since $\mathbf{A}$ and $\mathbf{C}$ are Hermitian positive definite, the adjoint of $\mathbf{M}$ is

$$\mathbf{M}^* = \mathbf{C}^{-1/2}\mathbf{B}^*\mathbf{A}^{-1/2}.$$

Consequently, its action on an output-space vector $\mathbf{u}$ is evaluated as

$$\begin{aligned} \mathbf{q} &= \mathbf{A}^{-1/2}\mathbf{u}, \\ \mathbf{s} &= \mathbf{B}^*\mathbf{q}, \\ \mathbf{v} &= \mathbf{C}^{-1/2}\mathbf{s} = \mathbf{M}^*\mathbf{u}. \end{aligned} \tag{5.4.29}$$

The adjoint action $\mathbf{B}^*\mathbf{q}$ is obtained using the experimental adjoint construction introduced in Section 5.3.2.

As in Approach 1, the forward evaluation of $\mathbf{B}\mathbf{t}$ in (5.4.28) provides not only the difference signal but also the common input $\mathbf{t}$ and the individual responses $\mathbf{G}_0\mathbf{t}$ and $\mathbf{G}\mathbf{t}$. During the first N_{acc} GKL iterations, these signals are retained and their frequency-domain information is accumulated according to (5.4.19). Once N_{acc} experiments have been collected, the admissibility condition is verified using Algorithm 5.3. Thus, the same data-driven index test is shared by both proposed approaches and requires no additional plant experiments.

Reduced dominant-singular-value estimation

When the GKL procedure of Section 5.3.1 is applied to $\mathbf{M}$, the coefficients α_j and β_j are generated iteratively, defining after k iterations the upper-bidiagonal matrix

$$\mathbf{B}_k = \begin{bmatrix} \alpha_1 & \beta_1 & & \\ & \alpha_2 & \beta_2 & \\ & & \ddots & \ddots \\ & & & \alpha_k \end{bmatrix}. \tag{5.4.30}$$

Rather than computing the singular values of $\mathbf{B}_k$ explicitly at every GKL iteration, we exploit the identity $\bar{\sigma}(\mathbf{B}_k)^2 = \lambda_{\max}\left(\mathbf{B}_k^*\mathbf{B}_k\right)$, and define the reduced Hermitian matrix

$$\mathbf{T}_k := \mathbf{B}_k^*\mathbf{B}_k. \tag{5.4.31}$$

Since $\mathbf{B}_k$ is upper bidiagonal, $\mathbf{T}_k$ is Hermitian tridiagonal. Its diagonal and off-diagonal entries can be obtained directly from the GKL coefficients without explicitly forming the product $\mathbf{B}_k^*\mathbf{B}_k$. In particular,

$$\begin{aligned} [\mathbf{T}_k]_{1,1} &= \alpha_1^2, \\ [\mathbf{T}_k]_{j,j} &= \alpha_j^2 + \beta_{j-1}^2, && j = 2, \dots, k, \\ [\mathbf{T}_k]_{j,j+1} &= [\mathbf{T}_k]_{j+1,j} = \alpha_j \beta_j, && j = 1, \dots, k-1. \end{aligned} \tag{5.4.32}$$

Thus, the reduced eigenvalue problem can be updated at each GKL iteration using only the newly generated coefficients α_k and β_k.

Although $\mathbf{T}_k$ is sufficiently small to allow direct eigendecomposition, we estimate its dominant eigenvalue iteratively as the GKL subspace expands, using a power iteration update in the reduced k-dimensional space. To implement this reduced-space iteration, at $k = 1$ the initial eigenvector estimate is simply $\mathbf{z}_1 = [1]$. For $k > 1$, the estimate obtained at the previous GKL iteration, $\mathbf{z}_{k-1} \in \mathbb{C}^{k-1}$, must be embedded in the enlarged k-dimensional reduced space. We therefore retain its previous components and initialize the newly introduced coordinate, associated with the additional GKL basis direction, to zero:

$$\tilde{\mathbf{z}}_k = \frac{\begin{bmatrix} \mathbf{z}_{k-1} \\ 0 \end{bmatrix}}{\left\| \begin{bmatrix} \mathbf{z}_{k-1} \\ 0 \end{bmatrix} \right\|_2}. \tag{5.4.33}$$

Rather than running the inner power iteration to convergence for every $\mathbf{T}_k$, we perform a single power iteration update per outer GKL step, using (5.4.33) as a warm start:

$$\mathbf{z}_k = \frac{\mathbf{T}_k \tilde{\mathbf{z}}_k}{\|\mathbf{T}_k \tilde{\mathbf{z}}_k\|_2}. \tag{5.4.34}$$

The corresponding Rayleigh quotient is

$$\hat{\lambda}_k = \mathbf{z}_k^* \mathbf{T}_k \mathbf{z}_k, \qquad \hat{\sigma}_k = \sqrt{\hat{\lambda}_k}. \tag{5.4.35}$$

This differs fundamentally from Approach 1 introduced in Section 5.4.1. There, power iteration is performed directly on the large finite-horizon operator $\mathbf{M}^*\mathbf{M}$, so each update requires a complete matrix-free evaluation of that composite operator. Here, the large-scale operator actions are used to expand the GKL bases, while the power iteration update is performed only on the small tridiagonal matrix $\mathbf{T}_k$. Moreover, the dominant eigenvector estimate from the previous reduced problem is reused as a warm start when the dimension is increased from $k-1$ to k. Convergence of the outer GKL iteration is assessed from the relative variation of the reduced singular-value estimate,

$$\frac{|\hat{\sigma}_k - \hat{\sigma}_{k-1}|}{\max(\hat{\sigma}_k, \varepsilon_0)} < \varepsilon_{\text{GKL}}, \tag{5.4.36}$$

where $\varepsilon_{\mathrm{GKL}} > 0$ is the prescribed GKL tolerance and $\varepsilon_0 > 0$ is a small constant used to avoid division by zero.

Upon convergence, the approximate dominant right singular vector can be mapped back to the original input space through

$$\hat{\mathbf{v}} = \mathbf{V}_k \mathbf{z}_k, \tag{5.4.37}$$

where $\mathbf{V}_k$ contains the right GKL basis vectors. A final consistency evaluation of the dominant singular value is then obtained as

$$\hat{\sigma}_{\max} = \|\mathbf{M}\hat{\mathbf{v}}\|_2. \tag{5.4.38}$$

Provided that the admissibility condition is satisfied, this value gives the finite-horizon estimate of the Vinnicombe metric.

Iterative MIMO ν-gap estimation via GKL

The complete procedure alternates the matrix-free actions of $\mathbf{M}$ and $\mathbf{M}^*$ according to the GKL bidiagonalization of Section 5.3.1. At each outer iteration, the newly obtained coefficients α_k and β_k are used to update the reduced tridiagonal matrix $\mathbf{T}_k$ through (5.4.32). A single power iteration update is then performed in this reduced space using (5.4.34), and the current singular-value estimate is computed from (5.4.35).

As in Approach 1, the first N_{acc} forward evaluations of $\mathbf{M}$ provide the common input and individual plant responses required to update the spectral accumulators in (5.4.19). Algorithm 5.3 is therefore invoked after N_{acc} iterations to verify the admissibility condition. If the condition fails, the procedure terminates and the Vinnicombe metric is set to one. Otherwise, the GKL iteration continues until the singular-value estimate converges.

The complete approach is summarized in Algorithm 5.5.

5.5 Experiments

This section presents simulation studies to assess Algorithms 5.3, 5.4, and 5.5. Using a textbook MIMO example, we evaluate the two proposed estimators of the Vinnicombe metric and compare their results against the `nugap` output returned by MATLAB's `gapmetric` routine.

5.5.1 Experimental Setting

We consider the continuous-time nominal MIMO plant

$$\mathbf{G}_0(s) = \begin{bmatrix} \dfrac{2}{1.4427s+1} & 1 \\ 1 & \dfrac{6.6667}{2.8037s+1} \end{bmatrix}, \tag{5.5.39}$$

Algorithm 5.5 Data-driven MIMO ν-gap estimation via GKL

Require: A normalized initial vector $\mathbf{v}_1$; maximum number of GKL iterations M_{GKL}; tolerance ε_{GKL}; number of accumulation experiments N_{acc}

Set $\beta_0 \leftarrow 0$ and initialize the spectral accumulators
for $k = 1, 2, \dots, M_{\text{GKL}}$ **do**
    Perform the forward GKL step using $\mathbf{M}\mathbf{v}_k$ via (5.4.28)
    Compute α_k and normalize $\mathbf{u}_k$ according to Algorithm 5.1
    if $k \leq N_{\text{acc}}$ **then**
        Update the spectral accumulators via (5.4.19)
    end if
    if $k = N_{\text{acc}}$ **then**
        Run Algorithm 5.3
        if inC is false **then**
            return $\hat{\delta}_\nu \leftarrow 1$
        end if
    end if
    Perform the adjoint GKL step using $\mathbf{M}^*\mathbf{u}_k$ via (5.4.29)
    Compute β_k and normalize $\mathbf{v}_{k+1}$ according to Algorithm 5.1
    Construct $\mathbf{T}_k$ via (5.4.32)
    Update $\mathbf{z}_k$ via (5.4.34)
    Compute $\widehat{\sigma}_k$ via (5.4.35)
    if (5.4.36) is satisfied **then**
        Compute $\widehat{\mathbf{v}}$ via (5.4.37)
        return $\hat{\delta}_\nu \leftarrow \|\mathbf{M}\widehat{\mathbf{v}}\|_2$
    end if
end for
Compute $\widehat{\mathbf{v}}$ via (5.4.37)
return $\hat{\delta}_\nu \leftarrow \|\mathbf{M}\widehat{\mathbf{v}}\|_2$

together with two alternative real plants,

$$\mathbf{G}_1(s) = \begin{bmatrix} \dfrac{2}{1.4427s+1} & \dfrac{1}{} \\ 1 & \dfrac{4.92}{1.9134s+1} \end{bmatrix}, \qquad \mathbf{G}_2(s) = \begin{bmatrix} \dfrac{2}{1.4427s+1} & \dfrac{1.6667}{1.0914s+1} \\ \dfrac{2}{1.4427s+1} & \dfrac{8}{3.4761s+1} \end{bmatrix}. \tag{5.5.40}$$

The first comparison, $(\mathbf{G}_0, \mathbf{G}_1)$, considers a relatively small deviation from the nominal dynamics, whereas the second, $(\mathbf{G}_0, \mathbf{G}_2)$, introduces more substantial differences across the multivariable input–output channels.

All continuous-time plants are discretized using a zero-order hold with sampling period $T_s = 1$. The data-driven computations are performed over finite-length input–output trajectories of $N = 500$ samples. In both approaches, the actions of the inverse-square-root normalization operators $\mathbf{A}^{-1/2}$ and $\mathbf{C}^{-1/2}$ are approximated using up to 40 Lanczos iterations, corresponding to Krylov subspaces of maximum dimension 40.

For Approach 1, we set $M_{\mathrm{PI}} = 50$ and $\varepsilon_{\mathrm{PI}} = 10^{-10}$, whereas for Approach 2, we set $M_{\mathrm{GKL}} = 100$ and $\varepsilon_{\mathrm{GKL}} = 10^{-10}$. The same initialization seed is used for both cases. For each plant pair, the estimates obtained with Algorithms 5.4 and 5.5 are compared with the corresponding model-based MATLAB implementation of the Vinnicombe metric.

5.5.2 Experimental Results

Figure 5.5.1 shows the estimates of $\delta_\nu(\mathbf{G}_0, \mathbf{G}_1)$ obtained with both proposed approaches. The nominal and true plants are relatively close, as they differ only in the $(2,2)$ entry, where $\mathbf{G}_1$ has a lower static gain and a shorter time constant. Both approaches converge closely to the MATLAB reference value, $\delta_\nu(\mathbf{G}_0, \mathbf{G}_1) = 0.0702$. This relatively small value also admits a direct robustness interpretation: from Proposition 2.1, any controller $\mathbf{C}$ that stabilizes $\mathbf{G}_0$ with generalized stability margin $b_{\mathbf{G}_0,\mathbf{C}} > 0.0702$ is guaranteed to stabilize $\mathbf{G}_1$ as well. Moreover, the data-driven admissibility test is satisfied, and hence $(\mathbf{G}_0, \mathbf{G}_1) \in \mathcal{C}$. Consequently, Algorithm 5.3 does not terminate the estimation by assigning $\delta_\nu = 1$, and both proposed approaches proceed with the corresponding norm estimation. Figure 5.5.2 considers the larger mismatch between $\mathbf{G}_0$ and $\mathbf{G}_2$, showing that both proposed approaches converge closely to the MATLAB reference value $\delta_\nu(\mathbf{G}_0, \mathbf{G}_2) = 0.8769$. Since the Vinnicombe metric lies between zero and one, this value indicates a substantial control-relevant discrepancy between the nominal and true plants. According to Proposition 2.1, a sufficient guarantee that the same controller stabilizes $\mathbf{G}_2$ would require a nominal stability margin $b_{\mathbf{G}_0,C} > 0.8769$, which represents a considerably more demanding robustness requirement than in the previous case. As a practical rule of thumb, Åström and Murray report that Vinnicombe recommended values of the ν-gap below $1/3$ for reasonable robustness [4, Example 12.4]. The value obtained here is well above this heuristic threshold, consistently indicating that $\mathbf{G}_0$ and $\mathbf{G}_2$ should not be regarded as close from a robust-control perspective. Nevertheless, the admissibility condition is also satisfied in this case, so $(\mathbf{G}_0, \mathbf{G}_2) \in \mathcal{C}$.

Regarding convergence, the numerical results suggest that, in terms of iteration

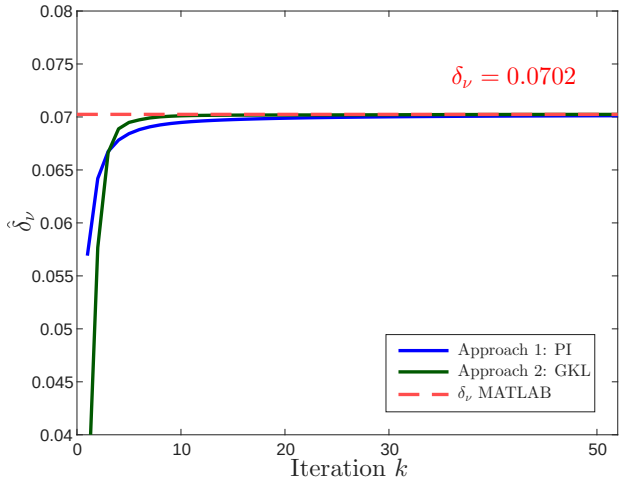


Figure 5.5.1: Estimation of $\delta_\nu(\mathbf{G}_0, \mathbf{G}_1)$. Approach #1 estimate (blue, solid), approach #2 estimate (dark-green, solid), and MATLAB reference δ_ν (red, dashed).

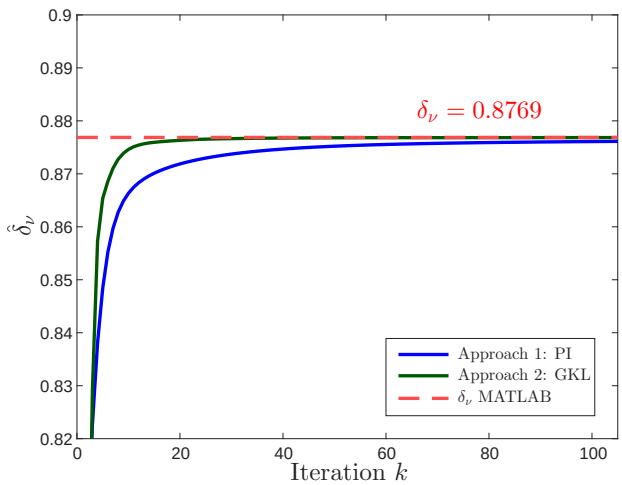


Figure 5.5.2: Estimation of $\delta_\nu(\mathbf{G}_0, \mathbf{G}_2)$. Approach #1 estimate (blue, solid), approach #2 estimate (dark-green, solid), and MATLAB reference δ_ν (red, dashed).

count, the GKL-based method approaches the reference value more rapidly than the power iteration approach, particularly for the larger mismatch $(\mathbf{G}_0, \mathbf{G}_2)$, whereas the difference is less pronounced for $(\mathbf{G}_0, \mathbf{G}_1)$. These observations are specific to the numerical examples considered here, and a more systematic comparison of the convergence properties of both approaches is currently under investigation as part of the ongoing development of this work.

5.6 Summary and Future Work

This chapter has extended the data-driven estimation of the Vinnicombe metric developed in Chapter 4 to the MIMO setting. By reformulating the problem from a finite-horizon operator perspective, two matrix-free approaches are proposed. The first estimates the dominant eigenvalue of $\mathbf{M}^*\mathbf{M}$ through power iteration, while using Lanczos approximations for the inverse-square-root normalization operators. The second applies Golub–Kahan–Lanczos bidiagonalization directly to $\mathbf{M}$ and $\mathbf{M}^*$, reducing the dominant singular-value estimation to a progressively growing low-dimensional problem. Both approaches are combined with the same data-driven admissibility test and require only input–output experiments rather than explicit lifted system matrices. The numerical examples of Section 5.5.2 show close agreement with MATLAB's `gapmetric` implementation for plant pairs with both small and large ν-gap values.

Although the numerical examples considered here are limited to 2×2 systems, the proposed MIMO formulation naturally extends to higher-dimensional settings, opening the possibility of studying models that better represent practical processes. In particular, a natural extension of the SISO gas-turbine example studied in Chapter 4 is to formulate a multivariable representation that incorporates additional process variables, such as pressure, flow, and temperature. Such a formulation would provide a more realistic description of the gas-turbine dynamics and would allow the proposed estimators to be evaluated beyond low-dimensional benchmark systems.

Future work will therefore focus on validating the proposed approaches on larger and industrially motivated MIMO systems, together with a more detailed statistical study of their estimation accuracy and convergence behavior. Of particular interest is the comparison of the number of iterations and experiments required by the two approaches. Although the operator evaluations are entirely matrix-free, repeated forward and adjoint experiments are currently required as the iterative procedures converge. Reducing this experimental burden, for example through more efficient Krylov updates, improved stopping criteria, or reuse of previously generated subspace information, constitutes an important direction for making the data-driven computation of the MIMO ν-gap more efficient in practical applications.

Part II

Counterfactual Explanations and Safe Perturbation Search

6 Counterfactual Explanations: Adaptive Search in Black-Box Models

Chapters 3–5 investigated how fundamental robustness properties of an unknown dynamical system can be inferred directly from input–output data, without requiring an explicit parametric description of the underlying plant. The structured singular value characterizes the system's vulnerability to structured modeling uncertainty, whereas the ν-gap metric quantifies the control-relevant discrepancy between a nominal model and the true plant. Taken together, these results address a central question in data-driven robustness analysis: *how much uncertainty or modeling mismatch can be tolerated before the desired closed-loop properties are lost?*

The second part of this thesis shifts the perspective from the analysis of unknown systems to purposeful interaction with them. Rather than asking only which perturbations may degrade stability or performance, we ask which carefully selected modifications can produce a desired change in behavior. The emphasis is therefore no longer on constructing a complete model of the system, but on extracting the information required to make an informed decision through a limited number of strategically designed queries. In this sense, the problem becomes one of intelligent input design: each new evaluation is selected not merely to collect additional data, but to maximize the task-relevant information gained.

This chapter introduces *Adaptive sampling Counterfactual Explanations* (ACE), a novel framework for explaining the behavior of black-box models through sequential and informative queries. Rooted in the principles of explainable artificial intelligence, ACE searches for small and meaningful perturbations capable of changing a model's outcome while minimizing the number of required evaluations. This viewpoint addresses the broader problem of counterfactual explainability for black-box classification models in machine learning and also provides the conceptual foundation for the controller-tuning application presented in the following chapter.

6.1 Introduction

The increasing complexity and widespread use of Artificial Intelligence have intensified the need for models whose decisions can be understood and justified. This need is reflected not only in regulations and guidelines promoting transparency and explainability [81, 82], but also in the growing development of Explainable Artificial Intelligence methods [46, 56, 83].

Model-agnostic XAI methods seek to explain black-box predictions without requiring access to the internal structure of the underlying model. Methods such as LIME [33] and SHAP [34], for instance, identify the features that most influence a prediction. However, they do not directly answer a fundamentally different and more actionable question: *what is the smallest change required to obtain a desired alternative outcome?* Counterfactual Explanations (CFEs) [16, 84] address precisely this problem by searching for small and feasible input perturbations capable of changing a model's prediction.

In the binary-classification setting considered here, the black-box model is accessed only through queries and their corresponding binary outcomes h. This setting is particularly relevant when the model is proprietary, inaccessible, or costly to evaluate. Existing CFE methods rely on geometric search [85], evolutionary optimization [39, 86, 87], constraint-guided methods [88, 40, 89, 90], or surrogate models [41, 91]. Nevertheless, many of these approaches devote limited attention to the number of queries required to identify a counterfactual, and often assume access to sufficiently large datasets for their calibration.

To address these limitations, this chapter introduces *Adaptive Sampling for Counterfactual Explanations* (ACE), a query-efficient method for classification problems. ACE constructs a Gaussian-process surrogate and uses Bayesian optimization to sequentially prioritize informative points, thereby reducing the number of black-box evaluations required to identify a counterfactual explanation. A Gaussian prior limits the need for extensive hyperparameter tuning, while a penalty formulation is employed to support global convergence.

In summary, in this chapter,

1. We propose ACE, a query-efficient method for generating counterfactual explanations using only limited access to the black-box model, while promoting plausible, feasible, and actionable solutions.
2. We conduct a comprehensive quantitative and qualitative evaluation of ACE on multiple real-world datasets and a control-oriented case study, benchmarking its performance against state-of-the-art CFE methods.

The remainder of this chapter is organized as follows: Section 6.2 formulates the CFE problem, Section 6.3 presents the ACE algorithm, Sections 6.4–6.6 provide the empirical evaluation (quantitative, control study, qualitative), and Section 6.7 concludes with final remarks. An additional appendix section is provided at the end of this chapter (cf. Section 6.7).

6.2 Problem Statement

Consider a trained classifier $h\colon \mathcal{X} \to \{0,1\}$, where $\mathcal{X}$ is an input space endowed with a metric d, and let $\tilde{\mathbf{x}} \in \mathcal{X}$ be a fixed input, referred to as the *instance*, whose predicted value $\tilde{y} = h(\tilde{\mathbf{x}})$ we aim to "explain". In this context, we define a *counterfactual explanation* (CFE) as the solution to the following optimization problem:

$$\underset{\mathbf{x}}{\text{minimize}} \quad d(\mathbf{x}, \tilde{\mathbf{x}}) \quad \text{s.t.} \quad \mathbf{x} \in \mathcal{S}_{db}, \tag{6.2.1}$$

where $\mathcal{S}_{db} = \{\mathbf{x} \in \mathcal{X}\colon h(\mathbf{x}) \neq h(\tilde{\mathbf{x}})\}$ represents the decision boundary of the classifier. The solution of (6.2.1) corresponds to identifying the point $\mathbf{x}$ closest to $\tilde{\mathbf{x}}$ that lies on the decision boundary of the classifier ($\mathcal{S}_{db}$), representing the smallest change needed in the input features to induce a classification flip (*i.e.*, $h(\mathbf{x}) \neq h(\tilde{\mathbf{x}})$).

6.3 ACE Algorithm

In this section we present our novel algorithm, Adaptive sampling for Counterfactual Explanations (ACE), which relies on Bayesian optimization. To construct ACE, we address two central challenges: formulating a cost function J that captures proximity and structural constraints within the input space $\mathcal{X}$, and approximating the behavior of the black-box classifier h through a continuous surrogate model. Following standard Bayesian classification [92], we assume that the binary classifier is a thresholded version of a smooth latent function:

$$h(\mathbf{x}) = \mathbb{H}(f(\mathbf{x})), \tag{6.3.2}$$

where $\mathbb{H}\colon \mathbb{R} \to \{0,1\}$ is the Heaviside step function centered at 0.5 (*i.e.*, $\mathbb{H}(a) = 1$ if $a \geq 0.5$, and $\mathbb{H}(a) = 0$ otherwise), and $f\colon \mathcal{X} \to \mathbb{R}$ is a smooth map that we call the *black-box target function*. The next subsections detail the main components of ACE: (i) reformulating the constrained counterfactual problem via a penalized objective over f; (ii) modeling f with a Gaussian Process to account for the black-box nature of h; (iii) leveraging Expected Improvement to guide Bayesian Optimization; (iv) optimizing mixed-type inputs via gradient-based and combinatorial methods.

6.3.1 Lagrangian Cost Function

As introduced in the previous section, we express the classifier as $h(\mathbf{x}) = \mathbb{H}(f(\mathbf{x}))$, where f is a real-valued latent function. Since f will later be modeled as a probabilistic surrogate (see Section 6.3.2), we interpret the decision boundary as the set of points satisfying $f(\mathbf{x}) = 0.5$. Then, the CFE problem (6.2.1) can be written in terms of f as

$$\underset{x}{\text{minimize}}\ d(\mathbf{x}, \tilde{\mathbf{x}}) \quad \text{subject to } f(\mathbf{x}) = 0.5. \tag{6.3.3}$$

In order to solve this problem with Bayesian Optimization, we reformulate it as an unconstrained problem through a Lagrangian formulation. Let us define the cost function $J(x) = d(\mathbf{x}, \tilde{\mathbf{x}}) + \lambda |f(\mathbf{x}) - 0.5|$, and let $\{\lambda_k\}_{k=1,2,\dots}$ be a non-negative, increasing sequence tending to infinity. At each iteration k, we solve

$$\underset{\mathbf{x}}{\text{minimize}} \quad d(\mathbf{x}, \tilde{\mathbf{x}}) + \lambda_k |f(\mathbf{x}) - 0.5|, \tag{6.3.4}$$

by employing an optimization algorithm to identify the minimizer for the current penalty value λ_k. Ideally, as $\lambda_k \to \infty$, the solution of problem (6.3.4) is expected to approach the solution of the original constrained problem (6.3.3). This is the so-called "penalty function method" [93, Sec. 13.1]. The terms $d(\mathbf{x}, \tilde{\mathbf{x}})$ and $|f(\mathbf{x}) - 0.5|$ jointly enforce *proximity*: the first term keeps the candidate point close to the instance in the feature space, while the second one drives queries toward the decision boundary where the label flips. decision boundary. Additional constraints are then incorporated to satisfy the key properties for counterfactual explanations discussed in Section 2.3.2.

Actionability, Sparsity, and Plausibility. To enhance interpretability and feasibility, we impose three structural constraints (see more details in Appendix 6.A): (i) *Actionability*—only actionable features may change; let $\mathcal{A} \subseteq \mathcal{X}$ be the set where immutable features remain fixed, i.e., $\mathbf{x} \in \mathcal{A}$ enforces domain constraints (e.g., age, gender) [40]. (ii) *Sparsity*—a penalty $g(\mathbf{x} - \tilde{\mathbf{x}})$ (typically ℓ_1 or ℓ_0) promotes minimal modifications. (iii) *Plausibility*—a hard plausibility filter, $l(\mathbf{x}; \mathbf{X})$, is implemented via the Local Outlier Factor (LOF) [94], which measures how isolated a candidate is relative to the local density of its neighbors in the collected data $\mathbf{X}$. Thus, ACE keeps inliers and discards points that lie far from the observed data manifold. Beyond these constraints, ACE also satisfies key desiderata [95] such as validity, diversity, and scalability [57, 58] (cf. Appendix 6.A).

Extended Optimization Problem. Building upon the three structural constraints introduced above, we define the extended optimization problem as:

$$\underbrace{\arg\min_{\mathbf{x} \in \mathcal{A} \subset \mathcal{X}}}_{\text{actionability}} \underbrace{d(\mathbf{x}, \tilde{\mathbf{x}}) + \lambda_k |f(\mathbf{x}) - 0.5|}_{\text{proximity}} + \Theta(\mathbf{x}) \;, \quad \Theta(\mathbf{x}) = \underbrace{\beta\, g(\mathbf{x} - \tilde{\mathbf{x}})}_{\text{sparsity}} + \underbrace{l(\mathbf{x}; X)}_{\text{plausibility}} \;, \tag{6.3.5}$$

where the hyperparameter $\beta > 0$ controls the trade-off with sparsity.

6.3.2 Surrogate Model

To solve optimization problem (6.3.5), we need to approximate f based on samples. However, we may only have access to the classifier output $h(\mathbf{x}) \in \{0, 1\}$ at a given sample $\mathbf{x} \in \mathcal{X}$, rather than the underlying value $f(\mathbf{x})$ required to solve (6.3.5). Therefore, we approximate the true underlying function f with a surrogate $\hat{f}$. In our setting, inspired by Gaussian Process Classifiers (GPC) [96, Sec. 6.4], $\hat{f}$ is a realization of a Gaussian process that returns the likelihood of Class 1 membership, taking values in $[0, 1]$.

For classification problems, we estimate the class probability $p(t = 1 | \mathbf{x}, \mathbf{x}_*, \mathbf{t}_*)$, where $t \in \{0, 1\}$ is the label of a test input $\mathbf{x} \in \mathcal{X}$, $\mathbf{x}_* = \mathbf{x}_{1:n}$ are the training inputs, and $\mathbf{t}_* = \mathbf{t}_{1:n}$ their binary labels. We denote the observed dataset by $(\mathbf{X}, \mathbf{y}) = (\mathbf{x}_*, \mathbf{t}_*)$, the kernel matrix by $\mathbf{K} \in \mathbb{R}^{n \times n}$ with entries $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$, and the latent values at the training inputs by $\mathbf{a}_* = \hat{f}(\mathbf{x}_*)$. The posterior distribution $p(\mathbf{a}_* | \mathbf{x}, \mathbf{x}_*, \mathbf{t}_*)$ can be replaced by the Laplace approximation [92, Sec. 3.4]

$$p(\mathbf{a}_* | \mathbf{x}, \mathbf{x}_*, \mathbf{t}_*) \approx \mathcal{N}\left(\mathbf{a}_*; \hat{\mathbf{a}},\ (\mathbf{W}(\hat{\mathbf{a}}) + \mathbf{K}^{-1})^{-1}\right),$$

where $\mathbf{W}(\mathbf{a}) = \text{diag}(\sigma_s(a_i)[1 - \sigma_s(a_i)])$, with $\sigma_s(a) = 1/(1 + \exp(-a))$ denoting the logistic sigmoid function, and $\hat{\mathbf{a}}$ is obtained by iterating the Newton update in [92, Sec. 3.4, Eq. (3.18)] until convergence. Given this approximation for the latent values at the training inputs, we can now compute the posterior distribution over the latent function value $a = \hat{f}(\mathbf{x})$ at a test input $\mathbf{x}$, which is approximately Gaussian, $\mathcal{N}(\mu_a, \sigma_a^2)$, where μ_a and σ_a^2 are the logit mean and the logit variance, respectively, defined as

$$\mu_a = \mathbf{k}_n^T(\mathbf{t}_* - \boldsymbol{\sigma}(\hat{\mathbf{a}})), \qquad \sigma_a^2 = \kappa(\mathbf{x}, \mathbf{x}) - \mathbf{k}_n^T \left(\mathbf{W}(\hat{\mathbf{a}})^{-1} + \mathbf{K}\right)^{-1} \mathbf{k}_n. \tag{6.3.6}$$

with $\mathbf{k}_n := [\kappa(\mathbf{x}, \mathbf{x}_1), \dots, \kappa(\mathbf{x}, \mathbf{x}_n)]^T$ denoting the kernel vector between x and the training inputs. The predictive class-1 probability is then approximated using the inverse probit transformation, i.e., $p(t = 1 \mid \mathbf{x}, \mathbf{x}_*, \mathbf{t}_*) = \sigma_s\big(\mu_a\big(1 + \frac{\pi\sigma_a^2}{8}\big)^{-1/2}\big)$ (cf. Appendix 6.C). The posterior mean μ_a in logit space is transformed into a probability via the sigmoid function, yielding $\mu = \sigma_s(\mu_a)$. The variance in probability space is then $\sigma^2 = \sigma_a^2 \mu^2 (1 - \mu)^2$, obtained via the delta method [97, p. 240], as discussed in Appendix 6.D.

6.3.3 Expected Improvement

Given the previous observations and the constructed surrogate model, we adaptively pick new points using Expected Improvement (EI), a Bayesian optimization acquisition function that selects the most informative sampling point. EI measures the gap between the current optimum and the surrogate function at a point $\mathbf{x} \in \mathcal{X}$, *i.e.*,

$$\text{EI}_n(\mathbf{x}) := \mathbb{E}_n \left\{ [J_n^* - J(\mathbf{x})]_+ \right\} \tag{6.3.7}$$

where $\mathbb{E}_n$ denotes expectation conditioned on the previously observed data, $J(\mathbf{x})$ is defined in Equation (6.3.5), and $\mathbf{x}^* = \arg\min_{\mathbf{x}_i \in \mathbf{x}_{1:n}} J(\mathbf{x}_i)$ is the cost minimizer among the previously observed inputs. To estimate $\text{EI}_n(\mathbf{x})$, we use a correlated Monte Carlo sampling method [96, Sec. 11.1]. Accordingly, we define $\hat{q}(\mathbf{x})$ as

$$\hat{q}(\mathbf{x}) = \max(0, d(\mathbf{x}^*, \tilde{\mathbf{x}}) + \lambda|\hat{f}(\mathbf{x}^*) - 0.5| + \Theta(\mathbf{x}^*) - d(\mathbf{x}, \tilde{\mathbf{x}}) - \lambda|\hat{f}(\mathbf{x}) - 0.5| - \Theta(\mathbf{x})).$$

We approximate Eq. (6.3.7) by a Monte Carlo average over m draws of $\hat{q}(\mathbf{x})$. To obtain *correlated* GP samples of the latent values $\hat{f}(\mathbf{x})$ and $\hat{f}(\mathbf{x}^*)$, we use their joint Gaussian specified by the posterior means $\mu(\mathbf{x}), \mu(\mathbf{x}^*)$, variances $\sigma^2(\mathbf{x}), \sigma^2(\mathbf{x}^*)$, and cross-covariance $\text{Cov}(\mathbf{x}, \mathbf{x}^*)$, and then form the 2×2 covariance matrix $\boldsymbol{\Sigma}_f$. To generate correlated samples, we apply Cholesky decomposition to $\boldsymbol{\Sigma}_f$, i.e., $\boldsymbol{\Sigma}_f = \mathbf{L}\mathbf{L}^T$, where $\mathbf{L}$ is a lower triangular matrix. Sampling $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, we obtain correlated samples via $[\hat{f}(\mathbf{x}), \hat{f}(\mathbf{x}^*)]^T = [\mu(\mathbf{x}), \mu(\mathbf{x}^*)]^T + \mathbf{L}\mathbf{z}$. The derivation of the cross-covariance term $\text{Cov}(\mathbf{x}, \mathbf{x}^*)$ is detailed in Appendix 6.D. Finally, we approximate the maximizer of EI_n using a general-purpose optimization method such as the L-BFGS-B algorithm [98].

6.3.4 Branch and Bound Method for Mixed Variables

ACE handles mixed variables by combining L-BFGS-B for continuous parameters with a Branch-and-Bound (B&B) search over categorical ones. Starting from the

continuous optimum (root), B&B fixes the categorical values and re-optimizes the remaining continuous variables to maximize the acquisition function; categorical features use ordinal/label encoding when appropriate. Full details and a schematic example appear in Appendix 6.E.

6.3.5 Overall Algorithm

The ACE pseudo-code is outlined in Algorithm 6.1, and the code is available on ACE repository.

Initialization. In line 1 from Algorithm 6.1, we draw n_0 samples to fit a Gaussian process. We assume mild prior information (bounds/summary stats): truncated normal for continuous features and uniform over categories, or alternatively select n_0 points from a known dataset.

Optimize Acquisition Function. After fitting the kernel to $(\mathbf{X}, \mathbf{y})$, we maximize the acquisition function with L-BFGS-B (line 5), initialized around $\tilde{\mathbf{x}}$ via a truncated normal. For high-dimensional data, we initialize in a PCA space [96, Sec. 12.1] and project back before optimizing.

Filtered Monte-Carlo Expected Improvement. ACE calculates the EI using Monte Carlo sampling, cf. Section 6.3.3, with *MC* samples. Both $d(\mathbf{x}, \tilde{\mathbf{x}})$ and $g(\mathbf{x} - \tilde{\mathbf{x}})$ in (6.3.5) are computed using feature-normalized norms, where each dimension is scaled by the standard deviation of the corresponding feature in the current dataset $\mathbf{X}$.

Best Posterior CFE. ACE evaluates the posterior mean on a Sobol low discrepancy grid [99] to find points with posterior probability closest to 0.5, as shown in line 10. Among these, the CFE is the one with minimal Euclidean distance to $\tilde{\mathbf{x}}$; sparsity is already encouraged by the cost in (6.3.5). If the selected candidate ($\mathbf{x}_s^n$) has the desired label but is farther from $\tilde{\mathbf{x}}$ than the previous CFE found ($\mathbf{x}_s^o$) (line 2), the algorithm terminates; otherwise, the process continues until a closer CFE is identified.

Algorithm 6.1 ACE algorithm

Input: Initial data $\mathbf{x}_{1:n_0}$, instance to explain $\tilde{\mathbf{x}}$
Parameters: λ_0 (init. penalty), λ_{max} (max penalty), κ (kernel), MC (MC samples), SS (Sobol samples), ϵ (tolerance), p (penalty growth)
Output: CFE $\mathbf{x}_s^n$

1: $\mathbf{X}, \mathbf{y} \leftarrow$ UpdateInitialData$(\mathbf{x}_{1:n_0}, h(\mathbf{x}_{1:n_0}))$
2: **while** $h(\tilde{\mathbf{x}}) = h(\mathbf{x}_s^n) \wedge \|\mathbf{x}_s^n - \tilde{\mathbf{x}}\| < \|\mathbf{x}_s^o - \tilde{\mathbf{x}}\|$ **do**
3: $\quad \mathbf{x}_s^o \leftarrow \mathbf{x}_s^n, \quad \lambda_k \leftarrow \lambda_0$
4: $\quad$ **while** $\|\mathbf{x}_k - \mathbf{x}_{k-1}\| > \epsilon \vee \lambda_k < \lambda_{\text{max}}$ **do**
5: $\quad\quad \mathbf{x}_k \leftarrow \arg\max_{\mathbf{x}} \text{EI}_k(\mathbf{X}, \mathbf{y}, \lambda_k, \kappa, MC)$
6: $\quad\quad$ Observe $h(\mathbf{x}_k)$
7: $\quad\quad \mathbf{X}, \mathbf{y} \leftarrow$ UpdateData$(\mathbf{x}_k, h(\mathbf{x}_k))$
8: $\quad\quad k \leftarrow k + 1, \quad \lambda_k \leftarrow (\lambda_{k-1})^p$
9: $\quad$ **end while**
10: $\quad \mathbf{x}_s^n \leftarrow$ SampleDecisionBoundary$(\mathbf{X}, \mathbf{y}, \kappa, SS)$
11: **end while**
12: **return** $\mathbf{x}_s^n$

6.4 Quantitative Evaluation

In this section, we define evaluation criteria based on the key properties of CFEs (Section 6.3.1, Appendix 6.A) and introduce an aggregated score for comparison. Using these metrics, we benchmark ACE against three state-of-the-art methods across eight binary classification datasets.

Table 6.4.1: Summary of real-world datasets.

Dataset	Features (Num/Cat)	Samples
Diabetes	8/0	768
KC2	21/0	522
Breast Cancer	9/0	683
Blood	4/0	748
Tic-Tac-Toe	0/9	958
Nursery	0/8	12,961
CMC	2/7	1,473
Credit	4/5	1,000

6.4.1 Experimental Setup

We evaluate three state-of-the-art methods: BayCon [41], MOC [39], and Growing Spheres (GS) [100], with the latter implemented using CARLA [101]. Experiments use eight real-world datasets (Table 6.4.1); due to space constraints, we report in the main text only four datasets—one label per setting—while the remaining ones and complementary-label results appear in Appendix 6.F. Datasets with only numerical features are labeled *continuous*, while those mixing numerical and categorical features are *heterogeneous*.

For each dataset, we perform two evaluations: (i) *fixed-instance*, where a single input instance is selected and each method is executed 100 times using different random seeds to assess robustness and variability; and (ii) *mixed-instance*, which involves generating one counterfactual explanation for each of 100 randomly sampled instances and target labels to evaluate general performance. For BayCon and MOC, which yield multiple candidates, we report the one minimizing the cost in (6.3.5), ensuring a fair comparison with ACE and GS, which return a single CFE per run.

Across all experiments, CFEs are generated for a Random Forest black-box model trained via bootstrap aggregation. Hyperparameter details are provided in Appendix 6.G. All experiments were conducted in Python 3.12.7, using SciPy for optimization[1].

[1]The GP classifier in ACE was implemented from scratch, as standard libraries (e.g., `scikit-learn`) do not expose the latent posterior—*i.e.*, the mean and variance in (6.3.6)—required to compute covariances between candidates $\mathbf{x}$ and the current best $\mathbf{x}^*$ (see Appendix 6.D).

6.4.2 Evaluation Metrics

The main evaluation metric is the *number of black-box evaluations* $h_{\#}$, used as a proxy for sample efficiency. We measure *proximity* via $d_2 = \|\mathbf{x} - \tilde{\mathbf{x}}\|_2$, and *sparsity* by $g_1 = \|\mathbf{x} - \tilde{\mathbf{x}}\|_1$. In the mixed-instance test setting, both metrics are normalized by the standard deviation of each input feature computed over the entire dataset to account for feature scale differences, and are denoted by d_{2_N} and g_{1_N}, respectively. *Plausibility* uses the affinity score $\alpha(\mathbf{x}) := \texttt{clip}\{\exp(1 + \text{LOF}_k(\mathbf{x}))\}$ (cf. Appendix 6.B) with values close to 1 indicating inliers. We also report *validity* $\mathcal{V}$, the proportion of successful counterfactual generations.

CFE Score

To compare methods, we define a scalar *CFE Score* $\mathcal{S}$ that aggregates all metrics into a single value using min-max normalization. Since affinity and validity lie in $[0, 1]$ and are to be maximized, we use $(1 - \texttt{metric})$ to align interpretation across terms, where lower values are preferred.

Based on the normalized metrics, the CFE Score is defined as: $\mathcal{S} = w_1 \cdot \tilde{h}_{\#} + w_2 \cdot \tilde{d}_2 + w_3 \cdot \tilde{g}_1 + w_4 \cdot (1 - \alpha) + w_5 \cdot (1 - \mathcal{V})$, where the tilde $\tilde{(\cdot)}$ denotes min-max normalization, and w_i are weights reflecting metric priorities. Since ACE prioritizes sample efficiency, we set $w_1 = 0.35$ for the number of queries $h_{\#}$. The remaining weights are 0.25 for d_2, 0.15 for g_1, and 0.125 for both affinity and validity, with $\sum w_i = 1$. A lower $\mathcal{S}$ indicates better performance.

6.4.3 Benchmark Results

Fixed Test Results

Tables 6.4.2 and 6.4.3 summarize the evaluation results for continuous and heterogeneous datasets. Each method explains two fixed instances 100 times to assess consistency (mean) and variability (standard deviation in parentheses). Non-actionable features—such as age or gender—are kept fixed throughout the optimization to guarantee feasibility and interpretability. ACE is initialized with $n_0 = 30$ points for all datasets (counted in the total evaluations). Across all four datasets, ACE consistently outperforms competing methods in terms of the number of black-box evaluations ($h_{\#}$), successfully generating counterfactuals in every experiment.

Continuous Datasets. On *Diabetes* and *Breast*, ACE achieves the lowest CFE Score, offering a better trade-off across all metrics. It outperforms BayCon and MOC in proximity while maintaining similar sparsity. Although GS obtains slightly better d_2, it requires over 28,000 and 40,883 queries—compared to around 71 and 60 for ACE—making it highly inefficient. Notably, ACE consistently achieves $\alpha(x) \approx 1$ and is the only method to produce valid CFEs in all 200 queries (same for all 600 runs in Appendix 6.F).

Table 6.4.2: Fixed test for continuous datasets with 100 tested points per experiment. Best CFE score are highlighted.

Dataset	Method	$h_{\#}$ (std)	d_2 (std)	g_1 (std)	$\alpha(x)$ (std)	$\mathcal{V}$ (std)	$\mathcal{S}$
Diabetes	ACE	71 (16)	34.09 (14.30)	49.57 (22.34)	1 (0.06)	1 (0)	**0.24**
	BayCon	1211 (110)	37.13 (9.23)	46.87 (12.19)	0.25 (0.08)	1 (0)	0.36
	MOC	2038 (1065)	58.95 (18.46)	62.91 (16.77)	0.84 (0.19)	0.98 (0.14)	0.45
	CARLA_{GS}	28013 (4455)	5.51 (0.90)	12.46 (2.24)	0.78 (0.03)	1 (0)	0.38
Breast	ACE	60 (16)	9.33 (1.05)	27.89 (2.87)	1 (0.10)	1 (0)	**0.20**
	BayCon	3567 (1058)	9.60 (1.25)	21.12 (1.90)	0.31 (0.03)	0.03 (0.20)	0.35
	MOC	845 (748)	10.28 (1.19)	20.81 (3.11)	0.09 (0.06)	0.97 (0.17)	0.37
	CARLA_{GS}	40883 (3545)	9.08 (0.71)	21.41 (2.25)	0.77 (0.03)	1 (0)	0.39

Heterogeneous Datasets. On *CMC* and *Nursery*, ACE identifies CFEs by changing only one feature by a single unit, while using far fewer evaluations, demonstrating the efficiency of its B&B strategy. Once again, ACE maintains high plausibility and validity, confirming its robustness on mixed-variable datasets.

Mixed Test Results

Tables 6.4.4 and 6.4.5 present the results of the mixed-instance test, where as in the fixed test, ACE retains its sample-efficiency edge, matching or surpassing baselines with far fewer evaluations.

Table 6.4.3: Fixed test for heterogeneous datasets with 100 tested points per experiment. Best CFE score are highlighted.

Dataset	Method	$h_{\#}$ (std)	d_2 (std)	g_1 (std)	$\alpha(x)$ (std)	$\mathcal{V}$ (std)	$\mathcal{S}$
CMC	ACE	65 (6)	1 (0)	1 (0)	1 (0.01)	1 (0)	**0**
	BayCon	1021 (102)	1 (0)	1 (0)	0.33 (0.02)	1 (0)	0.43
	MOC	590 (212)	1 (0.1)	1 (0.2)	0.9 (0.1)	1 (0)	0.2
	CARLA_{GS}	1005 (0)	3.87 (0)	7 (0)	1 (0)	1 (0)	0.74
Nursery	ACE	57 (7)	1 (0.04)	1.01 (0.1)	1 (0)	1 (0)	**0**
	BayCon	836 (29)	1 (0)	1 (0)	0.37 (0.01)	1 (0)	0.37
	MOC	220 (0)	- (-)	- (-)	- (-)	0 (0)	-
	CARLA_{GS}	1005 (0)	3.32 (0)	5 (0)	1 (0)	1 (0)	0.75

Continuous Datasets. ACE tops CFE Score across datasets, while BayCon attains slightly better proximity and sparsity. However, BayCon's CFEs are less plausible ($\alpha(\mathbf{x}) = 0.27 - 0.30$ vs. $0.98 - 1$ for ACE), indicating they lie farther from the data manifold. ACE also requires only 5.77% of BayCon's queries on average, highlighting its efficiency.

Table 6.4.4: Mixed test for continuous datasets. Best CFE scores are highlighted.

Dataset	Method	$h_{\#}$(std)	d_{2_N}(std)	g_{1_N}(std)	α(std)	$\mathcal{V}$ (std)	$\mathcal{S}$
Diabetes	ACE	72 (18)	2.46 (0.84)	2.7 (1.87)	0.98 (0.09)	1 (0)	**0.10**
	BayCon	1239 (238)	2.04 (0.97)	2.99 (1.89)	0.27 (0.09)	1 (0)	0.14
	MOC	1753 (1011)	2.09 (0.91)	2.62 (1.34)	0.58 (0.36)	0.91 (0.29)	**0.10**
	CARLA$_{GS}$	27463 (23054)	3.2 (2.66)	4.26 (3.41)	0.74 (0.08)	1 (0)	0.78
Breast	ACE	67 (22)	2.45 (1.37)	7.6 (3.95)	1 (0.01)	1 (0)	**0.16**
	BayCon	2045 (952)	2.42 (1.25)	4.25 (2.63)	0.30 (0.08)	0.60 (0.49)	**0.16**
	MOC	890 (804)	3.36 (1.19)	5.7 (2.51)	0.47 (0.33)	1 (0)	0.39
	CARLA$_{GS}$	36803 (20489)	2.62 (1.46)	6.56 (3.81)	0.76 (0.05)	1 (0)	0.54

Heterogeneous Datasets. Again, ACE obtains the best CFE Score on all datasets. Nevertheless, a similar trade-off is observed: while BayCon achieves slightly better proximity or sparsity, its affinity score is markedly lower. Thus, ACE yields feasible, actionable CFEs under tighter query budgets.

Table 6.4.5: Mixed test for heterogeneous datasets. Best CFE scores are highlighted.

Dataset	Method	$h_{\#}$ (std)	d_{2_N} (std)	g_{1_N} (std)	α (std)	$\mathcal{V}$ (std)	$\mathcal{S}$
CMC	ACE	70 (12)	2.12 (0.92)	2.99 (1.8)	1 (0.01)	0.97 (0.17)	**0.07**
	BayCon	1024 (241)	1.85 (0.71)	2.39 (1.24)	0.32 (0.05)	0.98 (0.14)	0.12
	MOC	332 (124)	1.57 (0.68)	1.8 (1.01)	0.8 (0.24)	0.18 (0.38)	0.13
	CARLA$_{GS}$	330674 (469741)	5.04 (0.9)	9.19 (1.96)	1 (0)	0.67 (0.47)	0.79
Nursery	ACE	56 (7)	1.22 (0)	1.22 (0)	1 (0.01)	1 (0)	**0**
	BayCon	840 (31)	1.22 (0)	1.22 (0)	0.37 (0)	1 (0)	0.37
	MOC	226 (10.47)	- (-)	- (-)	- (-)	0 (0)	-
	CARLA$_{GS}$	1005 (0)	2.09 (0.5)	3.4 (1.27)	1 (0)	1 (0)	0.75

6.5 Adaptive Control Tuning

In this section we apply ACE to iterative controller retuning. Starting from a controller-parameter baseline $\boldsymbol{\theta}_0$ that fails time–domain specs, we cast "meeting the specifications" as a binary oracle $h(\boldsymbol{\theta}) \in \{0, 1\}$ and seek the smallest actionable change $\boldsymbol{\theta} - \boldsymbol{\theta}_0$ that flips 0 (fail) to 1 (pass) under tight trial budgets.

6.5.1 Methodology

We consider a critical process that cannot be halted during operation; thus, parameter tuning must rely on short *closed-loop* trials with small, limited updates and without aggressive excitation.

Let $\boldsymbol{\theta}_i \in \mathbb{R}^p$ denote the controller parameters used in the i-th closed-loop trial. For a unit step, we compute the vector of time-domain metrics $\pi(\boldsymbol{\theta}_i) = (e_{\text{ss}}(\boldsymbol{\theta}_i), t_{\text{rise}}(\boldsymbol{\theta}_i), t_{\text{settle}}(\boldsymbol{\theta}_i), \text{OS}, \|u\|_{\text{rms}}(\boldsymbol{\theta}_i))$, where e_{ss} is the steady-state error, t_{rise} the rise time, t_{settle} the settling time (for a 5% band), OS the overshoot percentage,

and $\|u\|_{\text{rms}}$ is the control effort. The pass/fail binary oracle is

$$h(\boldsymbol{\theta}_i) = \mathbb{I}\{\pi(\boldsymbol{\theta}_i) \preceq \boldsymbol{\tau}\}, \quad i = 1, \dots, N, \tag{6.5.8}$$

where $\boldsymbol{\tau} \in \mathbb{R}^5$ collects the per-metric thresholds and $h = 1$ iff $\pi(\boldsymbol{\theta}_i) \preceq \tau$ (all specs met), and $h = 0$ otherwise. We treat (6.5.8) as the classifier from Section 6.3 (cf. (6.3.2)) and apply ACE via Algorithm 6.1.

6.5.2 Experimental Setting

Process & Controller. We study shell–and–tube heat–exchange temperature control in which steam flow is manipulated to regulate outlet–water temperature. The controller is a classical cascaded lead–lag compensator $F(s)$; the plant dynamics $G(s)$ are approximated by a standard second-order plus dead-time (SOPDT) model [102]:

$$G(s) = \frac{5\, e^{-1s}}{90s^2 + 33s + 1}, \qquad F(s) = K\, \frac{T_d s + 1}{\beta T_d s + 1}\, \frac{T_i s + 1}{T_i s + \gamma}, \quad 0 < \beta, \gamma < 1.$$

The tunable parameter vector is $\boldsymbol{\theta} = (K,\, T_d,\, \beta,\, T_i,\, \gamma)$. For ACE, $G(s)$ is unknown—the plant is treated as a black box (cf. (6.3.2)).

Spec thresholds. By Eq. (6.5.8), we set $\boldsymbol{\tau} = (\tau_e, \tau_{\text{rise}}, \tau_{\text{settle}}, \tau_{\text{OS}}, \tau_{\|u\|_{\text{rms}}}) = (0.01,\ 20\text{s},\ 50\text{s}, 20\%, 2)$.

Configurations. We initialize with 30 historical controllers identified from logged batches $(\mathbf{r}^{(i)}_{1:N_i}, \mathbf{u}^{(i)}_{1:N_i}, \mathbf{y}^{(i)}_{1:N_i})$. For each batch, the parameters are fitted via output–error NLS [7, Ch. 7]; details in Appendix 6.G. We add Gaussian noise σ_y^2=0.01 and use a unit step (r_{step}=1). Plant and controller are discretized via the bilinear transform with sampling period $T_s = 0.1\,\text{s}$.

6.5.3 Results

We perform 100 independent ACE runs, with different random seeds, from the same baseline $\boldsymbol{\theta}_0 = [4.7770, 2.4354, 0.0132, 27.5408, 0.3100]$. Figure 6.6.1 shows the component-wise deviations $\Delta = \boldsymbol{\theta}_{\text{cand}} - \boldsymbol{\theta}_0$, with $\boldsymbol{\theta}_{\text{cand}}$ collecting *all* candidate evaluations produced during the search—not only the final CFEs. The boxes show the interquartile range (25–75%), the orange line is the median, whiskers cover the non-outlier range, and circles are outliers. Across parameters—except T_I—typical magnitudes stay small ($|\Delta| \lesssim 0.45$), with only modest positive drifts in T_D and γ. The median trend indicates a mild increase in T_I, consistent with using a gentle lag to improve low-frequency behavior while keeping the phase-margin impact limited (per lead–lag guidelines [103, Ch. 6]). Overall, the distributions are tightly centered around zero, showing that ACE searches in a compact neighborhood and proposes small, local retunings; on average, ~ 16 proposals per run are needed before a CFE is found.

Figure 6.6.2 compares the baseline $\boldsymbol{\theta}$ (red) against a representative CFE $[3.4294,\ 2.2920, 0.0428,\ 28.9531, 0.1605]$ (blue). Before tuning, four specs were violated:

$e_{ss} = 0.0128$, $t_{settle} \approx 797$ s, OS $= 33\%$, and $\|u\|_{rms} = 2.8355$. After ACE, they improve to $e_{ss} = 0.0093$, $t_{settle} \approx 8.0$ s, OS $= 18.56\%$, and $\|u\|_{rms} = 0.9620$, consistent with Fig. 6.6.2, showing that a few targeted tweaks bring the loop in-spec.

6.6 Qualitative Analysis

We qualitatively analyze ACE's outputs in low- and high-dimensional settings, as visual inspection is key to understanding counterfactuals.

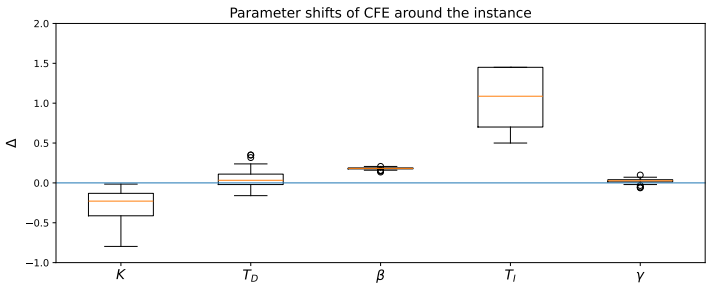


Figure 6.6.1: Box plot of component-wise shifts $\boldsymbol{\Delta} = \boldsymbol{\theta}_{\text{cand}} - \boldsymbol{\theta}_0$ across 100 ACE runs.

6.6.1 Low-Dimensional Visualization

Synthetic Dataset. We compare ACE to Growing Spheres (GS) [85] on a 2D `make_moons` example [104]. The black-box is an RBF-SVC with $\gamma = 1$, where $\gamma = 1/(2\sigma^2)$ controls the bias–variance trade-off [105, p. 47].

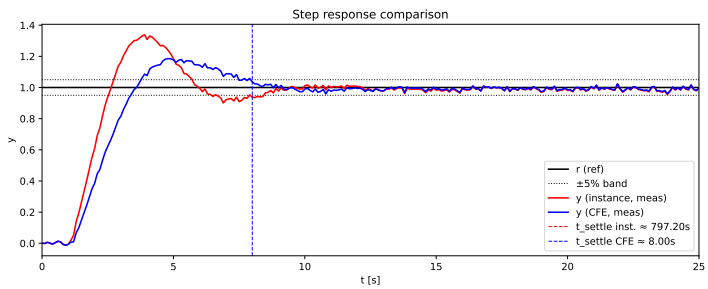


Figure 6.6.2: Step responses: baseline $\boldsymbol{\theta}_0$ (red) vs. one representative CFE (blue).

Figure 6.6.3 compares ACE (first figure) and GS (second figure) on `make_moons`. The black dashed 0.5 contour approximates the black-box boundary (red dotted). The instance $\tilde{\mathbf{x}} = [0.55, 0.25]^T$ is green and CFEs are yellow. With $n_0 = 4$, ACE reaches $[1.23, 0.25]^T$ in 14 queries vs. GS's $[1.14, 0.20]^T$ in 501; distances are 0.68 vs. 0.592. ACE moves on one axis $[+0.68, 0]$ while GS perturbs two $[+0.59, -0.05]$, yielding a sparser, simpler change despite a slightly larger distance.

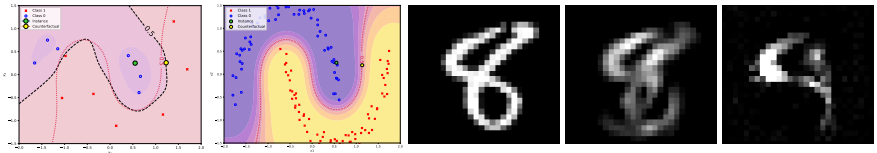


Figure 6.6.3: First: ACE on Make Moons. Second: GS on Make Moons. Third-Fifth: MNIST results (original, ACE CFE, and OmniXAI CFE).

6.6.2 High-Dimensional Visualization

MNIST. We use 28×28 grayscale digits (784-d) and focus on number 8 vs. 9. The task is to find the minimal ℓ_2 perturbation to turning an 8 into a 9. We compare ACE to OmniXAI [16] a library that generates counterfactual examples using a CNN trained specifically for this task. The black-box model is a CNN trained for binary classification between 8 and 9. In Fig. 6.6.3 (3 rightmost figures)—Class 0 corresponds to digit 8 and Class 1 to digit 9—ACE (with $n_0 = 50$) finds a valid CFE after only 9 extra queries (59 total), yielding a digit with an opened lower loop that visually resembles a 9 and is confidently classified as Class 1 by both the surrogate and black-box models. OmniXAI fails at 50 queries but succeeds after retraining on the same 59 points. ACE attains $\ell_2 = 5.47$ with posterior 55.19% (near the 50% boundary), vs. OmniXAI's $\ell_2 = 7.44$ and 64.93%. Overall, ACE demonstrates superior sample efficiency and plausibility, yielding concise and actionable explanations.

6.7 Conclusions and Future Work

In this chapter, we have introduced *Adaptive Sampling for Counterfactual Explanations* (ACE), a query-efficient framework for generating reliable and meaningful counterfactual explanations in binary classification problems. Through an extensive numerical study involving synthetic and real-world datasets, ACE is shown to consistently reduce the number of black-box evaluations required to identify a counterfactual while preserving desirable properties such as proximity, feasibility, plausibility, and actionability. These results demonstrate that informative sequential sampling can provide an effective alternative to exhaustive exploration when access to the underlying model is limited or expensive.

Beyond its original role as an explainability method, ACE also opens a natural connection between XAI and control. This intersection remains comparatively young, with only limited work devoted to exploiting explainability methods for the modification of closed-loop behavior. The lead–lag tuning example presented in this chapter constitutes a first step in this direction: controller parameters are interpreted as features of a black-box classifier, and a counterfactual corresponds to a small parameter modification capable of achieving the desired closed-loop specifications. To the best of our knowledge, this represents the first use of counterfactual explanations explicitly as a controller-tuning mechanism. Importantly, the formulation itself is not tied to a particular controller structure or to linear dynamics, suggesting a broader potential for counterfactual reasoning in data-driven controller adjustment.

Several directions naturally follow from these results. From the XAI perspective, multi-objective formulations could be explored to promote diversity among counterfactual explanations and further theoretical properties of ACE remain to be characterized. From the control perspective, a particularly important direction is to incorporate prior experimental knowledge and operational constraints into the search, so that informative controller modifications can be identified from only a small number of experiments while avoiding poorly explored or potentially unsafe regions of the parameter space. Chapter 7 takes a first step toward this objective by applying ACE to the fine-tuning of a pressure-control loop for an industrially relevant axial compressor. Although this application again considers a linear controller, the black-box nature of ACE does not inherently restrict the framework to linear controllers or plants, leaving its extension to more complex and nonlinear closed-loop systems as a promising direction for future research.

Appendix

6.A Desiderata for Counterfactual Explanations

As identified in prior literature [57, 58], high-quality counterfactual explanations (CFEs) should satisfy the following key properties, which are effectively addressed by our proposed method:

Validity. *The counterfactual must change the predicted label.* ACE guarantees validity by explicitly searching for counterfactuals $\mathbf{x}$ such that the predicted outcome differs from the original instance $\tilde{\mathbf{x}}$, i.e., $h(\mathbf{x}) \neq h(\tilde{\mathbf{x}})$.

Sparsity. *Minimal number of features should be altered.* To encourage sparse solutions, ACE incorporates an ℓ_1-norm penalty in its cost function, promoting counterfactuals that modify as few features as possible.

Proximity. *Counterfactuals should be close to the original input.* ACE measures proximity via a distance metric $d(\mathbf{x}, \tilde{\mathbf{x}})$, typically the Euclidean norm, though other metrics can be used depending on the data characteristics. This ensures that explanations remain within a small, interpretable neighborhood ε around $\tilde{\mathbf{x}}$, i.e., $d(\mathbf{x}, \tilde{\mathbf{x}}) < \varepsilon$.

Actionability. *Only mutable features should be changed.* ACE supports user-defined feature constraints and restricts modifications to actionable features, leaving immutable attributes (e.g., age or gender) unchanged throughout the optimization process to ensure the generation of feasible CFEs.

Diversity. *Providing multiple distinct counterfactuals improves user choice.* ACE combines Monte Carlo sampling with GP uncertainty to explore the decision boundary globally, while prioritizing diverse candidates near $\tilde{\mathbf{x}}$. This dual mechanism enables the generation of multiple, semantically distinct yet plausible CFEs.

Plausibility. *Counterfactuals should respect feature constraints and data distribution.* ACE enforces domain constraints and avoids unrealistic combinations by sampling within the input domain and promoting CFEs in high-density regions with respect to the training data.

Scalability. *Efficient generation across multiple instances.* Thanks to its Bayesian surrogate model, ACE reuses learned structures across similar queries and scales to both low- and high-dimensional datasets, supporting simultaneous generation of multiple CFEs.

6.B Extended Cost Function

Plausibility Term. To ensure closeness to the data manifold, we penalize counterfactuals that lie in low-density regions relative to the collected data $\mathbf{X}$ using the *Local Outlier Factor (LOF)* [94]. LOF quantifies the degree to which a point is isolated from its neighbors, with scores typically interpreted as

$$\mathrm{LOF}_k(\mathbf{x}) := \frac{1}{|N_k(\mathbf{x})|} \sum_{\mathbf{z} \in N_k(\mathbf{x})} \frac{\mathrm{lrd}(\mathbf{z})}{\mathrm{lrd}(\mathbf{x})}, \tag{6.B.9}$$

where $N_k(\mathbf{x})$ is the k-nearest neighborhood of $\mathbf{x}$, and $\mathrm{lrd}(\mathbf{z})$ denotes the local reachability density. In our implementation, we adopt `scikit-learn`'s `Local Outlier Factor` with `novelty = True`, which outputs negative LOF scores. A

point $\mathbf{x}$ is considered an *inlier* if its LOF score exceeds a threshold τ (in ACE, we use the default threshold, typically $\tau = -1.5$); otherwise, it is deemed implausible. Thus, we define the penalty or plausibility term $l(\mathbf{x}; \mathbf{X})$ as

$$l(\mathbf{x}; \mathbf{X}) := \begin{cases} 0, & \text{if } \mathrm{LOF}_k(\mathbf{x}) > \tau \text{ (inlier)}, \\ \infty, & \text{otherwise (outlier)}, \end{cases}$$

which effectively acts as a hard constraint that discards implausible candidates.

6.C Class-1 Posterior Calculations

For classification problems, we aim to estimate the probability $p(t = 1|\mathbf{x}, \mathbf{x}_*, \mathbf{t}_*)$, where $l \in \{0, 1\}$ is the label corresponding to input $\mathbf{x} \in \mathcal{X}$, and $\mathbf{t}_* \in \{0, 1\}^n$ is a vector of labels at the inputs in $\mathbf{x}_* \in \mathcal{X}^n$. This probability will be approximated as

$$p(t = 1|\mathbf{x}, \mathbf{x}_*, \mathbf{t}_*) = \int p(t = 1|a)p(a|\mathbf{x}, \mathbf{x}_*, \mathbf{t}_*)da,$$

where $a = \hat{f}(\mathbf{x})$, $p(t = 1|a) = \sigma(a) := 1/(1 + \exp(-a))$ is the logistic sigmoid function, and $\hat{f}$ is a Gaussian process. We are actually interested only on $p(a|\mathbf{x}, \mathbf{x}_*, \mathbf{t}_*)$, given by

$$p(a|\mathbf{x}, \mathbf{x}_*, \mathbf{t}_*) = \int p(a|\mathbf{x}, \mathbf{x}_*, \mathbf{a}_*)p(\mathbf{a}_*|\mathbf{x}, \mathbf{x}_*, \mathbf{t}_*)d\mathbf{a}_*,$$

where $p(a|\mathbf{x}, \mathbf{x}_*, \mathbf{a}_*)$ is a Gaussian distribution given by

$$p(a|\mathbf{x}, \mathbf{x}_*, \mathbf{a}_*) = \mathcal{N}(a; \mathbf{k}_n^T \mathbf{K}^{-1} \mathbf{a}_*, \kappa(\mathbf{x}, \mathbf{x}) - \mathbf{k}_n^T \mathbf{K}^{-1} \mathbf{k}_n).$$

The density $p(\mathbf{a}_*|\mathbf{x}, \mathbf{x}_*, \mathbf{t}_*)$, on the other hand, can be replaced by the Laplace approximation [92, Sec. 3.4]

$$p(\mathbf{a}_*|\mathbf{x}, \mathbf{x}_*, \mathbf{t}_*) \approx \mathcal{N}\left(\mathbf{a}_*; \hat{\mathbf{a}},\ (\mathbf{W}(\hat{\mathbf{a}}) + \mathbf{K}^{-1})^{-1}\right),$$

where $\mathbf{W}(\mathbf{a}) = \mathrm{diag}(\sigma(a_i)[1 - \sigma(a_i)])$, and $\hat{\mathbf{a}}$ is obtained by iterating until convergence the equation

$$\hat{\mathbf{a}}_{m+1} = \mathbf{K}(\mathbf{I} + \mathbf{W}(\hat{\mathbf{a}}_m)\mathbf{K})^{-1}(\mathbf{t}_* - \boldsymbol{\sigma}(\hat{\mathbf{a}}_m) + \mathbf{W}(\hat{\mathbf{a}}_m)\hat{\mathbf{a}}_m),$$

where $\boldsymbol{\sigma}$ consists in the entry-wise application of σ. Finally,

$$p(a|\mathbf{x}, \mathbf{x}_*, \mathbf{t}_*) \approx \mathcal{N}\left(a;\ \mu_a,\ \sigma_a^2\right),$$

where μ_a and σ_a^2 are the logit mean and the logit variance, respectively, defined as

$$\mu_a = \mathbf{k}_n^T(\mathbf{t}_* - \boldsymbol{\sigma}(\hat{\mathbf{a}})), \tag{6.C.10}$$

$$\sigma_a^2 = \kappa(\mathbf{x}, \mathbf{x}) - \mathbf{k}_n^T(\mathbf{W}(\hat{\mathbf{a}})^{-1} + \mathbf{K})^{-1}\mathbf{k}_n. \tag{6.C.11}$$

The Class 1 probability at input $\mathbf{x}$ given $\mathbf{x}_*$ and $\mathbf{t}_*$ is calculated using the inverse probit function $\sigma(a) \simeq \Phi(\lambda a)$, obtaining the approximate predictive distribution of the form

$$p(t = 1|\mathbf{x}, \mathbf{x}_*, \mathbf{t}_*) = \sigma(\mu_a(1 + \frac{\pi\sigma_a^2}{8})^{-1/2}).$$

The posterior mean of the GPC in the logit space, μ_a, gives us the expected value of the latent function at point $\mathbf{x}$. By applying the sigmoid function, we transform it into the probability space. This transformed value, $\mu = \sigma(\mu_a)$ represents the posterior mean probability. Finally, the variance in the probability space is given by

$$\sigma^2 = \sigma_a^2 \, \mu^2(1 - \mu)^2,$$

which is obtained using the delta method [97, p. 240], as discussed in the next section.

6.D Delta Method for cross-covariance term

First, let $\mu(\mathbf{x})$ and $\sigma^2(\mathbf{x})$ be the posterior mean and variance of the GP model at point $\mathbf{x}$, and similarly, let $\mu(\mathbf{x}^*)$ and $\sigma^2(\mathbf{x}^*)$ be the mean and variance at point x^*. The covariance between $\mathbf{x}$ and $\mathbf{x}^*$ is denoted by $\text{Cov}(\mathbf{x}, \mathbf{x}^*)$. We can represent the joint distribution of $\hat{f}(\mathbf{x})$ and $\hat{f}(\mathbf{x}^*)$ as

$$\begin{bmatrix} \hat{f}(\mathbf{x}) \\ \hat{f}(\mathbf{x}^*) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu(\mathbf{x}) \\ \mu(\mathbf{x}^*) \end{bmatrix}, \begin{bmatrix} \sigma^2(\mathbf{x}) & \text{Cov}(\mathbf{x}, \mathbf{x}^*) \\ \text{Cov}(\mathbf{x}, \mathbf{x}^*) & \sigma^2(\mathbf{x}^*) \end{bmatrix} \right).$$

To generate correlated samples, we perform the Cholesky decomposition on the covariance matrix

$$\begin{bmatrix} \sigma^2(\mathbf{x}) & \text{Cov}(\mathbf{x}, \mathbf{x}^*) \\ \text{Cov}(\mathbf{x}, \mathbf{x}^*) & \sigma^2(\mathbf{x}^*) \end{bmatrix} = \mathbf{L}\mathbf{L}^T,$$

where $\mathbf{L}$ is a lower triangular matrix.

To calculate the covariance between $\mathbf{x}$ and $\mathbf{x}^*$, we apply the delta method which uses the first-order Taylor expansion to approximate the expectation of a function of random variables, in particular, of $g(f(\mathbf{x}))$ and $g(f(\mathbf{x}^*))$, where g is a nonlinear function. Using the first-order Taylor expansion around the logit means $\mu_a(\mathbf{x})$ and $\mu_a(\mathbf{x}^*)$ (cf. (6.C.10)) yields

$$\begin{aligned} g(f(\mathbf{x})) &\approx g(\mu_a(\mathbf{x})) + g'(\mu_a(\mathbf{x}))(\hat{f}(\mathbf{x}) - \mu_a(\mathbf{x})), \\ g(f(\mathbf{x}^*)) &\approx g(\mu_a(\mathbf{x}^*)) + g'(\mu_a(\mathbf{x}^*))(\hat{f}(\mathbf{x}^*) - \mu_a(\mathbf{x}^*)). \end{aligned}$$

The covariance is then approximated by

$$\begin{aligned} &\text{Cov}(g(\hat{f}(\mathbf{x})), g(\hat{f}(\mathbf{x}^*))) \\ &\quad \approx \text{Cov}(g(\mu_a(\mathbf{x})) + g'(\mu_a(\mathbf{x}))(\hat{f}(\mathbf{x}) - \mu_a(\mathbf{x})), \\ &\qquad\quad g(\mu_a(\mathbf{x}^*)) + g'(\mu_a(\mathbf{x}^*))(\hat{f}(\mathbf{x}^*) - \mu_a(\mathbf{x}^*))) \\ &\quad = g'(\mu_a(\mathbf{x}))g'(\mu_a(\mathbf{x}^*)) \, \text{Cov}(\hat{f}(\mathbf{x}), \hat{f}(\mathbf{x}^*)), \end{aligned}$$

where $\text{Cov}((\hat{f}(\mathbf{x}), \hat{f}(\mathbf{x}^*))$ is calculated using (6.C.11) as

$$\text{Cov}\Big(\hat{f}(\mathbf{x}), \hat{f}(\mathbf{x}^*)\Big) = \kappa(\mathbf{x}, \mathbf{x}^*) - \kappa(\mathbf{x}_{1:n}, \mathbf{x})^T \left(\mathbf{W}(\hat{\mathbf{a}})^{-1} + \mathbf{K}\right)^{-1} \kappa(\mathbf{x}_{1:n}, \mathbf{x}^*).$$

For GPC, $g(z)$ is represented by the sigmoid function σ, converting logits into probabilities. Consequently, the derivatives are given by $g'(z) = \sigma(z)(1 - \sigma(z))$. Finally,

$$\begin{aligned}
&\text{Cov}(\sigma(\hat{f}(\mathbf{x})), \sigma(\hat{f}(\mathbf{x}^*))) \\
&\approx \mu_x(1 - \mu_x)\mu_{x^*}(1 - \mu_{x^*}) \, \text{Cov}(\hat{f}(\mathbf{x}), \hat{f}(\mathbf{x}^*)) \\
&\approx \text{Cov}(\mathbf{x}, \mathbf{x}^*).
\end{aligned}$$

We then draw samples from standard normal distributions $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and transform them into correlated samples:

$$\begin{bmatrix} \hat{f}(\mathbf{x}) \\ \hat{f}(\mathbf{x}^*) \end{bmatrix} = \begin{bmatrix} \mu(\mathbf{x}) \\ \mu(\mathbf{x}^*) \end{bmatrix} + \mathbf{L}\mathbf{z}.$$

6.E Branch and Bound Example

The ACE algorithm employs a hybrid optimization strategy to handle both continuous and categorical variables. For continuous features, we apply the quasi-Newton L-BFGS-B algorithm to maximize the acquisition function, whereas for categorical variables, which are inherently discrete, ACE employs a Branch and Bound (B&B) strategy.

Categorical features are encoded using ordinal or label encoding [104], depending on whether they exhibit a natural order. Starting from the solution obtained via continuous optimization (the *root node*), the B&B method systematically partitions the categorical space into smaller subproblems by introducing integer constraints. At each branch, the algorithm runs L-BFGS-B over the remaining continuous variables while fixing categorical ones, selecting the candidate that maximizes the acquisition function. For a visual example, see Appendix E.

Figure 6.E.4 illustrates the Branch and Bound algorithm for two categorical variables. Each node represents a constrained maximization, and the yellow-highlighted box corresponds to the best candidate found ($J(x) = 18.0$, $x_1 = 2.0$, $x_2 = 2.0$). This structured refinement ensures a principled and efficient exploration of the solution space, enabling the identification of near-global optima in problems involving discrete variables and non-convex structures, while effectively pruning suboptimal branches.

6.F Datasets

The quantitative evaluation is conducted on eight real-world classification datasets sourced from OpenML [106], Kaggle [107], and the UCI ML Repository [108]. The selected datasets largely align with those used in the original evaluations of the

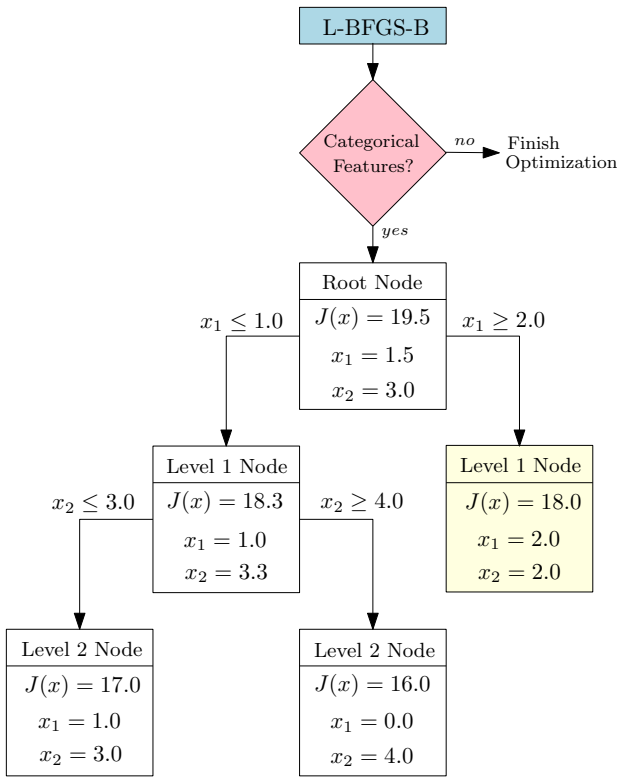


Figure 6.E.4: Branch and Bound method with L-BFGS-B Initialization.

compared methods. Growing Spheres was evaluated on MNIST (in the original paper) and on Make Moons (illustrative examples provided in its public repository). For the eight benchmark datasets, we included those used in the MOC and BayCon studies, which also compared against each other, and added Growing Spheres to enable both qualitative and quantitative comparisons. For OmniXAI, the MNIST dataset was included to complement the visual comparison with Growing Spheres.

The websites corresponding to the datasets used in Section 6.4 are the following: Diabetes · Breast · Blood · KC2 · Tic-Tac-Toe · Nursery · CMC · German Credit

6.F.1 Hyperparameter Search

An exhaustive grid search is performed over the ranges in Table 6.F.6. Continuous ranges ("to") are sampled at regular intervals, and bracketed values denote discrete candidate sets. Final values are selected from the subrange where further changes have negligible impact on the generated CFE, ensuring robustness and efficiency.

Table 6.F.6: Hyperparameter ranges explored.

Name	Symbol	Range Tried
Initial Penalty	λ_0	2 to 12
Kernel (ν)	κ	Matérn $[\frac{1}{2}, \frac{3}{2}, \frac{5}{2}]$
Monte-Carlo Samples	MC	800 to 2000
Penalty Growth	p	1.1 to 1.9
Sobol Samples	SS	1000 to 10000
Sparsity Trade-off	β	$[0.1, 1, 3, 5, 7, 10, 100]$

Notably, a key design choice is the selection of the kernel function for the Gaussian Process surrogate, which fundamentally shapes the quality of the acquisition strategy in Bayesian optimization. We employ the Matérn 5/2 kernel, which models C^2 functions—unlike the RBF kernel, which assumes infinite differentiability, or the linear kernel, which fails to capture non-linear structure—making it better suited to model complex and irregular decision boundaries in high-dimensional settings [92, Sec. 4.2.1].

As noted above, we chose hyperparameters so that small perturbations would not materially change the resulting CFE, ensuring stable convergence within a well-defined region; the final values used in our experiments are reported in Table 6.F.7. The algorithm starts with an initial penalty value, λ_0, in the vicinity of 1—set to 10 in our runs—ensuring that the search begins near the instance to be explained and helps the algorithm efficiently find a nearby minimizer. After each acquisition function maximization, λ is scaled by $p = 1.5$, gradually increasing to guide the search toward the decision boundary. Sobol sampling ($SS = 8000$) is utilized to densely cover high-dimensional spaces, ensuring thorough exploration

across all features in the dataset, while Monte Carlo sampling ($MC = 1000$) provides an efficient approximation of the Expected Improvement (EI) integral, balancing computational cost and precision.

Table 6.F.7: Summary of Hyperparameters.

Name	Symbol	Value
Initial Penalty	λ_0	10
Maximum Penalty	λ_{max}	1e15
Kernel	κ	Matern $\frac{5}{2}$
Monte-Carlo Samples	MC	1000
Convergence Tolerance	ϵ	0.001
Penalty Growth	p	1.5
Sobol Samples	SS	8000
Sparsity trade-off parameter	β	5

6.F.2 Computing Infrastructure

All experiments are conducted on a system with an **AMD Ryzen 7 4800HS CPU @ 2.90 GHz**, **16 GB RAM**, running **Windows 11 Home Single Language**, with computation performed on CPU only and no GPU acceleration (integrated or discrete) used. The implementation is in **Python 3.12.7** with the following key libraries: NumPy (v1.26.4), SciPy (v1.13.1), scikit-learn (v1.5.1), pandas (v2.2.2), matplotlib (v3.9.2), scikit-image (v0. 24.0), and threadpoolctl (v3.5.0).

6.F.3 Quantitative Evaluation - Real-World datasets

Complete results for all eight real-world datasets—using two randomly selected labels per dataset—are reported in Tables 6.F.8, 6.F.9, 6.F.10 and 6.F.11. Distances for categorical features use consistent encodings (label or ordinal), and identical random seeds ensure reproducibility. Non-actionable features—such as age or gender— are held fixed throughout the optimization to guarante feasibility and interpretability

ACE is initialized with $n_0 = 30$ points for all datasets, except for the low-dimensional Blood Test dataset (4 features), where $n_0 = 15$ is used. These initial samples are included in the total evaluation count. Across all eight datasets, ACE consistently outperforms competing methods in terms of the number of black-box evaluations ($h_{\#}$), successfully generating counterfactuals in every experiment.

6.G System Identification

We begin by considering a noise source solely in the plant output, as in Figure 6.G.5. An approach to estimating $\{C_i(q)\}_{i=1}^{K}$ is to minimize the input prediction

Table 6.F.8: Fixed test for continuous datasets with 100 tested points per experiment. The best Score CFE values are highlighted.

Dataset	Method	Label 1						Label 2					
		$h_{\#}$ (std)	d_x (std)	g_1 (std)	$\alpha(x)$ (std)	$\mathcal{V}$ (std)	$\mathcal{S}$	$h_{\#}$ (std)	d_x (std)	g_1 (std)	$\alpha(x)$ (std)	$\mathcal{V}$ (std)	$\mathcal{S}$
Diabetes	ACE	71 (16)	34.09 (14.3)	49.57 (22.34)	1 (0.06)	1 (0)	**0.24**	70 (15)	37.10 (6.88)	70.70 (13.33)	0.97 (0.04)	1 (0)	**0.17**
	BayCon	1211 (110)	37.13 (9.23)	46.87 (12.19)	0.25 (0.08)	1 (0)	0.36	1458 (225)	52.92 (10.5)	77.56 (14.59)	0.24 (0.10)	1 (0)	0.34
	MOC	2038 (1065)	58.95 (18.46)	62.91 (16.77)	0.84 (0.19)	0.98 (0.14)	0.45	2269 (1247)	87.00 (70.22)	98.66 (82.49)	0.55 (0.16)	0.98 (0.14)	0.47
	CARLA_{GS}	28013 (4455)	5.51 (0.90)	12.46 (2.24)	0.78 (0.03)	1 (0)	0.38	68937 (7978)	13.70 (1.60)	30.20 (4.30)	0.80 (0)	1 (0)	0.38
Breast	ACE	60 (16)	9.33 (1.05)	27.89 (2.87)	1 (0.10)	1 (0)	**0.2**	78 (27)	15.05 (0.90)	40.03 (3.24)	1 (0.01)	1 (0)	0.34
	BayCon	3567 (1058)	9.60 (1.25)	21.12 (1.90)	0.31 (0.03)	0.03 (0.20)	0.35	2960 (488)	13.81 (0.76)	33.30 (4.61)	0.29 (0.03)	0.03 (0.20)	0.35
	MOC	845 (748)	10.28 (1.19)	20.81 (3.11)	0.09 (0.06)	0.97 (0.17)	0.37	372 (507)	15.65 (0.78)	29.97 (3.71)	0.52 (0.19)	0.99 (0.10)	**0.31**
	CARLA_{GS}	40883 (3545)	9.08 (0.71)	21.41 (2.25)	0.77 (0.03)	1 (0)	0.39	65703 (5583)	13.03 (1.12)	32.88 (3.33)	0.79 (0.02)	1 (0)	0.42
KC2	ACE	56 (12)	626.76 (170.35)	1436.83 (328.41)	0.90 (0.22)	1 (0)	**0.02**	57 (13)	662.05 (170.25)	1529.80 (322.74)	0.89 (0.23)	1 (0)	**0.12**
	BayCon	3715 (1498)	2448.56 (2672.30)	2843.70 (3083.79)	0.13 (0.15)	0.98 (0.14)	0.14	3575 (1177)	3441.04 (2359.45)	3988.32 (2674.31)	0.05 (0.12)	1 (0)	0.53
	MOC	457 (672)	62570.15 (165673.31)	64514.02 (167510.11)	0.02 (0.08)	0.19 (0.39)	0.62	1277 (1093)	1652.42 (1597.79)	2069.07 (1808.94)	0.09 (0.19)	0.13 (0.34)	0.42
	CARLA_{GS}	139093 (22089)	27.73 (4.42)	101.57 (18.79)	0.17 (0.02)	1 (0)	0.45	102423 (42302)	20.39 (8.45)	73.65 (30.27)	0.16 (0.02)	1 (0)	0.46
Blood	ACE	54 (15)	2035.74 (596.52)	2072.76 (595.77)	0.99 (0.12)	1 (0)	**0.14**	57 (15)	78.15 (207.30)	70.63 (207.07)	1 (0.02)	1 (0)	0.38
	BayCon	919 (159)	2017.03 (104.66)	2046.13 (107.42)	0 (0.03)	1 (0)	0.31	772 (105)	75.42 (8.98)	78.77 (10.57)	0 (0.01)	1 (0)	0.6
	MOC	1790 (970)	5999.42 (1407.57)	6044.28 (1418.61)	0.05 (0.20)	0.85 (0.36)	0.63	632 (396)	8.12 (43.56)	8.27 (43.94)	0.99 (0.05)	0.66 (0.47)	**0.15**
	CARLA_{GS}	6313 (1239)	1.15 (0.23)	1.91 (0.42)	0.64 (0.16)	1 (0)	0.4	3083 (271)	0.55 (0.06)	0.76 (0.16)	0.8 (0.03)	1 (0)	0.38

Table 6.F.9: Fixed test for heterogeneous datasets with 100 tested points per experiment. The best Score CFE values are highlighted.

Dataset	Method	Label 1						Label 2					
		$h_{\#}$ (std)	d_x (std)	g_1 (std)	$\alpha(x)$ (std)	$\mathcal{V}$ (std)	$\mathcal{S}$	$h_{\#}$ (std)	d_x (std)	g_1 (std)	$\alpha(x)$ (std)	$\mathcal{V}$ (std)	$\mathcal{S}$
CMC	ACE	65 (6)	1 (0)	1 (0)	1 (0.01)	1 (0)	**0**	63 (7)	1 (0)	1 (0)	1 (0.01)	1 (0)	**0**
	BayCon	1021 (102)	1 (0)	1 (0)	0.33 (0.02)	1 (0)	0.43	925 (46)	1 (0)	1 (0)	0.29 (0.04)	1 (0)	0.41
	MOC	590 (212)	1 (0.1)	1 (0.2)	0.9 (0.1)	1 (0)	0.2	528 (184)	1 (0)	1 (0)	0.7 (0)	1 (0)	0.21
	CARLA_{GS}	1005 (0)	3.87 (0)	7 (0)	1 (0)	1 (0)	0.74	1005 (0)	4.24 (0)	8 (0)	1 (0)	1 (0)	0.75
Nursery	ACE	57 (7)	1 (0.04)	1.01 (0.1)	1 (0)	1 (0)	**0**	60 (8)	1 (0)	1 (0)	1 (0.01)	1 (0)	**0**
	BayCon	836 (29)	1 (0)	1 (0)	0.37 (0.01)	1 (0)	0.37	822 (29)	1 (0)	1 (0)	0.37 (0.01)	1 (0)	0.36
	MOC	220 (0)	- (-)	- (-)	- (-)	0 (0)	-	220 (0)	- (-)	- (-)	- (-)	0 (0)	-
	CARLA_{GS}	1005 (0)	3.32 (0)	5 (0)	1 (0)	1 (0)	0.75	1005 (0)	3.32 (0)	5 (0)	1 (0)	1 (0)	0.75
German Credit	ACE	68 (26)	7.48 (9.05)	12.88 (10.44)	0.94 (0.08)	1 (0)	**0.01**	54 (10)	6.11 (2.82)	11.4 (4.67)	1 (0.01)	1 (0)	**0**
	BayCon	973 (98)	1 (0)	1 (0)	0.23 (0.05)	1 (0)	0.33	1013 (69)	1 (0)	1 (0)	0.23 (0.08)	1 (0)	0.3
	MOC	1436 (753)	4159.51 (1039.12)	4200.26 (1047.8)	0.5 (0.14)	0.35 (0.48)	0.89	1739 (702)	3993.84 (473.19)	4013.8 (472.05)	0.26 (0.06)	0.05 (0.22)	0.96
	CARLA_{GS}	1005 (0)	2.13 (0.27)	3.49 (0.5)	1 (0.01)	1 (0)	0.24	1005 (0)	1.41 (0)	2.01 (0.01)	0.99 (0.07)	1 (0)	0.2
Tic-Tac-Toe	ACE	67 (11)	1 (0.01)	1 (0.03)	1 (0.01)	1 (0)	**0**	61 (9)	1 (0.01)	1 (0.02)	1 (0.01)	1 (0)	**0**
	BayCon	1526 (274)	1.41 (0)	2 (0)	0.27 (0.02)	1 (0)	0.44	930 (42)	1 (0)	1 (0)	0.28 (0.02)	1 (0)	0.44
	MOC	304 (71)	1.47 (0.17)	2.19 (0.59)	0.25 (0.03)	0.21 (0.41)	0.59	289 (81)	1.73 (0)	3 (0)	0.28 (0.06)	0.02 (0.14)	0.7
	CARLA_{GS}	890111 (312576)	1.22 (0.25)	1.55 (0.66)	1 (0)	0.11 (0.31)	0.65	1000001 (0)	- (-)	- (-)	- (-)	0 (0)	-

error:

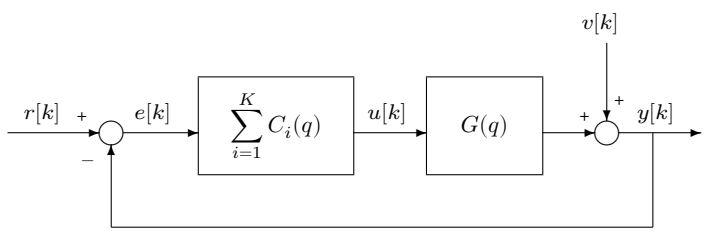


Figure 6.G.5: Closed-loop system setting, noiseless input, noisy output case.

$$\hat{\beta}_{\text{LS}} = \arg\min_{\beta} \frac{1}{N} \sum_{k=1}^{N} \left(u[k] - \sum_{i=1}^{K} C_i(q, \theta_i) e[k] \right)^2, \tag{6.G.12}$$

where $e[k] := y[k] - r[k]$, and where we have made explicit the dependence on θ_i of $C_i(q)$. Under technical conditions on reference and noise stationarity and closed-loop stability [109], the cost being minimized in (6.G.12) converges uniformly to its expected value on a compact set of parameters β. Thus, $\hat{\beta}_{\text{LS}}$ converges with

Table 6.F.10: Mixed test for continuous datasets with 100 tested points. The best Score CFE values are highlighted.

Dataset	Method	$h_{\#}$ (std)	d_{2_N} (std)	g_{1_N} (std)	α (std)	$\mathcal{V}$ (std)	$\mathcal{S}$
Diabetes	ACE	72 (18)	2.46 (0.84)	2.7 (1.87)	0.98 (0.09)	1 (0)	**0.1**
	BayCon	1239 (238)	2.04 (0.97)	2.99 (1.89)	0.27 (0.09)	1 (0)	0.14
	MOC	1753 (1011)	2.09 (0.91)	2.62 (1.34)	0.58 (0.36)	0.91 (0.29)	**0.1**
	CARLA$_{GS}$	27463 (23054)	3.2 (2.66)	4.26 (3.41)	0.74 (0.08)	1 (0)	0.78
Breast	ACE	67 (22)	2.45 (1.37)	7.6 (3.95)	1 (0.01)	1 (0)	**0.16**
	BayCon	2045 (952)	2.42 (1.25)	4.25 (2.63)	0.30 (0.08)	0.60 (0.49)	**0.16**
	MOC	890 (804)	3.36 (1.19)	5.7 (2.51)	0.47 (0.33)	1 (0)	0.39
	CARLA$_{GS}$	36803 (20489)	2.62 (1.46)	6.56 (3.81)	0.76 (0.05)	1 (0)	0.54
KC2	ACE	57 (13)	4.05 (1.06)	14.37 (4.06)	0.85 (0.23)	1 (0)	**0.12**
	BayCon	3730 (1408)	2.57 (1.26)	6.43 (4.01)	0.14 (0.14)	0.97 (0.17)	0.13
	MOC	1104 (1165)	11.87 (9.46)	24.49 (22.74)	0.12 (0.28)	0.23 (0.42)	0.61
	CARLA$_{GS}$	79763 (39687)	9.59 (6.57)	15.67 (8.8)	0.18 (0.09)	1 (0)	0.72
Blood	ACE	63 (23)	1.83 (0.72)	2.9 (1.24)	0.85 (0.32)	0.99 (0.1)	0.42
	BayCon	1091 (741)	0.82 (0.65)	1.02 (0.9)	0.32 (0.1)	0.91 (0.29)	**0.16**
	MOC	504 (402)	1.8 (2.75)	2.48 (4.28)	0.37 (0.43)	0.09 (0.29)	0.56
	CARLA$_{GS}$	30633 (28509)	0.65 (0.7)	0.85 (0.87)	0.33 (0.3)	1 (0)	0.43

probability one as $N \to \infty$ to

$$\hat{\beta}_{\text{LS}} \xrightarrow{\text{w.p.1}} \hat{\beta}_{\text{LS}}^{\infty} = \arg\min_{\beta} \mathbb{E}\left\{ \left(u[k] - \sum_{i=1}^{K} C_i(q, \theta_i) e[k] \right)^2 \right\} \text{ as } N \to \infty. \tag{6.G.13}$$

Due to the input satisfying

$$u[k] = \sum_{i=1}^{K} C_i(q) e[k],$$

the cost function being minimized in (6.G.13) can reach 0, the global optimum, when $e[k]$ is persistently exciting of sufficient order. After some algebraic procedures, a short derivation yields

$$\sum_{i=1}^{K} [C_i(q) - C_i(q, \hat{\theta}_i^{\infty})] = 0,$$

that is, $C_i(q, \hat{\theta}_i^{\infty}) = C_i(q)$ for all $i = 1, \ldots, K$, or in other words, $\hat{\beta}_{\text{LS}} \xrightarrow{\text{w.p.1}} \beta$.

In summary, a consistent estimator of the controller transfer function can be obtained without knowledge of the plant dynamics. This requires solving the nonlinear least squares problem in (6.G.12), which can be done using, for example, the `oe` command in MATLAB, with $e[k] = r[k] - y[k]$ as input and $u[k]$ as output.

Table 6.F.11: Mixed test for heterogeneous datasets with 100 tested points. The best Score CFE values are highlighted.

Dataset	**Method**	$h_{\#}$ (std)	d_{2_N} (std)	g_{1_N} (std)	α (std)	$\mathcal{V}$ (std)	$\mathcal{S}$
CMC	ACE	70 (12)	2.12 (0.92)	2.99 (1.8)	1 (0.01)	0.97 (0.17)	**0.07**
	BayCon	1024 (241)	1.85 (0.71)	2.39 (1.24)	0.32 (0.05)	0.98 (0.14)	0.12
	MOC	332 (124)	1.57 (0.68)	1.8 (1.01)	0.8 (0.24)	0.18 (0.38)	0.13
	CARLA$_{GS}$	330674 (469741)	5.04 (0.9)	9.19 (1.96)	1 (0)	0.67 (0.47)	0.79
Nursery	ACE	56 (7)	1.22 (0)	1.22 (0)	1 (0.01)	1 (0)	**0**
	BayCon	840 (31)	1.22 (0)	1.22 (0)	0.37 (0)	1 (0)	0.37
	MOC	226 (10.47)	- (-)	- (-)	- (-)	0 (0)	-
	CARLA$_{GS}$	1005 (0)	2.09 (0.5)	3.4 (1.27)	1 (0)	1 (0)	0.75
German Credit	ACE	73 (24)	2.61 (1)	2.47 (2.24)	0.94 (0.2)	1 (0)	0.21
	BayCon	1006 (143)	2.01 (0.57)	2.11 (0.76)	0.23 (0.1)	0.95 (0.22)	**0.2**
	MOC	1502 (763)	2.83 (0.93)	4.35 (2.02)	0.27 (0.32)	0.4 (0.49)	0.71
	CARLA$_{GS}$	3575 (12320)	2.7 (0.83)	4.28 (1.6)	0.95 (0.16)	1 (0)	0.71

7 Data-Driven Controller Tuning: Counterfactual Search for an Axial Compressor

As discussed in the conclusions of Chapter 6, ACE opens a promising avenue for combining ideas from XAI and controller tuning, an intersection that remains relatively unexplored in the literature. Building on this perspective, this chapter applies ACE to an industrially relevant setting in which controller tuning is particularly critical: the system constitutes a process bottleneck, and its operation cannot be interrupted or driven toward unstable regimes. In this context, ACE provides a sample-efficient framework for identifying improved controller settings through only a limited number of carefully selected closed-loop experiments.

7.1 Introduction

In industrial facilities, not all units carry the same operational importance. A relatively small number of critical assets often determine whether the plant remains on-spec and online, directly affecting availability, product quality, and downstream operation. Among these assets, compressors play a central role across a wide range of industries, including natural-gas transmission and gas liquefaction, where pressure and flow must be maintained around prescribed operating conditions. In this chapter, we focus on axial compressors, which are employed in applications ranging from grid-scale gas-turbine power plants to large aero engines [110]. Their admissible operating envelope is bounded by aerodynamic phenomena such as rotating stall, a localized loss of stable flow, and surge, a global instability characterized by large oscillations in pressure and mass flow [111].

Control loops around compressors are therefore critical for maintaining stable flow and adequate pressurization; however, tuning these loops remains particularly challenging in practice [112]. Opening the loop for plant identification or relay testing

is often infeasible: the unit may constitute a process bottleneck, interruptions are costly, and aggressive excitation can drive the compressor toward undesirable operating regimes. Even during maintenance, the operating conditions experienced by the unit are rarely representative of normal production. Consequently, controller adjustments must often be performed from limited prior information and through short closed-loop experiments.

A natural source of such prior information is the process historian associated with the Distributed Control System (DCS) or Programmable Logic Controller (PLC) [113]. These databases typically contain long-term records of the process variable (PV), setpoint (SP), controller output (OP), and operating mode, providing a valuable basis for characterizing normal operation, selecting representative operating windows, and benchmarking closed-loop performance. Two practical difficulties nevertheless arise. First, numerical controller parameters, such as PID gains or lead–lag time constants, are not always recorded and may therefore need to be reconstructed through a parameter-identification procedure. Second, historical segments must be representative of approximately time-invariant plant conditions; otherwise, observed changes may reflect plant drift rather than controller action, biasing identification and tuning. These considerations motivate the combination of historian-informed prior knowledge with a small number of carefully selected closed-loop tests around the current operating point.

Within this landscape, the axial-compressor pressure loop, regulated via Inlet Guide Vanes (IGVs)—adjustable inlet blades that modulate mass flow and incidence—offers an ideal testbed. The compressor's unsteady behavior near healthy operating points admits compact physics–based reductions such as the Moore-Greitzer compressor-plenum-throttle model [114], while under correct operation and standard maintenance the hardware envelope remains remarkably stable over multi-year horizons [115, p. 29], making DCS historian records comparable across long spans. Retuning, however, remains strongly constrained by unit availability and the need to avoid operating conditions associated with surge, making long or aggressive excitation undesirable.

In this chapter, we address the controller retuning dilemma by formulating it as a counterfactual explanation problem and applying ACE, introduced in Chapter 6, to identify small and operator-actionable modifications of the controller parameters that achieve prescribed time-domain closed-loop specifications. In this interpretation, controller parameters constitute the features of the black-box classifier, while a counterfactual represents a minimal parameter modification capable of changing the closed-loop outcome from unsatisfactory to satisfactory. To summarize, in this chapter

- We formulate compressor controller retuning as a historian-informed counterfactual explanation problem, seeking minimal modifications of the controller parameters that satisfy prescribed closed-loop specifications.
- We develop an ACE-based Bayesian tuning procedure that requires only a limited number of closed-loop step tests and restricts the search to a trust region around the baseline controller.
- We demonstrate the proposed approach on a compressor–plenum–throttle

system, showing that the pressure-control loop can be efficiently retuned using only a limited number of evaluations while keeping the search within an operationally relevant region and away from surge.

The remainder of this chapter is structured as follows. Section 7.2 describes the compressor system and formulates the controller-retuning problem. Section 7.3 presents the counterfactual formulation and the proposed ACE-based tuning procedure. Simulation results are presented in Section 7.4, and conclusions are drawn in Section 7.5.

7.2 Process Description and Problem Formulation

In this section, we present the canonical air system, an axial compressor discharging into a downstream plenum, and the closed-loop architecture used for controller retuning.

7.2.1 Physical Setting

In industrial process and turbomachinery applications, compressors raise total pressure to deliver the target mass flow against downstream demand. In axial compressors, rotor–stator rows add work and guide the flow, while IGVs set inlet pre-swirl to adjust incidence and mass flow at a fixed shaft speed, shifting the operating point along the compressor characteristic ψ_c (the quasi-steady map of pressure-rise coefficient versus flow coefficient at fixed speed and IGV setting).

Within this framework, we consider an axial compressor installed in a duct and discharging into a downstream plenum (Fig. 7.2.1). The plenum volume is large relative to the compressor duct; hence, flow speeds and fluid accelerations in the plenum are negligible, and the plenum pressure p_S can be treated as spatially uniform, albeit time-varying. We denote by p_T the upstream (ambient/atmospheric) pressure. A throttle at the plenum exit represents the downstream consumer (e.g., a gas turbine) and imposes a linear pressure–flow slope k_T. The compressor IGVs are actuated by a servo valve; by modulating the IGV position u, the operator sets the pressure rise that sustains the downstream unit. This lumped description is standard in the Moore–Greitzer model [114] and captures the key interactions of the compressor–plenum–throttle system.

A key operational constraint is *surge*: if plenum pressure rises excessively—*i.e.*, the atmosphere-to-plenum pressure rise Ψ increases beyond the compressor's capability—the annulus-averaged flow may reverse, driving pressure oscillations that threaten both blades and downstream hardware. Our study focuses on a healthy operating region—far from surge—while acknowledging that pressure excursions can move the system toward this boundary. In industrial practice, the hardware and its operating envelope remain remarkably stable over multi-year periods, thus archived process data provide a suitable prior information for identification and controller retuning.

Under the Moore-Greitzer assumptions—axisymmetric, annulus-averaged flow; a

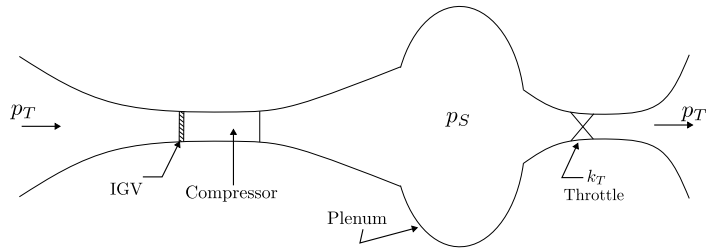


Figure 7.2.1: Schematic of the compressor system geometry.

quasi-steady compressor characteristic $\psi_c(\Phi)$ at fixed shaft speed and IGV setting; and the physical setting of Fig. 7.2.1—the compressor–plenum system reduces to a second-order system for the annulus-averaged axial-flow coefficient Φ:

$$\frac{d^2\Phi}{d\xi^2} + \frac{1}{\ell_c}\left(\frac{1}{4B^2 k_T} - \frac{d\psi_c}{d\Phi}\right)\frac{d\Phi}{d\xi} + \frac{1}{4\ell_c^2 B^2}\left[(\Phi - \bar{\Phi}) - \frac{\Psi - \bar{\Psi}}{k_T}\right] = 0, \tag{7.2.1}$$

where ψ_c is the compressor characteristic, Ψ the pressure rise coefficient proportional to $p_S - p_T$ (plenum relative to atmosphere), $\xi = Ut/R$ the nondimensional time with U the rotor peripheral speed at the mean radius R, and u the IGV position (per unit). The constants are $\ell_c = L_c/R$ (duct-length parameter), B (Greitzer parameter), and k_T (throttle slope). All variables are nondimensional, and overbars (e.g., $\bar{\Phi}$, $\bar{\Psi}$) denote operating-point values. Equation (7.2.1) is the well-known Moore–Greitzer pure-surge reduction, highlighting how B influences both the damping and stiffness of the lumped dynamics.

7.2.2 IGV Control

The pressure-rise from atmosphere to plenum must be controlled in a feedback loop. To this end, starting from the Moore-Greitzer dynamics in (7.2.1), a small-signal model is derived around a healthy operating point under IGV actuation. Linearizing around $(\bar{\Phi}, \bar{u}, \bar{\Psi})$ far from surge and defining $\tilde{\Phi} := \Phi - \bar{\Phi}$, $\tilde{u} := u - \bar{u}$, and $\tilde{\Psi} := \Psi - \bar{\Psi}$, the compressor map is locally

$$\psi_c(\Phi, u) \;\approx\; \psi_c(\bar{\Phi}, \bar{u}) \;+\; a\,\tilde{\Phi} \;+\; \alpha\,\tilde{u}, \tag{7.2.2}$$

with $a = \frac{\partial \psi_c}{\partial \Phi}\big|_{(\bar{\Phi},\bar{u})}$, $\alpha = \frac{\partial \psi_c}{\partial u}\big|_{(\bar{\Phi},\bar{u})}$. The linearized momentum equation in [114, Chap. 4, Eq. (4.2)] yields the small-signal relation

$$\tilde{\Psi} = a\,\tilde{\Phi} + \alpha\,\tilde{u} - \ell_c\,\frac{d\tilde{\Phi}}{d\xi}. \tag{7.2.3}$$

Combining (7.2.1) with (7.2.2)–(7.2.3) and taking Laplace transforms gives a second-order transfer function $G_1(s)$ from IGV to pressure rise,

$$G_1(s) = \frac{c_1\, s + c_0}{s^2 + 2\zeta\omega_n s + \omega_n^2},$$

with s being the Laplace transform variable, and ω_n, ζ, c_0, and c_1 set by $(\ell_c, B, k_T, a, \alpha)$.

On the other hand, the IGV actuator is modeled as a first-order lag with static gain k_v and time constant T_v, a standard approximation for electro–hydraulic servo valves:

$$G_2(s) \;=\; \frac{k_v}{1 + T_v s}.$$

The dynamics of the complete compressor-plenum-throttle system are controlled by a discrete-time cascaded control structure operating at the sampling frequency set by the DCS (see Fig. 7.2.2). Under this architecture the continuous-time transfer functions $G_1(s)$ and $G_2(s)$ are replaced by their discrete-time equivalents [116] $G_1^{\mathrm{d}}(q)$ and $G_2^{\mathrm{d}}(q)$, where q denotes the forward shift operator. The blocks $C_1(q)$ and $C_2(q)$ are designable linear and time-invariant controllers, and v_1 with v_2 are stationary stochastic processes that model output disturbances and sensor noise. In addition, y_1 is the pressure-related controlled variable (e.g., normalized pressure rise), and r_1 is the reference pressure rise. In practice, r_1, u_1, u_2, y_1 and y_2 are typically available from the DCS historian, whereas $v_1, v_2, G_1^{\mathrm{d}}(q)$ and $G_2^{\mathrm{d}}(q)$ are not.

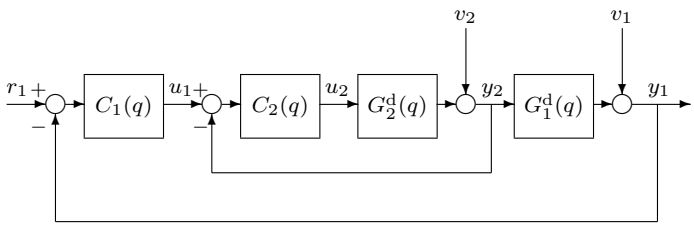


Figure 7.2.2: Cascaded system description for IGV control, with $C_1(q)$, $C_2(q)$, $G_1^{\mathrm{d}}(q)$ and $G_2^{\mathrm{d}}(q)$ being the outer controller, inner controller, compressor-plenum system, and IGV actuator, respectively.

7.2.3 Problem Formulation

This paper addresses the problem of controller retuning for critical feedback loops in axial-compressor (compressor-plenum-throttle) systems. The goal is to derive an iterative procedure that produces the smallest possible change in the baseline controller parameters while maintaining safe operation and ensuring control performance.

The controller is parameterized by a vector $\boldsymbol{\theta} \in \Theta$ (for example, PID gains), with $\boldsymbol{\theta}_0$ denoting the baseline control law. We assume the availability of N_{hist} batches of historical data $\{r_1^{(i)}[k], u_1^{(i)}[k], u_2^{(i)}[k], y_1^{(i)}[k], y_2^{(i)}[k]\}_{k=1}^{N_i}$, $i = 1, \dots, N_{\text{hist}}$, collected under distinct control laws. Since controller gains are not logged, they must be estimated from the data. The models $G_1^{\mathrm{d}}(q)$ and $G_2^{\mathrm{d}}(q)$ represent operation in a healthy region away from surge, and, because the hardware and operating envelope are remarkably stable over multi-year horizons, the plants are assumed to be time-invariant across the batches.

We seek a controller, parameterized by $\boldsymbol{\theta}$, that remains as close as possible to $\boldsymbol{\theta}_0$ under a prescribed metric d on Θ, while satisfying time-domain performance metrics such as fast response, stability, bounded overshoot, and small steady-state error. This leads to the optimization problem

$$\min_{\boldsymbol{\theta}\in\Theta} d(\boldsymbol{\theta}, \boldsymbol{\theta}_0) \quad \text{s.t.} \quad \text{performance specifications met.} \tag{7.2.4}$$

Because open-loop experiments on real compressor systems are costly and safety critical, only closed-loop data is collected, and the number of experiments must be kept minimal.

7.3 CFE Approach to Sample-Efficient Control Tuning

The objective in (7.2.4) is naturally cast as a counterfactual-explanation task on the controller space: starting from the factual $\boldsymbol{\theta}_0$ (label "fail"), and the available historical data, find the nearest counterfactual—*i.e.*, a solution to (7.2.4)—that flips the label to "pass". This formulation is well aligned with closed-loop tuning, since evaluating a candidate $\boldsymbol{\theta}$ yields a binary outcome $h(\boldsymbol{\theta}) \in \{0, 1\}$ that provides an intuitive indicator of specification compliance for retuning. In the sequel we operationalize this idea with an ACE-style, Bayesian optimization scheme that (i) leverages the DCS historian to build a prior on feasible controllers around historically feasible behavior and (ii) searches within a trust region near the current operating point, thereby proposing safe minimal-change controller updates without opening the loop.

7.3.1 Controller Identification and Performance Evaluation

A key step in the proposed approach is the collection of a labeled dataset from historical closed-loop data. Specifically, the controller parameters $\boldsymbol{\nu}_1^{(i)}$ and $\boldsymbol{\nu}_2^{(i)}$ for $C_1(q)$ and $C_2(q)$ must be estimated, given the ith batch of cascaded system data in Fig. 7.2.2, $(\mathbf{r}_1^{(i)}, \mathbf{u}_1^{(i)}, \mathbf{u}_2^{(i)}, \mathbf{y}_1^{(i)}, \mathbf{y}_2^{(i)})$, where $\mathbf{r}_1^{(i)} = \{r_1[k]\}_{k=1}^{N_i}$ and analogously for the other signals. Methods for this identification step include instrumental variables, prediction error methods, or errors-in-variable methods [117]. Since both controller input and output signals are assumed to be measured, a straightforward approach that ensures reliable estimates is to solve the corresponding nonlinear least-squares problems:

$$\boldsymbol{\nu}_1^{(i)} = \arg\min_{\boldsymbol{\nu}_1} \sum_{k=1}^{N_i} \left(u_1^{(i)}[k] - C_1(q, \boldsymbol{\nu}_1)(r_1^{(i)}[k] - y_1^{(i)}[k]) \right)^2,$$

$$\boldsymbol{\nu}_2^{(i)} = \arg\min_{\boldsymbol{\nu}_2} \sum_{k=1}^{N_i} \left(u_2^{(i)}[k] - C_2(q, \boldsymbol{\nu}_2)(u_1^{(i)}[k] - y_2^{(i)}[k]) \right)^2.$$

The quality of the identification step depends on the richness of the data and the sample size. The reference signal must be persistently exciting of sufficiently high order, with step tests designed to provide rich data according to standard input design rules [118]. In addition, in the presence of noise affecting the measurements,

the asymptotic covariance of the estimated parameters decreases proportionally to $1/N_i$ [117].

Let $\boldsymbol{\theta}^{(i)} = [\boldsymbol{\nu}_1^{(i)\top}, \boldsymbol{\nu}_2^{(i)\top}]^\top \in \mathbb{R}^p$ denote the controller parameters pertaining to the ith batch of data. Given a closed-loop step experiment, we evaluate a vector of time-domain specifications

$$\pi(\boldsymbol{\theta}^{(i)}) \;=\; (e_{\text{ss}}(\boldsymbol{\theta}^{(i)}),\, t_{\text{rise}}(\boldsymbol{\theta}^{(i)}),\, t_{\text{settle}}(\boldsymbol{\theta}^{(i)}),\, \text{OS}(\boldsymbol{\theta}^{(i)})),$$

where e_{ss} is the steady-state error, t_{rise} is the rise time, t_{settle} is the settling time for a fixed amplitude margin, and OS is the overshoot percent. Given $\pi(\boldsymbol{\theta}^{(i)})$ for each $i = 1, \dots, N_{\text{hist}}$, we define the binary labels

$$h(\boldsymbol{\theta}^{(i)}) \;=\; \mathbb{I}\{\pi(\boldsymbol{\theta}^{(i)}) \preceq \tau\}, \quad i = 1, \dots, N_{\text{hist}},$$

where $\tau = (\tau_e,\, \tau_{\text{rise}},\, \tau_{\text{settle}},\, \tau_{\text{OS}})$ describes the performance specifications for each metric, and $\preceq$ denotes componentwise inequality, *i.e.*, each component of $\pi(\boldsymbol{\theta}^{(i)})$ must be no greater than the corresponding component of τ. Note that $h(\boldsymbol{\theta}^{(i)})$ equals 1 if and only if all specifications are satisfied. The oracle $h(\cdot)$ is a black box (obtained via closed-loop simulation/operation) and costly to query; safety further requires that queries remain near historically observed, feasible behavior, since available closed-loop data (collected under healthy operation, away from surge) define the admissible operating envelope, *i.e.*, the range of operating conditions and controller settings observed to be safe. All historical data is thus condensed in the labeled dataset $\mathcal{D}_{\text{hist}} = \{(\boldsymbol{\theta}^{(i)}, \ell_i)\}_{i=1}^{N_{\text{hist}}}$, where $\ell_i = h(\boldsymbol{\theta}^{(i)}) \in \{0, 1\}$.

7.3.2 Counterfactual Explainability Step

Having identified controllers from historical data and assembled a labeled dataset $\mathcal{D}_{\text{hist}}$, we now use this information to construct the adaptive sampling procedure for counterfactual explanations. Given a current controller $\boldsymbol{\theta}_0$ that fails the time-domain specifications (*i.e.*, $h(\boldsymbol{\theta}_0) = 0$), the goal is to find a new controller $\boldsymbol{\theta}$ that satisfies the specifications while remaining as close as possible to $\boldsymbol{\theta}_0$.

Following the Gaussian process classification method [119], we rewrite the pass/ fail rule from Section 7.3.1 as

$$h(\boldsymbol{\theta}) = \mathbb{H}(f(\boldsymbol{\theta}) - 0.5),$$

where $\mathbb{H}\colon \mathbb{R} \to \{0, 1\}$ is the Heaviside step function (*i.e.*, $\mathbb{H}(a) = 1$ if $a \geq 0$ and 0 otherwise), and $f\colon \Theta \to \mathbb{R}$ is a smooth latent function. We refer to f as the *black-box target function*, where the latent score $f(\boldsymbol{\theta})$ summarizes how the closed-loop step response under controller $\boldsymbol{\theta}$ compares against the time-domain specifications: values above 0.5 indicate "pass" and below 0.5 indicate "fail".

At a high level, using historical data we obtain prior knowledge over feasible behavior, then adaptively propose new controllers near the data manifold and the current setting $\boldsymbol{\theta}_0$. Each proposal triggers a short closed-loop step test; we record the pass/fail label and restrict subsequent queries to a trust region to keep trials safe and minimal. Thus, the historian anchors the prior, and ACE's acquisition steers a few

targeted experiments toward the nearest label-flipping retuning that meets the specifications.

In what follows, we instantiate ACE for compressor–IGV tuning, following the guidelines presented in Chapter 6.

Cost Function

The ACE method interprets the decision boundary as the set $\{\, \boldsymbol{\theta} \in \Theta : \, f(\boldsymbol{\theta}) = 0.5 \,\}$, *i.e.*, where pass and fail are equally likely. Hence, the counterfactual problem in (7.2.4) can be re-written as

$$\min_{\boldsymbol{\theta}} d(\boldsymbol{\theta}, \boldsymbol{\theta}_0) \quad \text{s.t. } f(\boldsymbol{\theta}) = 0.5. \tag{7.3.5}$$

Following (6.3.4), we adopt a penalty formulation to obtain an unconstrained program:

$$\min_{\boldsymbol{\theta}} \; J(\boldsymbol{\theta}) \;=\; d(\boldsymbol{\theta}, \boldsymbol{\theta}_0) \;+\; \lambda_k |f(\boldsymbol{\theta}) - 0.5|, \tag{7.3.6}$$

where $\lambda_k > 0$ can be increased across iterations $k \in \mathbb{N}$ (denoting the ACE iteration index, cf. Algorithm 7.1 in Section 7.3.3) to tighten the boundary constraint. The terms $d(\boldsymbol{\theta}, \boldsymbol{\theta}_0)$ and $|f(\boldsymbol{\theta}) - 0.5|$ jointly enforce *proximity*: the first term keeps the candidate controller close to the current one in the parameter space, while the second one drives queries toward the decision boundary. Enforcing the boundary yields label changes with minimal parameter movement, consistent with the counterfactual goal.

To encode additional structural preferences, we add three terms—*actionability* (search only over operator–changeable parameters), *sparsity* (alter as few parameters as possible), and *plausibility* (prefer controllers supported by historical data):

$$\underbrace{\arg\min_{\boldsymbol{\theta} \in \mathcal{A} \subset \Theta}}_{\text{actionability}} \; \underbrace{d(\boldsymbol{\theta}, \boldsymbol{\theta}_0) + \lambda_k \left| f(\boldsymbol{\theta}) - 0.5 \right|}_{\text{proximity}} \;+\; \underbrace{\beta\, g(\boldsymbol{\theta} - \boldsymbol{\theta}_0)}_{\text{sparsity}} \;+\; \underbrace{l(\boldsymbol{\theta};\, \mathcal{D})}_{\text{plausibility}} \tag{7.3.7}$$

where: (i) $\mathcal{A} \subset \Theta$ enforces *actionability* by fixing non-actionable parameters (e.g., keeping the derivative gain in a PID controller unchanged when only retuning the remaining parameters); (ii) $g(\boldsymbol{\theta} - \boldsymbol{\theta}_0)$ promotes *sparsity* of changes (in this work, an ℓ_1 norm); (iii) $l(\boldsymbol{\theta}; \mathcal{D})$ is a *hard plausibility filter* implemented via the Local Outlier Factor (LOF) [94], assigning zero cost to inliers and discarding outliers far from the empirical data manifold, computed over $\mathcal{D} = \mathcal{D}_{\text{hist}} \cup \mathcal{D}_{\text{online}}$ (historical DCS data plus newly collected closed-loop samples). As in (7.3.6), the penalty method increases λ_k across iterations; $\beta > 0$ trades off sparsity against proximity and plausibility, and $d(\cdot, \cdot)$ is the Euclidean distance.

To this end, we define the *trust region* as the set of admissible candidates explored by ACE around $\boldsymbol{\theta}_0$, constrained to remain within historically observed feasible behavior (initial and newly collected healthy data) and reinforced by the plausibility filter $l(\boldsymbol{\theta}; \mathcal{D})$ in (7.3.7).

Surrogate Model

Recall that f is unknown and we only observe a binary label $\ell = h(\boldsymbol{\theta}) \in \{0, 1\}$. Therefore, to solve (7.3.7) we approximate the true underlying function f with a surrogate $\hat{f}$. In our setting, $\hat{f}$ is a Gaussian Process Classifier (GPC), a non-parametric model capable of capturing complex relationships in the data [96, Sec. 6.4], that returns the estimated probability of Class 1, *i.e.*, passing the time-domain specifications, so that $\hat{f}(\boldsymbol{\theta}) \in [0, 1]$.

The Class 1 probability at new input $\boldsymbol{\theta}$ given observed $\boldsymbol{\theta}_{\mathcal{D}}$ with labels $\ell_{\mathcal{D}}$ is calculated by

$$p(\ell = 1|\boldsymbol{\theta}, \boldsymbol{\theta}_{\mathcal{D}}, \ell_{\mathcal{D}}) = \sigma(\mu_a(1 + \pi\sigma_a^2/8)^{-1/2}),$$

where $\sigma(a) \simeq \Phi(\lambda a)$ is the inverse probit function, while μ_a and σ_a^2 denote the posterior mean and variance of the latent variable at $\boldsymbol{\theta}$ under the Laplace approximation [119, Sec. 3.4], using data $\mathcal{D}_{\text{hist}}$ from the historian together with newly collected experiments, $\mathcal{D}_{\text{online}}$.

In short, $\hat{f}(\boldsymbol{\theta})$ provides the score we use to prioritize candidate controllers. The next step details how we select safe experiments via an acquisition function within a data-supported neighborhood around historical operation.

Smart Sampling: Expected Improvement

Given the previous observations and the constructed surrogate model, we adaptively pick new parameters within the trust region defined above to keep trials as minimal as possible. To this end, we use *Expected Improvement (EI)*, a Bayesian optimization acquisition function that selects the most informative sampling point.

EI measures the expected reduction of the objective relative to the current best at a new point $\boldsymbol{\theta} \in \Theta$, *i.e.*,

$$\begin{aligned} \text{EI}_n(\boldsymbol{\theta}) :&= \mathbb{E}_n\left\{[J_n^* - J(\boldsymbol{\theta})]_+\right\} \\ &= \mathbb{E}_n\big\{[d(\boldsymbol{\theta}^*, \boldsymbol{\theta}_0) + \lambda|\hat{f}(\boldsymbol{\theta}^*) - 0.5| + \Omega(\boldsymbol{\theta}^*) \\ &\qquad - d(\boldsymbol{\theta}, \boldsymbol{\theta}_0) - \lambda|\hat{f}(\boldsymbol{\theta}) - 0.5| - \Omega(\boldsymbol{\theta})]_+\big\}, \end{aligned} \tag{7.3.8}$$

where $\mathbb{E}_n$ denotes expectation conditioned on the previously observed data, $\Omega(\cdot)$ represents the sparsity and plausibility terms in (7.3.7), $[a]_+ := \max\{a, 0\}$ and $\boldsymbol{\theta}^* = \text{argmin}_{\boldsymbol{\theta}_i \in \boldsymbol{\theta}_{1:n}} J(\boldsymbol{\theta}_i)$, is the cost function minimizer among the previously observed inputs.

To evaluate (7.3.8), ACE uses a Monte Carlo scheme that draws samples from the GPC-based model and approximates EI_n by averaging over those samples. Finally, we approximate the maximizer of EI_n with a general-purpose optimizer such as L-BFGS-B [98]. Each EI proposal $\boldsymbol{\theta}_k$ is validated by a short closed-loop step test, after which we assign the pass/fail label $\ell_k = h(\boldsymbol{\theta}_k) \in \{0, 1\}$. The online dataset is then updated, *i.e.*, $\mathcal{D}_{\text{online}} \leftarrow \mathcal{D}_{\text{online}} \cup \{(\boldsymbol{\theta}_k, \ell_k)\}$, and the classifier $\hat{f}$ is retrained on the combined dataset $\mathcal{D} = \mathcal{D}_{\text{hist}} \cup \mathcal{D}_{\text{online}}$.

Algorithm 7.1 ACE-Based Controller Fine-Tuning

Input: Baseline controller parameters $\boldsymbol{\theta}_0$, historian archive $\mathcal{H} = \left\{\left(\mathbf{r}_1^{(i)}, \mathbf{u}_1^{(i)}, \mathbf{u}_2^{(i)}, \mathbf{y}_1^{(i)}, \mathbf{y}_2^{(i)}\right)\right\}_{i=1}^{N_{\text{hist}}}$
Parameters: λ_0 (init. penalty), $\lambda_{\max}$ (max penalty), ϵ (tolerance), p (penalty growth)
Output: CFE $\boldsymbol{\theta}_s^n$

$\{\boldsymbol{\theta}^{(i)}\}_{i=1}^{N_{\text{hist}}} \leftarrow$ IdentifyFromDCSHistorian$(\mathcal{H})$
$\{\ell_i\}_{i=1}^{N_{\text{hist}}} \leftarrow$ ClosedLoopMetrics$(\mathcal{H})$
$\mathcal{D} \leftarrow \{(\boldsymbol{\theta}^{(i)}, \ell_i)\}_{i=1}^{N_{\text{hist}}}$
$\hat{f} \leftarrow$ GPCFit$(\mathcal{D})$
while $h(\boldsymbol{\theta}_0) = h(\boldsymbol{\theta}_s^n) \wedge \|\boldsymbol{\theta}_s^n - \boldsymbol{\theta}_0\| < \|\boldsymbol{\theta}_s^o - \boldsymbol{\theta}_0\|$ **do**
  $\boldsymbol{\theta}_s^o \leftarrow \boldsymbol{\theta}_s^n, \quad \lambda_k \leftarrow \lambda_0$
  while $\|\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k-1}\| > \epsilon \vee \lambda_k < \lambda_{\max}$ **do**
    $\boldsymbol{\theta}_k \leftarrow$ ExpectedImprovement$(\mathcal{D}, \hat{f}, \boldsymbol{\theta}_0, \lambda_k)$
    $\ell_k \leftarrow$ ClosedLoopStepTest$(\boldsymbol{\theta}_k)$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\boldsymbol{\theta}_k, \ell_k)\}$
    $\hat{f} \leftarrow$ GPCUpdate$(\mathcal{D})$
    $k \leftarrow k + 1, \quad \lambda_k \leftarrow (\lambda_{k-1})^p$
  end while
  $\boldsymbol{\theta}_s^n \leftarrow$ SampleDecisionBoundary$(\mathcal{D}, \hat{f}, \boldsymbol{\theta}_0)$
end while
return $\boldsymbol{\theta}_s^n$

7.3.3 Algorithm

We present the full sample-efficient controller retuning algorithm based on Bayesian optimization and counterfactual explanations in Algorithm 7.1. The method is initialized from the DCS archive: IdentifyFromDCSHistorian identifies controller parameters, ClosedLoopMetrics assigns pass/fail labels, and a GPC prior $\hat{f}$ is fitted to the resulting dataset $\{(\boldsymbol{\theta}^{(i)}, \ell_i)\}_{i=1}^{N_{\text{hist}}}$, where $\ell_i = h(\boldsymbol{\theta}^{(i)})$. Then ExpectedImprovement proposes actionable, trust-region candidates; each candidate is evaluated via a short closed-loop step test (ClosedLoopStepTest), after which the dataset and $\hat{f}$ are updated, and the boundary penalty parameter λ_k in (7.3.7) is increased to steer queries toward $f(\boldsymbol{\theta}) = 0.5$. Finally, SampleDecisionBoundary selects a counterfactual near $f(\boldsymbol{\theta}) = 0.5$ with minimal proximity to $\boldsymbol{\theta}_0$, while sparsity and plausibility are enforced through the cost in (7.3.7); here, $\boldsymbol{\theta}_s^n$ is the current CFE candidate and $\boldsymbol{\theta}_s^o$ the previously accepted one.

7.4 Results and Discussions

In this section, we evaluate Algorithm 7.1 on the compressor–plenum–throttle system. Figure 7.4.3 shows the site Piping and Instrumentation Diagram (P&ID): the outer pressure controller (PIC) and the inner IGV position controller (ZIC) are the loops ACE retunes. We report results for two scenarios: **Case 1** (single-loop) in which IGV position feedback is unavailable and the PIC output drives the servo valve directly—hence ACE retunes only the outer loop; and **Case 2** (cascade), consistent with the P&ID schematic, where IGV position feedback is present, the inner loop (ZIC) closes the vane-position dynamics, and ACE jointly retunes both controllers.

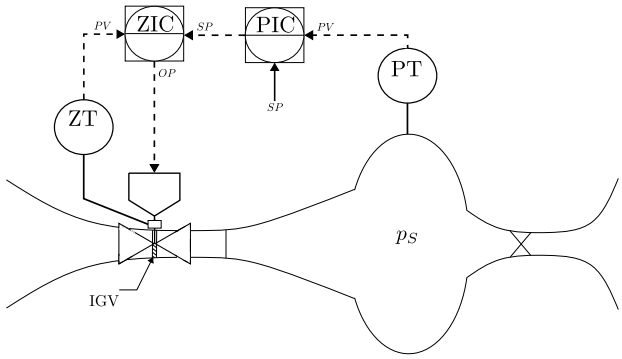


Figure 7.4.3: P&ID of the compressor system.

7.4.1 Experimental Setting

Controllers. In both scenarios the controllers are implemented as PIDs with derivative filtering, *i.e.*,

$$C(s) \;=\; K_p \;+\; \frac{K_i}{s} \;+\; \frac{K_d\, s}{1+T_f s}.$$

Accordingly, ACE searches for a four-parameter retuning in **Case 1**, with $\boldsymbol{\theta}_{case_1} = [K_p, K_i, K_d, T_f]^\top$, and for an eight-parameter retuning in **Case 2**, with $\boldsymbol{\theta}_{case_2} = [K_{p1}, K_{i1}, K_{d1}, T_{f1}, K_{p2}, K_{i2}, K_{d2}, T_{f2}]^\top$, where the subscripts 1 and 2 on the PID parameters map to the outer (C_1) and inner (C_2) controllers, respectively; in both cases, ACE selects counterfactual retunings that satisfy the *outer-loop* (PIC) closed-loop specifications.

Parameters. Unless otherwise noted, the compressor– plenum constants and IGV servo valve parameters are $(\ell_c, B, k_T, a, \alpha) = (2.0, 2.0, 0.20, 0.18, 0.30)$ and $(K_v, \tau) = (1.0, 0.5)$, respectively. We use a normalized step of amplitude $r_{\text{step}} = 1$. Measurement noises are zero-mean Gaussian with variances $(\sigma^2_{\text{outer}}, \sigma^2_{\text{inner}}) = (0.01, 0.005)$, corresponding to a outer-loop input signal-to-noise ratio of 20[dB]. ACE hyperparameters follow Appendix 6.F.1.

Continuous-to-Discrete conversion. The compressor–plenum path and the servo valve are mapped from $G_1(s)$ and $G_2(s)$ to $G_1^{\mathrm{d}}(q)$ and $G_2^{\mathrm{d}}(q)$ using the bilinear (Tustin) transform with sampling period $T_s = 0.1\,\mathrm{s}$. The PID controllers are implemented in position form: the derivative term uses a first–order filter discretized with Tustin–consistent coefficients, while the integral term uses the rectangular (forward Euler) rule.

Table 7.4.1: Case 1 (single loop): summary over 100 ACE runs per instance.

Inst.	Val.	LOF	iters.	$\Delta K_p; \Delta K_i; \Delta K_d; \Delta T_f$
$\boldsymbol{\theta}_{01}$	1	1.0276	12	$+0.0334;\ -0.0986;\ +0.0251;\ +0.0331$
$\boldsymbol{\theta}_{02}$	1	1.0251	12	$+0.0057;\ -0.0482;\ +0.0029;\ +0.0333$
$\boldsymbol{\theta}_{03}$	1	1.0205	18	$+0.0234;\ -0.0602;\ +0.0071;\ +0.0325$

Closed-loop criteria. We define the pass/fail function as

$$h(\boldsymbol{\theta}) = \begin{cases} 1, & \text{if } \begin{aligned} &|e_{\text{ss}}| && \le 0.01 \\ &\ \& \ t_{\text{rise}\,(10\to90)} && \le 20\ \text{s} \\ &\ \& \ t_{\text{settle}} && \le 50\ \text{s} \\ &\ \& \ \text{OS} && \le 20\% \end{aligned} \\ 0, & \text{otherwise.} \end{cases}$$

where e_{ss} is the steady-state error, $t_{\text{rise}(10\to90)}$ is the rise time between the first crossings of 10% and 90% of the final step change, t_{settle} is the first time for which $|y(t) - y_\infty| \le 0.05$ for all $t \ge t_{\text{settle}}$, and OS is the percent of overshoot.

7.4.2 Results

Case 1

Table 7.4.1 summarizes results over 100 ACE runs for three fixed instances $\boldsymbol{\theta}_{01:03}$, initialized with 30 historical controllers identified from the DCS archive. Each entry reports the mean *Validity* (fraction of runs that found a counterfactual), the mean *LOF* (Local Outlier Factor; values close to or greater than 1 indicate inliers to the data manifold) computed over every new $\boldsymbol{\theta}_k$ proposed during the search (including the final CFE), the mean number of additional closed-loop tests (*iters.*), and the mean component-wise parameter shift reported as $(K_p; K_i; K_d; T_f)$, where we define the shift as $\Delta = \boldsymbol{\theta}_{\text{CFE}} - \boldsymbol{\theta}_0$—*i.e.*, positive means increase w.r.t. baseline. Overall, average changes are small (typically $|\Delta| \lesssim 0.1$), with only a modest number of extra step tests per run. Over 100 trials, ACE requires 21.6 s per valid CFE on average, demonstrating practical deployability.

To examine the results in detail, we focus on the first instance in Table 7.4.1: $\boldsymbol{\theta}_0 = \boldsymbol{\theta}_{01} = [3.864, 1.112, 1.561, 0.015]$. Figure 7.4.4 reports the component-wise shifts $\Delta = \boldsymbol{\theta}_{\text{cand}} - \boldsymbol{\theta}_0$ across 100 ACE runs—including, but not limited to, the final CFEs. Note that, unlike Table 7.4.1—which averages only the *final* CFE per run—Figure 7.4.4 pools *all* candidate evaluations generated during the ACE search, with an average of $\sim$12 proposals per run until the CFE is found. The boxes span the interquartile range (25–75%), the solid orange line indicates the median and the dashed green line indicates the mean, whiskers extend to the non-outlier range, and circles denote outliers. Three patterns emerge. First, the distributions are tightly centered around zero, with few outliers and typical magnitudes $|\Delta| \lesssim 0.3$, indicating that ACE proposes small, local retunings consistent with the trust-region constraint. Second, the medians suggest a mild *increase* in K_p, a mild *decrease* in K_i, and small

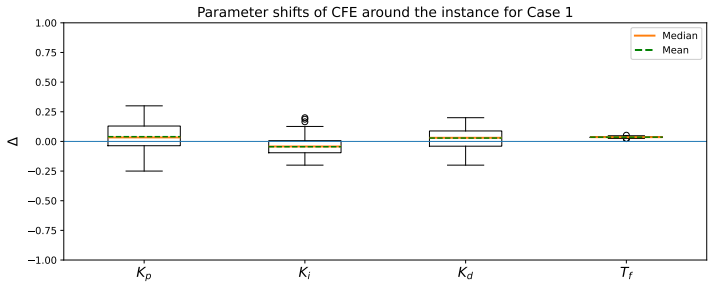


Figure 7.4.4: Case 1 - Box plot of component-wise shifts $\boldsymbol{\Delta} = \boldsymbol{\theta}_{\text{cand}} - \boldsymbol{\theta}_0$ across 100 ACE runs.

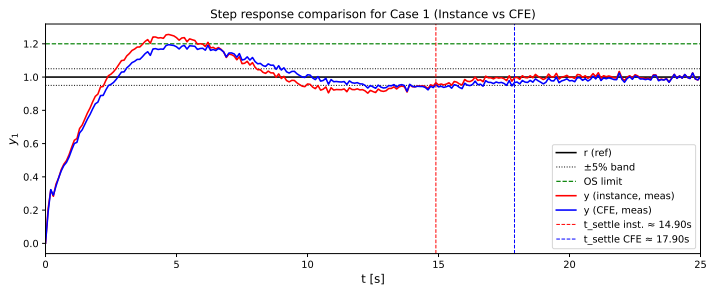


Figure 7.4.5: Case 1 - Step responses: baseline $\boldsymbol{\theta}_0$ (red) vs. one representative CFE (blue).

positive shifts for K_d and T_f—in line with making the loop slightly more responsive while filtering derivative action. Third, several CFEs exhibit $\Delta \approx 0$ on individual coordinates, showing that sparsity is naturally promoted: not every retuning needs to change all gains.

Figure 7.4.5 compares the instance $\boldsymbol{\theta}_0 = [3.864, 1.112, 1.561, 0.015]$ (red) against one representative counterfactual CFE $= [3.916, 1.007, 1.542, 0.022]$ (blue). Consistent with Figure 7.4.4, the counterfactual increases K_p slightly, reduces K_i, and applies modest adjustments to K_d and T_f. These small parameter changes are sufficient to flip the label from "fail" to "pass": overshoot is brought below the 20% limit (peak ≤ 1.20), while all other time–domain metrics are satisfied. The comparison illustrates the core CFE objective: achieving specification compliance with minimal movement in parameter space and without leaving the safe operating neighborhood; from a control perspective, this is achieved through small local gain adjustments that reduce overshoot and improve transient compliance without altering the closed-loop operating regime.

Case 2

As in Case 1, Table 7.4.2 summarizes 100 ACE runs for three fixed cascade instances—*Validity* is 1.0 for all instances and the mean *LOF* values remain close to 1, indicating that the proposed controllers lie on (or near) the historical manifold. Despite doubling the decision variables (eight gains instead of four), the mean number of additional closed-loop tests per run remains modest—∼15–18, comparable to Case 1—showing that ACE can jointly retune inner and outer loops without the usual two-step manual procedure used in industry [120, Chap. 16, p. 284]. Parameter shifts remain small in magnitude (typically $|\Delta| \lesssim 0.4$). Over 100 trials, ACE requires 29.5 s per valid CFE on average. In a sensitivity check, varying the historian archive size $\mathcal{H}$ yields broadly stable metrics in both cases, indicating a robust trust-region initialization. No closed-loop instability was observed for any candidate $\boldsymbol{\theta}_k$ evaluated in this study.

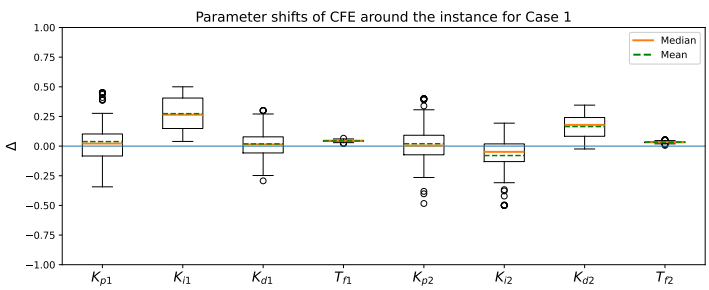


Figure 7.4.6: Case 2 - Box plot of component-wise shifts $\boldsymbol{\Delta} = \boldsymbol{\theta}_{\text{cand}} - \boldsymbol{\theta}_0$ across 100 ACE runs.

Figure 7.4.6 summarizes the component-wise shifts $\Delta = \boldsymbol{\theta}_{\text{cand}} - \boldsymbol{\theta}_0$ across 100 ACE runs for instance $\boldsymbol{\theta}_{01}$. As in Case 1, the distributions remain tightly centered around

Table 7.4.2: Case 2 (cascade): summary over 100 ACE runs per instance.

Inst.	Val.	LOF	iters.	C_1, C_2: $\Delta K_p; \Delta K_i; \Delta K_d; \Delta T_f$
$\boldsymbol{\theta}_{01}$	1	1.0218	15	$+0.0488;\ +0.3120;\ +0.0276;\ +0.0418$ $+0.0226;\ -0.0839;\ +0.0587;\ +0.0302$
$\boldsymbol{\theta}_{02}$	1	1.0123	17	$+0.1106;\ +0.1144;\ +0.2259;\ +0.0404$ $-0.1123;\ -0.0432;\ +0.1088;\ +0.0415$
$\boldsymbol{\theta}_{03}$	1	1.0285	18	$+0.3958;\ +0.1374;\ +0.0780;\ +0.0391$ $+0.0546;\ -0.1348;\ +0.0614;\ +0.0405$

zero with few outliers, indicating small, local moves consistent with the trust region. Two coordinates show a clear directional trend: K_{i1} and K_{d2} tend to increase, both typically within $|\Delta| \lesssim 0.5$. In contrast, K_{i2} exhibits a slight decrease, while T_{f1} and T_{f2} show mild positive shifts; the remaining gains are essentially centered at $\Delta \approx 0$. Overall, even in the 8-parameter search, ACE stays near the historical manifold and proposes sparse, low-magnitude adjustments.

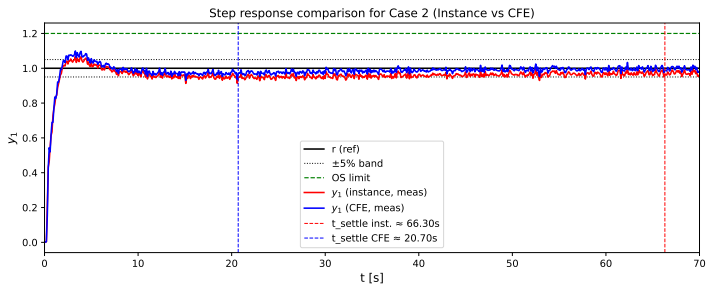


Figure 7.4.7: Case 2 - Step responses: baseline $\boldsymbol{\theta}_0$ (red) vs. one representative CFE (blue).

A representative comparison is shown in Fig. 7.4.7: the instance $\boldsymbol{\theta}_0 = [8.453, 0.136, 1.73, 0.010, 3.986, 2.842, 0.264, 0.02]$ versus one CFE $[8.519, 0.466, 1.73, 0.065, 3.84, 2.64, 0.569, 0.065]$. The parameter change is $\Delta_{\text{CFE}} = \text{CFE} - \boldsymbol{\theta}_0 = [+0.066, +0.331, +0.003, +0.055, -0.146, -0.202, +0.306, +0.045]$, *i.e.*, a modest boost in K_{i1} and K_{d2}, slight increases in T_{f1}/T_{f2}, and small shifts elsewhere. In PID terms, the larger K_{i1} removes steady-state error more decisively in the outer loop, while the higher K_{d2} improves inner-loop vane tracking and effective damping. The result is a substantial reduction of settling time—from ≈ 66.3 s down to ≈ 20.7 s—while keeping the overshoot within limits and satisfying all four time-domain metrics, achieved with small, interpretable moves in parameter space.

7.5 Conclusion and Future Work

In this chapter, we have presented a historian-informed counterfactual approach to sample-efficient controller retuning and demonstrated its application to the pressure-control loop of an axial compressor represented by a compressor–plenum–throttle system. Building on the ACE framework introduced in Chapter 6, the proposed methodology operates entirely in closed loop and searches

for minimal, operator-actionable modifications of the controller parameters within a trust region supported by historical operating data. In this way, controller retuning can be performed using only a limited number of short closed-loop experiments, without requiring an explicit parametric model of the underlying plant.

The results obtained for both single-loop and cascade architectures show that ACE can identify small and interpretable controller modifications that change the closed-loop classification from unsatisfactory to satisfactory while remaining close to historically observed operating conditions and preserving the surge margin in the simulations considered. More broadly, these results strengthen the connection between counterfactual explanations and controller tuning: rather than exhaustively exploring the parameter space, retuning can be formulated as a targeted search for the smallest intervention required to achieve the desired closed-loop behavior. This perspective is particularly attractive in industrial applications, where experiments are costly and operational interventions must remain limited.

Although this chapter focused on PID-based compressor control, the proposed formulation is not inherently restricted to either linear controllers or compressor systems. The same philosophy can be extended to other industrial processes for which historical data are available and controller evaluations are expensive, and may further incorporate operating context, including time-dependent information, to address nonstationary regimes. Future work includes improving the reconstruction of historian-based priors through instrumental-variable and kernel-regularized identification methods, extending the framework to more complex and nonlinear controller structures, and validating the approach using real DCS historian data and high-fidelity industrial simulators. Such extensions would further establish counterfactual reasoning as a practical tool for data-driven controller adjustment in systems where experimentation is intrinsically limited.

8 Summary and Future Research Directions

This thesis investigates data-driven methods for extracting task-relevant information from dynamical systems and machine learning models without requiring complete parametric descriptions of the underlying models. The developments follow two complementary directions; the first part of the thesis focuses on robustness analysis, where input–output experiments are used to characterize structured uncertainty and control-relevant model mismatch. The second part considers the identification of small and meaningful perturbations capable of producing desired changes in the behavior of black-box models, with particular emphasis on counterfactual explanations and controller adjustment.

8.1 Summary

In relation to the characterization of structured uncertainty, Chapter 3 studies the data-driven estimation of the structured singular value. The main objective is to assess the vulnerability of an unknown dynamical system to structured perturbations directly from input–output experiments. Based on a power iteration procedure, the proposed identification-free method estimates lower bounds on the structured singular value while explicitly accounting for the prescribed uncertainty structure. Numerical experiments over different block structures and system dimensions show close agreement with model-based computations under favorable conditions, while also revealing limitations associated with measurement noise and increasing system dimension.

Regarding control-relevant model mismatch, Chapters 4 and 5 study the data-driven estimation of the Vinnicombe metric between a nominal model and the true plant. Chapter 4 first considers discrete-time SISO systems and introduces an iterative estimator together with a data-driven admissibility test for the index condition. Building on this formulation, Chapter 5 extends the framework to MIMO systems by

reformulating the metric from a finite-horizon operator perspective. Two matrix-free approaches based on power iteration and Golub–Kahan–Lanczos bidiagonalization estimate the dominant singular value associated with the normalized plant mismatch. The numerical studies show close agreement with MATLAB's `gapmetric` implementation and demonstrate that control-relevant model mismatch can also be quantified directly from experimental input–output information in the multivariable setting.

Concerning the identification of purposeful perturbations, the focus in Chapters 6 and 7 shifts from characterizing changes that may degrade system behavior to identifying small modifications that produce a desired outcome. Chapter 6 introduces Adaptive Sampling for Counterfactual Explanations (ACE), a query-efficient framework that combines Gaussian-process classification and sequential sampling to identify feasible and actionable counterfactual explanations using a limited number of black-box evaluations. Building on this perspective, Chapter 7 applies counterfactual reasoning to data-driven controller adjustment in an axial compressor. By interpreting controller parameters as features of a black-box classification problem, ACE identifies small controller modifications that transform unsatisfactory closed-loop behavior into satisfactory operation while remaining close to historically observed operating conditions. These results illustrate how counterfactual reasoning provides a sample-efficient mechanism for controller retuning when an accurate model of the underlying process is not available.

8.2 Future Research Directions

We will now discuss possible paths for future work related to the topics studied in this thesis.

8.2.1 Data-Driven Robustness Metrics

Structured Singular Value

The developments of Chapter 3 provide a data-driven lower-bound estimate of the structured singular value, but several aspects of a complete data-driven robustness analysis remain open. In particular, the current method does not provide a corresponding data-driven upper bound, and the numerical performance of the proposed estimator deteriorates in the presence of measurement noise and for some higher-dimensional uncertainty structures. Moreover, parts of the computation rely on intermediate frequency-domain quantities, despite the information ultimately being obtained through time-domain experiments.

A natural direction for future research is therefore to develop a more complete time-domain formulation of structured singular-value estimation. Besides obtaining complementary upper bounds, such a formulation should reduce the dependence on frequency-domain reconstructions and improve robustness to noisy measurements, for example by incorporating instrumental-variable or regularization techniques.

A further challenge arises from the experimental implementation of transpose or

adjoint operator actions in multivariable systems, where direct evaluations may require a number of experiments that increases rapidly with the system dimension. One possible direction is to replace these exact actions by randomized approximations. In particular, stochastic perturbation methods could approximate the required gradient information using only a small number of directional experiments at each iteration [121]. Alternative stochastic optimization schemes, such as mirror-prox methods [122] or importance-sampling approximations for matrix–vector operations [123], also provide promising directions for reducing the experimental burden. Studying the convergence and statistical properties of these alternatives would constitute an important step toward scalable data-driven structured robustness analysis.

Data-Driven ν-Gap

Although Chapters 4 and 5 progressively extend the data-driven Vinnicombe metric from SISO to MIMO systems, the multivariable implementation has so far been evaluated only on relatively low-dimensional benchmark systems. Furthermore, the proposed matrix-free algorithms still require repeated forward and adjoint experiments as the power or Krylov iterations converge. This experimental burden can become significant when the true plant is expensive to excite or when the number of inputs and outputs increases.

An important direction for future work is therefore to investigate higher dimensional and industrially representative systems. In particular, the SISO gas-turbine benchmark considered in Chapter 4 could be extended to a multivariable description incorporating additional process variables such as pressure, flow, and temperature. This would allow the proposed methods to be evaluated on plant representations that more closely capture the interactions encountered in practical applications.

From an algorithmic perspective, a more detailed analysis of the convergence and finite-data behavior of the two MIMO approaches is also needed. Future work could investigate more efficient Krylov updates, reuse of previously generated subspace information, and alternative stopping criteria in order to reduce the number of required experiments. A statistical characterization under measurement noise and finite trajectory lengths would further clarify the trade-off between estimation accuracy and experimental effort. Finally, while the norm estimation in Chapter 5 admits an operator-based time-domain implementation, the admissibility test still relies on frequency-domain information. Therefore, developing an entirely data-driven time-domain characterization of this condition remains an interesting open direction.

8.2.2 Explainable AI

Counterfactual Explanations

Although ACE reduces the number of black-box evaluations required to identify a counterfactual explanation, several theoretical and methodological questions remain open. In particular, its convergence properties and query efficiency are primarily supported by empirical evidence, while the simultaneous treatment of proximity,

plausibility, actionability, and diversity introduces objectives that may compete with one another.

One possible direction is to formulate counterfactual generation as a multi-objective sequential optimization problem, allowing the trade-offs between these properties to be handled more explicitly and enabling the generation of diverse counterfactual explanations without repeatedly solving independent search problems. Further theoretical analysis of the convergence and sample-efficiency properties of ACE would also provide a stronger characterization of the number of black-box evaluations required to obtain reliable counterfactuals. Finally, extending the surrogate and acquisition mechanisms to better exploit structure in high-dimensional input spaces constitutes another relevant direction for improving the scalability of the method.

Controller Tuning and Safe Search

The controller-tuning applications considered in Chapters 6 and 7 show that counterfactual search can identify useful controller modifications with substantially fewer evaluations than an exhaustive exploration of the parameter space. However, sample efficiency alone is not sufficient when the queries correspond to experiments on a physical closed-loop system, since a newly proposed candidate may still lead to unstable operation or violate critical process constraints despite the surrogate being trained on prior stable controller configurations. Restricting the search to regions supported by historical data reduces this risk, but does not by itself provide an explicit stability guarantee.

A central direction for future research is therefore the development of a safe extension of ACE in which candidate counterfactuals are screened before being applied to the process. In such a framework, the sequential search would combine the objective of approaching the desired classification boundary with a data-driven assessment of closed-loop stability. Candidates that cannot be certified as sufficiently safe would be rejected or redirected before an experiment is performed, allowing the information collected from the system to guide both the counterfactual search and the evolving characterization of the safe region.

One possible implementation is to construct a probabilistic dynamical model from previously collected closed-loop trajectories and propagate its uncertainty into a stability test. For instance, posterior samples of the dynamics could be locally linearized under a candidate controller and used to estimate the probability that the resulting closed-loop system remains stable. This information could then operate as a safety filter within ACE, so that only controller modifications satisfying a prescribed confidence level are experimentally evaluated.

Beyond safety, future work should evaluate this framework using high-fidelity industrial simulators and ultimately real historian and DCS data. Extending the compressor study to nonlinear and multivariable dynamics, more general controller architectures, and changing operating conditions would provide an important step toward establishing counterfactual search as a practical methodology for data-driven controller adjustment under limited and safety-critical experimentation.

Bibliography

[1] Arturo Rosenblueth and Norbert Wiener. “The role of models in science”. In: *Philosophy of science* 12.4 (1945), pp. 316–321.

[2] Ludwig Von Bertalanffy. “General system theory”. In: *New York* 41973.1968 (1968), p. 40.

[3] Eduardo D Sontag. *Mathematical control theory: deterministic finite dimensional systems*. Vol. 6. Springer Science & Business Media, 2013.

[4] Karl Johan Åström and Richard Murray. *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, 2021.

[5] L. Ljung. *System Identification: Theory for the User, 2nd Edition*. Prentice Hall, 1999.

[6] Tamer Başar and Geert Jan Olsder. *Dynamic noncooperative game theory*. SIAM, 1998.

[7] Torsten Söderström and Petre Stoica. *System identification*. Prentice-Hall, 1989.

[8] Peter Overschee and Bart Moor. *Subspace identification for linear systems: Theory-Implementation-Applications*. Springer, 1996.

[9] Florian Dörfler, Jeremy Coulson, and Ivan Markovsky. “Bridging direct and indirect data-driven control formulations via regularizations and relaxations”. In: *IEEE Transactions on Automatic Control* 68.2 (2022), pp. 883–897.

[10] Zhong-Sheng Hou and Zhuo Wang. “From model-based control to data-driven control: Survey, classification and perspective”. In: *Information Sciences* 235 (2013), pp. 3–35.

[11] Claudio De Persis and Pietro Tesi. “Formulas for Data-Driven Control: Stabilization, Optimality, and Robustness”. In: *IEEE Transactions on Automatic Control* 65.3 (2020), pp. 909–924.

[12] K. Zhou, J. C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice-Hall, 1996.

[13] K. Zhou and J. C. Doyle. *Essentials of Robust Control*. Vol. 104. Prentice Hall Upper Saddle River, NJ, 1998.

[14] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, et al. “A survey of methods for explaining black box models”. In: *ACM computing surveys (CSUR)* 51.5 (2018), pp. 1–42.

[15] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, et al. “Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI”. In: *Information fusion* 58 (2020), pp. 82–115.

[16] Sandra Wachter, Brent Mittelstadt, and Chris Russell. “Counterfactual Explanations Without Opening the Black Box: Automated Decisions and the GDPR”. In: *Harvard Journal of Law & Technology* 31 (2017), pp. 841–887.

[17] A. Koch. *Determining input-output properties of linear time-invariant systems from data*. Logos Verlag, 2022.

[18] D.A. Bristow, M. Tharayil, and A.G. Alleyne. “A survey of iterative learning control”. In: *IEEE Control Systems Magazine* 26.3 (2006), pp. 96–114.

[19] H. Hjalmarsson, M. Gevers, S. Gunnarsson, et al. “Iterative feedback tuning: theory and applications”. In: *IEEE Control Systems Magazine* 18.4 (1998), pp. 26–41.

[20] P. Albertos and A. S. Piqueras. *Iterative identification and control: Advances in theory and applications*. Springer, 2012.

[21] Håkan Hjalmarsson. “From experiment design to closed-loop control”. In: *Automatica* 41.3 (2005), pp. 393–438.

[22] John Doyle. “Analysis of feedback systems with structured uncertainties”. In: *IEE Proceedings D (Control Theory and Applications)* 129.6 (1982), pp. 242–250.

[23] Glenn Vinnicombe. *Uncertainty and Feedback*. Imperial College Press, 2001.

[24] A. Packard, M. K. H. Fan, and J. Doyle. “A power method for the structured singular value”. In: *Proceedings of the IEEE Conference on Decision and Control*. 1988, pp. 2132–2137.

[25] Wolfgang Reinelt, Andrea Garulli, and Lennart Ljung. “Comparing different approaches to model error modeling in robust identification”. In: *Automatica* 38.5 (2002), pp. 787–803.

[26] Douwe K De Vries and Paul MJ Den Van Hof. “Quantification of model uncertainty from data”. In: *International Journal of Robust and Nonlinear Control* 4.2 (1994), pp. 301–319.

[27] Roy Smith and John C Doyle. “Model validation: A connection between robust control and identification”. In: *IEEE Transactions on automatic control* 37.7 (1992), pp. 942–952.

[28] Tom Oomen and Okko Bosgra. “Well-posed model uncertainty estimation by design of validation experiments”. In: *IFAC Proceedings Volumes* 42.10 (2009), pp. 1199–1204.

[29] Bo Wahlberg, Märta Barenthin Syberg, and Håkan Hjalmarsson. “Non-parametric methods for $\mathcal{L}_2$-gain estimation using iterative experiments”. In: *Automatica* 46.8 (2010), pp. 1376–1381.

[30] C. R. Rojas, T. Oomen, H. Hjalmarsson, et al. “Analyzing iterations in identification with application to nonparametric $\mathcal{H}_\infty$-norm estimation”. In: *Automatica* 48.11 (2012), pp. 2776–2790.

[31] T. Oomen, R. van der Maas, C. R. Rojas, et al. “Iterative Data-Driven $\mathcal{H}_\infty$ Norm Estimation of Multivariable Systems With Application to Robust

Active Vibration Isolation". In: *IEEE Transactions on Control Systems Technology* 22.6 (2014), pp. 2247–2260.

[32] T. Koenings, M. Krueger, H. Luo, et al. "A data-driven computation method for the gap metric and the optimal stability margin". In: *IEEE Transactions on Automatic Control* 63.3 (2017), pp. 805–810.

[33] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. ""Why Should I Trust You?": Explaining the Predictions of Any Classifier". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2016)*. 2016, pp. 1135–1144.

[34] Scott M. Lundberg and Su-In Lee. "A Unified Approach to Interpreting Model Predictions". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS 2017)*. 2017, pp. 4765–4774.

[35] Jean Pierre Allamaa, Panagiotis Patrinos, and Tong Duy Son. "ExAMPC: the Data-Driven Explainable and Approximate NMPC with Physical Insights". In: *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2025, pp. 2976–2983.

[36] Federico Porcari, Donatello Materassi, and Simone Formentin. "eXplainable AI for data driven control: an inverse optimal control approach". In: *2025 IEEE 64th Conference on Decision and Control (CDC)*. 2025, pp. 5186–5191.

[37] G. Riva and S. Formentin. "Toward eXplainable data-driven control (XDDC): The property-preserving framework". In: *IEEE Control Systems Letters* 8 (2024), pp. 478–483.

[38] Amir-Hossein Karimi, Bernhard Schölkopf, and Isabel Valera. "Algorithmic Recourse: from Counterfactual Explanations to Interventions". In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. 2021, pp. 353–362.

[39] Susanne Dandl, Christoph Molnar, Martin Binder, et al. "Multi-Objective Counterfactual Explanations". In: *Parallel Problem Solving from Nature – PPSN XVI. PPSN 2020*. 2020, pp. 448–469.

[40] Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, et al. "FACE: Feasible and Actionable Counterfactual Explanations". In: *Proceedings of the 2020 AAAI/ACM Conference on AI, Ethics, and Society (AIES)*. 2020, pp. 344–350.

[41] Piotr Romashov, Martin Gjoreski, Kacper Sokol, et al. "BayCon: Model-agnostic Bayesian Counterfactual Generator". In: *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*. 2022, pp. 740–746.

[42] Margarita A. Guerrero and Cristian R. Rojas. "ACE: Adapting sampling for Counterfactual Explanations". In: *Proceedings of The 8th Annual Learning for Dynamics and Control Conference*. Vol. 331. 2026, pp. 242–264.

[43] Margarita A. Guerrero, Rodrigo A. González, and Cristian R. Rojas. "Sample-efficient counterfactual tuning for compressor pressure control". In: *Journal of Process Control* 162 (2026), p. 103723.

[44] Yousef Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.

[45] Anne Greenbaum. *Iterative methods for solving linear systems*. SIAM, 1997.

[46] Wojciech Samek, Grégoire Montavon, Andrea Vedaldi, et al. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Springer, 2019.

[47] Geir E Dullerud and Fernando Paganini. *A course in robust control theory: a convex approach*. Springer Science & Business Media, 2013.

[48] Glenn Vinnicombe. "Structured uncertainty and the graph topology". In: *Proceedings of the 30th IEEE CDC*. 1991, pp. 541–542.

[49] Glenn Vinnicombe. "Frequency domain uncertainty and the graph topology". In: *IEEE Transactions on Automatic Control* 38.9 (1993), pp. 1371–1383.

[50] Glenn Vinnicombe. "On the Frequency Response Interpretation of an Indexed $\mathcal{L}_2$-gap metric". In: *American Control Conference*. 1992, pp. 1133–1137.

[51] G. H. Golub and C. F. Van Loan. *Matrix Computations, 4th Ed.* John Hopkins University Press, 2013.

[52] Walter Edwin Arnoldi. "The principle of minimized iterations in the solution of the matrix eigenvalue problem". In: *Quarterly of applied mathematics* 9.1 (1951), pp. 17–29.

[53] Yousef Saad. *Numerical methods for large eigenvalue problems: revised edition*. SIAM, 2011.

[54] Artan Borici. "A Lanczos approach to the inverse square root of a large and sparse matrix". In: *Journal of Computational Physics* 162.1 (2000), pp. 123–131.

[55] Giulia Vilone and Luca Longo. "Classification of explainable artificial intelligence methods through their output formats". In: *Machine Learning and Knowledge Extraction* 3.3 (2021), pp. 615–661.

[56] Christoph Molnar. *Interpretable machine learning: A guide for making Black Box models explainable*. Independently published, 2022.

[57] Vy Vo, Trung Le, Van Nguyen, et al. "Feature-based Learning for Diverse and Privacy-Preserving Counterfactual Explanations". In: *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2023, pp. 2211–2222.

[58] Riccardo Guidotti. "Counterfactual explanations and how to find them: literature review and benchmarking". In: *Data Min. Knowl. Discov.* 38 (2022), pp. 2770–2824.

[59] M. I Müller and C. R. Rojas. "Gain estimation of linear dynamical systems using Thompson sampling". In: *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*. 2019, pp. 1535–1543.

[60] T. Mushtaq, D. Bhattacharjee, P. Seiler, et al. "Structured singular value of a repeated complex full-block uncertainty". In: *International Journal of Robust and Nonlinear Control* 34.7 (2024), pp. 4881–4897.

[61] T. Mushtaq, P. Seiler, and Maziar S. Hemati. "Exact solution for the rank-one structured singular value with repeated complex full-block uncertainty". In: *Automatica* 167 (2024), p. 111717. issn: 0005-1098.

[62] A. Prajapati and P. Sharma. *Structured singular values and their application in computing eigenvalue backward errors of the Rosenbrock system matrix*. 2024. eprint: 2405.11974. url: https://arxiv.org/abs/2405.11974.

[63] Da-Wei Gu, Petko H. Petkov, and Mihail M. Konstantinov. *Robust Control Design with MATLAB®*. Springer, 2013.

[64] Y. Cheng and B. De Moor. “A continuous time power iteration method for computing a lower bound of μ”. In: *Proceedings of the 1994 American Control Conference (ACC)*. 1994, pp. 1416–1417.

[65] Charles A Desoer and Mathukumalli Vidyasagar. *Feedback Systems: Input-Output Properties*. SIAM, 2009.

[66] Alexandre Megretski and Anders Rantzer. “System analysis via integral quadratic constraints”. In: *IEEE Transactions on Automatic Control* 42 (1997), pp. 819–830.

[67] A. El-Sakkary. “The gap metric: Robustness of stabilization of feedback systems”. In: *IEEE Transactions on Automatic Control* 30 (1985), pp. 240–247.

[68] Michael Cantoni and Harald Pfifer. “Gap metric computation for time-varying linear systems on finite horizons”. In: *IFAC-PapersOnLine* 50.1 (2017), pp. 14513–14518.

[69] Peter Welch. “The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms”. In: *IEEE Transactions on Audio and Electroacoustics* 15.2 (1967), pp. 70–73.

[70] W. I. Rowen. “Simplified Mathematical Representations of Heavy-Duty Gas Turbines”. In: *Journal of Engineering for Power* 105 (1983), pp. 865–869.

[71] Mohammad Reza Bank Tavakoli, Behrooz Vahidi, and Wolfgang Gawlik. “An educational guide to extract the parameters of heavy duty gas turbines model in dynamic studies based on operational data”. In: *IEEE Transactions on Power Systems* 24.3 (2009), pp. 1366–1374.

[72] Aminollah Khormali, Iman Yousefi, Hassan Yahyaei, et al. “Identification of an industrial gas turbine based on Rowen’s model and using Multi-Objective Optimization method”. In: *3rd RSI ICROM*. 2015, pp. 482–487.

[73] Jeremy Coulson, John Lygeros, and Florian Dörfler. “Data-enabled predictive control: In the shallows of the DeePC”. In: *2019 18th European control conference (ECC)*. 2019, pp. 307–312.

[74] Mohammad Alsalti, Victor G Lopez, and Matthias A Müller. “Notes on data-driven output-feedback control of linear MIMO systems”. In: *IEEE Transactions on Automatic Control* 70.9 (2025), pp. 6143–6150.

[75] Yuan Guo, Zhongsheng Hou, Shida Liu, et al. “Data-driven model-free adaptive predictive control for a class of MIMO nonlinear discrete-time systems with stability analysis”. In: *IEEE Access* 7 (2019), pp. 102852–102866.

[76] Alireza Karimi and Christoph Kammer. “A data-driven approach to robust control of multivariable systems by convex optimization”. In: *Automatica* 85 (2017), pp. 227–233.

[77] Stephen Boyd, Martin Hast, and Karl Johan Åström. “MIMO PID tuning via iterated LMI restriction”. In: *International Journal of Robust and Nonlinear Control* 26.8 (2016), pp. 1718–1731.

[78] Zhaojun Bai, James Demmel, Jack Dongarra, et al. *Templates for the solution of algebraic eigenvalue problems: a practical guide*. SIAM, 2000.

[79] S. Skogestad and I. Postlethwaite. *Multivariable Feedback Control: Analysis and Design, 2nd Ed.* Wiley, 2005.

[80] Blair D Carlson. “Covariance matrix estimation errors and diagonal loading in adaptive arrays”. In: *IEEE Transactions on Aerospace and Electronic systems* 24.4 (1988), pp. 397–401.

[81] Paul Voigt and Axel von dem Bussche. *The EU General Data Protection Regulation (GDPR): A Practical Guide*. Springer, 2017.

[82] Content European Commission Directorate-General for Communications Networks and Technology. *Ethics Guidelines for Trustworthy AI*. Tech. rep. Publications Office, 2019. url: https://data.europa.eu/doi/10.2759/346720.

[83] Leilani H. Gilpin, David Bau, Ben Z. Yuan, et al. “Explaining explanations: An overview of interpretability of machine learning”. In: *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*. 2018, pp. 80–89.

[84] Sahil Verma, Varich Boonsanong, Minh Hoang, et al. “Counterfactual Explanations and Algorithmic Recourses for Machine Learning: A Review”. In: *ACM Computing Surveys* 56 (2024), Article 312, 42 pages.

[85] Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, et al. “Comparison-Based Inverse Classification for Interpretability in Machine Learning”. In: *Information Processing and Management of Uncertainty in Knowledge-Based Systems. Theory and Foundations. IPMU 2018*. 2018, pp. 100–111.

[86] Lyle Regenwetter, Yazan Abu Obaideh, and Faez Ahmed. “MCD: A Model-Agnostic Counterfactual Search Method For Multi-modal Design Modifications”. In: *Journal of Mechanical Design* 147 (2024), pp. 1–18.

[87] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, et al. “A fast and elitist multiobjective genetic algorithm: NSGA-II”. In: *IEEE Transactions on Evolutionary Computation* 6 (2002), pp. 182–197.

[88] Amir-Hossein Karimi, Gilles Barthe, Borja Balle, et al. “Model-Agnostic Counterfactual Explanations for Consequential Decisions”. In: *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*. 2020, pp. 895–905.

[89] Ana Lucic, Maartje A. Ter Hoeve, Gabriele Tolomei, et al. “CF-GNNExplainer: Counterfactual Explanations for Graph Neural Networks”. In: *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS)*. 2022, pp. 4499–4511.

[90] Martin Pawelczyk, Klaus Broelemann, and Gjergji Kasneci. “Learning Model-Agnostic Counterfactual Explanations for Tabular Data”. In: *Proceedings of The Web Conference 2020*. 2020, pp. 3126–3132.

[91] Thomas Spooner, Danial Dervovic, Jason Long, et al. *Counterfactual Explanations for Arbitrary Regression Models*. arXiv preprint 2106.15212, available at https://arxiv.org/abs/2106.15212. 2021.

[92] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[93] David G. Luenberger and Yinyu Ye. *Linear and Nonlinear Programming*. Springer, 2008.

[94] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, et al. “LOF: identifying density-based local outliers”. In: *Proceedings of the 2000 ACM*

SIGMOD International Conference on Management of Data. 2000, pp. 93–104.

[95] Finale Doshi-Velez and Been Kim. *Towards a rigorous science of interpretable machine learning*. arXiv preprint 1702.08608, available at https://arxiv.org/abs/1702.08608. 2017.

[96] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[97] George Casella and Roger L. Berger. *Statistical Inference, 2nd Ed.* CRC Press, 2024.

[98] Dong C. Liu and Jorge Nocedal. "On the limited memory BFGS method for large scale optimization". In: *Mathematical Programming* 45 (1989), pp. 503–528.

[99] Ilya M. Sobol'. "On the distribution of points in a cube and the approximate evaluation of integrals". In: *USSR Computational Mathematics and Mathematical Physics* 7 (1967), pp. 86–112.

[100] Thibault Laugel. *Growing Spheres: A Python Library for Adversarial Example Generation*. Available at https://github.com/thibaultlaugel/growingspheres. 2018.

[101] Martin Pawelczyk, Sascha Bielawski, Johannes van den Heuvel, et al. "CARLA: A Python Library to Benchmark Algorithmic Recourse and Counterfactual Explanation Algorithms". In: *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*. 2021, pp. 1–15.

[102] Subhransu Padhee. "Controller design for temperature control of heat exchanger system: simulation studies". In: *WSEAS Transactions on Systems and Control* 9.1 (2014), pp. 485–491.

[103] Katsuhiko Ogata. *Modern control engineering*. Prentice hall, 2010.

[104] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[105] Alexander J. Smola and Bernhard Schölkopf. *Learning with Lernels*. MIT Press, 2002.

[106] Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, et al. "OpenML: networked science in machine learning". In: *SIGKDD Explor. Newsl.* 15 (2014), pp. 49–60.

[107] Kaggle. *Kaggle: Your Machine Learning and Data Science Community*. 2010. url: https://www.kaggle.com.

[108] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*. 2019. url: https://archive.ics.uci.edu/.

[109] L. Ljung. "Convergence analysis of parametric identification methods". In: *IEEE Transactions on Automatic Control* 23.5 (1978), pp. 770–783.

[110] DE Brandt and RR Wesorick. "GE gas turbine design philosophy". In: *GER-3434, General Electric* (1994).

[111] Edward M Greitzer. "Surge and rotating stall in axial flow compressors: Theoretical compression system model". In: *Journal of Engineering for Power* 98.2 (1976), pp. 190–198.

[112] Bram De Jager. "Rotating stall and surge control: A survey". In: *34th IEEE Conference on Decision and Control*. Vol. 2. 1995, pp. 1857–1862.

[113] J. Crompton. “Data Management from the DCS to the Historian”. In: *Machine Learning and Data Science in the Oil and Gas Industry*. Ed. by Patrick Bangert. 2021, pp. 83–110.
[114] F. K. Moore and E. M. Greitzer. *A theory of post-stall transients in multistage axial compression systems*. Tech. rep. NASA, 1985.
[115] D. Beagle, B. Moran, M. McDufford, et al. “Heavy-Duty Gas Turbine Operating and Maintenance Considerations”. In: *GE Vernova Atlanta, GA* (2021).
[116] K. J. Åström and B. Wittenmark. *Computer Controlled Systems: Theory and Design*. Prentice-Hall, 1984.
[117] L. Ljung. *System Identification: Theory For The User*. Second. Prentice-Hall, 1999.
[118] T. Söderström and P. Stoica. *System Identification*. Prentice Hall, 2001.
[119] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.
[120] Dale E Seborg, Thomas F Edgar, Duncan A Mellichamp, et al. *Process Dynamics and Control*. John Wiley & Sons, 2016.
[121] James C Spall. “Multivariate stochastic approximation using a simultaneous perturbation gradient approximation”. In: *IEEE transactions on automatic control* 37.3 (1992), pp. 332–341.
[122] Michel Baes, Michael Burgisser, and Arkadi Nemirovski. “A randomized mirror-prox method for solving structured large-scale matrix saddle-point problems”. In: *SIAM Journal on Optimization* 23.2 (2013), pp. 934–962.
[123] Sylvester Eriksson-Bique, Mary Solbrig, Michael Stefanelli, et al. “Importance sampling for a Monte Carlo matrix multiplication algorithm, with application to information retrieval”. In: *SIAM Journal on Scientific Computing* 33.4 (2011), pp. 1689–1706.