On Greedy Reconstruction Algorithms

Dissertation zur Erlangung des
akademischen Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von

Simon Buchwald

an der

Universität Konstanz

Mathematisch-Naturwissenschaftliche Sektion

Fachbereich für Mathematik und Statistik

Tag der mündlichen Prüfung: 01.07.2025
1. Referent: Prof. Dr. Stefan Volkwein
2. Referent: Assoc. Prof. Dr. Gabriele Ciaramella

Konstanz, 2025

Zusammenfassung

Diese Arbeit widmet sich der Entwicklung und Analyse von Greedy Rekonstruktionsalgorithmen (GR). Mit diesen Verfahren können optimale Steuerungsfunktionen bestimmt werden, die eine exakte und effiziente Identifikation unbekannter Operatoren in dynamischen Systemen ermöglichen. Die Methoden basieren auf einer Zerlegung des Rekonstruktionsprozesses in Offline- und Online-Phasen, sowie auf einem Ansatz für den unbekannten Operator, der durch eine gegebene Familie an Operatoren definiert ist.

Der erste Teil dieser Arbeit konzentriert sich auf lineare dynamische Systeme. Eine detaillierte Analyse zeigt, dass die Verwendung von Steuerungen, die vom GR-Algorithmus berechnet wurden, zu strikter Konvexität des quadratischen Online-Rekonstruktionsproblems führt. Basierend auf diesen Ergebnissen wird ein neues Greedy-Verfahren (OGR) eingeführt, welches zusätzlich den Ansatz für den unbekannten Operator optimiert.

Im zweiten Teil werden allgemeinere nichtlineare Systeme betrachtet. Auf Grundlage der Linearisierungen dieser Systeme wird ein weiterer Greedy-Algorithmus (LGR) entwickelt. Es wird gezeigt, dass die von LGR berechneten Steuerungen lokale Konvergenz des klassischen Gauss-Newton-Verfahrens, angewandt auf das nichtlineare Online-Rekonstruktionsproblem, ermöglichen. Dieses Resultat wird auf die Steuerungen ausgeweitet, die vom ursprünglichen GR-Algorithmus, angewandt auf das nichtlineare System, berechnet werden. Numerische Experimente an dynamischen Systemen mit linearen und bilinearen Steuerungsstrukturen bestätigen die Effektivität der Greedy-Algorithmen.

Der dritte Teil der Arbeit befasst sich mit der Rekonstruktion nichttrivialer Wahrscheinlichkeitsverteilungen von zwei Parametern im Hamiltonian von Kernspinresonanz-Systemen. Obwohl die zugehörigen Spindynamiken durch ein bilineares System repräsentiert werden, ist das Online-Rekonstruktionsproblem quadratisch. Eine ausführliche Analyse zeigt, dass die vom GR-Algorithmus generierten Steuerungen strikte Konvexität dieses Identifikationsproblems garantieren. Diese Ergebnisse werden durch numerische Beispiele bestätigt, welche die Effizienz des OGR-Algorithmus hervorheben.

Im letzten Teil dieser Arbeit werden die GR-Algorithmen auf allgemeine Approximationsprobleme von Funktionen angewandt. Im Fall von Polynominterpolation zeigt sich, dass der ursprüngliche GR-Algorithmus genau die Leja-Interpolationspunkte (bzw. entsprechende Tensorprodukte davon) berechnet. Zudem kann eine Äquivalenz zwischen OGR und der etablierten empirischen Interpolationsmethode (EIM) bewiesen werden. Abschließend wird eine neue Klasse von Greedy-Algorithmen eingeführt, mit dem Ziel die kleinste Teilmenge eines zu approximierenden Datensatzes auszuwählen und gleichzeitig die Struktur des Nä-

herungsansatzes zu optimieren. Umfangreiche numerische Tests mit neuronalen Netzwerken belegen die Effizienz der neuen Algorithmen.

Abstract

This thesis is devoted to the development and analysis of greedy reconstruction (GR) algorithms. These procedures allow the design of optimized control functions that facilitate the accurate and efficient identification of unknown operators in dynamical systems. The approach is based on an offline/online decomposition of the reconstruction process, and on an ansatz for the unknown operator defined by a given family of operators.

The first part of this thesis focuses on linear dynamical systems. A detailed analysis demonstrates that the controls computed by the GR algorithm make the quadratic online reconstruction problem strictly convex. Based on these results, an optimized greedy strategy (OGR) is introduced that additionally optimizes the ansatz for the unknown operator.

In the second part, more general nonlinear systems are considered. Based on the linearization of these systems, a new greedy algorithm (LGR) is developed. It is shown that the controls computed by LGR lead to the local convergence of the classical Gauss-Newton method applied to the nonlinear online reconstruction problem. This result is extended to the controls obtained by the original GR algorithm applied directly to the nonlinear system. Numerical experiments on dynamical systems with linear and bilinear control structures confirm the effectiveness of the greedy algorithms.

The third part of this thesis considers the reconstruction of a non-trivial joint probability distribution for two Hamiltonian parameters in Nuclear Magnetic Resonance. While the corresponding spin dynamics can be represented by a bilinear dynamical system, the online reconstruction problem is fully quadratic. A rigorous analysis proves that the controls generated by the GR algorithm ensure strict convexity of this reconstruction problem. These findings are confirmed by numerical experiments that highlight the efficiency of the OGR algorithm.

In the final part of this work, the GR algorithms are applied to general function approximation problems. In the case of polynomial interpolation, the original GR algorithm is shown to compute exactly the (tensor-product) Leja points. Additionally, an equivalence between OGR and the well-established Empirical Interpolation Method is revealed. Finally, a new class of greedy algorithms is developed, that aim to select the smallest subset of a given data set, while simultaneously optimizing the structure of the approximation ansatz, within a given family of approximation functions. Extensive numerical tests on function approximation by neural networks demonstrate the efficiency of the new algorithms.

Acknowledgments

First and foremost, I want to thank my wife Selina from the bottom of my heart for always being there for me, especially during the time of writing this thesis. You are truly the love of my life!

I thank my family, my parents and my brother, for all your love and support throughout my entire life. You gave me the confidence to follow my heart in all of life's big decisions.

A special "grazie mille" is for Prof. Gabriele Ciaramella, for sparking my interest in GR algorithms and for guiding me through my PhD. Over the many years we worked together, you became my mentor, on a professional and also on a personal level. Whenever I got lost in details, you helped me regain my focus on the important matters at hand. I shall always remember to approach any problem in life "step-by-step".

A big thank is for Prof. Stefan Volkwein, for supporting me from the bachelor thesis to my PhD. Your encouraging style of teaching inspired me to choose the path of optimization in the first place. And even though I am still not a big coffee drinker, the "coffee breaks" in your office were always a welcome distraction and place of fruitful discussions.

I also want to thank Prof. Julien Salomon. You offered me new perspectives on the world of greedy algorithms, and showed me Paris from its most beautiful side.

I also thank Prof. Dominique Sugny. You showed me that the application of mathematics in the real world requires hard work, but also makes any success even more enjoyable.

I would like to thank Prof. Marco Verani. Working with you was a big inspiration, your views on mathematics are truly special.

A huge thanks is for all the current and former members of the AG Volkwein group. First of all, I have to thank Marco, my best friend for many years. I will never forget all the mathematical discussions we had during our many hiking trips in the alps. I also want to thank Jan B. for investing so much time into bringing the group closer together, and for your motivation and endurance. A "loaded" thanks is for Andrea, for the positive office vibe and the many insights into Italian culture. A last big thank is for Behzad, Carl, Chris, Georg, the other Jan, Lena, Luca, Lucas, Michi, Moritz and Stefan for the nice atmosphere in our group.

Finally, I gratefully acknowledge the financial support by the German Research Foundation (DFG) via the collaborative research center SFB1432, Project-ID 425217212.

Contents

Chapter 1

Introduction

Reconstruction/identification problems represent one of the main intersections between theoretical and applied science. The typical idea is to optimize operators or parameters inside a theoretical model such that its output fits best the data resulting from laboratory experiments. In general, this is understood as an inverse problem, where both the ansatz space for the operators and the experimental data are fixed. However, the data used in the fitting process can often be manipulated, for example when the experiment is affected by control functions (i.e., input forces such as magnetic fields, lasers or heat sources). If the controls are not chosen appropriately, the fitting process can result in very poor quality of the reconstructed operators. Thus, it is essential to identify a set of controls that allows one to generate good data, permitting the best possible reconstruction. This is a typical case in the field of Hamiltonian identification [17, 24, 27, 35, 38–41, 69, 72–74], as well as in engineering contexts such as state space realization [33, 50, 67] and optimal design of experiments [6, 11, 19]. In some settings, one is also interested in optimizing the structure of the ansatz space, within a given family of operators. For example, this can involve finding the smallest neural network able to learn the selected data, or simply reducing the complexity of the reconstruction problem.

This work is concerned with the development and analysis of so-called greedy reconstruction (GR) algorithms that are designed to address both of these aspects. More specifically, given the task to reconstruct an operator $B_\star$ inside a given model, the goal of our methods is to construct a set of optimal controls $\{\boldsymbol{u}^m\}_m$ while simultaneously optimizing the structure of the ansatz set $\mathcal{B}$ within a given family of operators in which one searches for $B_\star$.

This class of algorithms was first introduced in [57] for Hamiltonian identification problems. The approach divides the reconstruction process into two phases: an offline phase, where the controls are computed by a GR algorithm, and an online phase, where these controls are used to generate experimental data. This data is then used in an inverse problem for the (online) reconstruction of the unknown operator. However, [57] presented only preliminary numerical results.

We provide a detailed analysis of the strategy defined in [57] for the reconstruction of control matrices in linear dynamical systems (see [1]) which represents one of the main novelties of this thesis. Notably, we show that under appropriate observability assump-

tions, the controls generated by GR make the online reconstruction problem (globally) uniquely solvable. This analysis leads to the development of two new algorithms: optimized GR (OGR) and linearized GR (LGR). OGR extends the greedy character of GR to the ansatz space for $B_\star$. Given a family of operators $\mathcal{B}$, OGR is able to identify and eliminate redundant elements, thereby optimizing $\mathcal{B}$. Our investigation into OGR also reveals a lower bound for the number of control functions required to achieve uniqueness of solutions for the online reconstruction problem. LGR, on the other hand, builds the foundation of our analysis for bilinear and general nonlinear dynamical systems. Inspired by classical methods for solving nonlinear problems, such as the Gauss-Newton method (GN), LGR is formulated using linearizations of the dynamical systems. We demonstrate that the controls computed by LGR can ensure that GN is locally well-defined and convergent, thus making the online nonlinear reconstruction problem (locally) solvable. In particular, we provide a detailed analysis of LGR for two classes of problems: the reconstruction of a drift matrix in linear systems and the reconstruction of a Hamiltonian matrix in skew-symmetric bilinear systems, see [2]. Both cases represent truly nonlinear problems, since the unknown operators act on the states of the systems.

Our investigations into LGR also allow us to obtain a first analysis of the original GR algorithm applied to nonlinear systems, see [2]. More specifically, we prove that under appropriate controllability and observability assumptions, the controls generated by GR are also suitable for LGR, thereby make the online reconstruction problem (locally) solvable. This is demonstrated in detail for three classes of problems: the previously discussed linear and bilinear reconstruction problems, as well as the reconstruction of the control matrix in nonlinear dynamical systems with a linear control structure. Finally, we compare GR and LGR by direct numerical experiments and by a global convergence analysis in a specific case. These reveal that GR and LGR perform similarly when operating locally near the solution. However, GR is superior when applied directly to the original nonlinear system, especially when only poor information about the solution is available.

To test our algorithms in an applied setting, we adapt GR and OGR to the identification of probability distributions for parameters in nuclear magnetic resonance (NMR), see [3, 4], which marks the second main novelty of this thesis. In this context, the distributions that we aim to reconstruct are contained in the observable. We provide a rigorous analysis showing that the controls generated by the new GR algorithm (called GRPD) make the online reconstruction problem uniquely solvable. Numerical experiments show that the optimized version of GRPD (OGRPD) achieves comparable results to GRPD while utilizing only half the amount of controls, aligning with the theoretical lower bound that we found in our previous analysis. Notably, both methods are able to outperform a set of randomized controls, emphasizing their effectiveness.

The third and final novelty of the present thesis is the extension of GR and OGR to the field of function approximation, specifically focusing on polynomial interpolation and neural network applications, see [5]. We prove that GR and OGR applied to polynomial interpolation compute exactly a set of (tensor-product) Leja points. Moreover, we show that OGR is equivalent to the well-established Empirical Interpolation Method (EIM); see, e.g,. [10, 14, 20, 45, 58, 59] and references therein. Despite this equivalency, GR and OGR have one main advantage over EIM: EIM, as well as other greedy algorithms

commonly used for function approximation (see, e.g., [15]), is limited to linear parameterizations of the ansatz function. In contrast, GR and OGR can be formulated to also handle nonlinear parameterizations, such as weights and biases of neural networks. This observation motivates the introduction of a new class of greedy algorithms, called Data Greedy Reconstruction (DGR) algorithms. These methods assume that a (very) large data set is already available. The goal of these DGR algorithms is to select the smallest subset of data while simultaneously optimize the structure of the approximation ansatz, within a given family of approximation functions. In particular, we develop a combination of OGR and DGR tailored for neural network applications, called Network OGR (NOGR). We provide an in-depth discussion on the network structure optimization and illustrate by numerical tests how our algorithms effectively optimize the depth and number of neurons in a neural network.

1.1 Outline

Generally speaking, this thesis can be divided into two main parts. The first half (Chapters 3-5) is devoted to the introduction and analysis of the GR algorithms (see [3, 4]). In the second half (Chapters 6-8), we validate our theoretical results by numerical tests in specific fields of application.

Chapter 1 – Introduction. In the remaining part of this chapter, we list our own publications that this thesis is build upon, and fix the notation used throughout this work.

Chapter 2 – Greedy reconstruction algorithms for dynamical systems. In this chapter, we introduce our GR algorithms for the reconstruction of operators inside ordinary differential equations (ODEs) that are subject to control functions. We state the general framework and motivate the development of OGR, LGR and OLGR, where the latter corresponds to an optimized version of LGR.

Chapter 3 – Reconstruction of matrices in linear systems. In this chapter, we present our analysis of GR and LGR algorithms for the reconstruction of control and drift matrices characterizing linear input/output systems. Our analysis reveals that the performance of GR and LGR strongly depends on the observability and controllability properties of the underlying system.

Chapter 4 – Reconstruction of matrices in bilinear systems. In this chapter, we consider the LGR algorithm for the reconstruction of skew-symmetric matrices characterizing bilinear control systems. The special controllability properties of these systems, related to Lie algebras, enable us to apply an analysis similar to the one in Chapter 3.

Chapter 5 – Analysis for general dynamical systems. In this chapter, we present the first analysis of GR algorithms for nonlinear problems. While LGR is limited to linearized systems that only capture local characteristics of the original nonlinear system, we prove that the results obtained for LGR in Chapters 3 and 4 also hold locally for the standard GR algorithm.

Chapter 6 – Numerical experiments for linear and bilinear systems. Based on our analysis in Chapters 3 and 4, we compare the performance of GR, OGR, LGR and OLGR by numerical experiments using linear and bilinear reconstruction problems. Our results suggest that OGR is the most effective strategy of the four.

Chapter 7 – Reconstruction of parameter distributions in spin systems. In this chapter, we apply our GR algorithms in the field of Nuclear Magnetic Resonance (NMR) to reconstruct the joint probability distribution of two inhomogeneous Hamiltonian parameters. Once again, the controls computed by the optimized version of our GR algorithm turn out to be the best possible ones.

Chapter 8 – Greedy reconstruction algorithms for function approximation. In this chapter, we develop a new class of GR algorithms for function approximation problems. First, we demonstrate the effectiveness of our algorithms by relating them to well-established methods in the field of polynomial interpolation. Afterwards, we unite all our gained knowledge on GR algorithms and introduce NOGR, a novel strategy designed to simultaneously optimize a training data set and find a minimal neural network structure able to learn this data.

Chapter 9 – Conclusions and Outlook. In this final chapter, we draw our conclusions and outline the prospects for future research.

Appendix. The Appendix contains a local convergence result for the Gauss-Newton method used in Chapters 3 and 4.

1.2 Publications

[1] S. Buchwald, G. Ciaramella, and J. Salomon. “Analysis of a greedy reconstruction algorithm”. In: *SIAM J. Control Optim.* 59.6 (2021), pp. 4511–4537.

[2] S. Buchwald, G. Ciaramella, and J. Salomon. “Gauss-Newton oriented greedy algorithms for the reconstruction of operators in nonlinear dynamics”. In: *SIAM J. Control Optim.* 62.3 (2024), pp. 1343–1368.

[3] S. Buchwald, G. Ciaramella, J. Salomon, and D. Sugny. “A `SPIRED` code for the reconstruction of spin distribution”. In: *Computer Physics Communications* 299 (2024), p. 109126. DOI: `https://doi.org/10.1016/j.cpc.2024.109126`.

[4] S. Buchwald, G. Ciaramella, J. Salomon, and D. Sugny. "Greedy reconstruction algorithm for the identification of spin distribution". In: *Phys. Rev. A* 104 (6 Dec. 2021), p. 063112.

[5] S. Buchwald, G. Ciaramella, and M. Verani. "Greedy reconstruction algorithms for function approximation". Submitted, preprint available at `https://mox.polimi.it/reports-and-books/publication-results/?id=1323`.

The publications listed above serve as a foundation of this thesis. All authors contributed equally to the respective work. In particular, the author of the present thesis wrote all numerical codes and conducted all numerical tests. Moreover, this thesis contains a novel analysis for the reconstruction of drift matrices in partially observable and controllable linear systems in Section 3.2.4, and additional numerical experiments for a bilinear reconstruction problem in Section 6.2.

1.3 Notation

Consider a positive natural number N. We denote by $\langle \mathbf{v}, \mathbf{w} \rangle := \bar{\mathbf{v}}^\top \mathbf{w}$, for any $\mathbf{v}, \mathbf{w} \in \mathbb{C}^N$, the usual complex scalar product on $\mathbb{C}^N$, and by $\| \cdot \|_2$ the corresponding norm. For any $A \in \mathbb{C}^{N \times N}$, $\| A \|_2$ is the norm induced by the vector norm $\| \cdot \|_2$, $[A]_{j,k}$ is the j,k (with $j,k \leq N$) entry of A, and the notation $A_{[1:k,1:j]}$ indicates the upper left submatrix of A of size $k \times j$, namely, $[A_{[1:k,1:j]}]_{\ell,m} := [A]_{\ell,m}$ for $\ell = 1, \dots, k$ and $m = 1, \dots, j$. Similarly, $A_{[1:k,j]}$ denotes the column vector in $\mathbb{R}^k$ corresponding to the first k elements of the column j of A. Additionally, $\mathrm{im}(A)$ is the image of A, and $\ker(A)$ its kernel. We indicate by $\mathfrak{so}(N)$ the space of skew-symmetric matrices in $\mathbb{R}^{N \times N}$. Moreover, when talking about symmetric matrices, PD and PSD stand for positive definite and positive semi-definite, respectively. For a real number $x \in \mathbb{R}$ we denote by $\lceil x \rceil$ the ceiling function that maps x to the smallest integer greater than or equal to x.

Dynamical systems. By (A, B, C) we denote the input/output dynamical system

$$\boldsymbol{x}(t) = C\boldsymbol{y}(t), \quad \dot{\boldsymbol{y}}(t) = A\boldsymbol{y}(t) + B\boldsymbol{u}(t), \quad \boldsymbol{y}(0) = \boldsymbol{y}^0. \tag{1.1}$$

The controls $\boldsymbol{u}$ belong to the admissible set U_{ad}, which is assumed to be a non-empty and weakly compact subset of $L^2(0, T; \mathbb{R}^M)$, $M \in \mathbb{N}$. This guarantees well posedness of the optimal control problems considered in this work. Moreover, we assume that U_{ad} contains $\boldsymbol{u} \equiv 0$ as an interior point. This hypothesis is used in our analysis in Section 3.2, and Chapters 4 and 5. For an interval $X \subset \mathbb{R}$, the notation $\phi : X \rightrightarrows \mathbb{R}^N$ indicates that ϕ is a set-valued correspondence, i.e. $\phi(x) \subset \mathbb{R}^N$ is a set for $x \in X$. Finally, $\mathcal{B}_r^N(x) \subset \mathbb{R}^N$ is the ball with radius $r > 0$ and center $x \in \mathbb{R}^N$.

Polynomial interpolation. We denote by $\widetilde{x}_j \in \mathbb{R}$ and $\widetilde{\boldsymbol{x}}_j = [\widetilde{x}_{j,1}, \dots, \widetilde{x}_{j,N}]^\top \in \mathbb{R}^N$ any interpolation point, while using x and $\boldsymbol{x} = [x_1, \dots, x_N]^\top$ for general points in $\mathbb{R}$ and $\mathbb{R}^N$, respectively. For $N, M \in \mathbb{N}$, we let $\mathcal{G} = \mathcal{G}(X, \mathbb{R}^M)$ be a class of functions over some closed and bounded set $X \subset \mathbb{R}^N$. In particular, for $M = 1$ we denote by $\mathcal{P}_n := \mathcal{P}_n(X)$ the space

of polynomials p over X with degree $\deg(p) \leq n$, meaning that

$$p : X \to \mathbb{R}, \quad x \mapsto p(x) = \sum_{|\boldsymbol{\alpha}|=0}^{n} a_{\boldsymbol{\alpha}} \boldsymbol{x}^{\boldsymbol{\alpha}},$$

for coefficients $a_{\boldsymbol{\alpha}} \in \mathbb{R}$ with $\boldsymbol{\alpha} \in \mathbb{N}^N$, $|\boldsymbol{\alpha}| = \sum_{i=1}^{N} \alpha_i$, and

$$\boldsymbol{x}^{\boldsymbol{\alpha}} = \boldsymbol{x}^{(\alpha_1,\dots,\alpha_N)} := x_1^{\alpha_1} \cdots x_N^{\alpha_N}.$$

For a monomial $a_{\boldsymbol{\alpha}} x^{\boldsymbol{\alpha}}$, we call $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_N) \in \mathbb{N}^N$ its component-wise degree.

Neural networks. We denote by $F_{W,b}$ a residual neural network (compare, e.g., [46]) of depth m, with input size $N \in \mathbb{N}$, output size $M \in \mathbb{N}$ and hidden layers of (the same) width $d \in \mathbb{N}$, represented by weights $W = (W^{[1]}, \dots, W^{[m]}) \in \mathbb{R}^{N\times d} \times \cdots \times \mathbb{R}^{d\times M}$ and biases $b = (\boldsymbol{b}^{[1]}, \dots, \boldsymbol{b}^{[m]}) \in \mathbb{R}^d \times \cdots \times \mathbb{R}^d \times \mathbb{R}^M$. We call $\mathcal{W}(\boldsymbol{d}) := (\mathbb{R}^{N\times d} \times \cdots \times \mathbb{R}^{d\times M}) \times (\mathbb{R}^d \times \cdots \times \mathbb{R}^d \times \mathbb{R}^M)$ the joint space of the weight matrices W and the bias vectors b.

Chapter 2

Greedy reconstruction algorithms for dynamical systems

In this chapter, we introduce the three main classes of GR algorithms for operator reconstruction in dynamical systems. We begin by formulating the general setting in Section 2.1, which will serve as a reference throughout the following chapters. In Section 2.2, we state the standard GR algorithm as introduced by Maday and Salomon in [57]. We then present the OGR algorithm in Section 2.3, developed in [1] with the goal of optimizing the performance of GR and removing its dependence on the choice and order of the set $\mathcal{B}$. Finally, in Section 2.4, we discuss the LGR algorithm which was introduced in [2] to prove local convergence results for nonlinear reconstruction problems.

2.1 General setting

Consider the system of ordinary differential equations (ODE)

$$\begin{aligned} \dot{\boldsymbol{y}}(t) &= f(B_\star, \boldsymbol{y}(t), \boldsymbol{u}(t)), \quad t \in (0, T], \\ \boldsymbol{y}(0) &= \boldsymbol{y}^0, \end{aligned} \tag{2.1}$$

where $\boldsymbol{y}(t) \in \mathbb{R}^N$ with $N \in \mathbb{N}$, $\boldsymbol{y}^0 \in \mathbb{R}^N$ and $\boldsymbol{u} \in U_{ad}$. The operator $B_\star$ is unknown and assumed to lie in the linear space spanned by a finite-dimensional set $\mathcal{B} = \{B_1, \dots, B_K\}$, $K \in \mathbb{N}$, and we write $B_\star = \sum_{j=1}^K \beta_{\star,j} B_j =: B(\boldsymbol{\beta}_\star)$. We assume that $f : \operatorname{span}(\mathcal{B}) \times \mathbb{R}^N \times \mathbb{R}^M \to \mathbb{R}^N, (B, \boldsymbol{y}, \boldsymbol{u}) \mapsto f(B, \boldsymbol{y}, \boldsymbol{u})$ is differentiable in B and $\boldsymbol{y}$.

To identify the unknown operator $B_\star$ one uses a set of control functions $\{\boldsymbol{u}^m\}_{m=1}^K \subset U_{ad}$ to perform K laboratory experiments and obtain experimental data $\{\boldsymbol{\varphi}_\star^{data}(\boldsymbol{u}^m)\}_{m=1}^K$. For the purpose of this work, we assume that this data is given by

$$\boldsymbol{\varphi}_\star^{data}(\boldsymbol{u}^m) := C\boldsymbol{y}(B_\star, \boldsymbol{u}^m; T), \text{ for } m = 1, \dots, K, \tag{2.2}$$

where $\boldsymbol{y}(B_\star, \boldsymbol{u}; T)$ denotes the solution to (2.1) at time $T > 0$, corresponding to the operator $B_\star$ and a control function $\boldsymbol{u}$. The matrix $C \in \mathbb{R}^{P \times N}$ $(P \leq N)$ is a given observer matrix. The measurements are assumed not to be affected by noise.

Using the set $\{\boldsymbol{u}^m\}_{m=1}^K$ and the data $\{\boldsymbol{\varphi}_\star^{data}(\boldsymbol{u}^m)\}_{m=1}^K \subset \mathbb{R}^P$, the unknown vector $\boldsymbol{\beta}_\star$ is obtained by solving the least-squares problem

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^K} \frac{1}{2}\sum_{m=1}^{K} \|\boldsymbol{\varphi}(B(\boldsymbol{\beta}),\boldsymbol{u}^m) - \boldsymbol{\varphi}_\star^{data}(\boldsymbol{u}^m)\|_2^2, \tag{2.3}$$

where

$$\boldsymbol{\varphi}(B(\boldsymbol{\beta}),\boldsymbol{u}^m) := C\boldsymbol{y}(B(\boldsymbol{\beta}),\boldsymbol{u}^m;T). \tag{2.4}$$

Clearly $\boldsymbol{\beta}_\star$ is a global minimizer for (2.3).

In the presented setting, several variables are used. Let us clarify their roles in plain words:

- The elements of the set $\mathcal{B}$ can be arbitrarily chosen as data, but are assumed to be such that $B_\star \in \operatorname{span}\mathcal{B}$.
- Given a set $\mathcal{B}$, the true unknown of the reconstruction problem (2.3) is $\boldsymbol{\beta}_\star$ (or equivalently $B_\star$).
- The control functions $\{\boldsymbol{u}^m\}_{m=1}^K$ determine the experiments that are performed to produce the laboratory data $\{\boldsymbol{\varphi}_\star^{data}(\boldsymbol{u}^m)\}_{m=1}^K$, defined in (2.2). This data is necessary to assemble the (online) reconstruction problem (2.3).

If the control functions $\{\boldsymbol{u}^m\}_{m=1}^K$ and the data $\{\boldsymbol{\varphi}_\star^{data}(\boldsymbol{u}^m)\}_{m=1}^K$ are given, problem (2.3) is a standard parameter-identification inverse problem written in a least-squares minimization form. However, the choice of the control functions $\{\boldsymbol{u}^m\}_{m=1}^K$ can affect significantly the properties of problem (2.3) and the corresponding solutions.

2.2 The greedy reconstruction algorithm

To design an optimized set of control functions, in particular with the goal of improving local convexity properties of (2.3), Maday and Salomon introduced in [57] a numerical strategy which separates the reconstruction process in offline and online phases. During the offline phase, a GR algorithm computes a set of optimized control functions $\{\boldsymbol{u}^m\}_{m=1}^K$ by exploiting only the theoretical model (2.1), without using any laboratory data. In the online phase, these controls are used experimentally to produce the laboratory data $\{\boldsymbol{\varphi}_\star^{data}(\boldsymbol{u}^m)\}_{m=1}^K$ and to define the online reconstruction problem (2.3). Thus, the online phase consists (mathematically) in solving an inverse problem.

The ideal goal of this offline/online framework is to find a good approximation of the unknown operator, such that the difference at time T between observed experimental data and numerically computed data is minimized for any control $\boldsymbol{u}$. In other words, one aims at finding an operator B that solves

$$\min_{B\in\operatorname{span}\mathcal{B}} \max_{\boldsymbol{u}\in U_{ad}} \|\boldsymbol{\varphi}(B,\boldsymbol{u}) - \boldsymbol{\varphi}_\star^{data}(\boldsymbol{u})\|_2^2,$$

or equivalently an $\boldsymbol{\beta}$ that solves

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^K} \max_{\boldsymbol{u}\in U_{ad}} \|\boldsymbol{\varphi}(B(\boldsymbol{\beta}),\boldsymbol{u}) - \boldsymbol{\varphi}_\star^{data}(\boldsymbol{u})\|_2^2. \tag{2.5}$$

Algorithm 1 Greedy Reconstruction Algorithm (GR)

Require: A set of linearly independent operators $\mathcal{B} = \{B_1, \dots, B_K\}$.

1: Set $k = 1$ and find the control $\boldsymbol{u}^1$ that solves

$$\max_{\boldsymbol{u} \in U_{ad}} \|\boldsymbol{\varphi}(0, \boldsymbol{u}) - \boldsymbol{\varphi}(B_1, \boldsymbol{u})\|_2^2, \tag{2.7}$$

where $\boldsymbol{\varphi}(B, \boldsymbol{u})$ is defined in (2.4).

2: **while** $k \leq K - 1$ **do**

3: Fitting step: Find $\boldsymbol{\beta}^k = (\beta_j^k)_{j=1,\dots,k} \in \mathbb{R}^k$ that solves

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^k} \sum_{m=1}^{k} \|\boldsymbol{\varphi}(B^{(k)}(\boldsymbol{\beta}), \boldsymbol{u}^m) - \boldsymbol{\varphi}(B_{k+1}, \boldsymbol{u}^m)\|_2^2, \tag{2.8}$$

where $B^{(k)}(\boldsymbol{\beta}) := \sum_{j=1}^{k} \beta_j B_j$.

4: Splitting step: Find $\boldsymbol{u}^{k+1}$ that solves

$$\max_{\boldsymbol{u} \in U_{ad}} \|\boldsymbol{\varphi}(B^{(k)}(\boldsymbol{\beta}^k), \boldsymbol{u}) - \boldsymbol{\varphi}(B_{k+1}, \boldsymbol{u})\|_2^2. \tag{2.9}$$

5: Update $k \leftarrow k + 1$.

6: **end while**

Therefore, the goal of the GR algorithm is to generate a set of control functions $\{\boldsymbol{u}^m\}_{m=1}^K$ such that a computed solution to (2.3) also solves (2.5).

Since the GR algorithm does not have access to any laboratory data, we must assume that $\boldsymbol{\beta}^\star$ could be any vector in $\mathbb{R}^K$. To ensure that (2.5) has a unique solution $\boldsymbol{\beta} = \boldsymbol{\beta}^\star$ for any $\boldsymbol{\beta}^\star \in \mathbb{R}^K$, it must hold that

$$\forall \widetilde{\boldsymbol{\beta}}, \widehat{\boldsymbol{\beta}} \in \mathbb{R}^K,\ \widetilde{\boldsymbol{\beta}} \neq \widehat{\boldsymbol{\beta}},\ \exists \boldsymbol{u} \in U_{ad}\ :\ \boldsymbol{\varphi}(B(\widetilde{\boldsymbol{\beta}}), \boldsymbol{u}) \neq \boldsymbol{\varphi}(B(\widehat{\boldsymbol{\beta}}), \boldsymbol{u}).$$

Thus, the controls $\{\boldsymbol{u}^m\}_{m=1}^K$ generated by the GR algorithm must attempt to satisfy

$$\forall \widetilde{\boldsymbol{\beta}}, \widehat{\boldsymbol{\beta}} \in \mathbb{R}^K,\ \widetilde{\boldsymbol{\beta}} \neq \widehat{\boldsymbol{\beta}},\ \exists m \in \{1, \dots, K\}\ :\ \boldsymbol{\varphi}(B(\widetilde{\boldsymbol{\beta}}), \boldsymbol{u}^m) \neq \boldsymbol{\varphi}(B(\widehat{\boldsymbol{\beta}}), \boldsymbol{u}^m). \tag{2.6}$$

Following this idea, Maday and Salomon defined the GR algorithm as an iterative procedure that performs a sweep over the set $\mathcal{B} = \{B_k\}_{k=1}^K$ and computes a new control $\boldsymbol{u}^k$ at each iteration. The full GR algorithm is stated in Algorithm 1. The goal of this strategy is that, at iteration $k \in \{1, \dots, K\}$, the computed controls $\{\boldsymbol{u}^m\}_{m=1}^k$ satisfy

$$\forall \widetilde{\boldsymbol{\beta}}, \widehat{\boldsymbol{\beta}} \in \mathbb{R}^k,\ \widetilde{\boldsymbol{\beta}} \neq \widehat{\boldsymbol{\beta}},\ \exists m \in \{1, \dots, k\}\ :\ \boldsymbol{\varphi}(B^{(k)}(\widetilde{\boldsymbol{\beta}}), \boldsymbol{u}^m) \neq \boldsymbol{\varphi}(B^{(k)}(\widehat{\boldsymbol{\beta}}), \boldsymbol{u}^m), \tag{2.10}$$

where $B^{(k)}(\boldsymbol{\beta}) = \sum_{j=1}^{k} \beta_j B_j$. This is achieved by a series of alternating *fitting* and *splitting* steps that mimic the structure of the min-max problem (2.5).

Since the analysis performed in the following chapters focuses on Algorithm 1, let us explain it in more details. The initialization is performed by solving the optimal control problem (2.7), which aims to maximize the distance (at time T) between the observed states of the two systems

$$\begin{cases} \dot{\boldsymbol{y}}(t) = f(0, \boldsymbol{y}(t), \boldsymbol{u}(t)), \\ \boldsymbol{y}(0) = \boldsymbol{y}^0, \end{cases} \qquad \begin{cases} \dot{\boldsymbol{y}}(t) = f(B_1, \boldsymbol{y}(t), \boldsymbol{u}(t)), \\ \boldsymbol{y}(0) = \boldsymbol{y}^0. \end{cases}$$

The numerical solution of this maximization problem provides the first control function $\boldsymbol{u}^1$.

Assume now that the control functions $\boldsymbol{u}^1, \dots, \boldsymbol{u}^k$ have been computed. The new control $\boldsymbol{u}^{k+1}$ is obtained by performing a fitting step (namely solving problem (2.8)) and a splitting step (namely solving problem (2.9)). In the fitting step, one compares the two systems

$$\begin{cases} \dot{\boldsymbol{y}}(t) = f(\sum_{j=1}^{k} \beta_j B_j, \boldsymbol{y}(t), \boldsymbol{u}^m(t)), \\ \boldsymbol{y}(0) = \boldsymbol{y}^0, \end{cases} \qquad \begin{cases} \dot{\boldsymbol{y}}(t) = f(B_{k+1}, \boldsymbol{y}(t), \boldsymbol{u}^m(t)), \\ \boldsymbol{y}(0) = \boldsymbol{y}^0, \end{cases}$$

for $m \in \{1, \dots, k\}$, and looks for an $\boldsymbol{\beta} \in \mathbb{R}^k$ for which their observed solutions at time T are as similar as possible (ideally identical and hence indistinguishable). We denote by $\boldsymbol{\beta}^k = [\beta_1^k, \dots, \beta_k^k]^\top$ the vector computed by solving (2.8). This vector is used in the subsequent splitting step, which consists in solving the optimal control problem (2.9). Here, we compute a control function $\boldsymbol{u}^{k+1}$ that maximizes the distance (at time T) between the solutions of the two systems

$$\begin{cases} \dot{\boldsymbol{y}}(t) = f(\sum_{j=1}^{k} \beta_j^k B_j, \boldsymbol{y}(t), \boldsymbol{u}(t)), \\ \boldsymbol{y}(0) = \boldsymbol{y}^0, \end{cases} \qquad \begin{cases} \dot{\boldsymbol{y}}(t) = f(B_{k+1}, \boldsymbol{y}(t), \boldsymbol{u}(t)), \\ \boldsymbol{y}(0) = \boldsymbol{y}^0, \end{cases}$$

where now $\boldsymbol{\beta}^k$ is a fixed vector of coefficients and the optimization variable is the control function $\boldsymbol{u}$.

2.3 The optimized greedy reconstruction algorithm

The choice and ordering of the set $\mathcal{B} = \{B_1, \dots, B_K\}$ significantly influence the control functions generated by the standard GR Algorithm 1. Indeed, GR will always compute exactly K control functions, even if the sufficient condition (2.6) for the reconstruction problem (2.3) to be uniquely solvable is already satisfied by the first $k < K$ controls.

To reduce the dependency on $\mathcal{B}$ and to increase efficiency with respect to the number of computed controls, we introduce the OGR Algorithm 2. This new strategy extends the greedy nature of the original GR algorithm to the choice of the next element in $\mathcal{B}$. During each iteration, all remaining elements in $\mathcal{B}$ are considered as potential candidates. We then choose the one that yields the largest function value in the respective splitting step. In other words, we identify the basis element for which one can split the observation the most from all previous basis elements. It is important to remark that at each iteration, multiple fitting-step and splitting-step problems are solved. However, these computations can be performed in parallel since each individual problem is independent of the other ones.

OGR also introduces two new features related to the two tolerances tol_{fit} and tol_{split}, intended to further increase its efficiency. The first feature can be found in Lines 8-9. At iteration k, after all fitting step problems have been solved, we check whether there exists $\ell \in \{k+1, \dots, K\}$ for which the optimal cost function value is different from zero (i.e., larger than tol_{fit}). If this is the case, then there exists a control $\boldsymbol{u}^m$, $m \in \{1, \dots, \widetilde{k}\}$, that already satisfies $\boldsymbol{\varphi}(B^{(k)}(\boldsymbol{\beta}^\ell), \boldsymbol{u}^m) \neq \boldsymbol{\varphi}(B_\ell, \boldsymbol{u}^m; T)$. Thus, recalling (2.10), we can add the element B_ℓ to the already selected ones without computing a new control. The second

Algorithm 2 Optimized Greedy Reconstruction Algorithm (OGR)

Require: A set of operators $\mathcal{B} = \{B_1, \dots, B_K\}$ for $K \in \mathbb{N}$, and two tolerances $\text{tol}_{fit}, \text{tol}_{split} > 0$.

1: Find $\boldsymbol{u}^1$ and ℓ_1 that solve the initialization problem

$$\max_{\ell \in \{1,\dots,K\}} \max_{\boldsymbol{u} \in U_{ad}} \|\boldsymbol{\varphi}(0, \boldsymbol{u}) - \boldsymbol{\varphi}(B_\ell, \boldsymbol{u})\|_2^2. \tag{2.11}$$

2: Swap B_1 and B_{ℓ_1} in $\mathcal{B}$, and set $k = 1$, $\widetilde{k} = 1$ and $f_{split} = \|\boldsymbol{\varphi}(0, \boldsymbol{u}^1) - \boldsymbol{\varphi}(B_1, \boldsymbol{u}^1)\|_2^2$.

3: **while** $k \leq K - 1$ and $f_{split} \geq \text{tol}_{split}$ **do**

4: (Optional: Orthogonalize all elements $\{B_{k+1}, \dots, B_K\}$ with respect to $\{B_1, \dots, B_k\}$, remove any that are linearly dependent and update K accordingly.)

5: **for** $\ell = k+1, \dots, K$ **do**

6: Fitting step: Find $(\beta_j^\ell)_{j=1,\dots,k}$ that solve the problem

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^k} \sum_{m=1}^{\widetilde{k}} \|\boldsymbol{\varphi}(B^{(k)}(\boldsymbol{\beta}), \boldsymbol{u}^m) - \boldsymbol{\varphi}(B_\ell, \boldsymbol{u}^m; T)\|_2^2, \tag{2.12}$$

and set $f_\ell = \sum_{m=1}^{\widetilde{k}} \|\boldsymbol{\varphi}(B^{(k)}(\boldsymbol{\beta}^\ell), \boldsymbol{u}^m) - \boldsymbol{\varphi}(B_\ell, \boldsymbol{u}^m)\|_2^2$.

7: **end for**

8: **if** $\max_{\ell=k+1,\dots,K} f_\ell > \text{tol}_{fit}$ **then**

9: Set $\ell_{k+1} = \arg\max_{\ell=k+1,\dots,K} f_\ell$.

10: **else**

11: Splitting step: Find $\boldsymbol{u}^{\widetilde{k}+1}$ and ℓ_{k+1} that solve the problem

$$\max_{\ell \in \{k+1,\dots,K\}} \max_{\boldsymbol{u} \in U_{ad}} \|\boldsymbol{\varphi}(B^{(k)}(\boldsymbol{\beta}^\ell), \boldsymbol{u}) - \boldsymbol{\varphi}(B_\ell, \boldsymbol{u})\|_2^2. \tag{2.13}$$

12: Update $\widetilde{k} \leftarrow \widetilde{k} + 1$.

13: **end if**

14: Swap B_{k+1} and $B_{\ell_{k+1}}$ in $\mathcal{B}$.

15: Update $f_{split} \leftarrow \|\boldsymbol{\varphi}(B^{(k)}(\boldsymbol{\beta}^{\ell_{k+1}}), \boldsymbol{u}^{k+1}) - \boldsymbol{\varphi}(B_{k+1}, \boldsymbol{u}^{k+1})\|_2^2$ and $k \leftarrow k + 1$.

16: **end while**

feature can be found in Line 3. The algorithm terminates if the newly computed control is not able to split the observations, meaning that the cost function value of the splitting step is smaller than tol_{split}. This implies that the effect of any remaining element in $\mathcal{B}$ on the observations (2.4) is indistinguishable from the already selected ones.

Finally, we note that OGR returns not only an efficient set of controls $\{\boldsymbol{u}^m\}_{m=1}^{\widetilde{k}}$ but also an optimized (i.e, reordered and possibly reduced) set $\mathcal{B}$.

2.4 Linearized greedy reconstruction algorithms

This class of methods is inspired directly from the numerical process of solving the online reconstruction problem (2.3). A typical iterative strategy to solve nonlinear problems of the form (2.3) is the Gauss-Newton method (GN). Its process is initialized by a vector which we will call $\boldsymbol{\beta}_\circ \in \mathbb{R}^K$. We denote by $\boldsymbol{\beta}_c \in \mathbb{R}^K$ the GN iterate, and define $f_m(\boldsymbol{\beta}) :=$

$\frac{1}{2}\sum_{i=1}^{P}([R_m(\boldsymbol{\beta})]_i)^2 = \frac{1}{2}R_m(\boldsymbol{\beta})^\top R_m(\boldsymbol{\beta})$, where

$$R_m(\boldsymbol{\beta}) := \boldsymbol{\varphi}(B(\boldsymbol{\beta}), \boldsymbol{u}^m) - \boldsymbol{\varphi}_\star^{data}(\boldsymbol{u}^m), \quad \text{for } m \in \{1, \dots, K\}. \tag{2.14}$$

Thus, the online reconstruction problem (2.3) is equivalent to

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^K} \sum_{m=1}^{K} f_m(\boldsymbol{\beta}). \tag{2.15}$$

Given an iterate $\boldsymbol{\beta}_c \in \mathbb{R}^K$ and the Jacobian $R_m'(\boldsymbol{\beta}_c) \in \mathbb{R}^{P\times K}$ of R_m at $\boldsymbol{\beta}_c$, GN finds the new iterate by solving

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^K} \frac{1}{2}\sum_{m=1}^{K} \|R_m'(\boldsymbol{\beta}_c)(\boldsymbol{\beta}-\boldsymbol{\beta}_c) - R_m(\boldsymbol{\beta}_c)\|_2^2, \tag{2.16}$$

whose first-order necessary optimality condition is

$$\widehat{W}_c\boldsymbol{\beta} = \sum_{m=1}^{K} R_m'(\boldsymbol{\beta}_c)^\top R_m(\boldsymbol{\beta}_c), \tag{2.17}$$

where $\widehat{W}_c := \sum_{m=1}^{K} R_m'(\boldsymbol{\beta}_c)^\top R_m'(\boldsymbol{\beta}_c) \in \mathbb{R}^{K\times K}$ is symmetric PSD. Now, we recall the following convergence result from [51, Thm. 2.4.1] (for a proof see also Appendix A).

Lemma 2.4.1 (local convergence of GN) *Let $\boldsymbol{\beta}_\star$ be a minimizer of* (2.15) *such that for all $m \in \{1, \dots, K\}$ the function R_m is Lipschitz continuously differentiable near $\boldsymbol{\beta}_\star$ and $R_m(\boldsymbol{\beta}_\star) = 0$. If the initialization vector $\boldsymbol{\beta}_\circ \in \mathbb{R}^K$ is sufficiently close to $\boldsymbol{\beta}_\star$, and $\widehat{W}_c$ is PD for all iterates $\boldsymbol{\beta}_c \in \mathbb{R}^K$, then GN converges lcoally quadratic to $\boldsymbol{\beta}_\star$.*

Lemma 2.4.1 implies that, given an initialization vector $\boldsymbol{\beta}_\circ$ sufficiently close to the solution $\boldsymbol{\beta}_\star$, the functions $\{\boldsymbol{u}^m\}_{m=1}^K$ should be chosen such that the GN matrix

$$\widehat{W}_c = \sum_{m=1}^{K} R_{\boldsymbol{u}^m}'(\boldsymbol{\beta}_c)^\top R_{\boldsymbol{u}^m}'(\boldsymbol{\beta}_c)$$

is PD for all $\boldsymbol{\beta}_c \in \mathbb{R}^K$ in a neighborhood of $\boldsymbol{\beta}_\star$. Notice that $\widehat{W}_c$ being PD is equivalent to (2.16) being uniquely solvable. Using (2.14) and (2.4), we can write (2.16) more explicitly. For a direction $\delta\boldsymbol{\beta} \in \mathbb{R}^K$, we have

$$R_m'(\boldsymbol{\beta}_c)(\delta\boldsymbol{\beta}) = C\delta\boldsymbol{y}_c(B(\delta\boldsymbol{\beta}), \boldsymbol{u}^m; T), \tag{2.18}$$

where $\delta\boldsymbol{y}_c(B(\delta\boldsymbol{\beta}), \boldsymbol{u}^m; T)$ denotes the solution at time T to the linearized equation

$$\begin{cases} \dot{\delta\boldsymbol{y}}_c = \partial_{\boldsymbol{y}} f(B(\boldsymbol{\beta}_c), \boldsymbol{y}_c, \boldsymbol{u})\delta\boldsymbol{y}_c + \sum_{j=1}^{K} \delta\beta_j \Big(\partial_B f(B(\boldsymbol{\beta}_c), \boldsymbol{y}_c, \boldsymbol{u})(B_j)\Big), \ \delta\boldsymbol{y}_c(0) = 0, \\ \dot{\boldsymbol{y}}_c = f(B(\boldsymbol{\beta}_c), \boldsymbol{y}_c, \boldsymbol{u}), \quad \boldsymbol{y}_c(0) = \boldsymbol{y}^0, \end{cases} \tag{2.19}$$

at $\boldsymbol{u} = \boldsymbol{u}^m$. Hence, problem (2.16) can be written as

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^K} \sum_{m=1}^{K} \|C\delta\boldsymbol{y}_c(B(\boldsymbol{\beta}-\boldsymbol{\beta}_c), \boldsymbol{u}^m; T) - R_m(\boldsymbol{\beta}_c)\|_2^2. \tag{2.20}$$

Algorithm 3 Linearized Greedy Reconstruction Algorithm (LGR)

Require: A set of linearly independent operators $\mathcal{B} = \{B_1, \dots, B_K\}$.

1: Set $k = 1$ and find the control $\boldsymbol{u}^1$ that solves

$$\max_{\boldsymbol{u}\in U_{ad}} \|C\delta\boldsymbol{y}_\circ(B_1, \boldsymbol{u}; T)\|_2^2. \tag{2.23}$$

2: **while** $k \leq K-1$ **do**

3: Fitting step: Find $(\beta_j^k)_{j=1,\dots,k}$ that solves

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^k} \sum_{m=1}^{k} \|C\delta\boldsymbol{y}_\circ(B^{(k)}(\boldsymbol{\beta}), \boldsymbol{u}^m; T) - C\delta\boldsymbol{y}_\circ(B_{k+1}, \boldsymbol{u}^m; T)\|_2^2. \tag{2.24}$$

4: Splitting step: Find $\boldsymbol{u}^{k+1}$ that solves

$$\max_{\boldsymbol{u}\in U_{ad}} \|C\delta\boldsymbol{y}_\circ(B^{(k)}(\boldsymbol{\beta}^k), \boldsymbol{u}; T) - C\delta\boldsymbol{y}_\circ(B_{k+1}, \boldsymbol{u}; T)\|_2^2. \tag{2.25}$$

5: Update $k \leftarrow k+1$.

6: **end while**

The vectors $R_m(\boldsymbol{\beta}_c) \in \mathbb{R}^P$ are independent of $\boldsymbol{\beta}$ and can be considered as fixed data when solving (2.20).

Now, we recall that the GR algorithm was designed to generate controls $\{\boldsymbol{u}^m\}_{m=1}^K$ that make problems of the form (2.20) uniquely solvable. Thus, let us assume to be provided with a vector $\boldsymbol{\beta}_\circ$ sufficiently close to $\boldsymbol{\beta}_\star$ (recall from Section 2.1 that $B_\star = B(\boldsymbol{\beta}_\star)$). Further, let $\delta\boldsymbol{y}_\circ(B(\boldsymbol{\beta}-\boldsymbol{\beta}_\circ), \boldsymbol{u}^m; T)$ denote the solution at time T to

$$\begin{cases} \dot{\delta\boldsymbol{y}}_\circ = \partial_{\boldsymbol{y}} f(B(\boldsymbol{\beta}_\circ), \boldsymbol{y}_\circ, \boldsymbol{u})\delta\boldsymbol{y}_\circ + \sum_{j=1}^K (\beta_j - \beta_{\circ,j})\Big(\partial_B f(B(\boldsymbol{\beta}_\circ), \boldsymbol{y}_\circ, \boldsymbol{u})(B_j)\Big), \ \delta\boldsymbol{y}_\circ(0) = 0, \\ \dot{\boldsymbol{y}}_\circ = f(B(\boldsymbol{\beta}_\circ), \boldsymbol{y}_\circ, \boldsymbol{u}), \quad \boldsymbol{y}_\circ(0) = \boldsymbol{y}^0. \end{cases} \tag{2.21}$$

Then, the goal is to generate control functions $\{\boldsymbol{u}^m\}_{m=1}^K$ such that (2.20) in $\boldsymbol{\beta}_c = \boldsymbol{\beta}_\circ$, that is

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^K} \sum_{m=1}^{K} \|C\delta\boldsymbol{y}_\circ(B(\boldsymbol{\beta}-\boldsymbol{\beta}_\circ), \boldsymbol{u}^m; T) - R_m(\boldsymbol{\beta}_\circ)\|_2^2, \tag{2.22}$$

is uniquely solvable. In Section 3.2.5, we show that if (2.22) is uniquely solvable, the same holds for (2.20) at all $\boldsymbol{\beta}$ in a neighborhood of $\boldsymbol{\beta}_\circ$. Thus, if $\boldsymbol{\beta}_\circ$ is an initialization vector for GN, then (2.20) is uniquely solvable for all iterates $\boldsymbol{\beta}_c$ of GN.

The set $\{\boldsymbol{u}^m\}_{m=1}^K$ is computed by Algorithm 3. Since it is related to the linearized equations (2.21), we call it Linearized Greedy Reconstruction algorithm (LGR). Notice that $C\delta\boldsymbol{y}_\circ(0, \boldsymbol{u}; T) = 0$ for any $\boldsymbol{u}$ and therefore $C\delta\boldsymbol{y}_\circ(0, \boldsymbol{u}; T) - C\delta\boldsymbol{y}_\circ(B_1, \boldsymbol{u}; T) = C\delta\boldsymbol{y}_\circ(B_1, \boldsymbol{u}; T)$.

Analogously to the idea of OGR in Section 2.3, we can also formulate an optimized version of LGR, called OLGR, by replacing $\boldsymbol{\varphi}(B, \boldsymbol{u})$ in Algorithm 2 with the linearization $C\delta\boldsymbol{y}_\circ(B, \boldsymbol{u}; T)$.

Chapter 3

Reconstruction of matrices in linear systems

In this chapter, we develop the main analysis of the GR algorithms for dynamical systems. We consider a simple linear input/output system

$$\dot{\boldsymbol{y}}(t) = A\boldsymbol{y}(t) + B\boldsymbol{u}(t),\ t \in (0,T], \quad \boldsymbol{y}(0) = \boldsymbol{y}^0, \tag{3.1}$$

characterized by a drift matrix $A \in \mathbb{R}^{N\times N}$, a control matrix $B \in \mathbb{R}^{N\times M}$, $N, M \in \mathbb{N}^+$, an initial condition $\boldsymbol{y}^0 \in \mathbb{R}^N$ and a control $\boldsymbol{u} \in U_{ad}$.

We begin with the reconstruction of the control matrix B in Section 3.1. Since the solution of (3.1) is linear in B, the corresponding online reconstruction problem (2.3) becomes quadratic in $\boldsymbol{\beta}$. This allows us to derive sufficient conditions for the controls generated by GR to ensure that (2.3) is (globally) strictly convex. These conditions mainly revolve around the observability properties of the system (3.1). Therefore, we divide our analysis into fully observable systems, discussed in Section 3.1.2, and partially observable systems, discussed in Section 3.1.3. We also provide a brief analysis of OGR, revealing a lower bound on the number of controls required to ensure strict convexity of (2.3).

Afterwards, we discuss the reconstruction of the drift matrix A in Section 3.2. Since A acts on the state $\boldsymbol{y}$, the corresponding online reconstruction problem (2.3) becomes nonlinear in $\boldsymbol{\beta}$. Hence, we apply the linearized approach of LGR, as presented in Section 2.4. We show that sufficient conditions for local convergence of GN involve both, observability and controllability properties of the system (3.1). Thus, we again split our analysis into fully observable and controllable systems, discussed in Section 3.2.3, and partially observable and controllable systems, discussed in Section 3.2.4.

3.1 Reconstruction of a control matrix

Consider (2.1) with $f(B, \boldsymbol{y}, \boldsymbol{u}) = A\boldsymbol{y} + B\boldsymbol{u}$, that is,

$$\dot{\boldsymbol{y}}(t) = A\boldsymbol{y}(t) + B_\star\boldsymbol{u}(t),\ t \in (0,T], \quad \boldsymbol{y}(0) = \boldsymbol{y}^0, \tag{3.2}$$

where $A \in \mathbb{R}^{N\times N}$ is a given matrix for $N \in \mathbb{N}^+$, $\boldsymbol{y}^0 \in \mathbb{R}^N$, and $\boldsymbol{u} \in U_{ad}$. The control matrix $B_\star \in \mathbb{R}^{N\times M}$, $M \in \mathbb{N}^+$, is unknown and assumed to lie in the space spanned by a finite set of linearly independent matrices $\mathcal{B} = \{B_1, \dots, B_K\} \subset \mathbb{R}^{N\times M}$, $1 \le K \le NM$, and we write $B_\star = \sum_{j=1}^K \beta_{\star,j} B_j =: B(\boldsymbol{\beta}_\star)$.

Now, we apply the strategy introduced in Sections 2.1-2.2, i.e., use the GR Algorithm 1 to generate a set of control functions $\{\boldsymbol{u}^m\}_{m=1}^K \subset U_{ad}$ that are used to generate the experimental data

$$\boldsymbol{\varphi}_\star^{data}(\boldsymbol{u}^m) = C\boldsymbol{y}(B_\star, \boldsymbol{u}^m; T), \qquad m = 1, \dots, K,$$

where $C \in \mathbb{R}^{P\times N}$ is the observer matrix (compare (2.2)), and then solve the online reconstruction problem (2.3).

3.1.1 Matrix-vector formulation of GR

Our analysis begins by proving the following lemma, which gives us an equivalent matrix-vector formulation of the online reconstruction problem (2.3).

Lemma 3.1.1 (Online reconstruction problem in matrix form) *Problem* (2.3) *for the linear system* (3.2) *is equivalent to*

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^K} \langle \boldsymbol{\beta} - \boldsymbol{\beta}_\star, \widehat{W}(\boldsymbol{\beta} - \boldsymbol{\beta}_\star)\rangle, \tag{3.3}$$

where $\widehat{W} \in \mathbb{R}^{K\times K}$ *is defined as*

$$\widehat{W} := \sum_{m=1}^K W(\boldsymbol{u}^m), \tag{3.4}$$

with $W(\boldsymbol{u}^m) \in \mathbb{R}^{K\times K}$ *given by*

$$[W(\boldsymbol{u}^m)]_{\ell,j} := \langle \boldsymbol{\gamma}_\ell(\boldsymbol{u}^m), \boldsymbol{\gamma}_j(\boldsymbol{u}^m)\rangle, \ \textit{for } \ell, j = 1, \dots, K, \tag{3.5}$$

$$\boldsymbol{\gamma}_\ell(\boldsymbol{u}^m) := \int_0^T Ce^{(T-s)A} B_\ell \boldsymbol{u}^m(s) ds, \ \textit{for } m, \ell = 1, \dots, K. \tag{3.6}$$

Proof. Define $J(\boldsymbol{\beta}) := \sum_{m=1}^K \|\boldsymbol{\varphi}(B(\boldsymbol{\beta}), \boldsymbol{u}^m) - \boldsymbol{\varphi}_\star^{data}(\boldsymbol{u}^m)\|_2^2$ and notice that

$$\boldsymbol{\varphi}(B(\boldsymbol{\beta}), \boldsymbol{u}^m) = C\boldsymbol{y}(B(\boldsymbol{\beta}), \boldsymbol{u}^m; T) = Ce^{TA}\boldsymbol{y}^0 + \int_0^T Ce^{(T-s)A} B(\boldsymbol{\beta})\boldsymbol{u}^m(s) ds,$$

$$\boldsymbol{\varphi}_\star^{data}(\boldsymbol{u}^m) = C\boldsymbol{y}(B_\star, \boldsymbol{u}^m; T) = Ce^{TA}\boldsymbol{y}^0 + \int_0^T Ce^{(T-s)A} B(\boldsymbol{\beta}_\star)\boldsymbol{u}^m(s) ds.$$

Recalling that $B(\boldsymbol{\beta}) = \sum_{j=1}^K \beta_j B_j$, the function $J(\boldsymbol{\beta})$ can be written as

$$\begin{aligned} J(\boldsymbol{\beta}) &= \sum_{m=1}^K \Big\| \int_0^T Ce^{(T-s)A} \Big(\sum_{j=1}^K (\beta_j - \beta_{\star,j}) B_\ell \Big) \boldsymbol{u}^m(s) ds \Big\|_2^2 \\ &= \sum_{m=1}^K \sum_{\ell=1}^K \sum_{j=1}^K (\beta_\ell - \beta_{\star,\ell})(\beta_j - \beta_{\star,j}) \langle \boldsymbol{\gamma}_\ell(\boldsymbol{u}^m), \boldsymbol{\gamma}_j(\boldsymbol{u}^m)\rangle, \end{aligned}$$

where the vectors $\boldsymbol{\gamma}_\ell(\boldsymbol{u}^m)$ are defined in (3.6). We can now write

$$J(\boldsymbol{\beta}) = \sum_{\ell=1}^{K}\sum_{j=1}^{K}(\beta_\ell - \beta_{\star,\ell})(\beta_j - \beta_{\star,j})\sum_{m=1}^{K}\langle\boldsymbol{\gamma}_\ell(\boldsymbol{u}^m), \boldsymbol{\gamma}_j(\boldsymbol{u}^m)\rangle$$
$$= \Big\langle \boldsymbol{\beta} - \boldsymbol{\beta}_\star, \sum_{m=1}^{K} W(\boldsymbol{u}^m)(\boldsymbol{\beta} - \boldsymbol{\beta}_\star)\Big\rangle = \langle \boldsymbol{\beta} - \boldsymbol{\beta}_\star, \widehat{W}(\boldsymbol{\beta} - \boldsymbol{\beta}_\star)\rangle,$$

and the result follows. □

Notice that, for any $m = 1, \dots, K$, the matrix $W(\boldsymbol{u}^m)$ defined in (3.5) can be written as $W(\boldsymbol{u}^m) = \Gamma(\boldsymbol{u}^m)^\top\Gamma(\boldsymbol{u}^m)$, where $\Gamma(\boldsymbol{u}^m) = [\boldsymbol{\gamma}_1(\boldsymbol{u}^m) \cdots \boldsymbol{\gamma}_K(\boldsymbol{u}^m)]$. Hence, $W(\boldsymbol{u}^m)$ is symmetric and PSD. This guarantees that $\widehat{W}$ is also symmetric and PSD. Therefore, problem (3.3) is uniquely solved by $\boldsymbol{\beta} = \boldsymbol{\beta}_\star$ if and only if $\widehat{W}$ is PD, meaning that the GR algorithm actually aims at computing a set of control functions $\{\boldsymbol{u}^m\}_{m=1}^K$ that makes $\widehat{W}$ PD. We then need to study how the positivity of $\widehat{W}$ evolves during the iteration of the algorithm. To do so, the first step is to rewrite the problems (2.7), (2.8) and (2.9) also in a matrix form.

Lemma 3.1.2 (GR in matrix form) *Consider Algorithm 1. It holds that:*

- *The initialization problem* (2.7) *is equivalent to*

$$\max_{\boldsymbol{u}\in U_{ad}} [W(\boldsymbol{u})]_{1,1}. \tag{3.7}$$

- *The fitting-step problem* (2.8) *is equivalent to*

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^k} \langle\boldsymbol{\beta}, \widehat{W}^{(k)}_{[1:k,1:k]}\boldsymbol{\beta}\rangle - 2\langle\widehat{W}^{(k)}_{[1:k,k+1]}, \boldsymbol{\beta}\rangle, \tag{3.8}$$

  where $\widehat{W}^{(k)} = \sum_{m=1}^{k} W(\boldsymbol{u}^m)$*, and (recalling Section 1.3)* $\widehat{W}^{(k)}_{[1:k,1:k]} \in \mathbb{R}^{k\times k}$ *denotes the* $k \times k$ *upper-left block of* $\widehat{W}^{(k)}$ *and* $\widehat{W}^{(k)}_{[1:k,k+1]} \in \mathbb{R}^k$ *is a vector containing the first* k *components of the* $k+1$*-th column of* $\widehat{W}^{(k)}$*.*

- *The discriminatory-step problem* (2.9) *is equivalent to*

$$\max_{\boldsymbol{u}\in U_{ad}} \langle\boldsymbol{v}, [W(\boldsymbol{u})]_{[1:k+1,1:k+1]}\boldsymbol{v}\rangle, \tag{3.9}$$

  where $W(\boldsymbol{u})$ *is defined in* (3.5) *and* $\boldsymbol{v} := [(\boldsymbol{\beta}^k)^\top, -1]^\top$*.*

Moreover, problems (2.7)/(3.7)*,* (2.8)/(3.8)*, and* (2.9)/(3.9) *are well posed.*

Proof. The equivalences between (2.7), (2.8), (2.9) and (3.7), (3.8), and (3.9), respectively, can be proved by similar calculations to the one used in the proof of Lemma 3.1.1. We omit them for brevity.

Problem (2.8)/(3.8) is a quadratic minimization problem with quadratic function bounded from below by zero. Hence the existence of a minimizer follows.

Problems (2.7)/(3.7) and (2.9)/(3.9) are two classical optimal control problems. Since the admissible set U_{ad} is a weakly compact subset of $L^2(0,T;\mathbb{R}^M)$, the existence of a maximizer follows by standard arguments based on maximizing sequences and weak compactness; see, e.g., [25] and references therein. □

Using the matrix representation given in Lemma 3.1.2, we can now sketch the mathematical meaning of the iterations of GR. Assume that at the k-th iteration the submatrix $\widehat{W}^{(k)}_{[1:k,1:k]}$ is PD, but $\widehat{W}^{(k)}_{[1:k+1,1:k+1]}$ has a non-trivial (one-dimensional) kernel. The GR algorithm first tries to identify (by solving problem (3.8)) the kernel of $\widehat{W}^{(k)}_{[1:k+1,1:k+1]}$, and then attempts to compute (by solving problem (3.9)) a new control function $\boldsymbol{u}^{k+1}$ such that the matrix $W_{[1:k+1,1:k+1]}(\boldsymbol{u}^{k+1})$ is positive on the kernel $\widehat{W}^{(k)}_{[1:k+1,1:k+1]}$. If these happen, then the new updated matrix $\widehat{W}^{(k+1)} = \widehat{W}^{(k)} + W(\boldsymbol{u}^{k+1})$ has a PD upper-left block $\widehat{W}^{(k+1)}_{[1:k+1,1:k+1]}$. Moreover, if these two steps hold for any k, then we obtain after the $(K-1)$-th iteration the matrix $\widehat{W} = \widehat{W}^{(K)}$ results to be PD. Hence, two questions clearly arise:

1. Does the fitting step of the algorithm always compute the non-trivial kernel of $\widehat{W}^{(k)}_{[1:k+1,1:k+1]}$ (in case it is truly non trivial)?
2. Does the discriminatory step of the algorithm always compute a control function $\boldsymbol{u}^{k+1}$ that makes $\widehat{W}^{(k+1)}_{[1:k+1,1:k+1]}$ PD?

The first question can be answered with the help of the following technical lemma.

Lemma 3.1.3 (On the kernel of symmetric PSD matrices) *Consider a symmetric PSD matrix $\widetilde{G} \in \mathbb{R}^{n\times n}$ of the form*

$$\widetilde{G} = \begin{bmatrix} G & \boldsymbol{b} \\ \boldsymbol{b}^\top & c \end{bmatrix},$$

where $G \in \mathbb{R}^{(n-1)\times(n-1)}$ is symmetric and PD, and $\boldsymbol{b} \in \mathbb{R}^{n-1}$ and $c \in \mathbb{R}$ are such that the kernel of $\widetilde{G}$ is non-trivial. Then

$$\ker(\widetilde{G}) = \operatorname{span}\left\{ \begin{bmatrix} G^{-1}\boldsymbol{b} \\ -1 \end{bmatrix} \right\}.$$

Proof. Since the kernel of $\widetilde{G}$ is non-trivial, there exists a non-zero vector $\boldsymbol{u} = \begin{bmatrix} \boldsymbol{v} \\ d \end{bmatrix} \in \mathbb{R}^n \setminus \{0\}$ (with $\boldsymbol{v} \in \mathbb{R}^{n-1}$ and $d \in \mathbb{R}$) such that $\widetilde{G}\boldsymbol{u} = 0$. Moreover, since G is PD, the kernel of $\widetilde{G}$ must be one-dimensional and equal to the span of $\{\boldsymbol{u}\}$. Using the structure of $\boldsymbol{u}$, we write $\widetilde{G}\boldsymbol{u} = 0$ as

$$\begin{cases} G\boldsymbol{v} + d\,\boldsymbol{b} = 0, \\ \boldsymbol{b}^\top \boldsymbol{v} + dc = 0, \end{cases} \quad \overset{G \text{ invertible}}{\Longleftrightarrow} \quad \begin{cases} \boldsymbol{v} = -dG^{-1}\boldsymbol{b}, \\ -d\,\boldsymbol{b}^\top G^{-1}\boldsymbol{b} + dc = 0. \end{cases} \tag{3.10}$$

Now, suppose that $d = 0$. This implies that $\boldsymbol{v} = -dG^{-1}\boldsymbol{b} = 0$, which in turn implies that $\boldsymbol{u} = 0$. However, this is a contradiction to the fact that $\boldsymbol{u} \neq 0$. Hence $d \neq 0$. The result follows by the right equations in (3.10) (divided by $-d$). □

Recalling the equivalent form (3.8) of the fitting-step problem (2.8), one can clearly see that, if $\widehat{W}^{(k)}_{[1:k,1:k]}$ is PD, then the unique solution to (3.8) is given by $\boldsymbol{\beta}^k = (\widehat{W}^{(k)}_{[1:k,1:k]})^{-1}\widehat{W}^{(k)}_{[1:k,k+1]}$. On the other hand, if we set

$$\widetilde{G} = \widehat{W}^{(k)}_{[1:k+1,1:k+1]},\ G = \widehat{W}^{(k)}_{[1:k,1:k]},\ \boldsymbol{b} = \widehat{W}^{(k)}_{[1:k,k+1]},\ c = \widehat{W}^{(k)}_{[k+1,k+1]},$$

then Lemma 3.1.3 guarantees that the vector $\boldsymbol{v} := [(\boldsymbol{\beta}^k)^\top,\ -1]^\top$ spans the kernel of $\widehat{W}^{(k)}_{[1:k+1,1:k+1]}$, if this is non-trivial. Therefore, we have

$$\ker(\widehat{W}^{(k)}_{[1:k+1,1:k+1]}) = \operatorname{span}\Big\{ \begin{bmatrix} (\widehat{W}^{(k)}_{[1:k,1:k]})^{-1}\widehat{W}^{(k)}_{[1:k,k+1]} \\ -1 \end{bmatrix} \Big\} = \operatorname{span}\Big\{ \boldsymbol{v} := \begin{bmatrix} \boldsymbol{\beta}^k \\ -1 \end{bmatrix} \Big\}.$$

This means that, if $\widehat{W}^{(k)}_{[1:k+1,1:k+1]}$ has a rank defect, then the GR algorithm finds this defect by the fitting step.

The answer to the second question posed above is more complicated. In order to formulate it properly, we need to recall the definition of observability of an input/output dynamical system (A, B, C) of the form (1.1) with $A \in \mathbb{R}^{N\times N}$, $B \in \mathbb{R}^{N\times M}$, $C \in \mathbb{R}^{P\times N}$; see, e.g., [67].

Definition & Lemma 3.1.4 (observable input-output linear systems) *The input-output linear system* (1.1) *is said to be* <u>observable</u> *if the initial state* $\boldsymbol{y}(0) = \boldsymbol{y}^0$ *can be uniquely determined from input/output measurements. Equivalently,* (1.1) *is observable if and only if the observability matrix*

$$\mathcal{O}_N(C, A) := \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{N-1} \end{bmatrix}$$

has full column rank.

Notice that the matrix B does not affect the observability of system (1.1).

We now analyze the performance of GR in the case of fully observable systems (namely rank $\mathcal{O}_N(C, A) = N$) in Section 3.1.2 and in case of partially observable systems (namely rank $\mathcal{O}_N(C, A) < N$) in Section 3.1.3.

3.1.2 Analysis for fully observable systems

Let us assume that the system is observable, namely that rank $\mathcal{O}_N(C, A) = N$. We show in this section that this is a sufficient condition for the GR algorithm to make the matrix $\widehat{W}$ PD. To do so, we first prove the following lemma regarding the splitting step. Notice that the proof of this result is inspired by classical Kalmann controllability theory; see, e.g., [31].

Lemma 3.1.5 (Splitting-step problem for fully observable systems) *Assume that the matrices* $A \in \mathbb{R}^{N\times N}$ *and* $C \in \mathbb{R}^{P\times N}$ *are such that* rank $\mathcal{O}_N(C, A) = N$. *Let* $\widehat{W}^{(k)}_{[1:k,1:k]}$

be PD, $\boldsymbol{\beta}^k$ the solution to the fitting-step problem (2.8)*, and $\boldsymbol{v} = [(\boldsymbol{\beta}^k)^\top, -1]^\top$. Then any solution $\boldsymbol{u}^{k+1}$ of the splitting-step problem* (2.9) *satisfies*

$$\langle \boldsymbol{v}, W_{[1:k+1,1:k+1]}(\boldsymbol{u}^{k+1})\boldsymbol{v}\rangle = \Big\| \int_0^T Ce^{(T-s)A}\Big(\sum_{j=1}^{k} \beta_j^k B_j - B_{k+1}\Big)\boldsymbol{u}^{k+1}(s)ds\Big\|_2^2 > 0,$$

for $k = 0, 1, \ldots, K-1$.

Proof. Let us define $\widetilde{B} := \sum_{j=1}^{k} \beta_j^k B_j - B_{k+1}$. Since the matrices $B_1, \ldots, B_{k+1}$ are assumed to be linearly independent, $\widetilde{B}$ is non-zero.

Now, we consider an arbitrary $\delta \in (0, T)$ and define a control function $\widetilde{\boldsymbol{u}} \in U_{ad}$ as

$$\widetilde{\boldsymbol{u}}(s) := \begin{cases} 0, & 0 \le s < \delta, \\ \boldsymbol{e}_i, & \delta \le s \le T, \end{cases}$$

where $\boldsymbol{e}_i \in \mathbb{R}^M$ is the i-th canonical vector for some index $1 \le i \le M$. Further, we denote by $\widetilde{\boldsymbol{b}}_i$ the i-th column of $\widetilde{B}$. Since $\widetilde{B}$ is non-zero, we can choose the index i such that $\widetilde{\boldsymbol{b}}_i \neq 0$. Now, we compute

$$\begin{aligned}
\int_0^T Ce^{(T-s)A}\widetilde{B}\widetilde{\boldsymbol{u}}(s)ds &= \int_\delta^T Ce^{(T-s)A}\widetilde{\boldsymbol{b}}_i ds = \int_\delta^T C\Big[\sum_{j=0}^{\infty} \frac{(T-s)^j A^j}{j!}\Big]\widetilde{\boldsymbol{b}}_i ds \\
&\overset{(\star)}{=} \Big[\sum_{j=0}^{\infty} \int_\delta^T \frac{(T-s)^j}{j!} ds\, CA^j\Big]\widetilde{\boldsymbol{b}}_i = \Big[\sum_{j=0}^{\infty} \frac{(T-\delta)^{j+1}}{(j+1)!} CA^j\Big]\widetilde{\boldsymbol{b}}_i \\
&= \sum_{j=0}^{\infty} \beta_j(\delta) CA^j \widetilde{\boldsymbol{b}}_i,
\end{aligned}$$

where $\beta_j(\delta) := \frac{(T-\delta)^{j+1}}{(j+1)!}$ and we used the dominated convergence theorem (see, e.g., [64, Theorem 1.34]) to interchange integral and infinite sum and obtain the equality $(\star)$. Since the observability matrix $\mathcal{O}_N(C, A)$ has full rank and $\widetilde{\boldsymbol{b}}_i \neq 0$, there exists an index $0 \le j \le N-1$ such that $CA^j\widetilde{\boldsymbol{b}}_i \neq 0$. Hence, $f(\delta) := \sum_{j=0}^{\infty} \beta_j(\delta) CA^j\widetilde{\boldsymbol{b}}_i$ is an analytic function for $\delta \in (0, T)$ and such that $f \neq 0$.[1] We also know that (non-constant) analytic functions have isolated roots; see, e.g., [64, Theorem 10.18]. Therefore we can find a $\delta \in (0, T)$ such that $\sum_{j=0}^{\infty} \beta_j(\delta) CA^j\widetilde{\boldsymbol{b}}_i \neq 0$, and obtain the existence of a $\widetilde{\boldsymbol{u}} \in U_{ad}$ such that

$$\int_0^T Ce^{(T-s)A}\widetilde{B}\widetilde{\boldsymbol{u}}(s)ds \neq 0.$$

[1] To see it, recall that $\beta_j(\delta) = \frac{(T-\delta)^{j+1}}{(j+1)!}$, consider a function $g(x) = \sum_{j=0}^{\infty} \frac{x^{j+1}}{(j+1)!}\gamma_j$, and assume that there exists at least one integer k such that $\gamma_k \neq 0$. Now, if we pick the minimum integer $\widehat{k}$ such that $\gamma_{\widehat{k}} \neq 0$, we have that $g(x) = \frac{x^{\widehat{k}+1}}{(\widehat{k}+1)!}\gamma_{\widehat{k}} + \sum_{j=\widehat{k}+1}^{\infty} \frac{x^{j+1}}{(j+1)!}\gamma_j$. For $x \to 0$, the first term behaves as $O(x^{\widehat{k}+1})$, while the second term as $O(x^{\widehat{k}+2})$. Hence, there exists a point $y > 0$ such that $g(y) \neq 0$.

This implies that

$$\begin{aligned}\langle \boldsymbol{v}, W_{[1:k+1,1:k+1]}(\boldsymbol{u}^{k+1})\boldsymbol{v}\rangle &= \Big\| \int_0^T Ce^{(T-s)A}\Big(\sum_{\ell=1}^{k} \beta_\ell^k B_\ell - B_{k+1}\Big)\boldsymbol{u}^{k+1}(s)ds\Big\|_2^2 \\ &\geq \Big\| \int_0^T Ce^{(T-s)A}\Big(\sum_{\ell=1}^{k} \beta_\ell^k B_\ell - B_{k+1}\Big)\widetilde{\boldsymbol{u}}(s)ds\Big\|_2^2 \\ &= \Big\| \int_0^T Ce^{(T-s)A}\widetilde{B}\widetilde{\boldsymbol{u}}(s)ds\Big\|_2^2 > 0,\end{aligned}$$

where we have used that $\boldsymbol{u}^{k+1}$ is a maximizer for problem (2.9). □

Now, we can prove our first main result.

Theorem 3.1.6 (GR for fully observable systems) *Assume that the matrices $A \in \mathbb{R}^{N\times N}$ and $C \in \mathbb{R}^{P\times N}$ are such that* $\operatorname{rank} \mathcal{O}_N(C,A) = N$. *Let $K \in \{1,\dots,MN\}$ be arbitrary and let $\{\boldsymbol{u}^1,\dots,\boldsymbol{u}^K\} \subset U_{ad}$ be a set of controls generated by GR. Then the matrix $\widehat{W}$ defined in* (3.4) *is PD and the online reconstruction problem* (2.3) *is uniquely solvable by $\boldsymbol{\beta} = \boldsymbol{\beta}_\star$.*

Proof. By Lemma 3.1.1 it is sufficient to show that the matrix $\widehat{W}$ corresponding to the controls $\boldsymbol{u}^1,\dots,\boldsymbol{u}^K$ generated by the algorithm is PD. The proof of this claim proceeds by induction.

Lemma 3.1.5 guarantees that there exists a $\boldsymbol{u}^1$ such that $[W(\boldsymbol{u}^1)]_{1,1} > 0$. Now, we assume that $\widehat{W}^{(k)}_{[1:k,1:k]}$ is PD, and we show that $\widehat{W}^{(k+1)}_{[1:k+1,1:k+1]}$ is PD as well.

If $\widehat{W}^{k}_{[1:k+1,1:k+1]}$ is PD, then

$$\widehat{W}^{(k+1)}_{[1:k+1,1:k+1]} = \widehat{W}^{k}_{[1:k+1,1:k+1]} + W(\boldsymbol{u}^k)_{[1:k+1,1:k+1]}$$

is PD as well, since $W(\boldsymbol{u}^k)_{[1:k+1,1:k+1]}$ is PSD.

Assume now that the submatrix $\widehat{W}^{(k)}_{[1:k+1,1:k+1]}$ has a non-trivial kernel. Since $\widehat{W}^{(k)}_{[1:k,1:k]}$ is PD (induction hypothesis), problem (3.8) is uniquely solvable with solution $\boldsymbol{\beta}^k$. Then, by Lemma 3.1.3 the (one-dimensional) kernel of $\widehat{W}^{(k)}_{[1:k+1,1:k+1]}$ is the span of the vector $\boldsymbol{v} = [(\boldsymbol{\beta}^k)^\top,\ -1]^\top$. Finally, using Lemma 3.1.5 we obtain that the solution $\boldsymbol{u}^{k+1}$ to the splitting-step problem satisfies

$$0 < \langle \boldsymbol{v}, [W(\boldsymbol{u}^{k+1})]_{[1:k+1,1:k+1]}\boldsymbol{v}\rangle.$$

Hence, the matrix $[W(\boldsymbol{u}^{k+1})]_{[1:k+1,1:k+1]}$ is PD on the span of $\boldsymbol{v}$. Therefore $\widehat{W}^{(k+1)}_{[1:k+1,1:k+1]} = \widehat{W}^{(k)}_{[1:k+1,1:k+1]} + [W(\boldsymbol{u}^{k+1})]_{[1:k+1,1:k+1]}$ is PD. □

Remark 3.1.7 (Uniqueness of solution of the min-max problem (2.5)**)** *Under the assumption that the system is fully observable, the min-max problem* (2.5) *is also uniquely*

solvable with $\boldsymbol{\beta} = \boldsymbol{\beta}_\star$. To see this, we first note that (2.5) *can be written in terms of* $W(\boldsymbol{u})$*:*

$$\begin{aligned}\|C\boldsymbol{y}(B(\boldsymbol{\beta}),\boldsymbol{u};T) - C\boldsymbol{y}(B_\star,\boldsymbol{u};T)\|_2^2 &= \Big\| \int_0^T Ce^{(T-s)A}\Big(\sum_{j=1}^K(\beta_j - \beta_{\star,j})B_j\Big)\boldsymbol{u}(s)ds\Big\|_2^2 \\ &= \langle(\boldsymbol{\beta}-\boldsymbol{\beta}_\star), W(\boldsymbol{u})(\boldsymbol{\beta}-\boldsymbol{\beta}_\star)\rangle.\end{aligned}$$

Now, similarly as in the proof of Lemma 3.1.5 and using the full observability of the system, one can show that for any $\widehat{\boldsymbol{\beta}} \in \mathbb{R}^{NM}$ with $\widehat{\boldsymbol{\beta}} \neq \boldsymbol{\beta}_\star$ there exists a control $\boldsymbol{u}(\widehat{\boldsymbol{\beta}})$ such that $\langle(\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta}_\star), W(\boldsymbol{u}(\widehat{\boldsymbol{\beta}}))(\widehat{\boldsymbol{\beta}}-\boldsymbol{\beta}_\star)\rangle > 0$. Therefore the unique solution to (2.5) *is $\boldsymbol{\beta} = \boldsymbol{\beta}_\star$.*

Notice that, Theorem 3.1.6 does not require any particular assumption on the matrices $B_1, \dots, B_K$, which can be arbitrarily chosen with the only constraint to be linearly independent. Moreover, the number $K \in \{1, \dots, MN\}$ can be fixed arbitrarily and the GR algorithm will compute control functions that permit the exact reconstruction of the coefficients of the linear combination of the first K components of $B_\star$ in a basis $\{B_1, \dots, B_{MN}\}$. To be more precise, if the unknown $B_\star$ belongs to the span of K the linearly independent matrices $B_1, \dots, B_K$ used by the algorithm, then, using the control functions generated by the GR algorithm, the unknown $B_\star$ can be fully reconstructed. If $B_\star$ lies in the span of $\widetilde{K} \in \{K+1, K+2, \dots, MN\}$ linearly independent matrices $B_1, \dots, B_{\widetilde{K}}$, but only the first K of these are used by the algorithm (and in the online reconstruction problem), then one reconstructs exactly the K coefficients corresponding to the first elements $B_1, \dots, B_K$. Furthermore, the ordering of the K considered matrices does not affect the result of Theorem 3.1.6.

3.1.3 Analysis for partially observable systems

The observations and results of Section 3.1.2 are no longer true if the system is only partially observable, that is $\operatorname{rank}\mathcal{O}_N(C,A) = \mathcal{R} < N$. In this case, the choice of the linearly independent matrices $B_1, \dots, B_K$ and their ordering become crucial for the algorithm. In particular, we are going to show that the method can recover at most $K = \mathcal{R}M$ components of the unknown vector $\boldsymbol{\beta}_\star$, if appropriate matrices $B_1, \dots, B_K$ are chosen. Moreover, we will see that an inappropriate choice of matrices $B_1, \dots, B_K$ can lead to completely wrong results corresponding to an arbitrarily large error.

For our analysis, we begin by choosing a set of $K = NM$ matrices by exploiting the kernel of the observability matrix. In particular, recalling that $\operatorname{rank}\mathcal{O}_N(C,A) = \mathcal{R} < N$, the rank-nullity theorem allows us to consider a basis $\{\boldsymbol{v}_j\}_{j=1}^N \subset \mathbb{R}^N$ of $\mathbb{R}^N$, such that

$$\boldsymbol{v}_j \notin \ker\mathcal{O}_N(C,A), \qquad j = 1, \dots, \mathcal{R}, \tag{3.11}$$

$$\boldsymbol{v}_j \in \ker\mathcal{O}_N(C,A), \qquad j = \mathcal{R}+1, \dots, N, \tag{3.12}$$

where $\text{span}\{\boldsymbol{v}_j\}_{j=\mathcal{R}+1}^{N} = \ker \mathcal{O}_N(C,A)$. We now define a basis $\{B_k^{\mathcal{O}}\}_{k=1}^{NM}$ of $\mathbb{R}^{N\times M}$ as

$$\begin{aligned} B_1^{\mathcal{O}} = \boldsymbol{v}_1\boldsymbol{e}_1^\top,\ B_2^{\mathcal{O}} = \boldsymbol{v}_1\boldsymbol{e}_2^\top,\ \cdots,\ B_M^{\mathcal{O}} = \boldsymbol{v}_1\boldsymbol{e}_M^\top,\\ B_{M+1}^{\mathcal{O}} = \boldsymbol{v}_2\boldsymbol{e}_1^\top,\ B_{M+2}^{\mathcal{O}} = \boldsymbol{v}_2\boldsymbol{e}_2^\top,\ \cdots,\ B_{2M}^{\mathcal{O}} = \boldsymbol{v}_2\boldsymbol{e}_M^\top,\\ \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\ B_{(N-1)M+1}^{\mathcal{O}} = \boldsymbol{v}_N\boldsymbol{e}_1^\top,\ B_{(N-1)M+2}^{\mathcal{O}} = \boldsymbol{v}_N\boldsymbol{e}_2^\top,\ \cdots,\ B_{NM}^{\mathcal{O}} = \boldsymbol{v}_N\boldsymbol{e}_M^\top, \end{aligned} \tag{3.13}$$

where $\boldsymbol{e}_\ell \in \mathbb{R}^M$, for $\ell = 1,\dots,M$, are the canonical vectors in $\mathbb{R}^M$. Notice that, since the vectors $\{\boldsymbol{v}_j\}_{j=1}^{N}$ are linearly independent, the set $\{B_k^{\mathcal{O}}\}_{k=1}^{NM}$ is a basis of $\mathbb{R}^{N\times M}$.

From a computational point of view, the vectors $\boldsymbol{v}_j$ can be obtained by a singular value decomposition (SVD) of the observability matrix $\mathcal{O}_N(C,A) = U\Sigma V^\top$, where the columns of V form a basis of $\mathbb{R}^N$ and the last $N-\mathcal{R}$ columns of V span the kernel of $\mathcal{O}_N(C,A)$; see, e.g., [68, Theorem 5.2]. Therefore, one can set $\boldsymbol{v}_j = V_{[:,j]}$, $j = 1,\dots,N$.

Our first result for partially observable systems says that, if the basis $\{B_k^{\mathcal{O}}\}_{k=1}^{NM}$ is considered, then we can reduce the reconstruction of $B_\star = \sum_{j=1}^{MN}\beta_{\star,j}B_j^{\mathcal{O}}$ only to the first $\mathcal{R}M$ coefficients $\beta_1,\dots,\beta_{\mathcal{R}M}$. This is proved in the next lemma, where we use the notation

$$B_{\mathcal{R}}(\boldsymbol{\beta}_\star) := \sum_{j=1}^{\mathcal{R}M}\beta_{\star,j}B_j^{\mathcal{O}}. \tag{3.14}$$

Lemma 3.1.8 (reconstruction problem for partially observable systems) *Consider the basis $\{B_k^{\mathcal{O}}\}_{k=1}^{NM}$ constructed as in* (3.13) *(with vectors $\boldsymbol{v}_j$, $j=1,\dots,N$, as in* (3.11)-(3.12)*). The online least-squares problem* (2.3) *(with $K = MN$) is equivalent to*

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^{\mathcal{R}M}}\sum_{m=1}^{NM}\|C\boldsymbol{y}(B_{\mathcal{R}}(\boldsymbol{\beta}),\boldsymbol{u}^m;T) - C\boldsymbol{y}(B_\star,\boldsymbol{u}^m;T)\|_2^2.$$

Proof. Notice that, for any $\ell\in\{1,2,\dots,NM\}$ and $s\in[0,T]$, there exist N functions $\widetilde{\beta}_j$ such that

$$\begin{aligned} Ce^{(T-s)A}B_\ell^{\mathcal{O}} &= C\sum_{j=0}^{\infty}\frac{(T-s)^j}{j!}A^jB_\ell^{\mathcal{O}} \overset{(\star)}{=} C\Big[\sum_{j=0}^{N-1}\widetilde{\beta}_j(s)A^j\Big]B_\ell^{\mathcal{O}}\\ &= \Big[\widetilde{\beta}_0(s)I_N,\widetilde{\beta}_1(s)I_N,\dots,\widetilde{\beta}_{N-1}(s)I_N\Big]\mathcal{O}_N(C,A)B_\ell^{\mathcal{O}}, \end{aligned}$$

where we have used the Cayley-Hamilton theorem (see, e.g., [48, p.109]) to obtain the equality $(\star)$. If $\ell\in\{\mathcal{R}M+1,\dots,NM\}$, then $B_\ell^{\mathcal{O}} = \boldsymbol{v}_j\boldsymbol{e}_i^\top$ with $j\geq\mathcal{R}+1$, hence $\boldsymbol{v}_j\in\ker\mathcal{O}_N(C,A)$ and therefore

$$\mathcal{O}_N(C,A)B_\ell^{\mathcal{O}} = \underbrace{\mathcal{O}_N(C,A)\boldsymbol{v}_j}_{=0}\boldsymbol{e}_i^\top = 0.$$

Hence, $Ce^{(T-s)A}B_\ell^{\mathcal{O}} = 0$ for all $\ell\in\{\mathcal{R}M+1,\dots,NM\}$ and $s\in[0,T]$. Thus

$$\int_0^T Ce^{(T-s)A}B_\ell^{\mathcal{O}}\boldsymbol{u}(s)ds = 0,$$

for any control function $\boldsymbol{u} \in U_{ad}$. Now, recalling the definition of $J(\boldsymbol{\beta})$ from the proof of Lemma 3.1.1, our claim follows by writing the least-squares problem (2.3) as

$$\begin{aligned} J(\boldsymbol{\beta}) &= \sum_{m=1}^{NM} \| \sum_{j=1}^{NM} (\beta_j - \beta_{\star,j}) \int_0^T C e^{(T-s)A} B_j^{\mathcal{O}} \boldsymbol{u}^m(s) ds \|_2^2 \\ &= \sum_{m=1}^{NM} \| \sum_{j=1}^{\mathcal{R}M} (\beta_j - \beta_{\star,j}) \int_0^T C e^{(T-s)A} B_j^{\mathcal{O}} \boldsymbol{u}^m(s) ds \|_2^2. \end{aligned}$$

□

Lemma 3.1.8 implies that the coefficients $\beta_{\mathcal{R}M+1}, \dots, \beta_{MN}$ do not affect the cost function to be minimized. Therefore, as shown in Corollary 3.1.11, any vector $\boldsymbol{\beta} \in \mathbb{R}^{MN}$ of the form

$$\boldsymbol{\beta} = [\beta_1^\star, \cdots, \beta_{\mathcal{R}M}^\star, \beta_{\mathcal{R}M+1}, \cdots, \beta_{MN}]^\top$$

is a global solution to (2.3), for arbitrary $\beta_j \in \mathbb{R}$, $j = \mathcal{R}M + 1, \dots, MN$. This implies that one uses only the first $\mathcal{R}M$ elements of the basis. In fact, as we are going to show in Lemma 3.1.9 and Theorem 3.1.10, only their corresponding coefficients can be reconstructed, while no information can be obtained for the remaining ones. It is therefore natural, for $\operatorname{rank} \mathcal{O}_N(C, A) = \mathcal{R} < N$, to use the GR algorithm with only the first $\mathcal{R}M$ basis elements $B_1^{\mathcal{O}}, \dots, B_{\mathcal{R}M}^{\mathcal{O}}$. In this case, the proof of our main result for the GR algorithm is analogous to the one of Theorem 3.1.6. We first prove a version of Lemma 3.1.5 adapted to partially observable systems.

Lemma 3.1.9 (Splitting-step problem for partially observable systems) *Assume that* $\operatorname{rank} \mathcal{O}_N(C, A) = \mathcal{R} < N$ *and that the GR algorithm is run until the* k*-th iteration, with* $k < \mathcal{R}M$*, using the linearly independent matrices* $B_1^{\mathcal{O}}, \dots, B_{\mathcal{R}M}^{\mathcal{O}}$ *defined in* (3.13)*. Let* $\widehat{W}_{[1:k,1:k]}^{(k)}$ *be PD, and let* $\boldsymbol{\beta}^k$ *be the solution to the fitting-step problem* (2.8)*. Then any solution* $\boldsymbol{u}^{k+1}$ *of the splitting-step problem* (2.9) *satisfies for* $k = 1, \dots \mathcal{R}M - 1$

$$\langle \boldsymbol{v}, W_{[1:k+1,1:k+1]}(\boldsymbol{u}^{k+1}) \boldsymbol{v} \rangle = \| \int_0^T C e^{(T-s)A} \Big(\sum_{j=1}^{k} \beta_j^k B_j^{\mathcal{O}} - B_{k+1}^{\mathcal{O}} \Big) \boldsymbol{u}^{k+1}(s) ds \|_2^2 > 0,$$

where $\boldsymbol{v} := [(\boldsymbol{\beta}^k)^\top, \ -1]^\top$*, for* $k = 0, 1, \dots, K-1$.

Proof. Notice that, since the matrices $B_1^{\mathcal{O}}, \dots, B_{\mathcal{R}M}^{\mathcal{O}}$ are linearly independent and defined as in (3.13), we have that $\mathcal{O}_N(C, A) \Big(\sum_{j=1}^{k} \beta_j^k B_j^{\mathcal{O}} - B_{k+1}^{\mathcal{O}} \Big) \neq 0$.

With this observation, the result can be proved exactly as Lemma 3.1.5. □

Using Lemma 3.1.9, we can prove the following main result for GR in case the matrices $B_1^{\mathcal{O}}, \dots, B_{\mathcal{R}M}^{\mathcal{O}}$ defined in (3.13) are used.

Theorem 3.1.10 (GR for partially observable systems) *Let* $\{\boldsymbol{u}^m\}_{m=1}^{\mathcal{R}M} \subset U_{ad}$ *be a family of controls generated by the GR Algorithm 1 with* $K = \mathcal{R}M$ *and using the matrices* $B_1^{\mathcal{O}}, \dots, B_{\mathcal{R}M}^{\mathcal{O}}$ *defined in* (3.13)*. Then the least-squares problem*

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^{\mathcal{R}M}} \sum_{m=1}^{\mathcal{R}M} \| C\boldsymbol{y}(B_{\mathcal{R}}(\boldsymbol{\beta}), \boldsymbol{u}^m; T) - C\boldsymbol{y}(B_\star, \boldsymbol{u}^m; T) \|_2^2, \tag{3.15}$$

where $B_{\mathcal{R}}(\boldsymbol{\beta})$ is defined in (3.14)*, is uniquely solvable with $\beta_j = \beta_{\star,j}$, $j = 1, \dots, \mathcal{R}M$.*

Proof. The proof is the same as that of Theorem 3.1.6, where one should use Lemma 3.1.9 instead of Lemma 3.1.5. □

Theorem 3.1.10 allows us to prove the next corollary, which characterizes the result of the GR algorithm when more than $\mathcal{R}M$ basis elements of (3.13) are used.

Corollary 3.1.11 (More on partially observable systems)
Let $\{\boldsymbol{u}^m\}_{m=1}^K \subset U_{ad}$, with $K > \mathcal{R}M$, be a family of controls generated by GR, using the matrices $B_1^{\mathcal{O}}, \dots, B_K^{\mathcal{O}}$ defined in (3.13)*. Then the set of all global minimum points for the least-squares problem*

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^K} \sum_{m=1}^{K} \|C\boldsymbol{y}(B^{(K)}(\boldsymbol{\beta}), \boldsymbol{u}^m; T) - C\boldsymbol{y}(B_\star, \boldsymbol{u}^m; T)\|_2^2,$$

is given by $\{\boldsymbol{\beta} \in \mathbb{R}^K \mid \beta_j = \beta_{\star,j},\ j = 1, \dots, \mathcal{R}M\}$.

Proof. Theorem 3.1.10 (and Theorem 3.1.6) and its proof allow us to obtain that, using the first $\mathcal{R}M$ controls generated by the GR algorithm, the matrix $\widehat{W}^{(\mathcal{R}M)} \in \mathbb{R}^{K \times K}$ has a PD upper-left submatrix $\widehat{W}^{(\mathcal{R}M)}_{[1:\mathcal{R}M, 1:\mathcal{R}M]}$ and all the other entries $[\widehat{W}^{(\mathcal{R}M)}]_{\ell,j}$ are zero. Indeed, recalling the vectors $\boldsymbol{\gamma}_k(\boldsymbol{u}^m)$ defined in (3.6), for any $B_k^{\mathcal{O}}$ with $k \geq \mathcal{R}M + 1$, we have that $\mathcal{O}_N(C, A)B_k^{\mathcal{O}} = 0$ and thus

$$\boldsymbol{\gamma}_k(\boldsymbol{u}^m) = \int_0^T Ce^{(T-s)A} B_k^{\mathcal{O}} \boldsymbol{u}^m(s) ds = 0,$$

for any $T > 0$ and any $m = 1, \dots, \mathcal{R}M$. Similarly, the matrices $W(\boldsymbol{u}^m)$ for $m > \mathcal{R}M$ have the same structure, namely that their only nonzero components can be the upper-left submatrices $[W(\boldsymbol{u}^m)]_{[1:\mathcal{R}M, 1:\mathcal{R}M]}$. Therefore, the matrix $\widehat{W} = \widehat{W}^{(K)}$ has a PD upper-left submatrix $\widehat{W}_{[1:\mathcal{R}M, 1:\mathcal{R}M]}$, while all its other entries are zero. Therefore, the result follows by Lemma 3.1.1. □

Remark 3.1.12 (More about the kernel of $\mathcal{O}_N(C, A)$ and identifiability)
Corollary 3.1.11 guarantees that, if the basis $\{B_j^{\mathcal{O}}\}_{j=1}^K$ is used with $K > \mathcal{R}M$, then one can reconstruct exactly $\mathcal{R}M$ coefficients, while nothing can be said about the coefficients β_j for $j > \mathcal{R}M$. This is due to the structure of the matrix $\widehat{W}^{(\mathcal{R}M)}$, which has a PD submatrix $\widehat{W}^{(\mathcal{R}M)}_{[1:\mathcal{R}M, 1:\mathcal{R}M]}$ and is zero elsewhere (as discussed in the proof of Corollary 3.1.11).

Remark 3.1.13 (A priori error estimate) *Let $\boldsymbol{\beta}^{approx}$ be the solution to* (3.15)*. Then we get the a-priori error formula*

$$B_\star - B_{\mathcal{R}}(\boldsymbol{\beta}^{approx}) = \sum_{j=\mathcal{R}M+1}^{NM} \beta_{\star,j} B_j^{\mathcal{O}}.$$

Remark 3.1.14 (Min-max problem) *Following the same arguments of the proof of Lemma 3.1.8, one can show that the min-max problem* (2.5) *is equivalent to*

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^{\mathcal{R}M}} \max_{\boldsymbol{u} \in U_{ad}} \|C\boldsymbol{y}(B_\star, \boldsymbol{u}; T) - C\boldsymbol{y}(B_{\mathcal{R}}(\boldsymbol{\beta}), \boldsymbol{u}; T)\|_2^2. \tag{3.16}$$

Analogously to Remark 3.1.7, we can conclude that, using the matrices $B_1^{\mathcal{O}}, \dots, B_{\mathcal{R}M}^{\mathcal{O}}$ *defined in* (3.13)*, problem* (3.16) *is uniquely solvable with* $\beta_j = \beta_{\star,j}$*,* $j = 1, \dots, \mathcal{R}M$*.*

The results proved so far for a partially observable system are obtained for the special basis $\{B_j\}_{j=1}^{MN}$ constructed in (3.13). However, it is natural to ask:

- Is there any basis that permits to reconstruct more than $\mathcal{R}M$ coefficients?
- Can one reconstruct at least $\mathcal{R}M$ coefficients for any arbitrarily chosen basis?

The answers to both questions are negative. The first one is given by Theorem 3.1.15.

Theorem 3.1.15 (Maximal number of identifiable elements) *Let the observability matrix* $\mathcal{O}_N(C, A)$ *be such that* $\operatorname{rank} \mathcal{O}_N(C, A) = \mathcal{R} < N$*. There exists no basis of* $\mathbb{R}^{N\times M}$ *for which one can exactly recover more than* $\mathcal{R}M$ *coefficients.*

Proof. Consider the basis $\mathcal{B} = \{B_k^{\mathcal{O}}\}_{k=1}^{NM} \subset \mathbb{R}^{N\times M}$ constructed as in (3.13) and another arbitrarily chosen basis $\widehat{\mathcal{B}} = \{\widehat{B}_k\}_{k=1}^{NM} \subset \mathbb{R}^{N\times M}$. Any element $\widehat{B} \in \widehat{\mathcal{B}}$ can be written as a linear combination of the elements of $\mathcal{B}$, that is $\widehat{B} = \sum_{j=1}^{NM} \lambda_j B_j^{\mathcal{O}}$, for appropriate $\lambda_j \in \mathbb{R}$, $j = 1, \dots, MN$. Multiplying $\widehat{B}$ with $\mathcal{O}_N(C, A)$, we get

$$\mathcal{O}_N(C, A)\widehat{B} = \mathcal{O}_N(C, A)\left[\sum_{j=1}^{NM} \lambda_j B_j^{\mathcal{O}}\right] = \sum_{j=1}^{NM} \lambda_j \mathcal{O}_N(C, A) B_j^{\mathcal{O}} = \sum_{j=1}^{\mathcal{R}M} \lambda_j \mathcal{O}_N(C, A) B_j^{\mathcal{O}},$$

where we used that $\mathcal{O}_N(C, A)B_j^{\mathcal{O}} = 0$, for $j \in \{\mathcal{R}+1, \dots, N\}$, to obtain the last equality. Now define the set $\mathcal{D} = \{D_k\}_{k=1}^{NM}$ as $D_k := \mathcal{O}_N(C, A)\widehat{B}_k, \quad k = 1, \dots, NM$. Hence, we can conclude that at most $\mathcal{R}M$ elements of $\mathcal{D}$ are linearly independent. Recalling the proof of Lemma 3.1.5 and Remark 3.1.12, this means that for $NM - \mathcal{R}M$ elements of $\widehat{\mathcal{B}}$ there exists a linear combination of the other $\mathcal{R}M$ elements, such that the observation at final time T is identical for any control $\boldsymbol{u}$. Therefore one can reconstruct at most $\mathcal{R}M$ coefficients for the basis $\widehat{\mathcal{B}}$. □

Let us now explain why the answer to the second question is also negative. To do so, we provide the following examples, which show that a wrong choice of a basis leads to inconclusive results.

Example 3.1.16 (Wrong bases lead to inconclusive results) *Consider a simple system with*

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad B_\star = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix},$$

and the following basis of $\mathbb{R}^{2\times 2}$

$$\widehat{B}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad \widehat{B}_2 = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}, \quad \widehat{B}_3 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \widehat{B}_4 = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}.$$

Notice that in this case the observability condition does not hold, since one can compute that

$$\mathcal{R} = \operatorname{rank} \mathcal{O}_N(C, A) = \operatorname{rank} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}^\top = 1.$$

Clearly we have that

$$B_\star = 0 \cdot B_1 + 1 \cdot B_2 + 0 \cdot B_3 + 1 \cdot B_4,$$

and hence $\boldsymbol{\beta}_\star = [0\,1\,0\,1]^\top$. *We can now compute for an arbitrary control* $\boldsymbol{u} \in U_{ad}$ *that*

$$\begin{aligned} C\boldsymbol{y}(B_\star, \boldsymbol{u}; T) - C\boldsymbol{y}(B(\boldsymbol{\beta}), \boldsymbol{u}; T) &= C\int_0^T e^{(T-s)A} B_\star \boldsymbol{u}(s) ds - C\int_0^T e^{(T-s)A} B(\boldsymbol{\beta})\boldsymbol{u}(s) ds \\ &= \int_0^T Ce^{(T-s)A}\Big(\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} - \begin{bmatrix} \beta_1+\beta_2 & \beta_3+\beta_4 \\ \beta_2 & \beta_4 \end{bmatrix} \Big)\boldsymbol{u}(s) ds \\ &= \int_0^T \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} e^{T-s} & 0 \\ 0 & e^{T-s} \end{bmatrix} \begin{bmatrix} 1-(\beta_1+\beta_2) & 1-(\beta_3+\beta_4) \\ 1-\beta_2 & 1-\beta_4 \end{bmatrix} \boldsymbol{u}(s) ds \\ &= \int_0^T \begin{bmatrix} e^{T-s}(1-(\beta_1+\beta_2)) & e^{T-s}(1-(\beta_3+\beta_4)) \\ 0 & 0 \end{bmatrix} \boldsymbol{u}(s) ds, \end{aligned}$$

which is zero for any $\boldsymbol{\beta} = [\beta_1\ \beta_2\ \beta_3\ \beta_4]^\top \in \mathbb{R}^4$ *with* $\beta_1 + \beta_2 = 1$ *and* $\beta_3 + \beta_4 = 1$ *(for any control* $\boldsymbol{u}$*). This means that any* $\boldsymbol{\beta} = [\beta_1\ \beta_2\ \beta_3\ \beta_4]$ *with* $\beta_1 + \beta_2 = 1$ *and* $\beta_3 + \beta_4 = 1$ *solves the least-squares problem* (2.3)*, independently on the control functions* $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_4$. *Since the online least-square problem has then infinitely many solutions,*[2] *one cannot conclude anything about the quality of a computed solution, which has the form*

$$\widehat{B}^{approx} = \begin{bmatrix} 1 & 1 \\ \beta_2 & \beta_4 \end{bmatrix},$$

leading to the error

$$\|B_\star - B_{\mathcal{R}}(\boldsymbol{\beta}^{approx})\|_F^2 = (1-\beta_2)^2 + (1-\beta_4)^2,$$

which can be arbitrarily large (here $\|\cdot\|_F$ *denotes the Frobenius norm). Even if one would by chance guess the right coefficients (in this case* $\beta_2 = 1, \beta_4 = 1$*), there would be no way to verify them, since their effect is not observable. Notice also that, even if the entries* $\widehat{B}_{1,1}^{approx}$ *and* $\widehat{B}_{1,2}^{approx}$ *are correct, it is not possible to certify this or to associate these correct entries to some precise elements of the chosen basis. This example shows that for an arbitrarily chosen basis, one can not conclude anything about the quality of the computed coefficients or the difference between* $B(\boldsymbol{\beta})$ *and* $B_\star$.

Example 3.1.17 (Good bases lead to certified results) *Consider the same system as in Example 3.1.16, but now let us use the SVD of the observability matrix,*

$$\mathcal{O}_2(C,A) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} & 0 \\ 0 & 1 & 0 & 0 \\ \frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sqrt{2} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = U\Sigma V^\top,$$

which gives $\boldsymbol{v}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \notin \ker \mathcal{O}_N(C,A)$, $\boldsymbol{v}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \in \ker \mathcal{O}_N(C,A)$, *leading to the basis*

$$B_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, B_2 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, B_3 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, B_4 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix},$$

[2] Notice that these solutions are also solution to the min-max problem (2.5).

constructed as in (3.13)*. In this case, we have* $\boldsymbol{\beta}_\star = [1\ 1\ 1\ 1]^\top$*. Since the GR algorithm considers only the first two basis elements, one gets the final result*

$$B^{approx} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}.$$

Similarly to Example 3.1.16, the two entries $\widehat{B}_{1,1}^{approx}$ *and* $\widehat{B}_{1,2}^{approx}$ *are correct, but now this is guaranteed by Theorem 3.1.10. Therefore, in this case, the results obtained are accompanied by precise information on their correctness.*

These examples show clearly that without an a priori knowledge about the observability of the system (and hence about the "quality" of the basis), GR leads to inconclusive results. This is the main motivation for the optimized version of GR (OGR), which attempts to identify automatically "good" elements in a given set of matrices.

3.1.4 OGR for linear-quadratic problems

Consider an arbitrary set of linearly independent matrices $\{B_j\}_{j=1}^K \subset \mathbb{R}^{N\times M}$. The idea of the OGR Algorithm 2 is to choose at every iteration one element B_j which leads to a control function capable of improving the rank of the matrix $\widehat{W}^{(k)}_{[1:k+1,1:k+1]}$. For this purpose, OGR replaces the sweeping process of GR with a more robust and parallel testing of all the matrices. At each iteration, the element associated with the maximal splitting-step function value is chosen and removed from the set $\{B_j\}_{j=1}^K$, while the corresponding control function is added to the set of already computed control functions. Therefore, the dimension of the set $\{B_j\}_{j=1}^K$ reduces by one at each iteration and the algorithm is stopped if either all the K matrices were chosen or as soon none of the remaining ones can be split by the others.

Notice that a selected element B_{k+1} will not be linearly dependent on previously chosen elements (after multiplication with the observability matrix). This is proven in the next theorem, which also motivates the stopping criterion used in line 3 of OGR.

Theorem 3.1.18 (Linearly independence of selected basis elements) *Assume that OGR selected already* k *linearly independent matrices* B_j*,* $j = 1, \dots, k$*. At iteration* $k+1$*, the new selected matrix* B_{k+1} *is such that* $\mathcal{O}_N(C, A)B_{k+1}$ *is linearly independent from the matrices* $\mathcal{O}_N(C, A)B_j$*,* $j = 1, \dots, k$*, if and only if*

$$\|C\boldsymbol{y}(B^{(k)}(\boldsymbol{\beta}^{\ell_k}), \boldsymbol{u}^{k+1}; T) - C\boldsymbol{y}(B_{\ell_{k+1}}, \boldsymbol{u}^{k+1}; T)\|_2^2 > 0.$$

Proof. If the matrix $\mathcal{O}_N(C, A)B_{k+1}$ is linearly independent from the matrices $\mathcal{O}_N(C, A)B_j$, $j = 1, \dots, k$, then one can show as in the proof of Lemma 3.1.9 that

$$\|C\boldsymbol{y}(B^{(k)}(\boldsymbol{\beta}^{\ell_k}), \boldsymbol{u}^{k+1}; T) - C\boldsymbol{y}(B_{\ell_{k+1}}, \boldsymbol{u}^{k+1}; T)\|_2^2 > 0.$$

Now, we prove the other implication by contraposition. Assume that there exists a vector $\boldsymbol{\beta} \in \mathbb{R}^k$ such that $\mathcal{O}_N(C, A)(\sum_{j=1}^k \beta_j B_j - B_{k+1}) = 0$ holds. This vector $\boldsymbol{\beta}$ is a solution of the fitting step problem with cost-function value equal to zero. However, the corresponding cost function of the splitting-step problem (2.13) results to be zero for any control function $\boldsymbol{u}$. The result follows by contraposition. □

Notice that, if OGR stops when first reaching line 3, then the chosen basis does not allow one to distinguish the states corresponding to controlled and uncontrolled systems. In this case, entering in the while loop would be useless since the first splitting step will certainly fail in producing a large enough splitting value.

Theorem 3.1.18 shows exactly that OGR manages to identify among the elements of the given set $\{B_j\}_{j=1}^K$ the ones that do not lie in the kernel of $\mathcal{O}_N(C,A)$. For instance, let us consider again the system of Example 3.1.16, for which we have shown that GR leads to inconclusive results. If we use instead OGR, this performs two iterations and selects only two basis elements, one among $\widehat{B}_1$ and $\widehat{B}_2$ and the other among $\widehat{B}_3$ and $\widehat{B}_4$. This can be shown by performing calculations similar to the ones of Example 3.1.16. In particular, in the initialization step the four matrices produce the same cost function value. Hence, any of them can be selected by the algorithm. Assume that the element $\widehat{B}_1$ is picked (hence $\ell_1 = 1$) and consider the first iteration of the algorithm ($k = 1$). At the fitting step the algorithm computes a coefficient $\beta_1^2 = 1$ for $\widehat{B}_2$, and some coefficients β_1^3 and β_1^4 corresponding to $\widehat{B}_3$ and $\widehat{B}_4$. Now, $\beta_1^2 = 1$ leads to a cost function of the splitting step which is zero for any control functions, while for β_1^3 and β_1^4 there exists a control function leading to a non-zero value of the splitting cost. Therefore, the algorithm selects either $\widehat{B}_3$ or $\widehat{B}_4$. Let us assume that $\widehat{B}_4$ is picked ($\ell_2 = 4$) and hence the two elements $\widehat{B}_2$ and $\widehat{B}_4$ are swapped. In the fitting step of the second iteration ($k = 2$), the algorithm computes $\boldsymbol{\beta}^3 = [\,0\,,\,1\,]^\top$ and $\boldsymbol{\beta}^4 = [\,1\,,\,0\,]^\top$. Both of these two vectors lead to a splitting cost that is zero for any control. Hence, since the splitting step does not find any positive function value, the algorithm stops and returns $\widehat{B}_{\ell_1} = \widehat{B}_1$ and $\widehat{B}_{\ell_2} = \widehat{B}_4$ and the corresponding controls. If one uses the two selected basis elements and the corresponding control functions in the online phase, then one obtains the result $\boldsymbol{\beta} = [\,1\,,\,1\,]^\top$, which is not the exact solution shown in Example 3.1.16. This is due to the non-full observability of the system, which implies that $\mathcal{O}_N(C,A)\widehat{B}_1 = \mathcal{O}_N(C,A)\widehat{B}_2$ and $\mathcal{O}_N(C,A)\widehat{B}_3 = \mathcal{O}_N(C,A)\widehat{B}_4$. This means that the observations generated by the elements $\widehat{B}_1$ and $\widehat{B}_3$ cannot be distinguished by the ones created by $\widehat{B}_2$ and $\widehat{B}_4$. The non-full observability of the system cannot be overcome by any numerical strategy. OGR can nevertheless identify automatically all the observable degrees of freedom of the considered system.

Let us now demonstrate the efficiency of OGR by direct numerical experiments. We consider an experiment with two randomly chosen $N \times N$ full-rank real matrices A and C with $N = 10$. The unknown $B_\star$ is a randomly chosen real $N \times N$ matrix. In this case the system is fully observable, nevertheless we construct the basis elements to be used in GR and OGR as in (3.13) (by an SVD of the observability matrix), but we order the elements randomly. We then run GR and compute the rank of the matrix $\widehat{W}^{(k)}$ at every iteration k. This leads to the results shown in Figure 3.1 by the blue curve. The rank increases monotonically during the iterations and becomes full after about 30 iterations. However, the curve is not strictly monotonically increasing since the rank does not increase at each iteration. If we repeat the same experiment (with the same matrices) using OGR, we obtain the red curve in Figure 3.1. This curve is strictly monotonically until $\widehat{W}^{(k)}$ is full-rank after 11 iterations. Thus OGR computes much less control functions, meaning that also fewer laboratory experiments are needed to fully reconstruct the unknown operator $B_\star$. This experiment clearly shows the high potential of the OGR algorithm, which is capable

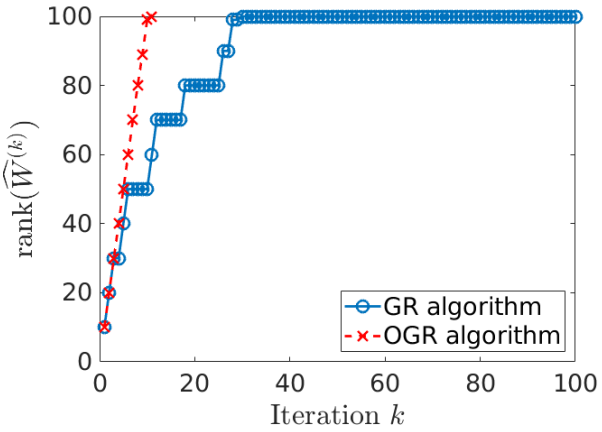

Figure 3.1: Rank of the matrix $\widehat{W}^{(k)}$ corresponding to GR (blue curve) and OGR (red curve) for a fully observable system. Both algorithms make use of a basis constructed as in (3.13).

to choose among the elements $B_1, \dots, B_K$ in an optimized fashion.

Looking more closely at the plots in Figure 3.1, it seems that the rank increase of the matrix $\widehat{W}^{(k)}$ is related to the dimension of the problem. Indeed, the rank of any matrix $W(\boldsymbol{u})$ is bounded by the number of rows P in the observer matrix $C \in \mathbb{R}^{P\times N}$, as we show in the following proposition.

Proposition 3.1.19 *Let $\widehat{W}^{(k)} = \sum_{m=1}^{k} W(\boldsymbol{u}^m)$ for $k \in \mathbb{N}$, $W(\boldsymbol{u})$ defined as in (3.5)-(3.6), and an arbitrary set of controls $\{\boldsymbol{u}^m\}_{m=1}^{k} \subset U_{ad}$. Then*

$$\operatorname{rank} \widehat{W}^{(k)} \leq kP,$$

where P is the number of rows in the observer matrix $C \in \mathbb{R}^{P\times N}$.

Proof. For any $\boldsymbol{u}$ we have

$$W(\boldsymbol{u}) = \Gamma(\boldsymbol{u})^\top \Gamma(\boldsymbol{u}),$$

where $\Gamma(\boldsymbol{u}) = [\gamma_1(\boldsymbol{u}) \ \cdots \ \gamma_K(\boldsymbol{u})] \in \mathbb{R}^{P\times K}$ for γ_j defined in (3.6). Using standard matrix-rank properties (compare, e.g., [48, Section 0.4.5]), we obtain $\operatorname{rank} W(\boldsymbol{u}) \leq \operatorname{rank} \Gamma(\boldsymbol{u}) \leq P$ and therefore

$$\operatorname{rank} \widehat{W}^{(k)} = \operatorname{rank} \Big(\sum_{m=1}^{k} W(\boldsymbol{u}^m) \Big) \leq \sum_{m=1}^{k} P = kP.$$

□

From Proposition 3.1.19 we can also directly derive the following result regarding the minimum number of iterations needed by GR and OGR.

Proposition 3.1.20 *Consider the online reconstruction problem (2.3) for the setting described at the beginning of Section 3.1. Let the observer matrix $C \in \mathbb{R}^{P\times N}$ be such that $\operatorname{rank} \mathcal{O}_N(C, A) = \mathcal{R} \leq N$, and consider the basis $\mathcal{B} = \{B_1^{\mathcal{O}}, \dots, B_{\mathcal{R}M}^{\mathcal{O}}\}$ with $B_j^{\mathcal{O}}$ defined as*

in (3.13). *Then GR and OGR need to compute at least* $\lceil \frac{\mathcal{R}M}{P} \rceil$ *controls in order to make* (2.3) *uniquely solvable.*

Recall that we considered $N = M = P = \mathcal{R} = 10$ to obtain the results in Figure 3.1. Thus, OGR is able to reduce the number of iterations almost down to the theoretical minimum $\lceil \frac{\mathcal{R}M}{P} \rceil = 10$.

Let us conclude this section with another important observation. The improvement proposed in OGR allows one to even enrich the set $\mathcal{B} = \{B_j\}_{j=1}^K$ used as input in OGR with other new elements that can be linearly dependent on $B_1, \dots, B_K$. In this case, if we denote by $\{B_j\}_{j=1}^{\widetilde{K}}$, for $\widetilde{K} > K$, the enriched set, then Theorem 3.1.18 guarantees that OGR will automatically pick some elements of the enriched set $\{B_j\}_{j=1}^{\widetilde{K}}$, such that $\mathcal{O}_N(C, A)B_j$ are linearly independent for all selected B_j. Hence, the corresponding splitting-step cost-function values will be strictly positive. Thus, OGR is particularly useful in more general nonlinear reconstruction problems, where an optimal choice of the set $\mathcal{B}$ can not be derived from the underlying system.

3.2 Reconstruction of a drift matrix

3.2.1 Model problem

We consider (2.1) with $f(A, \boldsymbol{y}, \boldsymbol{u}) = A\boldsymbol{y} + B\boldsymbol{u}$, that is,

$$\dot{\boldsymbol{y}}(t) = A_\star \boldsymbol{y}(t) + B\boldsymbol{u}(t),\ t \in (0, T], \quad \boldsymbol{y}(0) = 0. \tag{3.17}$$

The main difference to the setting in Section 3.1 is that we now assume the control matrix $B \in \mathbb{R}^{N\times M}$ to be given and the drift matrix $A_\star \in \mathbb{R}^{N\times N}$ to be unknown and to lie in the space spanned by a set of linearly independent matrices $\mathcal{B} = \{A_1, \dots, A_K\} \subset \mathbb{R}^{N\times N}$, $1 \le K \le N^2$. We write $A_\star = \sum_{j=1}^K \beta_{\star,j} A_j =: A(\boldsymbol{\beta}_\star)$. As stated in Section 2.1, we want to identify the unknown drift matrix $A_\star$ by using a set of control functions $\{\boldsymbol{u}^m\}_{m=1}^K \subset U_{ad}$ in order to perform K laboratory experiments and obtain the experimental data $\{\boldsymbol{\varphi}_\star^{data}(\boldsymbol{u}^m)\}_{m=1}^K \subset \mathbb{R}^P$, as defined in (2.2).

Remark 3.2.1 *The hypothesis* $\boldsymbol{y}(0) = 0$ *in* (3.17) *can be made without loss of generality. Indeed, if* $\boldsymbol{y}(0) = \boldsymbol{y}^0 \neq 0$, *one can use* $\boldsymbol{u} = 0$ *(case of uncontrolled system), generate the data* $\boldsymbol{\varphi}_\star^{data}(0)$, *and then subtract this from all other data* $\{\boldsymbol{\varphi}_\star^{data}(\boldsymbol{u}^m)\}_{m=1}^K$ *to get back (by linearity) to the case of system* (3.17) *with* $\boldsymbol{y}(0) = 0$.

Using $\{\boldsymbol{u}^m\}_{m=1}^K$ and $\{\boldsymbol{\varphi}_\star^{data}(\boldsymbol{u}^m)\}_{m=1}^K$, the unknown vector $\boldsymbol{\beta}_\star$ is obtained by solving the online reconstruction problem (2.3), in which $\boldsymbol{\varphi}(A(\boldsymbol{\beta}), \boldsymbol{u}^m) = \boldsymbol{y}(A(\boldsymbol{\beta}), \boldsymbol{u}^m; T)$ now solves (3.17), with $A_\star$ replaced by $A(\boldsymbol{\beta})$. Thus, we use LGR Algorithm 3 to generate $\{\boldsymbol{u}^m\}_{m=1}^K$ with the goal of making (2.22) uniquely solvable, that means making PD the GN matrix $\widehat{W}_\circ$, defined in (3.19). In (2.22), $\delta\boldsymbol{y}_\circ(A(\delta\boldsymbol{\beta}), \boldsymbol{u}; t)$ is now the solution to

$$\begin{cases} \dot{\delta\boldsymbol{y}}_\circ(t) = A(\boldsymbol{\beta}_\circ)\delta\boldsymbol{y}_\circ(t) + \sum_{j=1}^K \delta\beta_j A_j \boldsymbol{y}_\circ(t), \quad t \in (0, T], \quad \delta\boldsymbol{y}_\circ(0) = 0, \\ \dot{\boldsymbol{y}}_\circ(t) = A(\boldsymbol{\beta}_\circ)\boldsymbol{y}_\circ(t) + B\boldsymbol{u}(t), \quad t \in (0, T], \quad \boldsymbol{y}_\circ(0) = 0. \end{cases} \tag{3.18}$$

In what follows, we show that the LGR does produce $\{\boldsymbol{u}^m\}_{m=1}^K$ that make $\widehat{W}_\circ$ PD under appropriate assumptions. In contrast to the setting in Section 3.1, we now require both observability and controllability of linear systems. Thus, we recall the observability (see Definition & Lemma 3.1.4) and controllability properties for an input/output system (A, B, C) of the form (1.1) with $A \in \mathbb{R}^{N\times N}$, $B \in \mathbb{R}^{N\times M}$, $C \in \mathbb{R}^{P\times N}$; see, e.g., [67, Thm. 3, Thm. 23].

Definition & Lemma 3.2.2 (controllable input-output linear systems) *The linear system* (1.1) *is said to be* <u>controllable</u> *if for any final state $\boldsymbol{y}^f$ there exists an input sequence that transfers $\boldsymbol{y}^0$ to $\boldsymbol{y}^f$. Equivalently,* (1.1) *is controllable if and only if the controllability matrix*

$$\mathcal{C}_N(A, B) := \begin{bmatrix} B & AB & \cdots & A^{N-1}B \end{bmatrix}$$

has full rank.

Similar to Section 3.1, our analysis is based on a matrix-vector reformulation of LGR, which we introduce in Section 3.2.2. Detailed analyses for fully observable and controllable, and partially observable and controllable are provided in Section 3.2.3 and 3.2.4, respectively.

Notice that, in fact, the analysis we are going to present in the following sections is also valid in the setting of Section 3.1, and is therefore an extension of the results obtained in Sections 3.1.1-3.1.4.

3.2.2 Matrix-vector formulation of LGR

Recall the notation from Section 2.4. From (3.18), we have that $\delta\boldsymbol{y}_\circ$ is linear in $\boldsymbol{\beta}$. Thus, $R'_m(\boldsymbol{\beta}_\circ)(\delta\boldsymbol{\beta}) = \delta\boldsymbol{y}_\circ(A(\delta\boldsymbol{\beta}), \boldsymbol{u}^m; T) = \sum_{j=1}^K \delta\beta_j C\delta\boldsymbol{y}_\circ(A_j, \boldsymbol{u}^m; T)$. Hence, $R'_m(\boldsymbol{\beta}_\circ)$ is a matrix with columns $R'_m(\boldsymbol{\beta}_\circ)_j = C\delta\boldsymbol{y}_\circ(A_j, \boldsymbol{u}^m; T)$ for $j = 1, \ldots, K$, and hence

$$[\widehat{W}_\circ]_{i,j} = \sum_{m=1}^K \langle C\delta\boldsymbol{y}_\circ(A_i, \boldsymbol{u}^m; T), C\delta\boldsymbol{y}_\circ(A_j, \boldsymbol{u}^m; T)\rangle, \quad i, j \in \{1, \ldots, K\}. \tag{3.19}$$

Now, we can use (3.19) to rewrite the sub-problems of LGR (2.23), (2.24) and (2.25) in a matrix form in Lemma 3.2.3, similar to Lemma 3.1.2.

Lemma 3.2.3 (LGR in matrix form) *Consider Algorithm 3. Then:*

- *The initialization problem* (2.23) *is equivalent to*

$$\max_{\boldsymbol{u}\in U_{ad}} [W_\circ(\boldsymbol{u})]_{1,1}, \tag{3.20}$$

  where $[W_\circ(\boldsymbol{u})]_{i,j} := \langle C\delta\boldsymbol{y}_\circ(A_i, \boldsymbol{u}; T), C\delta\boldsymbol{y}_\circ(A_j, \boldsymbol{u}; T)\rangle$ for $i, j \in \{1, \ldots, K\}$.

- *Let $\widehat{W}_\circ^{(k)} := \sum_{m=1}^k W_\circ(\boldsymbol{u}^m)$, the fitting-step problem* (2.24) *is equivalent to*

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^k} \langle \boldsymbol{\beta}, [\widehat{W}_\circ^{(k)}]_{[1:k,1:k]}\boldsymbol{\beta}\rangle - 2\langle [\widehat{W}_\circ^{(k)}]_{[1:k,k+1]}, \boldsymbol{\beta}\rangle. \tag{3.21}$$

- *Let $\boldsymbol{v} := [(\boldsymbol{\beta}^k)^\top, -1]^\top$, the splitting-step problem* (2.25) *is equivalent to*

$$\max_{\boldsymbol{u}\in U_{ad}} \langle \boldsymbol{v}, [W_\circ(\boldsymbol{u})]_{[1:k+1,1:k+1]}\boldsymbol{v}\rangle. \tag{3.22}$$

Moreover, problems (2.23)-(3.20)*,* (2.24)-(3.21)*, and* (2.25)-(3.22) *are well posed.*

Proof. The proof is similar to the one of Lemma 3.1.2. For an arbitrary $k \in \{0, \dots, K-1\}$ let $\boldsymbol{v} \in \mathbb{R}^{k+1}$ and $A(\boldsymbol{v}) = \sum_{j=1}^{k+1} \boldsymbol{v}_j A_j$. We have

$$\|C\delta\boldsymbol{y}_\circ(A(\boldsymbol{v}), \boldsymbol{u}; T)\|_2^2 = \langle \boldsymbol{v}, [W_\circ(\boldsymbol{u})]_{[1:k+1,1:k+1]}\boldsymbol{v}\rangle.$$

Recalling that $\delta\boldsymbol{y}_\circ(A(\boldsymbol{v}), \boldsymbol{u}; T) = \sum_{j=1}^{k+1} \boldsymbol{v}_j \delta\boldsymbol{y}_\circ(A_j, \boldsymbol{u}; T)$, we obtain the equivalence between (3.20), (3.22), and (2.23), (2.25) for suitable k and $\boldsymbol{v}$. For the equivalence between (3.21) and (2.24), notice that for $\boldsymbol{v} = [\boldsymbol{\beta}^\top, -1]^\top \in \mathbb{R}^{k+1}$ and any $W \in \mathbb{R}^{(k+1)\times(k+1)}$ we have

$$\langle \boldsymbol{v}, W\boldsymbol{v}\rangle = \langle \boldsymbol{\beta}, [W]_{[1:k,1:k]}\boldsymbol{\beta}\rangle - 2\langle [W]_{[1:k,k+1]}, \boldsymbol{\beta}\rangle + [W]_{k+1,k+1}.$$

The well-posedness of the three problems follows by standard arguments; see, e.g., the proof of Lemma 3.1.2. □

Recalling the results from Section 3.1.1, the idea of LGR is to iteratively find control functions $\{\boldsymbol{u}_m\}_{m=1}^k$, $k \in \{1, \dots, K\}$, such that the matrix $[W_\circ(\boldsymbol{u})]_{[1:k,1:k]}$ is PD. A sufficient condition, for LGR to be successful, is that any new control $\boldsymbol{u}^{k+1}$ satisfies

$$\|C\delta\boldsymbol{y}_\circ(A^{(k)}(\boldsymbol{\beta}^k), \boldsymbol{u}^{k+1}; T) - C\delta\boldsymbol{y}_\circ(A_{k+1}, \boldsymbol{u}^{k+1}; T)\|_2^2 > 0.$$

The existence of such a control depends on the controllability and observability properties of system (3.18), as shown in Sections 3.2.3 and 3.2.4.

3.2.3 Analysis for fully observable and controllable systems

We define $\mathcal{O}_N^\circ := \mathcal{O}_N(C, A(\boldsymbol{\beta}_\circ))$ and $\mathcal{C}_N^\circ := \mathcal{C}_N(A(\boldsymbol{\beta}_\circ), B)$ and assume that the system $(A(\boldsymbol{\beta}_\circ), B, C)$ is observable and controllable, namely $\mathcal{R} := \operatorname{rank}(\mathcal{O}_N^\circ) \cdot \operatorname{rank}(\mathcal{C}_N^\circ) = N^2$. In what follows, we show that this is a sufficient condition for $\widehat{W}_\circ$ to be PD with the controls generated by the LGR Algorithm 3. First, we need the following result [9, Ch. 3, Thm. 2.11].

Lemma 3.2.4 (controllability of time-invariant systems) *Consider the system* $\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{u}$ *with* $\boldsymbol{x}(0) = 0$ *and its solution* $\boldsymbol{x}(\boldsymbol{u}, t) := \int_0^t e^{(t-s)A(\boldsymbol{\beta}_\circ)} B\boldsymbol{u}(s)ds$*. For any finite time* $t_0 > 0$*, there exists a control* $\boldsymbol{u}$ *that transfers the state to* $\boldsymbol{w}$ *in time* t_0*, i.e.* $\boldsymbol{x}(\boldsymbol{u}, t_0) = \boldsymbol{w}$*, if and only if* $\boldsymbol{w} \in \operatorname{im}\big(\mathcal{C}_N(A, B)\big)$*. Furthermore, an appropriate* $\boldsymbol{u}$ *that will accomplish this transfer in time* t_0 *is given by* $\boldsymbol{u}(t) = B^\top e^{(t_0-t)A^\top}\boldsymbol{\nu}$*, for* $t \in [0, t_0]$ *and* $\boldsymbol{\nu}$ *such that* $\mathcal{W}_c(0, t_0)\boldsymbol{\nu} = \boldsymbol{w}$*, where* $\mathcal{W}_c(0, T) := \int_0^T e^{\tau A} B B^\top e^{\tau A^\top} d\tau$*.*

Now, we prove the following lemma regarding the initialization problem (2.23) and the splitting step problem (2.25). Notice that the proof of this result is inspired by classical Kalman controllability theory; see, e.g., [31].

Lemma 3.2.5 (LGR initialization and splitting steps (linear systems)) *Assume that the matrices* $A(\boldsymbol{\beta}_\circ) \in \mathbb{R}^{N\times N}$*,* $B \in \mathbb{R}^{N\times M}$ *and* $C \in \mathbb{R}^{P\times N}$ *are such that* $\operatorname{rank}(\mathcal{O}_N^\circ) = \operatorname{rank}(\mathcal{C}_N^\circ) = N$*, and let* $\widetilde{A} \in \mathbb{R}^{N\times N} \setminus \{0\}$ *be arbitrary. Then any solution* $\widetilde{\boldsymbol{u}}$ *of the problem* $\max_{\boldsymbol{u}\in U_{ad}} \|C\delta\boldsymbol{y}_\circ(\widetilde{A}, \boldsymbol{u}; T)\|_2^2$ *satisfies*

$$\|C\delta\boldsymbol{y}_\circ(\widetilde{A}, \widetilde{\boldsymbol{u}}; T)\|_2^2 > 0,$$

where $\dot{\delta \boldsymbol{y}}_\circ = A(\boldsymbol{\beta}_\circ)\delta \boldsymbol{y}_\circ + \widetilde{A}\boldsymbol{y}^\circ$, *with* $\delta \boldsymbol{y}_\circ(0) = 0$, *and* $\dot{\boldsymbol{y}}_\circ = A(\boldsymbol{\beta}_\circ)\boldsymbol{y}_\circ + B\boldsymbol{u}$ *with* $\boldsymbol{y}_\circ(0) = 0$

Proof. It is sufficient to construct a $\widetilde{\boldsymbol{u}} \in U_{ad}$ such that $C\delta \boldsymbol{y}_\circ(\widetilde{A}, \widetilde{\boldsymbol{u}}; T) \neq 0$. Since $\widetilde{A} \neq 0$, there exists $\boldsymbol{w} \in \mathbb{R}^N \setminus \{0\}$ such that $\widetilde{A}\boldsymbol{w} \neq 0$. Since $(A(\boldsymbol{\beta}_\circ), B, C)$ is observable, there exists $\tilde{t} > 0$ such that $Ce^{\tilde{t}A(\boldsymbol{\beta}_\circ)}\widetilde{A}\boldsymbol{w} \neq 0$. The map $f : \mathbb{R} \to \mathbb{R}^P, t \mapsto Ce^{tA(\boldsymbol{\beta}_\circ)}\widetilde{A}\boldsymbol{w}$ is analytic with derivatives $f^{(i)}(t) = CA(\boldsymbol{\beta}_\circ)^i e^{tA(\boldsymbol{\beta}_\circ)}\widetilde{A}\boldsymbol{w}$. Since $\mathcal{O}_N^\circ$ has full rank and $e^{\tilde{t}A(\boldsymbol{\beta}_\circ)}\widetilde{A}\boldsymbol{w} \neq 0$, there exists $i \in \{0, \dots, N\}$ such that $f^{(i)}(\tilde{t}) = CA(\boldsymbol{\beta}_\circ)^i e^{\tilde{t}A(\boldsymbol{\beta}_\circ)}\widetilde{A}\boldsymbol{w} \neq 0$. Hence, f is nonconstant, and there exists $t_0 \in (0, T)$ with $Ce^{t_0 A(\boldsymbol{\beta}_\circ)}\widetilde{A}\boldsymbol{w} \neq 0$.

Now, we use that $\boldsymbol{y}_\circ(\boldsymbol{u}, s) := \int_0^s e^{(s-\tau)A(\boldsymbol{\beta}_\circ)}B\boldsymbol{u}(\tau)d\tau$ is the solution at time s of $\dot{\boldsymbol{y}}_\circ = A(\boldsymbol{\beta}_\circ)\boldsymbol{y}_\circ + B\boldsymbol{u}$, with $\boldsymbol{y}_\circ(0) = 0$. Since $\mathcal{C}_N^\circ$ has full rank, we have $\boldsymbol{w} \in \operatorname{im}\big(\mathcal{C}_N^\circ\big)$. Thus, Lemma 3.2.4 guarantees that $\widehat{\boldsymbol{u}}(t) = B^\top e^{(t_0-t)A(\boldsymbol{\beta}_\circ)^\top}\boldsymbol{\nu}$, for $t \in [0, t_0]$ and some $\boldsymbol{\nu} \in \mathbb{R}^N$, satisfies $\boldsymbol{y}_\circ(\widehat{\boldsymbol{u}}, t_0) = \boldsymbol{w}$. Clearly, $\widehat{\boldsymbol{u}}$ is analytic in $[0, t_0]$ and thereby the same holds for $\boldsymbol{y}_\circ(\widehat{\boldsymbol{u}}, s)$. Note that, since $\boldsymbol{u} \equiv 0$ is an interior point of U_{ad}, there exists $\lambda > 0$ such that $\lambda\widehat{\boldsymbol{u}} \in U_{ad}$ with $Ce^{t_0 A(\boldsymbol{\beta}_\circ)}\widetilde{A}\boldsymbol{y}_\circ(\lambda\widehat{\boldsymbol{u}}, t_0) = \lambda Ce^{t_0 A(\boldsymbol{\beta}_\circ)}\widetilde{A}\boldsymbol{y}_\circ(\widehat{\boldsymbol{u}}, t_0) \neq 0$. Hence, we can assume without loss of generality that $\widehat{\boldsymbol{u}} \in U_{ad}$.

In conclusion, we obtain that the map $\boldsymbol{g} : \mathbb{R} \to \mathbb{R}^P$ defined as

$$s \mapsto Ce^{(T-s)A(\boldsymbol{\beta}_\circ)}\widetilde{A}\int_0^s e^{(s-\tau)A(\boldsymbol{\beta}_\circ)}B\widehat{\boldsymbol{u}}(\tau)d\tau,$$

is analytic in $(0, t_0)$ with $\boldsymbol{g}(t_0) \neq 0$. Thus, $\boldsymbol{g}$ is nonzero in an open subinterval of $(0, t_0)$. Hence, there exists $t_1 \in (0, t_0)$ such that $\int_0^{t_1} \boldsymbol{g}(s)ds \neq 0$. By choosing

$$\widetilde{\boldsymbol{u}}(s) := \begin{cases} 0, & 0 \leq s < T - t_1, \\ \widehat{\boldsymbol{u}}(s - t_1), & T - t_1 \leq s \leq T, \end{cases}$$

and using that $C\delta \boldsymbol{y}_\circ(\widetilde{A}, \widetilde{\boldsymbol{u}}; T) = \int_0^T Ce^{(T-s)A(\boldsymbol{\beta}_\circ)}\widetilde{A}\int_0^s e^{(s-\tau)A(\boldsymbol{\beta}_\circ)}B\widetilde{\boldsymbol{u}}(\tau)d\tau ds$, we obtain

$$C\delta \boldsymbol{y}_\circ(\widetilde{A}, \widetilde{\boldsymbol{u}}; T) = \int_0^{t_1} Ce^{(t_1-s)A(\boldsymbol{\beta}_\circ)}\widetilde{A}\int_0^s e^{(s-\tau)A(\boldsymbol{\beta}_\circ)}B\widehat{\boldsymbol{u}}(\tau)d\tau ds = \int_0^{t_1} \boldsymbol{g}(s)ds \neq 0.$$

□

Lemma 3.2.5 can be applied to both (2.23) and (2.25), choosing $\widetilde{A} = A_1$ and $\widetilde{A} = \big(A^{(k)}(\boldsymbol{\beta}^k) - A_{k+1}\big)$, respectively. Now, we can prove the main result of this section.

Theorem 3.2.6 (positive definiteness of the GN matrix $\widehat{W}_\circ$ (linear systems)) *Assume that* $A(\boldsymbol{\beta}_\circ) \in \mathbb{R}^{N\times N}$, $B \in \mathbb{R}^{N\times M}$ *and* $C \in \mathbb{R}^{P\times N}$ *are such that* $\operatorname{rank}(\mathcal{O}_N^\circ) = \operatorname{rank}(\mathcal{C}_N^\circ) = N$. *For* $K \leq N^2$, *let* $\mathcal{B} = \{A_1, \dots, A_K\} \subset \mathbb{R}^{N\times N}$ *be a set of linearly independent matrices such that* $A(\boldsymbol{\beta}_\circ) \in \operatorname{span}(\mathcal{B})$, *and let* $\{\boldsymbol{u}^1, \dots, \boldsymbol{u}^K\} \subset U_{ad}$ *be generated by Algorithm 3. Then the GN matrix* $\widehat{W}_\circ$, *defined in* (3.19), *is PD.*

Proof. We proceed by induction. Lemma 3.2.5 guarantees that there exists a $\boldsymbol{u}^1$ such that $[W_\circ(\boldsymbol{u}^1)]_{1,1} = \|C\delta \boldsymbol{y}_\circ(A_1, \boldsymbol{u}^1; T)\|_2^2 > 0$. Now, let us assume that $[\widehat{W}_\circ^{(k)}]_{[1:k,1:k]} = \sum_{m=1}^k [W_\circ(\boldsymbol{u}^m)]_{[1:k,1:k]}$ is PD. By construction, $[\widehat{W}_\circ^{(k+1)}]_{[1:k+1,1:k+1]}$ is PSD. Thus, if $[\widehat{W}_\circ^{(k)}]_{[1:k+1,1:k+1]}$ is PD, then

$$[\widehat{W}_\circ^{(k+1)}]_{[1:k+1,1:k+1]} = [\widehat{W}_\circ^{(k)} + W_\circ(\boldsymbol{u}^{k+1})]_{[1:k+1,1:k+1]}$$

is PD as well, since $[W_\circ(\boldsymbol{u}^k)]_{[1:k+1,1:k+1]}$ is PSD. Assume now that the submatrix $[\widehat{W}_\circ^{(k)}]_{[1:k+1,1:k+1]}$ has a nontrivial kernel. Since $[\widehat{W}_\circ^{(k)}]_{[1:k,1:k]}$ is PD (induction hypothesis), problem (2.24) is uniquely solvable with solution $\boldsymbol{\beta}^k$. Then, by Lemma 3.1.3 the (one-dimensional) kernel of $[\widehat{W}_\circ^{(k)}]_{[1:k+1,1:k+1]}$ is the span of the vector $\boldsymbol{v} = [(\boldsymbol{\beta}^k)^\top,\ -1]^\top$. Using Lemma 3.2.5 we obtain that the solution $\boldsymbol{u}^{k+1}$ to the splitting step problem satisfies

$$\langle \boldsymbol{v}, [W_\circ(\boldsymbol{u}^{k+1})]_{[1:k+1,1:k+1]}\boldsymbol{v}\rangle = \|C\delta\boldsymbol{y}_\circ(A^{(k)}(\boldsymbol{\beta}^k) - A_{k+1}, \boldsymbol{u}; T)\|_2^2 > 0.$$

Thus, $[W(\boldsymbol{u}^{k+1})]_{[1:k+1,1:k+1]}$ is PD on the span of $\boldsymbol{v}$, and $[\widehat{W}_\circ^{(k+1)}]_{[1:k+1,1:k+1]}$ is PD. □

Notice that Theorem 3.2.6 does not require any assumption on the matrices $A_1, \dots, A_K$. These can be arbitrarily chosen with the only constraint to be linearly independent. Also the ordering of these matrices does not affect the result of Theorem 3.2.6. This is, however, different for partially observable and controllable systems, i.e. for $\mathcal{R} < N^2$.

3.2.4 Analysis for partially observable and controllable systems

The results of Section 3.2.3 are no longer true if $(A(\boldsymbol{\beta}_\circ), B, C)$ is not fully observable and controllable, that is, $\operatorname{rank}(\mathcal{O}_N^\circ)\cdot\operatorname{rank}(\mathcal{C}_N^\circ) < N^2$. We denote from now on $\mathcal{R} := \mathcal{R}_\mathcal{O}\mathcal{R}_\mathcal{C}$ with

$$\mathcal{R}_\mathcal{O} := \operatorname{rank}(\mathcal{O}_N^\circ), \quad \mathcal{R}_\mathcal{C} := \operatorname{rank}(\mathcal{C}_N^\circ).$$

In what follows, the choice of the linearly independent matrices $A_1, \dots, A_K$ and their ordering becomes crucial for the algorithm, and we show that one can make the matrix $\widehat{W}_\circ$ PD on an $\mathcal{R}$-dimensional subspace of $\mathbb{R}^K$, if $A_1, \dots, A_K$ are appropriately chosen. We begin by choosing a set of $K = N^2$ matrices by exploiting the kernel of the Kronecker product of $\mathcal{O}_n^\circ$ and $\mathcal{C}_N^\circ$. Using [49, Theorem 4.2.15], we obtain $\operatorname{rank}\Big((\mathcal{C}_N^\circ)^\top \otimes \mathcal{O}_N^\circ\Big) = \mathcal{R}_\mathcal{C}\mathcal{R}_\mathcal{O}$. Now, the rank-nullity theorem allows us to consider a basis $\{\boldsymbol{v}_j\}_{j=1}^{N^2}$ of $\mathbb{R}^{N^2}$ such that

$$\boldsymbol{v}_j \notin \ker\Big((\mathcal{C}_N^\circ)^\top \otimes \mathcal{O}_N^\circ\Big), j = 1, \dots, \mathcal{R}, \tag{3.23}$$

$$\boldsymbol{v}_j \in \ker\Big((\mathcal{C}_N^\circ)^\top \otimes \mathcal{O}_N^\circ\Big), j = \mathcal{R}+1, \dots, N^2, \tag{3.24}$$

with $\operatorname{span}\{\boldsymbol{v}_j\}_{j=\mathcal{R}+1}^{N^2} = \ker\Big((\mathcal{C}_N^\circ)^\top \otimes \mathcal{O}_N^\circ\Big)$. We introduce the operation

$$\operatorname{vec}(A) := \Big[A_{[1,1]}, \dots, A_{[N,1]}, A_{[1,2]}, \dots, A_{[N,2]}, \dots, A_{[1,N]}, \dots, A_{[N,N]}\Big]^\top,$$

that stacks the columns of a matrix $A \in \mathbb{R}^{N\times N}$ below each other. Using this notation, we define the matrices $A_1^\mathcal{O}, \dots, A_{N^2}^\mathcal{O}$ by rearranging $\boldsymbol{v}$ as

$$\operatorname{vec}(A_j^\mathcal{O}) = \boldsymbol{v}_j, \quad j = 1, \dots, N^2. \tag{3.25}$$

Since the vectors $\{\boldsymbol{v}_j\}_{j=1}^{N^2}$ are linearly independent, the set $\{A_j^\mathcal{O}\}_{j=1}^{N^2}$ is a basis of $\mathbb{R}^{N\times N}$.

From a computational point of view, the vectors $\boldsymbol{v}_j$ can be obtained by the singular value decomposition (SVD) of $(\mathcal{C}_N^\circ)^\top \otimes \mathcal{O}_N^\circ = U\Sigma V^\top$, where the columns of V form a basis of

$\mathbb{R}^{N^2}$. The last $N^2-\mathcal{R}$ columns of V span the kernel of $(\mathcal{C}_N^\circ)^\top \otimes \mathcal{O}_N^\circ$; see, e.g., [68, Theorem 5.2]. Hence, we set $\boldsymbol{v}_j = [V]_{[:,j]}$ for $j = 1, \dots, N^2$. Notice that for $\mathcal{C}_N^\circ = U_\mathcal{C}\Sigma_\mathcal{C}V_\mathcal{C}^\top$ and $\mathcal{O}_N^\circ = U_\mathcal{O}\Sigma_\mathcal{O}V_\mathcal{O}^\top$, the result from [49, Theorem 4.2.15] says that $V = U_\mathcal{C}^\top \otimes V_\mathcal{O}$. Thus, we can compute V using only the SVD of $\mathcal{O}_n^\circ$ and $\mathcal{C}_N^\circ$, which are also used to get $\mathcal{R} = \mathcal{R}_\mathcal{O}\mathcal{R}_\mathcal{C}$, and avoid computations with the large matrix $(\mathcal{C}_N^\circ)^\top \otimes \mathcal{O}_N^\circ \in \mathbb{R}^{N^2MP\times N^2}$.

Our first result says that $C\delta\boldsymbol{y}_\circ(A, \boldsymbol{u}; T) = 0$ for any $A \in \operatorname{span}\{A_j^\mathcal{O}\}_{j=\mathcal{R}+1}^{N^2}$.

Lemma 3.2.7 (reconstructing partially observable and controllable systems) *Consider $\{A_j^\mathcal{O}\}_{j=1}^{N^2}$ constructed as in (3.25), with vectors $\boldsymbol{v}_j$, $j = 1, \dots, N^2$ as in (3.23)-(3.24). Then $C\delta\boldsymbol{y}_\circ(A, \boldsymbol{u}; T) = 0$ for any $A \in \operatorname{span}\{A_j^\mathcal{O}\}_{j=\mathcal{R}+1}^{N^2}$ and $\boldsymbol{u} \in U_{ad}$, i.e. the last $N^2 - \mathcal{R}$ rows and columns of $\widehat{W}_\circ$, defined in (4.6), are zero.*

Proof. Notice that, by the Taylor expansion of the exponential matrix and the Cayley-Hamilton theorem (see, e.g., [49, page 109]), for any $i \in \{1, \dots, N^2\}$, $s \in [0, T]$ and $\tau \in [0, s]$, there exist N functions $\beta_\ell : [0, T] \times [0, T] \to \mathbb{R}$, $\ell = 0, \dots, N-1$, such that

$$Ce^{(T-s)A(\boldsymbol{\beta}_\circ)}A_i^\mathcal{O}e^{(s-\tau)A(\boldsymbol{\beta}_\circ)}B = \sum_{p=0}^{N-1}\sum_{\ell=0}^{N-1}\beta_p(T,s)\beta_\ell(s,\tau)\big(CA(\boldsymbol{\beta}_\circ)^p\big)A_i^\mathcal{O}\big(A(\boldsymbol{\beta}_\circ)^\ell B\big).$$

If $i \in \{\mathcal{R}+1, \dots, N^2\}$, then $\operatorname{vec}(A_i^\mathcal{O}) = \boldsymbol{v}_i \in \ker\Big((\mathcal{C}_N^\circ)^\top \otimes \mathcal{O}_N^\circ\Big)$ and therefore

$$0 = \Big((\mathcal{C}_N^\circ)^\top \otimes \mathcal{O}_N^\circ\Big)\operatorname{vec}(A_i^\mathcal{O}) = \operatorname{vec}\Big(\mathcal{O}_N^\circ A_i^\mathcal{O}\mathcal{C}_N^\circ\Big).$$

Thus, using the definitions of $\mathcal{O}_\circ$ and $\mathcal{C}_\circ$, we get $\big(CA(\boldsymbol{\beta}_\circ)^p\big)A_i^\mathcal{O}\big(A(\boldsymbol{\beta}_\circ)^\ell B\big) = 0$ for any $\ell, p \in \{0, \dots, N-1\}$. Hence, $(Ce^{(T-s)A(\boldsymbol{\beta}_\circ)})A_i^\mathcal{O}(e^{sA(\boldsymbol{\beta}_\circ)}B) = 0$ for all $i \in \{\mathcal{R}+1, \dots, N^2\}$ and $s \in [0, T]$. Now, let $\boldsymbol{u} \in U_{ad}$ be an arbitrary function. Then we have

$$C\delta\boldsymbol{y}_\circ(A_i^\mathcal{O}, \boldsymbol{u}; T) = \int_0^T Ce^{(T-s)A(\boldsymbol{\beta}_\circ)}A_i^\mathcal{O}\int_0^s e^{(s-\tau)A(\boldsymbol{\beta}_\circ)}B\boldsymbol{u}(\tau)d\tau ds = 0.$$

Recalling (4.6), we obtain that $[W_\circ(\boldsymbol{u})]_{i,j} = 0$ for $i, j \geq N^2 - \mathcal{R}$ and the result follows. □

Lemma 3.2.7 implies that one can only make the upper-left $\mathcal{R}\times\mathcal{R}$ submatrix of $\widehat{W}_\circ$ PD. This means that only the first $\mathcal{R}$ elements of the basis are of interest. In fact, as we are going to show in Lemma 3.2.8 and Theorem 3.2.9, it only makes sense to use LGR with the first $\mathcal{R}$ basis elements $A_1^\mathcal{O}, \dots, A_\mathcal{R}^\mathcal{O}$, and to consider the submatrix $[\widehat{W}_\circ]_{1:\mathcal{R},1:\mathcal{R}}$. In this case, the proof for our main result is similar to the one of Theorem 3.2.6. We first prove a version of Lemma 3.2.5 adapted to partially observable and controllable systems.

Lemma 3.2.8 (LGR initialization and splitting steps for $\mathcal{R} < N^2$) *Assume that the matrices $A(\boldsymbol{\beta}_\circ) \in \mathbb{R}^{N\times N}$, $B \in \mathbb{R}^{N\times M}$ and $C \in \mathbb{R}^{P\times N}$ are such that $\operatorname{rank}(\mathcal{O}_N^\circ) \cdot \operatorname{rank}(\mathcal{C}_N^\circ) = \mathcal{R} < N^2$. Let $\widetilde{A} \in \Big(\operatorname{span}\{A_j^\mathcal{O}\}_{j=1}^\mathcal{R}\Big) \setminus \{0\}$ for $A_j^\mathcal{O}$ as in (3.25). Any solution $\widetilde{\boldsymbol{u}}$ to the maximization problem $\max_{\boldsymbol{u}\in U_{ad}} \|C\delta\boldsymbol{y}_\circ(\widetilde{A}, \boldsymbol{u}; T)\|_2^2$ satisfies*

$$\|C\delta\boldsymbol{y}_\circ(\widetilde{A}, \widetilde{\boldsymbol{u}}; T)\|_2^2 > 0, \tag{3.26}$$

where $\dot{\delta\boldsymbol{y}}_\circ = A(\boldsymbol{\beta}_\circ)\delta\boldsymbol{y}_\circ + \widetilde{A}\boldsymbol{y}$ with $\delta\boldsymbol{y}_\circ(0) = 0$ and $\dot{\boldsymbol{y}} = A(\boldsymbol{\beta}_\circ)\boldsymbol{y} + B\boldsymbol{u}$ with $\boldsymbol{y}(0) = 0$.

Proof. Using the calculations in the proof of Lemma 3.2.7, we have $\mathcal{O}_N^\circ A_\ell^{\mathcal{O}} \mathcal{C}_N^\circ \neq 0$, for any $\ell \in \{1, \dots, \mathcal{R}\}$. Since $\widetilde{A} \in \left(\text{span}\{A_j^{\mathcal{O}}\}_{j=1}^{\mathcal{R}}\right) \setminus \{0\}$ and by the construction of the $A_j^{\mathcal{O}}$, we also obtain $\mathcal{O}_N^\circ \widetilde{A} \mathcal{C}_N^\circ \neq 0$. Hence, there exists $\boldsymbol{w} \in \text{im}(\mathcal{C}_N^\circ)$ such that $\widetilde{A}\boldsymbol{w} \notin \ker(\mathcal{O}_N^\circ)$ and therefore there exists $\tilde{t} > 0$ such that $Ce^{\tilde{t}A(\boldsymbol{\beta}_\circ)}\widetilde{A}w \neq 0$. With this observation, the result can be proved analogously to Lemma 5.5. □

Following exactly the proof of Theorem 3.2.6, in which Lemma 3.2.8 has to used instead of Lemma 3.2.5, we can prove our main result.

Theorem 3.2.9 (positive definiteness of the GN matrix $\widehat{W}_\circ$ for $\mathcal{R} < N^2$) *Let $A(\boldsymbol{\beta}_\circ) \in \mathbb{R}^{N\times N}$, $B \in \mathbb{R}^{N\times M}$ and $C \in \mathbb{R}^{P\times N}$ be such that $\text{rank}(\mathcal{O}_N^\circ) \cdot \text{rank}(\mathcal{C}_N^\circ) = \mathcal{R} < N^2$. Let $\{\boldsymbol{u}^m\}_{m=1}^{\mathcal{R}} \subset U_{ad}$ be generated by the LGR Algorithm 3 with $K = \mathcal{R}$ and using the matrices $A_1^{\mathcal{O}}, \dots, A_{\mathcal{R}}^{\mathcal{O}}$ defined in* (3.25)*. Then $[\widehat{W}_\circ]_{[1:\mathcal{R},1:\mathcal{R}]}$ is PD.*

Having proved that Algorithm 3 makes $\widehat{W}_\circ$ PD, the obvious question is whether this is sufficient for the convergence of GN (Lemma 2.4.1). We answer this question in Section 3.2.5.

3.2.5 Positive definiteness of the GN matrix

To account for both fully and partially observable and controllable cases, we assume for the rest of Section 3.2 that the basis $\mathcal{B} = \{A_1, \dots, A_K\}$ satisfies $K \leq \mathcal{R} = \text{rank}(\mathcal{O}_N^\circ) \cdot \text{rank}(\mathcal{C}_N^\circ) \leq N^2$ and if $\mathcal{R} < N^2$, then $A_j \in \{A_i^{\mathcal{O}}\}_{i=1}^{\mathcal{R}}$ for $j = 1, \dots, K$.

To guarantee convergence of GN, we need to show that $\widehat{W}(\boldsymbol{\beta}) := \sum_{m=1}^K R_m'(\boldsymbol{\beta})^\top R_m'(\boldsymbol{\beta})$ (defined in Section 2.4) remains PD in a neighborhood of $\boldsymbol{\beta}_\star$. In Sections 3.2.3 and 3.2.4, we proved that the controls generated by Algorithm 3 make $\widehat{W}_\circ = \widehat{W}(\boldsymbol{\beta}_\circ)$ PD. Thus, it is sufficient to prove that $\widehat{W}(\boldsymbol{\beta})$ is PD in a neighborhood of $\boldsymbol{\beta}_\circ$ containing $\boldsymbol{\beta}_\star$. To do so, we write $\widehat{W}(\boldsymbol{\beta})$ as

$$[\widehat{W}(\boldsymbol{\beta})]_{i,j} := \sum_{m=1}^K \langle \boldsymbol{\gamma}_i(\boldsymbol{\beta}, \boldsymbol{u}^m), \boldsymbol{\gamma}_j(\boldsymbol{\beta}, \boldsymbol{u}^m)\rangle, \quad i,j \in \{1, \dots, K\}, \tag{3.27}$$

$$\boldsymbol{\gamma}_j(\boldsymbol{\beta}, \boldsymbol{u}^m) := \int_0^T Ce^{(T-s)A(\boldsymbol{\beta})} A_j \boldsymbol{y}(A(\boldsymbol{\beta}), \boldsymbol{u}^m; s) ds, \quad j \in \{1, \dots, K\}, \tag{3.28}$$

and recall the next lemma, which follows from the Bauer-Fike theorem [18].

Lemma 3.2.10 (rank stability) *Consider $N_D, M_D \in \mathbb{N}^+$ with $N_D \geq M_D$, and a matrix $D \in \mathbb{R}^{N_D \times M_D}$ with rank $\mathcal{R}_{\mathcal{D}}$ and (positive) singular values $\sigma_1, \dots, \sigma_{\mathcal{R}_{\mathcal{D}}}$ in descending order. Then $\min_{\widehat{D}\in\mathbb{R}^{N_D\times M_D}} \{\|\widehat{D}\|_2 \mid \text{rank}(D + \widehat{D}) < \mathcal{R}_{\mathcal{D}}\} = \sigma_{\mathcal{R}_{\mathcal{D}}}$.*

Using this lemma, we can prove the following approximation result.

Lemma 3.2.11 (positive definiteness of $\widehat{W}(\boldsymbol{\beta})$ (linear systems)) *Let $\widehat{W}_\circ$ defined in* (3.19) *be PD and let $\sigma_K^\circ > 0$ be its smallest singular value. Then, there exists $\delta := \delta(\sigma_K^\circ) > 0$ such that $\widehat{W}(\boldsymbol{\beta})$ (in* (3.27)*) is PD for any $\boldsymbol{\beta} \in \mathbb{R}^K$ with $\|\boldsymbol{\beta} - \boldsymbol{\beta}_\circ\|_2 < \delta$.*

Proof. Our first goal is to show that $\widehat{W}(\boldsymbol{\beta})$ is continuous in $\boldsymbol{\beta}$. From (3.27) and (3.28) we know that $\widehat{W}(\boldsymbol{\beta})$ is the sum over products of $\int_0^T Ce^{(T-s)A(\boldsymbol{\beta})}A_j \boldsymbol{y}(A(\boldsymbol{\beta}), \boldsymbol{u}^m; s)ds$, where $\boldsymbol{y}(A(\boldsymbol{\beta}), \boldsymbol{u}^m; s) = \int_0^s e^{(s-\tau)A(\boldsymbol{\beta})}B\boldsymbol{u}^m(\tau)d\tau$. Now, recall that $A(\boldsymbol{\beta}) = \sum_{j=1}^K \beta_j A_j$, meaning that $A(\boldsymbol{\beta})$ is continuous in $\boldsymbol{\beta}$. Since the exponential map $\mathbb{R}^N \to \mathbb{R}^{N\times N}, \boldsymbol{\beta} \mapsto e^{sA(\boldsymbol{\beta})}$ and the integral map $\mathbb{R}^{N\times N} \to \mathbb{R}^N, X \mapsto \int_0^s XB\boldsymbol{u}(\tau)d\tau$ are continuous, we obtain that $\boldsymbol{y}(A(\boldsymbol{\beta}), \boldsymbol{u}^m; s)$ is continuous in $\boldsymbol{\beta}$. Since products of continuous functions are continuous, we obtain that $\widehat{W}(\boldsymbol{\beta})$ is continuous in $\boldsymbol{\beta}$.

By assumption, $\widehat{W}_\circ$ is PD, and therefore $\sigma_K^\circ > 0$. Since $\widehat{W}(\boldsymbol{\beta})$ is continuous in $\boldsymbol{\beta}$, we obtain that there exists a $\delta := \delta(\sigma_K^\circ) > 0$ such that for any $\boldsymbol{\beta}$ with $\|\boldsymbol{\beta} - \boldsymbol{\beta}_\circ\|_2 < \delta(\sigma_K^\circ)$ it holds that $\|\widehat{W}(\boldsymbol{\beta}) - \widehat{W}(\boldsymbol{\beta}_\circ)\|_2 < \sigma_K^\circ$. Now, let $\widehat{\boldsymbol{\beta}}$ be such that $\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_\circ\|_2 < \delta(\sigma_K^\circ)$ and hence $\|\widehat{W}(\widehat{\boldsymbol{\beta}}) - \widehat{W}(\boldsymbol{\beta}_\circ)\|_2 < \sigma_K^\circ$. Setting $D = \widehat{W}(\boldsymbol{\beta}_\circ)$ and $\widehat{D} = \widehat{W}(\widehat{\boldsymbol{\beta}}) - \widehat{W}(\boldsymbol{\beta}_\circ)$, Lemma 3.2.10 implies that $K = \operatorname{rank}(\widehat{W}(\boldsymbol{\beta}_\circ)) \leq \operatorname{rank}(\widehat{W}(\widehat{\boldsymbol{\beta}}))$. Because of (3.27), $\widehat{W}(\widehat{\boldsymbol{\beta}}) \in \mathbb{R}^{K\times K}$ meaning that $\operatorname{rank}(\widehat{W}(\widehat{\boldsymbol{\beta}})) = K$. Since $\widehat{W}(\boldsymbol{\beta})$ is PSD by construction, $\operatorname{rank}(\widehat{W}(\widehat{\boldsymbol{\beta}})) = K$ implies that $\widehat{W}(\widehat{\boldsymbol{\beta}})$ is PD. □

Lemma 3.2.11 implies that the positive definiteness of $\widehat{W}(\boldsymbol{\beta})$ is locally preserved near $\boldsymbol{\beta}_\circ$. Now, we can prove our main convergence result.

Theorem 3.2.12 (convergence of GN (linear systems)) *Let $\boldsymbol{\beta}_\circ \in \mathbb{R}^K$ be such that $A(\boldsymbol{\beta}_\circ) \in \mathbb{R}^{N\times N}$, $B \in \mathbb{R}^{N\times M}$ and $C \in \mathbb{R}^{P\times N}$ satisfy $\operatorname{rank}(\mathcal{O}_N^\circ) \cdot \operatorname{rank}(\mathcal{C}_N^\circ) = \mathcal{R} \leq N^2$. Let $\{\boldsymbol{u}^m\}_{m=1}^K \subset U_{ad}$ be a set of controls generated by Algorithm 3. Finally, let $\widehat{\sigma}_K$ be the K-th (smallest) singular value of $\widehat{W}_\circ$ defined in* (3.19)*. Then there exists $\delta = \delta(\widehat{\sigma}_K) > 0$ such that if $\boldsymbol{\beta}_\star \in \mathbb{R}^K$ satisfies $\|\boldsymbol{\beta}_\star - \boldsymbol{\beta}_\circ\|_2 < \delta$, then GN method for the problem*

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^K} \frac{1}{2} \sum_{m=1}^K \|C\boldsymbol{y}(A(\boldsymbol{\beta}_\star), \boldsymbol{u}^m; T) - C\boldsymbol{y}(A(\boldsymbol{\beta}), \boldsymbol{u}^m; T)\|_2^2, \tag{3.29}$$

initialized with $\boldsymbol{\beta}_\circ$, converges to $\boldsymbol{\beta}_\star$.

Proof. Theorem 3.2.6 guarantees that $\widehat{W}_\circ$ is PD. Thus, $\widehat{\sigma}_K > 0$ and by Lemma 3.2.11 there exists $\delta = \delta(\widehat{\sigma}_K) > 0$ such that $\widehat{W}(\boldsymbol{\beta})$ is PD for $\boldsymbol{\beta} \in \mathbb{R}^K$ with $\|\boldsymbol{\beta} - \boldsymbol{\beta}_\circ\|_2 < \delta$. Moreover, we know from Section 2.1 that $\widehat{W}(\boldsymbol{\beta}_c)$ is the GN matrix for the iterate $\boldsymbol{\beta}_c \in \mathbb{R}^K$ of GN for (2.3). As in the proof of Lemma 3.2.11, one can also show that the functions $R_m(\boldsymbol{\beta})$, defined in (2.14), are Lipschitz continuously differentiable in $\boldsymbol{\beta}$ for all $m \in \{1, \ldots, K\}$. Hence, if $\|\boldsymbol{\beta}_\star - \boldsymbol{\beta}_\circ\|_2 < \delta$, then the result follows by Lemma 2.4.1. □

However, the results in this section also imply that if $K > \mathcal{R}$, then the GN matrix $\widehat{W}_\circ$ can not be full rank. Thus, there is no more guarantee that GN converges to $\boldsymbol{\beta}_\star$ and the online reconstruction problem (2.3) can have infinitely many solutions.

3.2.6 Local uniqueness of solutions

Theorem 3.2.12 says that GN converges to $\boldsymbol{\beta}_\star$ if an appropriate initialization vector $\boldsymbol{\beta}_\circ$ is used. However, in the linear case corresponding to (3.17) we can specify the local properties of problem (2.3) around the solution $\boldsymbol{\beta}_\star$. To this end, we start by rewriting the cost function in a matrix form.

Lemma 3.2.13 (reconstruction problem in matrix form (linear systems)) *Problem* (2.3) *for the linear system* (3.17) *is equivalent to*

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^K} \frac{1}{2}\langle \boldsymbol{\beta}-\boldsymbol{\beta}_\star, \widetilde{W}(\boldsymbol{\beta}_\star,\boldsymbol{\beta})(\boldsymbol{\beta}-\boldsymbol{\beta}_\star)\rangle, \tag{3.30}$$

where $\widetilde{W}(\boldsymbol{\beta}_\star,\boldsymbol{\beta}) \in \mathbb{R}^{K\times K}$ *is defined as*[3] $\widetilde{W}(\boldsymbol{\beta}_\star,\boldsymbol{\beta}) := \sum_{m=1}^K W(\boldsymbol{\beta}_\star,\boldsymbol{\beta},\boldsymbol{u}^m)$*, with*

$$[W(\boldsymbol{\beta}_\star,\boldsymbol{\beta},\boldsymbol{u}^m)]_{i,j} := \langle \boldsymbol{\gamma}_i(\boldsymbol{\beta}_\star,\boldsymbol{\beta},\boldsymbol{u}^m), \boldsymbol{\gamma}_j(\boldsymbol{\beta}_\star,\boldsymbol{\beta},\boldsymbol{u}^m)\rangle, \quad i,j\in\{1,\dots,K\}, \tag{3.31}$$

$$\boldsymbol{\gamma}_j(\boldsymbol{\beta}_\star,\boldsymbol{\beta},\boldsymbol{u}^m) := \int_0^T Ce^{(T-s)A(\boldsymbol{\beta}_\star)}A_j\boldsymbol{y}(A(\boldsymbol{\beta}),\boldsymbol{u}^m;s)ds, \quad j\in\{1,\dots,K\}. \tag{3.32}$$

Proof. Let $J(\boldsymbol{\beta}) := \frac{1}{2}\sum_{m=1}^K \|C\boldsymbol{y}(A(\boldsymbol{\beta}),\boldsymbol{u}^m;T) - C\boldsymbol{y}(A_\star,\boldsymbol{u}^m;T)\|_2^2$. For $t\in[0,T]$ and $m\in\{1,\dots,K\}$ define $\Delta\boldsymbol{y}_m(t) := \boldsymbol{y}(A(\boldsymbol{\beta}),\boldsymbol{u}^m;t) - \boldsymbol{y}(A_\star,\boldsymbol{u}^m;t)$. Then we have

$$\begin{aligned}\frac{d}{dt}(\Delta\boldsymbol{y}_m)(t) &= A(\boldsymbol{\beta})\boldsymbol{y}(A(\boldsymbol{\beta}),\boldsymbol{u}^m;t) + B\boldsymbol{u}^m(t) - A(\boldsymbol{\beta}_\star)\boldsymbol{y}(A_\star,\boldsymbol{u}^m;t) - B\boldsymbol{u}^m(t)\\ &= A(\boldsymbol{\beta}_\star)\Delta\boldsymbol{y}_m(t) + A(\boldsymbol{\beta}-\boldsymbol{\beta}_\star)\boldsymbol{y}(A(\boldsymbol{\beta}),\boldsymbol{u}^m;t),\end{aligned}$$

whose solution at $t=T$ is

$$\Delta\boldsymbol{y}_m(T) = \int_0^T e^{(T-s)A(\boldsymbol{\beta}_\star)}\big[A(\boldsymbol{\beta}-\boldsymbol{\beta}_\star)\boldsymbol{y}(A(\boldsymbol{\beta}),\boldsymbol{u}^m;s)\big]ds.$$

Thus, recalling $A(\boldsymbol{\beta}) = \sum_{j=1}^K \beta_j A_j$, a direct calculation shows that $J(\boldsymbol{\beta})$ can be written as $J(\boldsymbol{\beta}) = \frac{1}{2}\langle \boldsymbol{\beta}-\boldsymbol{\beta}_\star, \widetilde{W}(\boldsymbol{\beta}_\star,\boldsymbol{\beta})(\boldsymbol{\beta}-\boldsymbol{\beta}_\star)\rangle$. □

Now, the set of global solutions to (3.30) is $\mathcal{S}_{global} := \big\{\boldsymbol{\beta}\in\mathbb{R}^K \mid (\boldsymbol{\beta}-\boldsymbol{\beta}_\star)\in\ker\widetilde{W}(\boldsymbol{\beta}_\star,\boldsymbol{\beta})\big\}$. Since $\widetilde{W}(\boldsymbol{\beta}_\star,\boldsymbol{\beta})$ is symmetric PSD, (3.30) is locally uniquely solvable if and only if $\widetilde{W}(\boldsymbol{\beta}_\star,\boldsymbol{\beta})$ is PD for $\boldsymbol{\beta}$ close to $\boldsymbol{\beta}_\star$. Now, Theorem 3.2.12 guarantees that Algorithm 3 computes $\{\boldsymbol{u}_m\}_{m=1}^K$ such that $\widehat{W}(\boldsymbol{\beta}_\star) = \widetilde{W}(\boldsymbol{\beta}_\star,\boldsymbol{\beta}_\star)$ is PD, if $\boldsymbol{\beta}_\star$ is close enough to $\boldsymbol{\beta}_\circ$. Similar to the proof of Lemma 3.2.11, one can prove that $\widetilde{W}(\boldsymbol{\beta}_\star,\boldsymbol{\beta})$ is continuous in $\boldsymbol{\beta}$. Hence, if $\widetilde{W}(\boldsymbol{\beta}_\star,\boldsymbol{\beta}_\star)$ is PD, then the same holds for $\widetilde{W}(\boldsymbol{\beta}_\star,\boldsymbol{\beta})$, when $\boldsymbol{\beta}$ is close to $\boldsymbol{\beta}_\star$, which implies that (3.30) is locally uniquely solvable with $\boldsymbol{\beta}=\boldsymbol{\beta}_\star$.

[3]Notice that this notation is related to (3.27) in the sense that $\widetilde{W}(\boldsymbol{\beta},\boldsymbol{\beta}) = \widehat{W}(\boldsymbol{\beta})$.

Chapter 4

Reconstruction of matrices in bilinear systems

In this chapter, we extend the results from Chapter 3, in particular Section 3.2, to skew-symmetric bilinear systems. We consider (2.1) with a right-hand side $f(A, \boldsymbol{y}, u) = (A + uB)\boldsymbol{y}$, that is,

$$\dot{\boldsymbol{y}}(t) = (A_\star + u(t)B)\boldsymbol{y}(t),\ t \in (0,T], \quad \boldsymbol{y}(0) = \boldsymbol{y}^0, \tag{4.1}$$

where $B \in \mathfrak{so}(N)$ is a given skew-symmetric matrix for $N \in \mathbb{N}^+$, the initial state is $\boldsymbol{y}^0 \in \mathbb{R}^N \setminus \{0\}$, and $u \in U_{ad} \subset L^2(0,T;\mathbb{R})$ (with $M = 1$; see Section 1.3) is a control function. The matrix $A_\star \in \mathfrak{so}(N)$ is unknown and assumed to lie in the space spanned by a set of linearly independent matrices $\mathcal{B} = \{A_1, \dots, A_K\} \subset \mathbb{R}^{N\times N}$, $1 \le K \le N^2$, and we write $A_\star = \sum_{j=1}^K \beta_{\star,j} A_j =: A(\boldsymbol{\beta}_\star)$. Since the matrices $A_\star$ and B are skew-symmetric, system (4.1) is norm preserving, i.e. $\|\boldsymbol{y}(t)\|_2 = \|\boldsymbol{y}^0\|_2$ for all $t \in [0,T]$.[1]

To identify the true matrix $A_\star$, one can consider a set of control functions $\{u^m\}_{m=1}^K \subset U_{ad}$ and use it experimentally to obtain the data $\{\boldsymbol{\varphi}_\star^{data}(u^m)\}_{m=1}^K \subset \mathbb{R}^P$, as defined in (2.2). The unknown vector $\boldsymbol{\beta}_\star$ is then obtained by solving the online reconstruction problem (2.3), in which $\boldsymbol{\varphi}(A(\boldsymbol{\beta}), u^m) = C\boldsymbol{y}(A(\boldsymbol{\beta}), u^m; T)$ now solves (4.1), with $A_\star$ replaced by $A(\boldsymbol{\beta})$. We assume to have an estimate $\boldsymbol{\beta}_\circ$ of $\boldsymbol{\beta}_\star$. We can derive the linearized equation in $\boldsymbol{\beta}_\circ$:

$$\begin{cases} \dot{\delta\boldsymbol{y}}_\circ(t) = (A_\circ + u(t)B)\delta\boldsymbol{y}_\circ(t) + \sum_{j=1}^K \delta\beta_j A_j \boldsymbol{y}_\circ(t), \quad t \in (0,T], \quad \delta\boldsymbol{y}_\circ(0) = 0, \\ \dot{\boldsymbol{y}}_\circ(t) = (A_\circ + u(t)B)\boldsymbol{y}_\circ(t), \quad t \in (0,T], \quad \boldsymbol{y}_\circ(0) = \boldsymbol{y}^0, \end{cases} \tag{4.2}$$

where $A_\circ := A(\boldsymbol{\beta}_\circ)$. Denoting by $\delta\boldsymbol{y}_\circ(A(\delta\boldsymbol{\beta}), u; t)$ the solution of (4.2) at time $t \in [0,T]$, the GN matrix $\widehat{W}_\circ$ is defined as in (3.19), and LGR is detailed in Algorithm 3.

Let us recall the following definition and result from [25, Corollary 4.11].

Definition & Lemma 4.0.1 (Controllability of skew-sym. bilinear systems) *Consider a system of the form*

$$\dot{\boldsymbol{y}}(t) = (A_\circ + u(t)B)\boldsymbol{y}(t), \quad \boldsymbol{y}(0) = \boldsymbol{y}^0, \tag{4.3}$$

[1] To see this, we observe that $\frac{1}{2}\frac{d}{dt}\|\boldsymbol{y}(t)\|_2^2 = \langle \boldsymbol{y}(t), \dot{\boldsymbol{y}}(t)\rangle = \langle \boldsymbol{y}(t), (A_\star + u(t)B)\boldsymbol{y}(t)\rangle = 0$.

where $A_\circ, B \in \mathfrak{so}(N)$. System (4.3) is said to be controllable if for any final state $\boldsymbol{y}^f$ that lies on the sphere of radius $\|\boldsymbol{y}^0\|_2$ there exists a control $u(t)$ that transfers $\boldsymbol{y}^0$ to $\boldsymbol{y}^f$. Furthermore, if the Lie algebra $Lie\{A_\circ, B\} \subset \mathfrak{so}(N)$, generated by the matrices $A_\circ$ and B, has dimension $\frac{N(N-1)}{2}$, then there exists a constant $\tilde{t} \geq 0$ such that for any $T \geq \tilde{t}$ controllability of (4.3) holds.

Notice that the Lie algebra $Lie\{A_\circ, B\}$ contains exactly the matrices $A_\circ, B$ and their iterated commutators

$$[A_\circ, B], [[A_\circ, B], A_\circ], [[A_\circ, B], B], \dots$$

see, e.g., [25, Section 4.1] and references therein, where $[A_\circ, B] := A_\circ B - BA_\circ$ denotes the matrix commutator. Since $[A_\circ, B]^\top = -[A_\circ, B]$, we have $Lie\{A_\circ, B\} \subset \mathfrak{so}(N)$. Recalling that the space of skew-symmetric matrices $\mathfrak{so}(N)$ has dimension $\frac{N(N-1)}{2}$, we obtain $dim(Lie\{A_\circ, B\}) \leq \frac{N(N-1)}{2}$. For further details on Lie algebras, we refer the interested reader to [25, Appendix A.3] and references therein.

As in Section 3.2, we also need to make some assumptions on the observability of the linearized equation in (4.2). However, recalling the proof of Lemma 3.2.5, these assumptions are only required to prove the existence of a control function that guarantees a positive cost function value in the splitting step. If we assume this function to be constant, at least on a subinterval of $[0, T]$, then we get a system of the form

$$\dot{\delta\boldsymbol{y}}_\circ(t) = (A_\circ + cB)\delta\boldsymbol{y}_\circ(t) + A(\delta\boldsymbol{\beta})\boldsymbol{y}_\circ(t), \tag{4.4}$$

for a scalar $c \in \mathbb{R}$. In this case, system (4.4) is again a linear system, for which observability is defined in Definition 3.1.4. Hence, the observability matrix is $\mathcal{O}_N(C, A_\circ + cB)$.

Let us state our assumptions on controllability and observability of (4.3) and (4.4).

Assumption 4.0.2 *Let the matrices $A_\circ$, B and C be such that*

1. *The Lie algebra $Lie\{A_\circ, B\} \subset \mathfrak{so}(N)$ has dimension $\frac{N(N-1)}{2}$ and $T > 0$ is large enough, such that the controllability result from Lemma 4.0.1 holds.*
2. *There exists $c \in \mathbb{R}$ such that $\mathcal{O}_N(C, A_\circ + cB)$ has full rank.*

In addition, let $U_{ad} \subset L^2(0, T; \mathbb{R})$ be chosen such that the controllability result from Lemma 4.0.1 holds, and such that $u \equiv c$ is an interior point of U_{ad} (for the above $c \in \mathbb{R}$).

Remark 4.0.3 *The analysis presented in the following sections can be applied to the case where the matrix $A = A_\star$ is assumed to be known and $B = B(\boldsymbol{\beta}) := \sum_{j=1}^K \beta_j B_j$ is unknown and to be identified. The main differences in the case of the reconstruction of B is that the state equation is linearized around an initial guess $B_\circ$, leading to*

$$\begin{cases} \dot{\delta\boldsymbol{y}}_\circ(t) = (A + u(t)B_\circ)\delta\boldsymbol{y}_\circ(t) + \sum_{j=1}^K \delta\beta_j u(t) B_j \boldsymbol{y}_\circ(t), \quad t \in (0, T], \quad \delta\boldsymbol{y}_\circ(0) = 0, \\ \dot{\boldsymbol{y}}_\circ(t) = (A + u(t)B_\circ)\boldsymbol{y}_\circ(t), \quad t \in (0, T], \quad \boldsymbol{y}_\circ(0) = \boldsymbol{y}^0. \end{cases}$$

Assumption 4.0.2 would be the same, only with A instead of $A_\circ$ and $B_\circ$ instead of B. Notice that, in this case, we also cover Schrödinger-type systems of the form

$$i\dot{\boldsymbol{\psi}}(t) = (H + u(t)\mu_\star)\boldsymbol{\psi}(t),\ t \in (0, T], \quad \boldsymbol{\psi}(0) = \boldsymbol{\psi}^0,$$

as considered in [57], for Hermitian matrices $H, \mu_\star \in \mathbb{C}^{N\times N}$. This can be seen by writing $\boldsymbol{\psi} = \boldsymbol{\psi}_R + i\boldsymbol{\psi}_I$, $\boldsymbol{\psi}^0 = \boldsymbol{\psi}^0_R + i\boldsymbol{\psi}^0_I$, $H = H_R + iH_I$ and $\mu_\star = \mu_{\star,R} + i\mu_{\star,I}$, to get

$$\dot{\boldsymbol{y}}(t) = \left(\underbrace{\begin{bmatrix} H_I & H_R \\ -H_R & H_I \end{bmatrix}}_{=:A} + u(t) \underbrace{\begin{bmatrix} \mu_{\star,I} & \mu_{\star,R} \\ -\mu_{\star,R} & \mu_{\star,I} \end{bmatrix}}_{:=B_\star} \right) \boldsymbol{y}(t), \tag{4.5}$$

for $\boldsymbol{y}(t) := \begin{bmatrix} \boldsymbol{\psi}_R(t) & \boldsymbol{\psi}_I(t) \end{bmatrix}^\top$ and skew-symmetric matrices $A, B_\star \in \mathbb{R}^{N\times N}$ (compare also [25, Section 2.12.2]).

4.1 Analysis for skew-symmetric bilinear systems

We show in this section that Assumption 4.0.2 is a sufficient condition for the GN matrix $\widehat{W}_\circ$, defined as in (3.19), to be PD if the controls generated by the LGR Algorithm 3 are used. The idea of the analysis is similar to the one considered in Section 3.2, meaning that we first have to show the existence of a control that makes the cost function of (2.25) strictly positive.

Lemma 4.1.1 (GR initialization and splitting steps (bilinear systems)) *Let the matrices $A_\circ$, B and C satisfy Assumption 4.0.2. Let $\widetilde{A} \in \operatorname{span}(\mathcal{B})$ be an arbitrary matrix. If $T > 0$ is sufficiently large, then $\|C\delta\boldsymbol{y}_\circ(\widetilde{A}, \widetilde{u}; T)\|_2^2 > 0$ for any solution $\widetilde{u}$ to*

$$\max_{u\in U_{ad}} \|C\delta\boldsymbol{y}_\circ(\widetilde{A}, u; T)\|_2^2, \text{ s.t. } \dot{\delta\boldsymbol{y}}_\circ(t) = (A_\circ + u(t)B)\delta\boldsymbol{y}_\circ(t) + \widetilde{A}\boldsymbol{y}_\circ(t), \quad \delta\boldsymbol{y}_\circ(0) = 0,$$
$$\dot{\boldsymbol{y}}_\circ(t) = (A_\circ + u(t)B)\boldsymbol{y}_\circ(t), \quad \boldsymbol{y}_\circ(0) = \boldsymbol{y}^0.$$

Proof. It is sufficient to construct a $\widehat{u}_c \in U_{ad}$ such that $C\delta\boldsymbol{y}_\circ(\widetilde{A}, \widehat{u}_c; T) \neq 0$ for T sufficiently large. Let us define

$$\widehat{u}_c(s) := \begin{cases} \widehat{u}(s), & \text{for } 0 \le s \le \widehat{t}, \\ c, & \text{for } \widehat{t} < s \le T, \end{cases}$$

where $c \in \mathbb{R}$, $\widehat{u} \in U_{ad}$, $T > 0$ and $\widehat{t} \in (0, T)$ are to be chosen. Since $\widetilde{A} \neq 0$, there exists $\boldsymbol{v} \in \mathbb{R}^N \setminus \{0\}, \|\boldsymbol{v}\|_2 = \|\boldsymbol{y}^0\|_2$ such that $\widetilde{A}\boldsymbol{v} \neq 0$. By the first part of Assumption 4.0.2, we know that (4.3) is controllable on the sphere of radius $\|\boldsymbol{y}^0\|_2$, meaning that there exist $\widehat{t} > 0$ and $\widehat{u} \in U_{ad}$ such that $\boldsymbol{y}_\circ(\widetilde{A}, \widehat{u}; \widehat{t}) = \boldsymbol{v}$. Defining $A_c := A_\circ + cB$, we notice that $f_{\boldsymbol{v}}(t) := \widetilde{A}e^{tA_c}\boldsymbol{v}$ is analytic in t, and since $f_{\boldsymbol{v}}(0) = \widetilde{A}\boldsymbol{v} \neq 0$, it is not equal to zero everywhere and therefore has only isolated roots, see, e.g., [64, Thm. 10.18]. Recalling that exponential matrices are always invertible (see, e.g., [49, Thm. 2.6.38]), we obtain that there exists $t_1 > 0$ such that $e^{-t_1(A_c)}\widetilde{A}e^{(t_1-\widehat{t})A_c}\boldsymbol{v} \neq 0$. By defining $\boldsymbol{w} := \delta\boldsymbol{y}_\circ(\widetilde{A}, \widehat{u}; \widehat{t})$ and $\boldsymbol{g}(t) := \int_{\widehat{t}}^t e^{-s(A_c)}\widetilde{A}e^{(s-\widehat{t})A_c}\boldsymbol{v}ds + e^{-\widehat{t}A_c}\boldsymbol{w}$, we observe that $\frac{d\boldsymbol{g}(t_1)}{dt} = e^{-t_1(A_c)}\widetilde{A}e^{(t_1-\widehat{t})A_c}\boldsymbol{v} \neq 0$. Since $\frac{d\boldsymbol{g}(t)}{dt}$ is analytic in t, the same holds for $\boldsymbol{g}(t)$,[2] and since $\frac{d\boldsymbol{g}(t_1)}{dt} \neq 0$ we obtain that $\boldsymbol{g}(t)$ has only isolated roots. Notice that

$$e^{-tA_c}\delta\boldsymbol{y}(\widetilde{A}, \widehat{u}_c; t) = e^{-tA_c}\int_{\widehat{t}}^t e^{(t-s)(A_c)}\widetilde{A}e^{(s-\widehat{t})A_c}\boldsymbol{v}ds + e^{(t-\widehat{t})A_c}\boldsymbol{w} = \boldsymbol{g}(t),$$

[2] This follows directly from the fundamental theorem of calculus.

for $t > \widehat{t}$. Thus, it remains to show that there exists $T > \widehat{t}$ such that $Ce^{TA_c}\boldsymbol{g}(T) \neq 0$. Assumption 4.0.2 guarantees that there exists $c \in \mathbb{R}$ such that the observability matrix $\mathcal{O}_N(C, A_\circ + cB)$ has full rank. Hence, for any $\boldsymbol{q} \in \mathbb{R}^N \setminus \{0\}$ there exists a $t_{\boldsymbol{q}} > \widehat{t}$ such that $Ce^{t_{\boldsymbol{q}}A_c}\boldsymbol{q} \neq 0$. Since $t \mapsto Ce^{tA_c}\boldsymbol{q}$ is analytic in t, $Ce^{t_{\boldsymbol{q}}A_c}\boldsymbol{q} \neq 0$ implies that it has only isolated roots. Thus, for $t > \widehat{t}$, $t \mapsto Ce^{tA_c}\boldsymbol{g}(t)$ is the composition of two analytic functions which both have only isolated roots, and is therefore also analytic with isolated roots. Hence, there exists $T > \widehat{t}$ such that $C\delta\boldsymbol{y}(\widetilde{A}, \widehat{u}_c; T) = Ce^{TA_c}\boldsymbol{g}(T) \neq 0$. □

Now, we can prove our main result, whose proof is the same as the one of Theorem 3.2.6, in which Lemma 4.1.1 has to be used instead of Lemma 3.2.5.

Theorem 4.1.2 (positive definiteness of the GN matrix $\widehat{W}_\circ$ (bilinear systems)) *Let $\boldsymbol{\beta}_\circ \in \mathbb{R}^K$ be such that the matrices $A(\boldsymbol{\beta}_\circ), B \in \mathfrak{so}(N)$ and $C \in \mathbb{R}^{P\times N}$ satisfy Assumption 4.0.2. For $K \leq N^2$, let $\mathcal{B} = \{A_1, \dots, A_K\} \subset \mathfrak{so}(N)$ be a set of linearly independent matrices such that $A_\star \in \operatorname{span} \mathcal{B}$, and let $\{u^1, \dots, u^K\} \subset U_{ad}$ be controls generated by Algorithm 3. Then the GN matrix $\widehat{W}_\circ$, defined in* (3.19)*, is PD.*

4.2 Positive definiteness of the GN matrix

As in Section 3.2.5, we show that if the GN matrix in $\boldsymbol{\beta}_\circ$ is PD, then the same is true locally, for all iterates $\boldsymbol{\beta}_c$ of GN. We start by writing the matrix $\widehat{W}(\boldsymbol{\beta})$ as a function of $\boldsymbol{\beta}$:

$$[\widehat{W}(\boldsymbol{\beta})]_{i,j} := \sum_{m=1}^{K} \langle C\delta\boldsymbol{y}(\boldsymbol{\beta}, A_i, u^m; T), C\delta\boldsymbol{y}(\boldsymbol{\beta}, A_j, u^m; T)\rangle, \quad i, j \in \{1, \dots, K\}, \tag{4.6}$$

where $\delta\boldsymbol{y}(\boldsymbol{\beta}, \widehat{A}, u; T)$ denotes the solution at time T of

$$\begin{cases} \dot{\delta\boldsymbol{y}}(t) &= (A(\boldsymbol{\beta}) + u(t)B)\delta\boldsymbol{y}(t) + \widehat{A}\boldsymbol{y}(t), \quad \delta\boldsymbol{y}(0) = 0, \\ \dot{\boldsymbol{y}}(t) &= (A(\boldsymbol{\beta}) + u(t)B)\boldsymbol{y}(t), \quad \boldsymbol{y}(0) = \boldsymbol{y}^0. \end{cases} \tag{4.7}$$

Now, we want to prove the same positive definiteness result as in Lemma 3.2.11.

Lemma 4.2.1 (positive definiteness of $\widehat{W}_\circ$ (bilinear systems)) *Let $\widehat{W}_\circ$, defined in* (3.19)*, be PD and denote by $\sigma_K^\circ > 0$ the smallest singular value of $\widehat{W}_\circ$. Then, there exists $\delta := \delta(\sigma_K^\circ) > 0$ such that for any $\boldsymbol{\beta} \in \mathbb{R}^K$ with $\|\boldsymbol{\beta} - \boldsymbol{\beta}_\circ\|_2 < \delta$, the matrix $\widehat{W}(\boldsymbol{\beta})$, defined as in* (4.6)*, is also PD.*

Proof. Recalling the proof of Lemma 3.2.11, it is sufficient to show that the solution $\delta\boldsymbol{y}(\boldsymbol{\beta}, \widehat{A}, u; T)$ of (4.7) is continuous in $\boldsymbol{\beta}$. By [25, Proposition 3.26],[3] we obtain continuity of the map $\boldsymbol{\beta} \mapsto \boldsymbol{y}(A(\boldsymbol{\beta}), u; T)$ and analogously the continuity of $\boldsymbol{\beta} \mapsto \delta\boldsymbol{y}(\boldsymbol{\beta}, \widehat{A}, u; T)$. □

Using the result from Lemma 4.2.1, we can directly prove our main result.

Theorem 4.2.2 (convergence of GN (bilinear systems)) *Let $\boldsymbol{\beta}_\circ \in \mathbb{R}^K$ be such that the matrices $A(\boldsymbol{\beta}_\circ)$, B and C satisfy Assumption 4.0.2, and let $\{u^m\}_{m=1}^K \subset U_{ad}$ be generated by Algorithm 3. Denote by $\widehat{\sigma}_K$ the smallest singular value of $\widehat{W}_\circ$, defined in* (3.19)*. Then*

[3] This result is a special case of the implicit function theorem; see, e.g., [25, Thm. 3.4].

there exists $\delta = \delta(\widehat{\sigma}_K) > 0$ *such that, if* $\boldsymbol{\beta}_\star \in \mathbb{R}^K$ *satisfies* $\|\boldsymbol{\beta}_\star - \boldsymbol{\beta}_\circ\|_2 \leq \delta$*, then GN for the solution of* (2.3)*, initialized with* $\boldsymbol{\beta}_\circ$*, converges to* $\boldsymbol{\beta}_\star$*.*

Proof. Theorem 4.1.2 implies that $\widehat{W}_\circ$ is PD, namely $\widehat{\sigma}_K > 0$. As in the proof of Lemma 4.2.1, one can show that $\boldsymbol{\beta} \mapsto R_m(\boldsymbol{\beta})$, defined in (2.14), are Lipschitz continuously differentiable in $\boldsymbol{\beta}$ for all $m \in \{1, \dots, K\}$. Thus, the result follows by Lemma 4.2.1. □

4.3 Local uniqueness of solutions

Let us study the local properties of the online reconstruction problem (2.3) around $\boldsymbol{\beta}_\star$. We use the same approach as in the linear case, and start by rewriting problem (2.3) in a matrix-vector form.

Lemma 4.3.1 (reconstruction problem in matrix form (bilinear systems)) *Problem* (2.3) *the bilinear system* (4.1) *is equivalent to*

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^K} \frac{1}{2} \langle \boldsymbol{\beta} - \boldsymbol{\beta}_\star, \widetilde{W}(\boldsymbol{\beta}_\star, \boldsymbol{\beta})(\boldsymbol{\beta} - \boldsymbol{\beta}_\star) \rangle, \tag{4.8}$$

where we define $\widetilde{W}(\boldsymbol{\beta}_\star, \boldsymbol{\beta}) := \sum_{m=1}^K W(\boldsymbol{\beta}_\star, \boldsymbol{\beta}, u^m) \in \mathbb{R}^{K \times K}$*,*

$$[W(\boldsymbol{\beta}_\star, \boldsymbol{\beta}, u^m)]_{i,j} := \langle C\delta\boldsymbol{y}_m(\boldsymbol{\beta}_\star, \boldsymbol{\beta}, A_j; T), C\delta\boldsymbol{y}_m(\boldsymbol{\beta}_\star, \boldsymbol{\beta}, A_i; T) \rangle, \tag{4.9}$$

for $i, j \in \{1, \dots, K\}$*, and* $\delta\boldsymbol{y}_m(\boldsymbol{\beta}_\star, \boldsymbol{\beta}, A; t)$ *solves*

$$\begin{cases} \dot{\delta\boldsymbol{y}}(t) &= (A(\boldsymbol{\beta}_\star) + u^m(t)B)\delta\boldsymbol{y}(t) + A\boldsymbol{y}(t), \quad \delta\boldsymbol{y}(0) = 0, \\ \dot{\boldsymbol{y}}(t) &= (A(\boldsymbol{\beta}) + u^m(t)B)\boldsymbol{y}(t), \quad \boldsymbol{y}(0) = \boldsymbol{y}^0. \end{cases} \tag{4.10}$$

Proof. Let $J(\boldsymbol{\beta}) = \frac{1}{2} \sum_{m=1}^K \|C\boldsymbol{y}(A(\boldsymbol{\beta}), u^m; T) - C\boldsymbol{y}(A_\star, u^m; T)\|_2^2$. For $t \in [0, T]$ and $m \in \{1, \dots, K\}$ define

$$\Delta\boldsymbol{y}_m(t) := \boldsymbol{y}(A(\boldsymbol{\beta}), u^m; t) - \boldsymbol{y}(A_\star, u^m; t). \tag{4.11}$$

Then we have

$$\begin{aligned} \dot{\Delta\boldsymbol{y}}_m(t) &= (A(\boldsymbol{\beta}) + u^m(t)B)\boldsymbol{y}(A(\boldsymbol{\beta}), u^m; t) - (A(\boldsymbol{\beta}_\star) + u^m(t)B)\boldsymbol{y}(A_\star, u^m; t) \\ &= (A(\boldsymbol{\beta}_\star) + u^m(t)B)\Delta\boldsymbol{y}_m(t) + A(\boldsymbol{\beta} - \boldsymbol{\beta}_\star)\boldsymbol{y}(A(\boldsymbol{\beta}), u^m; t) \\ &= \dot{\delta\boldsymbol{y}}(\boldsymbol{\beta}_\star, \boldsymbol{\beta}, A(\boldsymbol{\beta} - \boldsymbol{\beta}_\star), u^m; t). \end{aligned}$$

From (4.10) we get that $\dot{\delta\boldsymbol{y}}_m(\boldsymbol{\beta}_\star, \boldsymbol{\beta}, \widehat{A}; t)$ and $\delta\boldsymbol{y}_m(\boldsymbol{\beta}_\star, \boldsymbol{\beta}, \widehat{A}; t)$ are linear in $\widehat{A}$. Thus

$$\begin{aligned} J(\boldsymbol{\beta}) &= \frac{1}{2} \sum_{m=1}^K \|C\Delta\boldsymbol{y}_m(T)\|_2^2 = \frac{1}{2} \sum_{m=1}^K \|C\delta\boldsymbol{y}_m(\boldsymbol{\beta}_\star, \boldsymbol{\beta}, A(\boldsymbol{\beta} - \boldsymbol{\beta}_\star); T)\|_2^2 \\ &= \frac{1}{2} \sum_{m=1}^K \sum_{i=1}^K \sum_{j=1}^K (\beta_i - \beta_{\star,i})(\beta_j - \beta_{\star,j}) \langle C\delta\boldsymbol{y}_m(\boldsymbol{\beta}_\star, \boldsymbol{\beta}, A_i; T), C\delta\boldsymbol{y}_m(\boldsymbol{\beta}_\star, \boldsymbol{\beta}, A_j; T) \rangle \\ &\overset{(4.9)}{=} \frac{1}{2} \langle \boldsymbol{\beta} - \boldsymbol{\beta}_\star, \widetilde{W}(\boldsymbol{\beta}_\star, \boldsymbol{\beta})(\boldsymbol{\beta} - \boldsymbol{\beta}_\star) \rangle, \end{aligned}$$

which yields the equivalence of the problems (2.3) and (4.8). □

Notice that the notations in (4.6) and Lemma 4.3.1 are related in the sense that $\widehat{W}(\boldsymbol{\beta}) = \widetilde{W}(\boldsymbol{\beta}, \boldsymbol{\beta})$. Now, proceeding as in Section 3.2.6 and defining the set of all global solutions $\mathcal{S}_{global} := \left\{\boldsymbol{\beta} \in \mathbb{R}^K \mid (\boldsymbol{\beta} - \boldsymbol{\beta}_\star) \in \ker \widetilde{W}(\boldsymbol{\beta}_\star, \boldsymbol{\beta})\right\}$, we obtain the same local uniqueness of the solution $\boldsymbol{\beta}_\star$ to (2.3), meaning that if $\widehat{W}(\boldsymbol{\beta}_\star) = \widetilde{W}(\boldsymbol{\beta}_\star, \boldsymbol{\beta}_\star)$ is PD, the same holds for $\widetilde{W}(\boldsymbol{\beta}_\star, \boldsymbol{\beta})$ when $\boldsymbol{\beta}$ is close to $\boldsymbol{\beta}_\star$.

Chapter 5

Analysis for general dynamical systems

The LGR algorithm analyzed in the previous chapters only considers the linearized system. Thus it does not have access to the full (nonlinear) dynamics and can only capture the local characteristics of the considered system. Moreover, as we will show in Chapter 6, standard GR can outperform LGR when $\boldsymbol{\beta}_\circ$ is far from the solution. However, the analysis of LGR allows us to better understand the local behavior of GR and prove that locally it is capable to construct control functions that guarantee convergence of GN. This analysis is carried out in Section 5.1. This is the first analysis of GR algorithms for nonlinear problems. Afterwards, we discuss in Section 5.1.2 global convergence of GN for a specific bilinear system.

5.1 A local analysis for nonlinear GR algorithms

This section is concerned with general nonlinear systems of the form

$$\begin{aligned}\dot{\boldsymbol{y}}(t) &= f(A(\boldsymbol{\beta}_\circ) + A(\delta\boldsymbol{\beta}_\star), \boldsymbol{y}(t), \boldsymbol{u}(t)),\\ \boldsymbol{\varphi}_\star^{data}(\boldsymbol{u}) &= C\boldsymbol{y}(A(\boldsymbol{\beta}_\circ) + A(\delta\boldsymbol{\beta}_\star), \boldsymbol{u}; T),\end{aligned}$$

with the goal of reconstructing $A(\delta\boldsymbol{\beta}_\star) = A_\star - A(\boldsymbol{\beta}_\circ)$. Here, the shift of $A_\star$ is considered to perform a local analysis near $A(\boldsymbol{\beta}_\circ)$. The goal is to prove convergence of GN for the controls generated by the GR Algorithm 4 using a local analogy to the LGR Algorithm 3. Notice that there are a few differences between Algorithms 4 and 3. To derive a local analogy between them, all operators from the set $\mathcal{B} = \{A_1, \dots, A_K\}$ are shifted by $A(\boldsymbol{\beta}_\circ)$. Additionally, the fitting step problem (5.2) only minimizes over a compact set $\mathcal{K}_k \subset \mathbb{R}^k$. However, this is not restrictive since the set $\mathcal{K}_k$ can be chosen arbitrarily large.

We recall that, in order to obtain our main results for Algorithm 4, it is sufficient to prove two points. First, that the fitting step (5.2) identifies the kernel of the submatrix $[\widehat{W}_\circ^{(k)}]_{[1:k+1,1:k+1]}$. Second, that for the initialization and each splitting step in Algorithm 4

Algorithm 4 Nonlinear Greedy Reconstruction Algorithm

Require: A set of linearly independent operators $\mathcal{B} = \{A_1, \dots, A_K\}$, an (initial) operator $A(\boldsymbol{\beta}_\circ) \in \operatorname{span} \mathcal{B}$ and a family of compact sets $\mathcal{K}_j \subset \mathbb{R}^j$, $j = 1, \dots, K-1$.

1: Compute the control $\boldsymbol{u}^1$ by solving

$$\max_{\boldsymbol{u} \in U_{ad}} \|C\boldsymbol{y}(A(\boldsymbol{\beta}_\circ), \boldsymbol{u}; T) - C\boldsymbol{y}(A(\boldsymbol{\beta}_\circ) + A_1, \boldsymbol{u}; T)\|_2^2. \tag{5.1}$$

2: **while** $k \leq K-1$ **do**

3: Fitting step: Find $\boldsymbol{\beta} = (\beta_j^k)_{j=1,\dots,k}$ that solves

$$\min_{\boldsymbol{\beta} \in \mathcal{K}_k} \sum_{m=1}^{k} \|C\boldsymbol{y}(A(\boldsymbol{\beta}_\circ) + A^{(k)}(\boldsymbol{\beta}), \boldsymbol{u}^m; T) - C\boldsymbol{y}(A(\boldsymbol{\beta}_\circ) + A_{k+1}, \boldsymbol{u}^m; T)\|_2^2, \tag{5.2}$$

where $A^{(k)}(\boldsymbol{\beta}) = \sum_{j=1}^{k} \beta_j A_j$.

4: Splitting step: Find $\boldsymbol{u}^{k+1}$ that solves

$$\max_{\boldsymbol{u} \in U_{ad}} \|C\boldsymbol{y}(A(\boldsymbol{\beta}_\circ) + A^{(k)}(\boldsymbol{\beta}^k), \boldsymbol{u}; T) - C\boldsymbol{y}(A(\boldsymbol{\beta}_\circ) + A_{k+1}, \boldsymbol{u}; T)\|_2^2. \tag{5.3}$$

5: Update $k \leftarrow k+1$.

6: **end while**

there exists at least one control for which the corresponding cost function is strictly positive (making the submatrix $[\widehat{W}_\circ^{(k+1)}]_{[1:k+1,1:k+1]}$ PD).

To prove the fitting step result, we need some continuity properties of the argmin operator. For this purpose, we introduce the following definition of hemi-continuous set-valued correspondences (see, e.g., [21, Chapter VI,§1]).

Definition 5.1.1 (hemi-continuity) *Let $X \subset \mathbb{R}$ be an open set. A correspondence $c : X \rightrightarrows \mathbb{R}^k$ is called upper hemi-continuous (u.h.c.) if for each $x_0 \in X$ and each open set $G \subset \mathbb{R}^k$ with $c(x_0) \subset G$ there is a neighborhood $U(x_0) \subset X$ such that $x \in U(x_0) \Rightarrow c(x) \subset G$. The mapping is called lower hemi-continuous (l.h.c.) if for each $x_0 \in X$ and each open set $G \subset \mathbb{R}^k$ with $c(x_0) \cap G \neq \emptyset$ there is a neighborhood $U(x_0) \subset X$ such that $x \in U(x_0) \Rightarrow c(x) \cap G \neq \emptyset$. Furthermore, $c : X \rightrightarrows \mathbb{R}^k$ is hemi-continuous if it is u.h.c. and l.h.c.*

Using Definition 5.1.1, we can recall the Berge maximum theorem [7, Thm. 17.31].

Lemma 5.1.2 (Berge maximum theorem) *Let $X \subset \mathbb{R}$ be an open interval. Let $J : \mathbb{R}^k \times X \to \mathbb{R}$ be a continuous function and $\phi : X \rightrightarrows \mathbb{R}^k$ be a hemi-continuous, set-valued correspondence such that $\phi(x)$ is nonempty and compact for any $x \in X$. Then the correspondence $c : X \rightrightarrows \mathbb{R}^k$ defined by $c(x) := \arg\min_{z \in \phi(x)} J(z; x)$ is u.h.c.*

We will also need the following technical lemma.

Lemma 5.1.3 (limit of set-valued correspondance) *Let $X \subset \mathbb{R}$ be an open interval with $0 \in X$, and $c : X \rightrightarrows \mathbb{R}^k$ be a u.h.c. correspondence. If $c(0) = \{0\}$, then $\lim_{k\to\infty} c(x_k) = \{0\}$ for any sequence $(x_k)_{k=1}^{\infty}$ such that $\lim_{k\to\infty} x_k = 0$.*

Proof. Consider an arbitrary sequence $(x_k)_{k=1}^{\infty}$ with $\lim_{k\to\infty} x_k = 0$, and let $c(0) = \{0\}$.

It is sufficient to show that for any $u > 0$ there exists $n_u \in \mathbb{N}$ such that for all $k \geq n_u$ we have $c(x_k) \subset \mathcal{B}_u^k(0)$. Let $u > 0$ and define $G_u := \mathcal{B}_u^k(0)$. Since $c(0) = \{0\}$ and c is u.h.c., there exists a neighborhood $U_u(0) \subset \mathbb{R}$ such that $c(x) \subset G_u$ for any $x \in U_u(0)$. Since $U_u(0)$ is an open neighborhood of 0, there exists $\xi_u > 0$ such that $(-\xi_u, \xi_u) \subset U_u(0)$. Since $\lim_{k\to\infty} x_k = 0$, there exists n_u such that for all $k \geq n_u$ we have $x_k \in (-\xi_u, \xi_u)$ and hence $c(x_k) \subset \mathcal{B}_u^k(0)$. □

To use Lemmas 5.1.2 and 5.1.3, we make the following assumptions.

Assumption 5.1.4 *Let $k \in \{1, \ldots, K-1\}$ and define*

$$J_k(\boldsymbol{\beta}; A_{k+1}) := \sum_{m=1}^{k} \|C\boldsymbol{y}(A(\boldsymbol{\beta}_\circ) + A^{(k)}(\boldsymbol{\beta}), \boldsymbol{u}^m; T) - C\boldsymbol{y}(A(\boldsymbol{\beta}_\circ) + A_{k+1}, \boldsymbol{u}^m; T)\|_2^2.$$

- *If $\|A_{k+1}\|_2$ is small enough, then there exists a $\boldsymbol{\beta}^k = \boldsymbol{\beta}^k(A_{k+1})$ that solves (5.2) with $J_k(\boldsymbol{\beta}^k; A_{k+1}) = 0$.*
- *There exists $\nu > 0$ such that $\mathcal{B}_\nu^k(0) \subset \mathcal{K}_k$ and $\arg\min_{\boldsymbol{\beta} \in \overline{\mathcal{B}_\nu^k(0)}} J_k(\boldsymbol{\beta}; 0) = \{0\}$.*

The first point in Assumption 5.1.4 guarantees that locally near $A(\boldsymbol{\beta}_\circ)$, for $\|A_{k+1}\|_2$ small enough, one can solve (5.2) making the cost function zero, meaning that one can find a linear combination of the first k elements for which the final state cannot be distinguished from the $k+1$-th element by any of the k computed controls. On the other hand, if the minimum function value is strictly positive, then there already exists a control in the set $\{\boldsymbol{u}_m\}_{m=1}^k$ that discriminates (splits) these two states.

The second point in Assumption 5.1.4 ensures that $\{0\} = \arg\min\{J_k(\boldsymbol{\beta}, 0) \mid \boldsymbol{\beta} \in \overline{\mathcal{B}_\nu^k(0)}\}$. If this was not true, then for any $\nu > 0$, the ball $\mathcal{B}_\nu^k(0)$ would contain infinitely many $\boldsymbol{\beta} \in \mathbb{R}^k \setminus \{0\}$ satisfying $J_k(\boldsymbol{\beta}, 0) = 0$. Hence, for an infinite number of linear combinations in the set $\{A_1, \ldots, A_k\}$, the corresponding states could not be distinguished by any of the previously selected controls. This implies that one of the previous splitting steps was not successful, which contradicts what we assume to reach iteration k.

Now, we can show that the local nonlinear fitting step problem (5.2) is able to identify the kernel of the submatrix $[\widehat{W}_\circ^{(k)}]_{[1:k+1,1:k+1]}$, if it is non-trivial.

Theorem 5.1.5 (nonlinear GR fitting step problems) *Let $k \in \{1, \ldots, K\}$ and let $\boldsymbol{\beta}^k$ be a solution to (5.2). If $\|A_{k+1}\|_2$ is sufficiently small and Assumption 5.1.4 holds, then $\boldsymbol{\beta}^k$ solves (2.24) with $\sum_{m=1}^k \|C\delta\boldsymbol{y}_\circ(A^{(k)}(\boldsymbol{\beta}^k), \boldsymbol{u}^m; T) - C\delta\boldsymbol{y}_\circ(A_{k+1}, \boldsymbol{u}^m; T)\|_2^2 = 0$.*

Proof. Let $A_{k+1} = \delta_k \widetilde{A}_{k+1}$, where $\delta_k = \|A_{k+1}\|_2$ and $\|\widetilde{A}_{k+1}\|_2 = 1$. Now, define $\widehat{J}_k(\boldsymbol{\beta}, \delta_k) := J_k(\boldsymbol{\beta}, \delta_k \widetilde{A}_{k+1})$. The first point of Assumption 5.1.4 implies that there is a $\widehat{\delta}_k > 0$ such that $\widehat{J}_k(\boldsymbol{\beta}, \delta_k) = 0$ for all $|\delta_k| < \widehat{\delta}_k$. Thus, Lemma 5.1.2 guarantees that $c_k : (-\widehat{\delta}_k, \widehat{\delta}_k) \rightrightarrows \mathbb{R}^k, c_k(\delta_k) = \arg\min_{\boldsymbol{\beta} \in \mathcal{K}_k} \widehat{J}_k(\boldsymbol{\beta}; \delta_k)$ is u.h.c.[1] According to the second point of Assumption 5.1.4, $c_k(0) = 0$ is an isolated solution of (5.2). Hence, the upper hemi-continuity of c_k guarantees that for $\delta_k \to 0$ we have $\boldsymbol{\beta}^k \to 0$ for any corresponding solution $\boldsymbol{\beta}^k = \boldsymbol{\beta}^k(\delta_k)$

[1]Note that, in this setting, the correspondence $\phi : (-\widehat{\delta}_k, \widehat{\delta}_k) \rightrightarrows \mathbb{R}^k$ mentioned in Lemma 5.1.2 is defined as $\phi(x) = \mathcal{K}_k$ for any $x \in (-\widehat{\delta}_k, \widehat{\delta}_k)$ with $\mathcal{K}_k$ compact, and is therefore hemi-continuous.

of (5.2). Now, let $m \in \{1, \dots, k\}$. If $\widehat{J}_k(\boldsymbol{\beta}^k; \delta_k) = 0$, then

$$C\boldsymbol{y}(A(\boldsymbol{\beta}_\circ) + A^{(k)}(\boldsymbol{\beta}^k), \boldsymbol{u}^m; T) - C\boldsymbol{y}(A(\boldsymbol{\beta}_\circ) + \delta_k \widetilde{A}_{k+1}, \boldsymbol{u}^m; T) = 0. \tag{5.4}$$

We define $g(\boldsymbol{\beta}) := C\boldsymbol{y}(A(\boldsymbol{\beta}), \boldsymbol{u}^m; T)$. Since $f(A, \boldsymbol{y}, \boldsymbol{u})$ in (2.1) is assumed to be differentiable with respect to A and $\boldsymbol{y}$, we obtain that the map $A \mapsto \boldsymbol{y}(A, \boldsymbol{u}; T)$ is differentiable with respect to A by the implicit function theorem (see, e.g., [30, Thm. 17.13-1]). Hence, $C\boldsymbol{y}(A(\boldsymbol{\beta}), \boldsymbol{u}; T)$ is also differentiable with respect to $\boldsymbol{\beta}$. By Taylor's theorem, we get $g(\boldsymbol{\beta}_\circ + \boldsymbol{v}) = g(\boldsymbol{\beta}_\circ) + g'(\boldsymbol{\beta}_\circ)(\boldsymbol{v}) + O(\|\boldsymbol{v}\|_2^2)$ for $\boldsymbol{v} \in \mathbb{R}^k$. Defining $\widehat{\boldsymbol{\beta}^k}$ and $\widehat{\boldsymbol{\delta}}_k$ as $\widehat{\boldsymbol{\beta}^k} := [\boldsymbol{\beta}^k, 0, \cdots, 0]^\top \in \mathbb{R}^k$ and $\widehat{\boldsymbol{\delta}}_k := [0, \cdots, 0, \delta_k]^\top \in \mathbb{R}^k$, we can rewrite (5.4) as

$$0 = g(\boldsymbol{\beta}_\circ + \widehat{\boldsymbol{\beta}^k}) - g(\boldsymbol{\beta}_\circ + \widehat{\boldsymbol{\delta}}_{k+1}) = g'(\boldsymbol{\beta}_\circ)(\widehat{\boldsymbol{\beta}^k}) - g'(\boldsymbol{\beta}_\circ)(\widehat{\boldsymbol{\delta}}_{k+1}) + O(\|\widehat{\boldsymbol{\beta}^k}\|_2^2) + O(|\delta_k|^2).$$

Since $g'(\boldsymbol{\beta}_\circ)(\widehat{\boldsymbol{\beta}^k}) = C\delta\boldsymbol{y}_\circ(A^{(k)}(\boldsymbol{\beta}^k), \boldsymbol{u}^m; T)$ and $g'(\boldsymbol{\beta}_\circ)(\widehat{\boldsymbol{\delta}}_{k+1}) = C\delta\boldsymbol{y}_\circ(\delta_k \widetilde{A}_{k+1}, \boldsymbol{u}^m; T)$,

$$0 = C\delta\boldsymbol{y}_\circ(A^{(k)}(\boldsymbol{\beta}^k), \boldsymbol{u}^m; T) - C\delta\boldsymbol{y}_\circ(\delta_k \widetilde{A}_{k+1}, \boldsymbol{u}^m; T) + O(\|\widehat{\boldsymbol{\beta}^k}\|_2^2) + O(|\delta_k|^2). \tag{5.5}$$

Since $\boldsymbol{\beta}^k = \boldsymbol{\beta}^k(\delta_k) \to 0$ for $\delta_k \to 0$, we know that all four terms vanish for $\delta_k \to 0$. However, $O(|\delta_k|^2)$ converges faster than $C\delta\boldsymbol{y}_\circ(\delta_k \widetilde{A}_{k+1}, \boldsymbol{u}^m; T)$ and $O(\|\widehat{\boldsymbol{\beta}^k}\|_2^2)$ faster than $C\delta\boldsymbol{y}_\circ(A^{(k)}(\boldsymbol{\beta}^k), \boldsymbol{u}^m; T)$. Hence, (5.5) can only be true for $\delta_k \to 0$ if $C\delta\boldsymbol{y}_\circ(A^{(k)}(\boldsymbol{\beta}^k), \boldsymbol{u}^m; T) - C\delta\boldsymbol{y}_\circ(\delta_k \widetilde{A}_{k+1}, \boldsymbol{u}^m; T) = 0$ for δ_k small enough, which is equivalent to $C\delta\boldsymbol{y}_\circ(A^{(k)}(\boldsymbol{\beta}^k), \boldsymbol{u}^m; T) - C\delta\boldsymbol{y}_\circ(A_{k+1}, \boldsymbol{u}^m; T) = 0$ for $\|A_{k+1}\|_2$ sufficiently small. □

Let us now comment about the assumption on $\|A_{k+1}\|_2$. The goal of this section is to prove a local result around the approximation $A(\boldsymbol{\beta}_\circ)$. Since the map $\boldsymbol{\beta} \mapsto A(\boldsymbol{\beta})$ is linear, the terms $A(\boldsymbol{\beta}_\circ) + A^{(k)}(\boldsymbol{\beta}^k)$ and $A(\boldsymbol{\beta}_\circ) + A_{k+1}$ are perturbations of $A(\boldsymbol{\beta}_\circ)$. These are clearly used in the proof of Theorem 5.1.5. In order to remain close to $A(\boldsymbol{\beta}_\circ)$, we require that the basis elements A_{k+1} are sufficiently small. However, this is not a restrictive assumption since it can be interpreted as a simple rescaling of the operators $A_1, \dots, A_K$, which are only required to be linearly independent. Thus, the sufficiently small assumption of $\|A_{k+1}\|_2$ has to be understood in the sense that one has to perturb $A(\boldsymbol{\beta}_\circ)$ by stepping in direction of the (linearly independent) element A_{k+1}. However, the length of the step must be sufficiently small to remain in a neighborhood of $A(\boldsymbol{\beta}_\circ)$ (related to the Taylor expansion used in the proof of Theorem 5.1.5).

Regarding the initialization and splitting-step result, we make now the assumption that there always exists a control that makes the corresponding cost function value strictly positive, and discuss specific cases where this assumption holds.

Assumption 5.1.6 *Let $k \in \{1, \dots, K-1\}$ and $\boldsymbol{\beta}^k \in \mathbb{R}^k$ be the solution of (5.2). There exists a solution $\boldsymbol{u}^{k+1} \in U_{ad}$ to (5.3) that simultaneously satisfies*

$$\|C\boldsymbol{y}(A(\boldsymbol{\beta}_\circ) + A^{(k)}(\boldsymbol{\beta}^k), \boldsymbol{u}^{k+1}; T) - C\boldsymbol{y}(A(\boldsymbol{\beta}_\circ) + A_{k+1}, \boldsymbol{u}^{k+1}; T)\|_2^2 > 0, \text{ and} \tag{5.6}$$

$$\|C\delta\boldsymbol{y}_\circ(A^{(k)}(\boldsymbol{\beta}^k), \boldsymbol{u}^{k+1}; T) - C\delta\boldsymbol{y}_\circ(A_{k+1}, \boldsymbol{u}^{k+1}; T)\|_2^2 > 0. \tag{5.7}$$

Let (5.6)-(5.7) also hold for a solution $\boldsymbol{u}^1 \in U_{ad}$ to (5.1) with $k = 0$ and $\boldsymbol{\beta}^0 = 0$.

In Theorem 5.1.10, we will investigate Assumption 5.1.6 for the two settings considered in Section 3.2 and Chapter 4. Now, we state a result relating Algorithms 3 and 4.

Theorem 5.1.7 (positive definiteness of the GN matrix $\widehat{W}_\circ$ (general systems)) *Consider the general setting of* (2.1) *with* $\{A_1, \dots, A_K\}$ *linearly independent such that* $\|A_k\|_2$ *is small enough for all* $k \in \{1, \dots, K\}$. *Let* $\{\boldsymbol{u}^m\}_{m=1}^K \subset U_{ad}$ *be generated by Algorithm 4 such that Assumption 5.1.4 holds for all* $k \in \{1, \dots, K-1\}$ *and* $\boldsymbol{u}^m$ *satisfies Assumption 5.1.6 for all* $m \in \{1, \dots, K\}$. *Then* $\widehat{W}_\circ$ *is PD.*

The proof of Theorem 5.1.7 is similar to that of Theorem 3.2.6 and is omitted for brevity.

In the Sections 5.1.1 and 5.1.2 we discuss Assumption 5.1.6 for linear and bilinear systems and for a general class of nonliner systems, respectively.

5.1.1 Linear and bilinear control systems

In this section, we discuss Assumption 5.1.6 in the settings of Section 3.2 and Chapter 4. First, we require the following result from [71, p. 1079].

Lemma 5.1.8 (on analytic functions in Banach spaces) *Let* X, Y *denote real Banach spaces and* $\mathcal{B}_r(x) \subset X$ *the open ball with center* $x \in X$ *and radius* $r > 0$. *For an open set* $D \subset X$, *let the functions* $f, g : D \to Y$ *be analytic. If there exist* $x_f, x_g \in D$ *such that* $f(x_f) \neq 0$ *and* $g(x_g) \neq 0$, *then for any* $x \in D$ *and any* $r > 0$ *there exists a* $\widetilde{x} \in \mathcal{B}_r(x) \subset D$ *such that* $f(\widetilde{x}) \neq 0$ *and* $g(\widetilde{x}) \neq 0$.

We also require the following result about the analycity of control-to-state maps, which follows directly from the implicit function theorem (see, e.g., [71, p. 1081]).

Lemma 5.1.9 (analycity of control-to-state maps) *Consider system* (2.1) *and define the map* $c : U \times Y \to Z$ *as* $c(\boldsymbol{u}, \boldsymbol{y}) := [\dot{\boldsymbol{y}} - f(A, \boldsymbol{y}, \boldsymbol{u}), \boldsymbol{y}(0) - \boldsymbol{y}^0]$, *where* U *is the Hilbert space of control functions,* Y *is the (Banach) space, where solutions to* (2.1) *lie, and* Z *is a Banach space. If* c *is analytic in* $\boldsymbol{u}$ *and* $\boldsymbol{y}$, (2.1) *has a unique solution* $\boldsymbol{y} = \boldsymbol{y}(\boldsymbol{u}) \in Y$ *such that* $c(\boldsymbol{y}(\boldsymbol{u}), \boldsymbol{u}) = 0$ *for each* $\boldsymbol{u} \in U_{ad} \subset U$ *and the linearized state equation* $\dot{\delta \boldsymbol{y}} = \delta_{\boldsymbol{y}} f(A, \boldsymbol{y}(\boldsymbol{u}), \boldsymbol{u})(\delta \boldsymbol{y}) - \varphi$ *with* $\delta \boldsymbol{y}(0) = \varphi^0$ *is uniquely solvable for any* $[\varphi, \varphi^0] \in Z$, *then the control-to-state map* $L : U_{ad} \to Y, \boldsymbol{u} \mapsto \boldsymbol{y}(\boldsymbol{u})$ *is analytic. If the solution space* Y *is such that the evaluation map* $S_T : Y \to \mathbb{R}^N, \boldsymbol{y} \mapsto \boldsymbol{y}(T)$ *is linear and continuous, then also the map* $S : U_{ad} \to \mathbb{R}^N, \boldsymbol{u} \mapsto (\boldsymbol{y}(\boldsymbol{u}))(T)$ *is analytic.*

Proof. First, we prove that the control-to-state map $L : U_{ad} \to Y, \boldsymbol{u} \mapsto \boldsymbol{y}(\boldsymbol{u})$ is analytic. This follows directly from the implicit function theorem [71, p. 1081] if we can show that the map $D_{\boldsymbol{y}} c(\boldsymbol{u}, \boldsymbol{y})$ is an isomorphism of Y on Z for any pair $(\widetilde{\boldsymbol{u}}, \widetilde{\boldsymbol{y}}) \subset U \times Y$ such that $\widetilde{\boldsymbol{y}}$ is the unique solution to (2.1) for $\widetilde{\boldsymbol{u}}$, i.e. $c(\widetilde{\boldsymbol{u}}, \widetilde{\boldsymbol{y}}) = 0$. Since $D_{\boldsymbol{y}} c(\widetilde{\boldsymbol{u}}, \widetilde{\boldsymbol{y}})(\delta \boldsymbol{y}) = \varphi$, which is equivalent to $\dot{\delta \boldsymbol{y}} = \delta_{\boldsymbol{y}} f(A, \widetilde{\boldsymbol{y}}, \widetilde{\boldsymbol{u}})(\delta \boldsymbol{y}) - \varphi$ with $\delta \boldsymbol{y}(0) = \varphi^0$, admits a unique solution $\delta \boldsymbol{y} \in Y$ for any $[\varphi, \varphi^0] \in Z$, $D_{\boldsymbol{y}} c(\widetilde{\boldsymbol{u}}, \widetilde{\boldsymbol{y}})$ is bijective and hence an isomorphism of Y on Z. It remains to show that also the map $S : U_{ad} \to \mathbb{R}^N, \boldsymbol{u} \mapsto (\boldsymbol{y}(\boldsymbol{u}))(T)$ is analytic. Consider an arbitrary $\boldsymbol{u}_0 \in U_{ad}$. Since the control-to-state map L is analytic, there exist (by definition, see, e.g., [71, p. 1078]) ℓ-linear, symmetric and continuous maps $a_\ell : (U_{ad})^\ell \to \mathbb{R}^N, (\boldsymbol{u}_1, \dots, \boldsymbol{u}_\ell) \mapsto a_\ell(\boldsymbol{u}_1, \dots, \boldsymbol{u}_\ell)$ such that $\boldsymbol{y}(\boldsymbol{u}) = \sum_{\ell=0}^\infty a_\ell(\boldsymbol{u} - \boldsymbol{u}_0)^\ell$. Now, define the maps $b_\ell : (U_{ad})^\ell \to \mathbb{R}^N$ as $b_\ell(\boldsymbol{u})^\ell := (a_\ell(\boldsymbol{u})^\ell)(T)$, meaning that $\sum_{\ell=0}^\infty b_\ell(\boldsymbol{u} - \boldsymbol{u}_0)^\ell = (\boldsymbol{y}(\boldsymbol{u}))(T)$. Since $S_T : Y \to \mathbb{R}^N, \boldsymbol{y} \mapsto \boldsymbol{y}(T)$ is linear and continuous, the maps b_ℓ are ℓ-linear, symmetric

and continuous. Thus, the map $S : U_{ad} \to \mathbb{R}^N, \boldsymbol{u} \mapsto (\boldsymbol{y}(\boldsymbol{u}))(T) = \sum_{\ell=0}^{\infty} b_\ell(\boldsymbol{u}-\boldsymbol{u}_0)^\ell$ is analytic by definition. □

In our case, we consider $U = L^2(0,T;\mathbb{R}^M)$ in the linear setting and $U = L^2(0,T;\mathbb{R})$ in the bilinear setting, $Y = H^1(0,T;\mathbb{R}^N)$ and $Z = L^2(0,T;\mathbb{R}^N) \times \mathbb{R}^N$. Then, the assumptions in Lemma 5.1.9 on the ODE system and its linearization are satisfied for (3.17) and (3.18) in the linear setting, and for (4.1) and (4.2) in the bilinear setting.[2] Notice that all solutions lie in $H^1(0,T;\mathbb{R}^N) \Subset C(0,T;\mathbb{R}^N)$ (see, e.g., [30]), which implies that the evolution map $S_T : H^1(0,T;\mathbb{R}^N) \to \mathbb{R}^N, \boldsymbol{y} \mapsto \boldsymbol{y}(T)$ is also linear and continuous. Now, we can prove our main result.

Theorem 5.1.10 (analysis for linear and bilinear systems) *Consider the linear setting* (3.17) *or the bilinear setting* (4.1)*. Assume that the systems are sufficiently observable and controllable, i.e., fully observable and controllable in the linear case, and satisfying Assumption 4.0.2 in the bilinear case. If* $\|A_{k+1}\|_2$ *is sufficiently small, then the set of controls in* U_{ad} *that satisfy* (5.6)-(5.7) *in Assumption 5.1.6 is nonempty.*

Proof. For brevity, we denote $A_{\boldsymbol{\beta}} := A(\boldsymbol{\beta}_\circ) + A^{(k)}(\boldsymbol{\beta}^k)$, $A_+ := A(\boldsymbol{\beta}_\circ) + A_{k+1}$, $\boldsymbol{y}_{\boldsymbol{\beta}}(\boldsymbol{u};t) := \boldsymbol{y}(A_{\boldsymbol{\beta}},\boldsymbol{u};t)$ and $\boldsymbol{y}_+(\boldsymbol{u};t) := \boldsymbol{y}(A_+,\boldsymbol{u};t)$. We start with the linear setting (3.17) from Section 3.2. First, we derive observability and controllability properties for the systems (A_+,B,C) and $(A_{\boldsymbol{\beta}},B,C)$. Denote by $\sigma_k > 0$ the smallest singular value of $\mathcal{O}_N(C,A(\boldsymbol{\beta}_\circ))$. Let $k \in \{1,\dots,K\}$ and $\boldsymbol{\beta}^k \in \mathbb{R}^k$ be the solution of (5.2) for $\|A_{k+1}\|_2 > 0$ sufficiently small such that $\|\mathcal{O}_N(C,A(\boldsymbol{\beta}_\circ)) - \mathcal{O}_N(C,A_+)\|_2 < \sigma_k$. From the proof of Theorem 5.1.5, we obtain that also $\boldsymbol{\beta}^k$ can be assumed to be sufficiently small such that $\|\mathcal{O}_N(C,A(\boldsymbol{\beta}_\circ)) - \mathcal{O}_N(C,A_{\boldsymbol{\beta}})\|_2 < \sigma_k$. Now, Lemma 3.2.10 guarantees that

$$\operatorname{rank}(\mathcal{O}_N(C,A_+)) = \operatorname{rank}(\mathcal{O}_N(C,A_{\boldsymbol{\beta}})) = N.$$

Using the same argument for the rank of the controllability matrices, we obtain that the systems (A_+,B,C) and $(A_{\boldsymbol{\beta}},B,C)$ are fully observable and controllable.

Next, we consider the state of the difference $\boldsymbol{z}(t) = \boldsymbol{y}(A_+,\boldsymbol{u};t) - \boldsymbol{y}(A_{\boldsymbol{\beta}},\boldsymbol{u};t)$ with $\dot{\boldsymbol{z}} = A_+\boldsymbol{z} + (A_+ - A_{\boldsymbol{\beta}})\boldsymbol{y}(A_{\boldsymbol{\beta}},\boldsymbol{u};t)$. Since $A_+ \neq A_{\boldsymbol{\beta}}$, there exists $\boldsymbol{v} \in \mathbb{R}^N$ such that $(A_+ - A_{\boldsymbol{\beta}})\boldsymbol{v} \neq 0$. Recalling that $(A_{\boldsymbol{\beta}},B)$ is controllable, we can find $\boldsymbol{u}_{t_1}$ for any $t_1 \in (0,T]$ such that $\boldsymbol{y}_{\boldsymbol{\beta}}(\boldsymbol{u}_{t_1};) = \boldsymbol{v}$ and therefore $(A_+ - A_{\boldsymbol{\beta}})\boldsymbol{y}_{\boldsymbol{\beta}}(\boldsymbol{u}_{t_1};t_1) \neq 0$. We define

$$\widetilde{\boldsymbol{u}}(s) := \begin{cases} \boldsymbol{u}_{t_1}(s), & \text{for } 0 \le s < t_1, \\ \boldsymbol{c}, & \text{for } t_1 \le s \le T, \end{cases}$$

where $\boldsymbol{c} \in \mathbb{R}^N$ is chosen later. For $t > t_1$, we have

$$\boldsymbol{z}(t) = e^{(t-t_1)A_+}\boldsymbol{z}(t_1) + \int_{t_1}^{t} e^{(t-s)A_+}(A_+ - A_{\boldsymbol{\beta}})\boldsymbol{y}_{\boldsymbol{\beta}}(\widetilde{\boldsymbol{u}};s)ds.$$

Now, we define

$$\widetilde{\boldsymbol{z}}(t) := e^{-(t-t_1)A_+}\boldsymbol{z}(t) = \boldsymbol{z}(t_1) + \int_{t_1}^{t} e^{(t_1-s)A_+}(A_+ - A_{\boldsymbol{\beta}})\boldsymbol{y}_{\boldsymbol{\beta}}(\widetilde{\boldsymbol{u}};s)ds. \tag{5.8}$$

[2] Existence and uniqueness of all solutions $\boldsymbol{y}, \delta\boldsymbol{y}$ follow by Carathéodory's existence theorem (see, e.g., [67, Thm. 54] and related propositions). For $\boldsymbol{u} \in L^2(0,T;\mathbb{R}^M)$ in the linear and $u \in L^2(0,T;\mathbb{R})$ in the bilinear setting, we obtain $\dot{\boldsymbol{y}}, \delta\dot{\boldsymbol{y}} \in L^2(0,T;\mathbb{R}^N)$ and thus $\boldsymbol{y}, \delta\boldsymbol{y} \in H^1(0,T;\mathbb{R}^N)$.

Notice that for $s > t_1$, the terms $e^{(t_1-s)A_+}$ and $\boldsymbol{y}_{\boldsymbol{\beta}}(\widetilde{\boldsymbol{u}}; s) = e^{(s-t_1)A_{\boldsymbol{\beta}}}\boldsymbol{v} + \int_0^s e^{(s-\tau)A_{\boldsymbol{\beta}}} B\boldsymbol{c} ds$ are continuous in s. Since exponential matrices are invertible (see, e.g., [49, pag. 369, 5.6.P43]) and $\boldsymbol{z}(t_1)$ is independent of t, there exists a $t > t_1$ such that

$$\boldsymbol{z}(t_1) + \int_{t_1}^{t} e^{(t_1-s)A_+}(A_+ - A_{\boldsymbol{\beta}})\boldsymbol{y}_{\boldsymbol{\beta}}(\widetilde{\boldsymbol{u}}; s) ds \neq 0$$

and thus $\widetilde{\boldsymbol{z}}(t) \neq 0$. Using (5.8), we obtain

$$C\boldsymbol{z}(t) = Ce^{(t-t_1)A_+}\widetilde{\boldsymbol{z}}(t) = \sum_{j=0}^{\infty} \frac{(t-t_1)^j}{j!} CA_+^j \widetilde{\boldsymbol{z}}(t). \tag{5.9}$$

The observability of (A_+, B, C) guarantees the existence of $i \in \{0, \dots, N-1\}$ such that $CA_+^i \widetilde{\boldsymbol{z}}(t) \neq 0$. We have $\frac{(t-t_1)^i}{i!} > 0$ for $t > t_1$ and all summands in (5.9) converge to zero at different rates. Hence, there exists $t > t_1$ such that $C\boldsymbol{z}(t) \neq 0$. Since $t_1 \in (0, T]$ was chosen arbitrarily, we get $C\boldsymbol{z}(T) \neq 0$ and thus $C\boldsymbol{y}_{\boldsymbol{\beta}}(\widetilde{\boldsymbol{u}}; T) - C\boldsymbol{y}_+(\widetilde{\boldsymbol{u}}; T) \neq 0$.

Regarding the linearized system (3.18), we have already shown in Lemma 3.2.5 that there exists an $\boldsymbol{u} \in U_{ad}$ such that $C\delta\boldsymbol{y}_\circ(A^{(k)}(\boldsymbol{\beta}^k), \boldsymbol{u}; T) - C\delta\boldsymbol{y}_\circ(A_{k+1}, \boldsymbol{u}; T) \neq 0$.

Finally, the maps $S, S_\delta : L^2(0, T; \mathbb{R}^M) \to \mathbb{R}^N$, $S(\boldsymbol{u}) := C\boldsymbol{y}_{\boldsymbol{\beta}}(\boldsymbol{u}; T) - C\boldsymbol{y}_+(\boldsymbol{u}; T)$, $S_\delta(\boldsymbol{u}) := C\delta\boldsymbol{y}_\circ(A^{(k)}(\boldsymbol{\beta}^k), \boldsymbol{u}; T) - C\delta\boldsymbol{y}_\circ(A_{k+1}, \boldsymbol{u}; T)$ are analytic by Lemma 5.1.9. Using Lemma 5.1.8, we obtain the existence of an $\boldsymbol{u} \in U_{ad}$ such that $C\boldsymbol{y}(A_{\boldsymbol{\beta}}, \boldsymbol{u}; T) - C\boldsymbol{y}(A_+, \boldsymbol{u}; T) \neq 0$ and $C\delta\boldsymbol{y}_\circ(A^{(k)}(\boldsymbol{\beta}^k), \boldsymbol{u}; T) - C\delta\boldsymbol{y}_\circ(A_{k+1}, \boldsymbol{u}; T) \neq 0$.

Now, let us consider the bilinear setting (4.1) from Chapter 4. First, we derive the observability and controllability properties for the systems (A_+, B, C) and $(A_{\boldsymbol{\beta}}, B, C)$ from Assumption 4.0.2. Similar to the prove for the linear setting above, one can show that Assumption 4.0.2 holds for (A_+, B, C) and $(A_{\boldsymbol{\beta}}, B, C)$ if $\|A_{k+1}\|_2$, and thereby also $\|\boldsymbol{\beta}^k\|_2$, is small enough.

Next, we want to show the existence of a control $u \in L^2(0, T; \mathbb{R})$ such that $C\boldsymbol{y}(A_{\boldsymbol{\beta}}, u; T) \neq C\boldsymbol{y}(A_+, u; T)$. We consider the state $\boldsymbol{z}(t) = \boldsymbol{y}(A_+, u; t) - \boldsymbol{y}(A_{\boldsymbol{\beta}}, u; t)$ with

$$\dot{\boldsymbol{z}}(t) = (A_+ + u(t)B)\boldsymbol{z}(t) + (A_+ - A_{\boldsymbol{\beta}})\boldsymbol{y}(A_{\boldsymbol{\beta}}, u; t).$$

Since $A_+ \neq A_{\boldsymbol{\beta}}$ and the system $(A_{\boldsymbol{\beta}}, B, C)$ is controllable for $t > 0$ large enough, there exists a control u and a time $t_1 > 0$ such that $(A_+ - A_{\boldsymbol{\beta}})\boldsymbol{y}(A_{\boldsymbol{\beta}}, u; t_1) \neq 0$. We define

$$\widetilde{u}(s) := \begin{cases} u(s), & \text{for } 0 \leq s < t_1, \\ c, & \text{for } t_1 \leq s \leq T, \end{cases} \tag{5.10}$$

where $c \in \mathbb{R}$ is to be chosen later. For $t > t_1$, we obtain

$$\boldsymbol{z}(t) = e^{(t-t_1)(A_+ + cB)}\boldsymbol{z}(t_1) + \int_{t_1}^{t} e^{(t-s)(A_+ + cB)}(A_+ - A_{\boldsymbol{\beta}})\boldsymbol{y}(A_{\boldsymbol{\beta}}, \widetilde{u}; s) ds. \tag{5.11}$$

Now, we define

$$\widetilde{\boldsymbol{z}}(t) := e^{-(t-t_1)(A_+ + cB)}\boldsymbol{z}(t) = \boldsymbol{z}(t_1) + \int_{t_1}^{t} e^{(t_1-s)(A_+ + cB)}(A_+ - A_{\boldsymbol{\beta}})\boldsymbol{y}(A_{\boldsymbol{\beta}}, \widetilde{u}; s) ds. \tag{5.12}$$

Notice that, for $s > t_1$, the terms $e^{(t_1-s)(A_+ + cB)}$ and $\boldsymbol{y}(A_{\boldsymbol{\beta}}, \widetilde{u}; s) = e^{s(A_{\boldsymbol{\beta}}+cB)}\boldsymbol{y}(A_{\boldsymbol{\beta}}, u; t_1)$ are continuous in s. Since exponential matrices are invertible and $\boldsymbol{z}(t_1)$ is independent of t, there exists a $t > t_1$ such that

$$\boldsymbol{z}(t_1) + \int_{t_1}^{t} e^{(t_1-s)(A_+ + cB)}(A_+ - A_{\boldsymbol{\beta}})\boldsymbol{y}(A_{\boldsymbol{\beta}}, \widetilde{u}; s)ds \neq 0$$

and hence $\widetilde{\boldsymbol{z}}(t) \neq 0$. Using (5.12), we obtain

$$C\boldsymbol{z}(t) = Ce^{(t-t_1)(A_+ + cB)}\widetilde{\boldsymbol{z}}(t) = \sum_{j=0}^{\infty} \frac{(t-t_1)^j}{j!} C(A_+ + cB)^j \widetilde{\boldsymbol{z}}(t). \tag{5.13}$$

By the third point in Assumption 4.0.2, there exists a $c \in \mathbb{R}$ such that $\mathcal{O}_N(C, A_+ + cB)$ has full rank. This implies the existence of an $i \in \{0, \ldots, N-1\}$ such that

$$C(A_+ + cB)^i \widetilde{\boldsymbol{z}}(t) \neq 0.$$

Since $\frac{(t-t_1)^i}{i!} > 0$ for $t > t_1$, and all terms of the sum in (5.13) converge to zero for $t \to 0$ at different rates for different j, there exists $t > t_1$ such that $C\boldsymbol{z}(t) \neq 0$. By analycity of $C\boldsymbol{z}(t)$ in T and c, we obtain $C\boldsymbol{y}(A_{\boldsymbol{\beta}}, \widetilde{u}; T) \neq C\boldsymbol{y}(A_+, \widetilde{u}; T)$ for T sufficiently large.

Regarding the linearized system (4.2), Lemma 4.1.1 implies that there exists a $u \in L^2(0, T; \mathbb{R})$ such that $C\delta\boldsymbol{y}_\circ(A^{(k)}(\boldsymbol{\beta}^k), u; T) \neq C\delta\boldsymbol{y}_\circ(A_{k+1}, u; T)$.

By Lemma 5.1.9, the maps $S, S_\delta : L^2(0, T; \mathbb{R}) \to \mathbb{R}^N$, defined by $S(u) := \boldsymbol{y}(A, u; T)$, $S_\delta(u) := \delta\boldsymbol{y}_\circ(A, u; T)$ are analytic. Using Lemma 5.1.8, we obtain the existence of an $u \in U_{ad}$ such that $C\boldsymbol{y}(A_{\boldsymbol{\beta}}, u; T) \neq C\boldsymbol{y}(A_+, u; T)$ and $C\delta\boldsymbol{y}_\circ(A^{(k)}(\boldsymbol{\beta}^k), u; T) \neq C\delta\boldsymbol{y}_\circ(A_{k+1}, u; T)$. □

Remark 5.1.11 *Notice that we did not prove exactly Assumption 5.1.6 in Theorem 5.1.10, but only the existence of a general control $\boldsymbol{u} \in U_{ad}$ that satisfies* (5.6)-(5.7)*. However, this implies that any solution $\boldsymbol{u}^{k+1}$ to* (5.3) *always satisfies* (5.6)*. Additionally, we recall from the proof of Theorem 5.1.10 that the maps $S, S_\delta : L^2(0, T; \mathbb{R}^M) \to \mathbb{R}^N$, defined by $S(\boldsymbol{u}) := \boldsymbol{y}(A, \boldsymbol{u}; T)$, $S_\delta(\boldsymbol{u}) := \delta\boldsymbol{y}_\circ(A, \boldsymbol{u}; T)$ are nonzero. Thus, we obtain by Lemma 5.1.8 that any neighborhood of $\boldsymbol{u}^{k+1}$ can contain only isolated controls not satisfying* (5.7)*, and infinitely many $\boldsymbol{u}$ that do satisfy* (5.7)*. Thus, it is rather unlucky to choose an $\boldsymbol{u}^{k+1}$ not satisfying* (5.7)*. On the other hand, one can also add inequality* (5.7) *as a constraint to* (5.3) *to ensure that both inequalities are met by $\boldsymbol{u}^{k+1}$.*

As a consequence of Theorems 5.1.7, 5.1.10 and Remark 5.1.11, the controls generated by Algorithm 4 for (3.17) and (4.1) make the GN matrix $\widehat{W}_\circ$, defined in (3.19), PD under certain assumptions. Thus, the results from Sections 3.2.5 and 4.2 imply that GN for the reconstruction problems (3.29) and (2.3), initialized with $\boldsymbol{\beta}_\circ$, converges to $\boldsymbol{\beta}_\star$.

5.1.2 A general class of nonlinear systems

Now, we focus on a class of more general nonlinear systems and prove our second main result, which is analogous to Theorem 5.1.10. These systems (of the form (2.1)) are characterized by a function f of the form $f(A, \boldsymbol{y}, \boldsymbol{u}) = g(\boldsymbol{y}) + A\boldsymbol{u}$, and thus have the form

$$\dot{\boldsymbol{y}}(t) = g(\boldsymbol{y}(t)) + A(\boldsymbol{\beta}_\star)\boldsymbol{u}(t), \quad t \in [0, T], \tag{5.14}$$

where $\boldsymbol{y}(t) \in \mathbb{R}^N, N \in \mathbb{N}$. We make the following hypotheses.

Assumption 5.1.12 (on the system (5.14)**)** *The initial state is* $\boldsymbol{y}(0) = 0$. *The function* g *is in* $C^1(\mathbb{R}^N, \mathbb{R}^N)$ *and satisfies* $g(0) = 0$.[3] *Additionally,* g *is Lipschitz continuous with Lipschitz constant* L. *The approximation of* $\boldsymbol{\beta}_\star$ *is given by* $\boldsymbol{\beta}_\circ = 0$. *The state* $\boldsymbol{y}(T)$ *can be fully observed, meaning that the observer matrix is the identity:* $C = I \in \mathbb{R}^{N\times N}$.

Assumption 5.1.12 and $A(\boldsymbol{\beta}_\circ = 0) = 0$ imply that the linearized equation in $\boldsymbol{\beta}_\circ = 0$ is

$$\begin{aligned} \dot{\delta\boldsymbol{y}}_\circ(t) &= g'(\boldsymbol{y}_\circ(t))\delta\boldsymbol{y}_\circ(t) + A(\boldsymbol{\beta})\boldsymbol{u}(t), \quad \delta\boldsymbol{y}(0) = 0, \\ \dot{\boldsymbol{y}}_\circ(t) &= g(\boldsymbol{y}_\circ(t)), \quad \boldsymbol{y}_\circ(0) = 0. \end{aligned}$$

Using that $\boldsymbol{y}_\circ(0) = 0$ and $g(0) = 0$ from Assumption 5.1.12, we get $\boldsymbol{y}_\circ(t) = 0$ for all $t \in [0, T]$, and thus

$$\dot{\delta\boldsymbol{y}}_\circ(t) = g'(0)\delta\boldsymbol{y}_\circ(t) + A(\boldsymbol{\beta})\boldsymbol{u}(t).$$

Now, we show that there exists a control satisfying inequality (5.6) in Assumption 5.1.6. For this purpose, we introduce the notation $\widetilde{A} := A^{(k)}(\boldsymbol{\beta}^k) - A_{k+1}$, $\boldsymbol{y}_{\boldsymbol{\beta}}(t) := \boldsymbol{y}(A^{(k)}(\boldsymbol{\beta}^k), \boldsymbol{u}^{k+1}; t)$, $\boldsymbol{y}_{k+1}(t) := \boldsymbol{y}(A_{k+1}, \boldsymbol{u}^{k+1}; t)$, and $\Delta\boldsymbol{y}(t) := \boldsymbol{y}_{\boldsymbol{\beta}}(t) - \boldsymbol{y}_{k+1}(t)$. Then, $\Delta\boldsymbol{y}(t)$ satisfies

$$\dot{\Delta\boldsymbol{y}}(t) = g(\boldsymbol{y}_{\boldsymbol{\beta}}(t)) - g(\boldsymbol{y}_{k+1}(t)) + \widetilde{A}\boldsymbol{u}^{k+1}(t), \quad \Delta\boldsymbol{y}(0) = 0. \tag{5.15}$$

By multiplying (5.15) with $\widetilde{A}\boldsymbol{u}^{k+1}(t)$, and using the Lipschitz continuity of g, we get

$$\begin{aligned} \|\widetilde{A}\boldsymbol{u}^{k+1}(t)\|_2^2 &= \langle \widetilde{A}\boldsymbol{u}^{k+1}(t), \dot{\Delta\boldsymbol{y}}(t)\rangle + \langle \widetilde{A}\boldsymbol{u}^{k+1}(t), g(\boldsymbol{y}_{k+1}(t)) - g(\boldsymbol{y}_{\boldsymbol{\beta}}(t))\rangle \\ &\le \|\widetilde{A}\boldsymbol{u}^{k+1}(t)\|_2 \Big(\|\dot{\Delta\boldsymbol{y}}(t)\|_2 + L\|\Delta\boldsymbol{y}(t)\|_2 \Big), \end{aligned}$$

where we used Cauchy-Schwarz and triangle inequalities. Thus, we obtain

$$\|\widetilde{A}\boldsymbol{u}^{k+1}(t)\|_2 \le \|\dot{\Delta\boldsymbol{y}}(t)\|_2 + L\|\Delta\boldsymbol{y}(t)\|_2. \tag{5.16}$$

Since $A^{(k)}(\boldsymbol{\beta}^k)$ and A_{k+1} are linearly independent, we have that $\widetilde{A} \neq 0$. Thus, there exists $\boldsymbol{v} \in \mathbb{R}^M$ such that $\widetilde{A}\boldsymbol{v} \neq 0$. Now, we define the control

$$\boldsymbol{u}_{\boldsymbol{v},\tau}(t) := \begin{cases} 0, & t \in [0, \tau), \\ \boldsymbol{v}, & t \in [\tau, T], \end{cases}$$

for an arbitrary $\tau \in (0, T)$. By (5.16) it holds that

$$\|\dot{\Delta\boldsymbol{y}}(t)\|_2 + L\|\Delta\boldsymbol{y}(t)\|_2 \ge \|\widetilde{A}\boldsymbol{u}_{\boldsymbol{v},\tau}(t)\|_2 > 0$$

for all $t > \tau$. Recalling that $C = I$ by Assumption 5.1.12, if $\|\Delta\boldsymbol{y}(T)\|_2 > 0$, the first inequality (5.6) is satisfied and we could move to inequality (5.7). Thus, assume that $\|\Delta\boldsymbol{y}(T)\|_2 = 0$. Then $\|\dot{\Delta\boldsymbol{y}}(T)\|_2 \neq 0$ and therefore $\|\Delta\boldsymbol{y}(T - \eta)\|_2 \neq 0$ for some $\eta > 0$. Notice that η is independent of τ. Since $\Delta\boldsymbol{y}(t) = 0$ for $t \in [0, \tau)$, we can simply shift τ by η, namely considering $\boldsymbol{u}_{\boldsymbol{v},\tau+\eta}$, to obtain $\|\Delta\boldsymbol{y}(T)\|_2 > 0$. In conclusion, we have found a control, namely $\boldsymbol{u}^{k+1} = \boldsymbol{u}_{\boldsymbol{v},\tau+\eta}$, satisfying (5.6).

[3] The first two hypotheses imply that the system with $\boldsymbol{u} = 0$ has an equilibrium point at zero.

Now, we show that $\boldsymbol{u}^{k+1} = \boldsymbol{u}_{\boldsymbol{v},\tau+\eta}$ satisfies (5.7) in Assumption 5.1.6 as well. To do so, let $\Delta\boldsymbol{y}_\delta(t) := \delta\boldsymbol{y}_\circ(A^{(k)}(\boldsymbol{\beta}^k), \boldsymbol{u}^{k+1}; t) - \delta\boldsymbol{y}_\circ(A_{k+1}, \boldsymbol{u}^{k+1}; t)$, which clearly satisfies

$$\dot{\Delta\boldsymbol{y}}_\delta(t) = g'(0)\Delta\boldsymbol{y}_\delta(t) + \widetilde{A}\boldsymbol{u}^{k+1}(t), \qquad \Delta\boldsymbol{y}_\delta(0) = 0.$$

Now, since $\widetilde{A}\boldsymbol{u}^{k+1}(t) = \widetilde{A}\boldsymbol{u}_{\boldsymbol{v},\tau+\eta}(t) \neq 0$ for $t \in [\tau+\eta, T]$, thus $\dot{\Delta\boldsymbol{y}}_\delta(\tau+\eta) \neq 0$. Hence, since $t \mapsto \Delta\boldsymbol{y}_\delta(t)$ is continuous, there exists an $\widetilde{\eta} > 0$ such that $\tau+\widetilde{\eta} \in (\tau+\eta, T]$ and $\Delta\boldsymbol{y}_\delta(t) \neq 0$ for all $t \in (\tau+\eta, \tau+\widetilde{\eta}]$. If $\Delta\boldsymbol{y}_\delta(T) \neq 0$ we proved our claim. Thus, we assume that $\tau+\widetilde{\eta} < T$ and $\Delta\boldsymbol{y}_\delta(T) = 0$. In this case, we can use the fact that τ is arbitrary and choose it to shift the interval $(\tau+\eta, \tau+\widetilde{\eta}]$ in order to get that $T \in (\tau+\eta, \tau+\widetilde{\eta}]$. This implies that the control $\boldsymbol{u}^{k+1} = \boldsymbol{u}_{\boldsymbol{v},\tau+\eta}$ satisfies both inequalities (5.6) and (5.7) in Assumption 5.1.6.

We summarize our findings in the next theorem (analogue of Theorem 5.1.10).

Theorem 5.1.13 (analysis for systems of the form (5.14)**)** *Consider system* (5.14) *and let Assumption 5.1.12 hold. Then the set of controls in U_{ad} that satisfy both* (5.6) *and* (5.7) *in Assumption 5.1.6 is nonempty.*

Remark 5.1.14 *Theorem 5.1.13 is an alternative convergence proof for the linear setting considered in Section 3.1, where the unknown is the control matrix. However, it does not cover the setting of Section 3.2, where the unknown is the drift matrix. Moreover, as in Remark 5.1.11, Theorem 5.1.13 does not exactly prove Assumption 5.1.6, but rather that there exist controls that simultaneously satisfy* (5.6) *and* (5.7)*. On the other hand, the map $\boldsymbol{u} \mapsto \Delta\boldsymbol{y}_\delta(T) = \delta\boldsymbol{y}_\circ(A^{(k)}(\boldsymbol{\beta}^k), \boldsymbol{u}; T) - \delta\boldsymbol{y}_\circ(A_{k+1}, \boldsymbol{u}; T)$ is analytic by Lemma 5.1.9, and the map $t \mapsto \Delta\boldsymbol{y}_\delta(t)$ is (at least) continuous. In this case, the control functions $\boldsymbol{u}$ that violate simultaneously* (5.6) *and* (5.7) *are isolated points. Therefore, it is theoretically possible but also very unlikely that the solutions $\boldsymbol{u}^{k+1}$ to the nonlinear splitting step problem* (5.3) *do not satisfy* (5.7) *in Assumption 5.1.6.*

5.2 Global convergence in a specific case

In the previous sections, we discussed the performance of the GR algorithm on the local convergence of GN for the online reconstruction problem. In the context of inverse problems for nonlinear ODEs, global convergence can generally not be expected, independently of the method used to generate control functions or data. A sufficient condition for the global convergence of GN applied to such problems is strict (global) convexity of the online reconstruction problem (2.3). Thus, if GR can produce controls making the cost of (2.3) strictly convex, then GN (or other optimization solvers) exhibits global convergence. One example, where strict convexity can be proven under certain assumptions, is given by the linear-quadratic setting discussed in Section 3.1. In what follows, we discuss global convergence (convexity) for a specific case of a bilinear system.

The goal is to reconstruct a matrix $A_\star = \beta_{\star,1}A_1 + \beta_{\star,2}A_2 \in \mathbb{R}^{2\times 2}$ of the system

$$\dot{\boldsymbol{y}}(t) = u(t)A_\star\boldsymbol{y}(t), \quad t \in [0,T], \quad \boldsymbol{y}(0) = \boldsymbol{y}^0, \tag{5.17}$$

where $T = 1$. Assume that

$$\boldsymbol{y}^0 = \begin{bmatrix}1\\0\end{bmatrix}, \quad \boldsymbol{\beta}_\star = \boldsymbol{\beta}_\circ = \begin{bmatrix}0\\0\end{bmatrix}, \quad A_1 = \begin{bmatrix}1 & 0\\0 & -1\end{bmatrix}, \quad A_2 = \begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}.$$

We also assume that we can observe the full state $\boldsymbol{y}(T)$, i.e., $C = I \in \mathbb{R}^{2\times 2}$. We consider that U_{ad} is a set of constant controls that are bounded in absolute value by 1, i.e., $u(t) \equiv u \in U_{ad} = [-1, 1]$. Thus, the solution to (5.17) can be written explicitly as

$$\boldsymbol{y}(\boldsymbol{\beta}, u; t) = e^{tuA(\boldsymbol{\beta})}\boldsymbol{y}^0.$$

Hence, the corresponding online reconstruction problem (2.3) reads as

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^2} J(\boldsymbol{\beta}) := \sum_{m=1}^{2} \|e^{(u_m(\beta_1 A_1+\beta_2 A_2))}\boldsymbol{y}^0 - e^{u_m A_\star}\boldsymbol{y}^0\|_2^2. \tag{5.18}$$

Since $A_\star$ is zero, we have $e^{uA_\star}\boldsymbol{y}^0 = \boldsymbol{y}^0$. Using the expressions for A_1 and A_2 and the result [22, Cor. 2.3], one gets

$$e^{u(\beta_1 A_1+\beta_2 A_2)}\boldsymbol{y}^0 = \frac{1}{\|\boldsymbol{\beta}\|_2}\begin{bmatrix} \|\boldsymbol{\beta}\|_2 \cosh(\|\boldsymbol{\beta}\|_2 u) + \beta_1 \sinh(\|\boldsymbol{\beta}\|_2 u) \\ \beta_2 \sinh(\|\boldsymbol{\beta}\|_2 u) \end{bmatrix},$$

for $\boldsymbol{\beta} \neq 0$. Thus, by a direct calculation, we can write the cost function J in (5.18) as

$$\begin{aligned} J(\boldsymbol{\beta}) = \sum_{m=1}^{2} \Big(&\cosh(\|\boldsymbol{\beta}\|_2 u_m)^2 + \sinh(\|\boldsymbol{\beta}\|_2 u_m)^2 - 2\cosh(\|\boldsymbol{\beta}\|_2 u_m) \\ &+ \frac{2\beta_1 \sinh(\|\boldsymbol{\beta}\|_2 u_m)}{\|\boldsymbol{\beta}\|_2}(\cosh(\|\boldsymbol{\beta}\|_2 u_m) - 1) + 1\Big), \end{aligned}$$

with $J(0) := \lim_{\beta \searrow 0} J(\beta) = 0$. Assume now that $u_1 = -1$ and $u_2 = 1$. Changing the variable to $x = \|\boldsymbol{\beta}\|_2$, and using that $\sinh(-x) = -\sinh(x)$ and $\cosh(-x) = \cosh(x)$, we get

$$J(\boldsymbol{\beta}) = 2\cosh(\|\boldsymbol{\beta}\|_2)^2 + 2\sinh(\|\boldsymbol{\beta}\|_2)^2 - 4\cosh(\|\boldsymbol{\beta}\|_2) + 2 =: \widetilde{J}(x = \|\boldsymbol{\beta}\|_2).$$

Thus, we have

$$\widetilde{J}''(x) = 8\cosh(x)^2 + 8\sinh(x)^2 - 4\cosh(x).$$

Since $\cosh(x) \geq 1$ and $\sinh(x) \geq 0$ for $x \geq 0$, it holds that $\widetilde{J}''(x) > 0$ for $x \geq 0$. Therefore, J is strictly convex. Hence, a sufficient condition for global convergence of GN is that our proposed methods select the two controls $u_1 = -1$ and $u_2 = 1$ (or $u_1 = 1$ and $u_2 = -1$). Notice that, if only one control is used among 1 and -1, then J is not convex.

Let us now study the GR algorithm. The initialization problem is given by

$$\max_{u\in[-1,1]} J_I(u) := \|\boldsymbol{y}^0 - e^{uA_1}\boldsymbol{y}^0\|_2^2.$$

One can show that $J_I(u) = (\exp(u) - 1)^2$, which attains its unique global maximum at $u = 1$. Thus, assuming that the splitting-step solver converges to the unique global maximizer, GR will choose $u_1 = 1$. Let us now consider the fitting step problem

$$\min_{\beta\in\mathbb{R}} J_F(\beta) := \|e^{u_1\beta A_1}\boldsymbol{y}^0 - e^{u_1 A_2}\boldsymbol{y}^0\|_2^2.$$

Direct calculations show that

$$J_F(\beta) = e^{2\beta} - 2\cosh(1)e^{\beta} + \cosh(1)^2 + \sinh(1)^2,$$
$$J_F'(\beta) = 2e^{\beta}(e^{\beta} - \cosh(1)).$$

Clearly, $J_F'(\log(\cosh(1))) = 0$ and $J_F'(\beta)$ is negative for $\beta < \log(\cosh(1))$ and positive for $\beta > \log(\cosh(1))$. Since it is continuous in 0, J_F has a global minimum in $\beta_1 := \log(\cosh(1))$. Now, consider the splitting step:

$$\max_{u\in[-1,1]} J_S(u) := \|e^{u\beta_1 A_1}\boldsymbol{y}^0 - e^{uA_2}\boldsymbol{y}^0\|_2^2. \tag{5.19}$$

Proceeding as before, one can show that (5.19) has a unique global maximizer given by $u = -1$. Thus, assuming again that the splitting step solver converges to the unique global maximizer, GR will choose $u_2 = 1$.

In conclusion, GR chooses $u_1 = 1$ and $u_2 = -1$, making the online reconstruction problem (5.18) strictly convex and hence leading to global convergence of GN.

The above example is very instructive. On the one hand, as we are going to see, it can be used to reveal some properties (and weaknesses) of GR. On the other hand, it is straightforward to carry out similar calculations to compare GR, LGR and OGR. These calculations (omitted here for brevity) lead to the following remarks.

The above results about GR for the specific case (5.17) depend heavily on the order of the basis elements A_1 and A_2. If one repeats the calculation with the order of A_1 and A_2 reversed, then the initialization problem has two global maximizers at $u = \pm 1$. Independently of which of these is chosen, the fitting step has two global minimizers. Each of them leads to a splitting step problem having two global maximizers located again at $u = \pm 1$, independently of the chosen value in the initialization step. Thus, even if one assumes that the optimizations solvers used for initialization, fitting and splitting steps converge to global (maximum/minimum) points, it can happen that the control computed at the splitting step is equal to the one obtained at the initialization step, making the cost of the online reconstruction problem not convex. The importance of the ordering of the elements in $\mathcal{B}$ was already discussed in detail in Section 3.1 for the reconstruction of the control matrix B in case of linear systems. This was exactly the motivation for designing OGR. In fact, if one repeats the above calculation for OGR, the two controls $u = 1$ and $u = -1$ are always obtained, independently of the ordering of A_1 and A_2. Finally, for LGR one can show that, independently of the ordering of A_1 and A_2, it can happen that the controls computed at the initialization and splitting steps are equal, leading to a non-convex online reconstruction problem. The reason is that the linearization process does not allow to fully capture the dynamics of the system. This is also apparent in the numerical experiments in the next chapter.

Chapter 6

Numerical experiments for linear and bilinear systems

We study efficiency and robustness of the GR and OGR algorithms by direct numerical experiments. In Section 6.1, we consider the reconstruction of a drift matrix. In Section 6.2, we address a Hamiltonian identification problem, as discussed in [57]. In Section 6.3, we focus on a multi-spin reconstruction problem. All fitting-step problems in the GR algorithms are solved by the standard MATLAB function `fminunc` that uses a BFGS (Broyden, Fletcher, Goldfarb, Shanno) descent-direction method (see [28, 37, 43, 65]) with a cubic line search procedure.[1] All splitting-step problems are solved by a projected BFGS descent-direction method with an Armijo line search (see, e.g., [51, Section 5.5.3]). The online reconstruction problems are solved by GN.

6.1 Reconstruction of drift matrices

We consider system (3.17) with (full rank) randomly generated matrices $A_\star, B, C \in \mathbb{R}^{3\times 3}$. The final time is $T = 1$ and the initial value is $\boldsymbol{y}^0 = [0, 0, 0]^\top$. For the set of admissible controls U_{ad} we use box constraints, i.e.

$$U_{ad} = \{\boldsymbol{u} \in L^2(0, T; \mathbb{R}^3) \mid |u_j(t)| \leq 100 \text{ a.e in } (0, T), j = 1, 2, 3\}.$$

First, we study the algorithms for system (3.18). This is obtained by linearizing (3.17) around two different $A_\circ$, which are randomly chosen approximations to $A_\star$, one with 1% and the other with 10% relative error, meaning that, e.g., $\frac{\|A_\star - A_\circ\|_F}{\|A_\star\|_F} = 0.01$ for the one with 1% error, where $\|\cdot\|_F$ is the Frobenius norm. The LGR Algorithm 3 is run for two different choices for the basis $\mathcal{B}$: the canonical basis of $\mathbb{R}^{3\times 3}$ and a basis consisting of 9 randomly generated (linearly independent) 3×3 matrices. LGR is also compared with OLGR, which is run with a set of 18 matrices, namely, the 9 canonical basis elements and the 9 random matrices. The controls generated by the respective algorithms are then used

[1] Notice that we implicitly assume that the set $\mathcal{K}_k$ in the nonlinear GR Algorithm 4 and its optimized version is chosen large enough such that the BFGS method never reaches the boundary of $\mathcal{K}_k$.

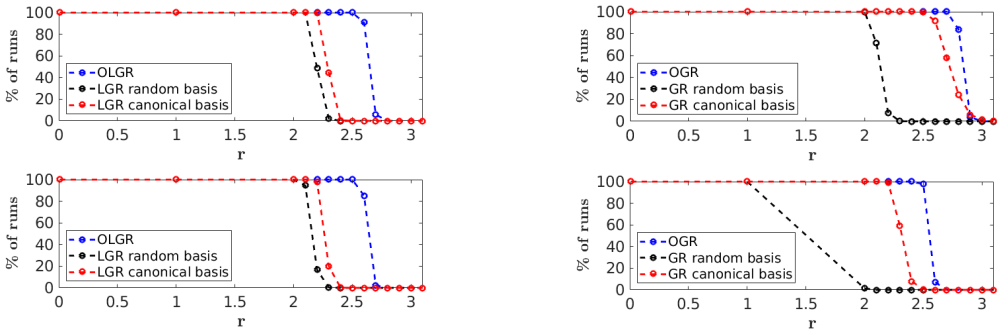

Figure 6.1: Percentage of runs that converged to $A_\star$ initialized by randomly chosen vectors on a sphere with radius r, for controls generated by LGR and OLGR for 1% (top left) and 10% (bottom left) relative error between $A_\star$ and $A_\circ$, and GR and OGR with (bottom right) and without the shift by $A_\circ$ (top right).

to reconstruct the matrix $A_\star$ by solving the online least-squares problem (2.3) with GN. To test the robustness of the control functions, we consider a nine-dimensional sphere centered in the global minimum $A_\star$ and with given relative radius r, and repeat the minimization for 1000 initialization vectors randomly chosen on this sphere. We then count the percentage of times that GN converges to the global solution $A_\star = A(\boldsymbol{\beta}_\star)$ up to a tolerance of $Tol = 0.005$ (half of the smallest considered radius), meaning that $\frac{\|A_\star - A(\boldsymbol{\beta}_{comp})\|_F}{\|A_\star\|_F} \leq Tol$, where $\boldsymbol{\beta}_{comp}$ denotes the solution computed by GN. Repeating this experiment for different radii of the sphere, we obtain the results reported in Figure 6.1. All control sets make GN capable of reliably reconstructing the global minimum $A_\star$ up to a relative radius $r = 2$, which corresponds to a relative error of 200%. This demonstrates that the choice of the basis is not crucial for fully observable and controllable systems. However, OLGR is able to reduce the number of controls down to 3 and still outperforms any set of 9 controls generated by LGR, while staying reliable up to a relative error of 250%. Thus, OLGR is able to compute a better basis and to omit unnecessary controls.

Next, we repeat the same experiments for GR and OGR. However, we replace the case for the approximation $A_\circ$ with a relative error of 1% by $A_\circ = 0$. This effectively removes the shift and makes the algorithm independent of the choice of $A_\circ$, which is the version of the algorithm considered in [57] and [1]. We obtain the results shown in the two panels on the right in Figure 6.1. The performance of the control sets is similar to the ones for the linearized system, with an increase in performance for GR with the canonical basis, without the shift by $A_\circ$, and a decrease in performance for GR with the random basis and an $A_\circ$ that has a 10% relative error with respect to $A_\star$. As in the linearized setting, OGR is able to reduce the number of controls to 3 and still outperforms any set of 9 controls generated by LGR.

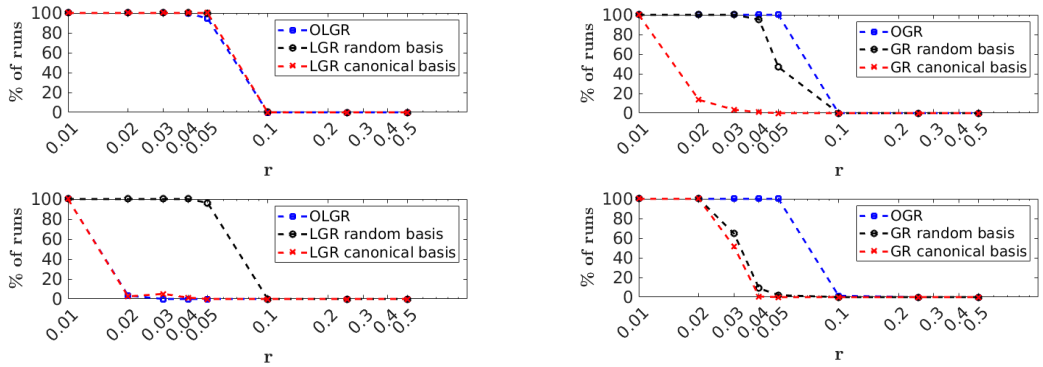

Figure 6.2: Percentage of runs that converged to $\mu_\star \in \mathbb{C}^{3\times 3}$ initialized by randomly chosen vectors on a sphere of radius r, for controls generated by LGR and OLGR for 1% (top left) and 10% (bottom left) relative error between $\mu_\star$ and $\mu_\circ$, and GR and OGR with (bottom right) and without the shift by $\mu_\circ$ (top right).

6.2 Bilinear reconstruction problem

Similar to [57], we consider a Schrödinger-type equation, written as a real system as in (4.5) with matrices

$$H = H_R = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 8 & 0 \\ 0 & 0 & 16 \end{bmatrix},$$

$$\mu_\star = \begin{bmatrix} -0.3243 & -3.4790 + 0.7359i & -0.5338 + 1.9254i \\ -3.4790 - 0.7359i & -3.8342 & -1.1697 + 2.0256i \\ -0.5338 - 1.9254i & -1.1697 - 2.0256i & 1.0551 \end{bmatrix}.$$

The final time is $T = 10\pi$ and the initial state is $\boldsymbol{\psi}_0 = [1, 0, 0]^\top$. The observer matrix is $C = [\boldsymbol{\psi}_1, i\boldsymbol{\psi}_1]$, which means that the final state is measured against the fixed state $\boldsymbol{\psi}_1 = \frac{1}{\sqrt{3}}[1, 1, 1]^\top$. For U_{ad} we use the box constraints $|u(t)| \leq 1$. Again, we consider two bases, each consisting of 9 elements: the canonical and a random one for the space of Hermitian matrices in $\mathbb{C}^{3\times 3}$. We then perform the same experiments as in Section 6.1. The results are reported in Figure 6.2. We observe that the radii, up to which the control sets make GN capable of reliably reconstructing the global minimum, are much smaller than for the linear setting in Section 6.1. When the initial relative error between $\mu_\circ = \mu(\boldsymbol{\beta}_\circ)$ and $\mu_\star = \mu(\boldsymbol{\beta}_\star)$ is very small (1%) then LGR and OLGR have the most stable performance regarding the choice of the basis, making GN capable of reliably reconstructing the global minimum $\mu_\star$ up to a relative error of $4 - 5\%$. However, when the initial relative error is larger (10%) then only the LGR algorithm for the random basis can keep its performance, while even OLGR fails at errors of over 1%. The results for OGR, on the other hand, show the best performance, with and without a shift by $\mu_\circ$. The controls generated by the GR algorithms can not match OGR or LGR and OLGR for small initial errors, but are still more stable with respect to larger initial errors.

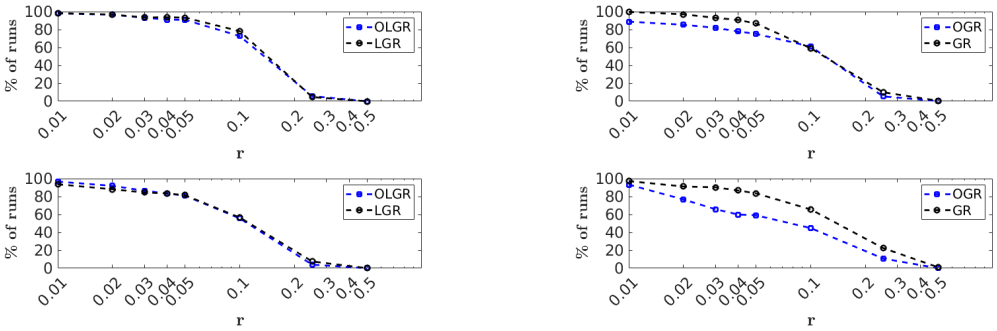

Figure 6.3: Same as Figure 6.1 but for the experiments of Section 6.3.

6.3 Multi-spin (/multi-qubit) reconstruction problem

Let us consider the case of a 3-qubit system. We use the 64-dimensional real representation of the Liouville master equation (compare, e.g., [25, Section 2.12.1]):

$$\dot{\boldsymbol{y}}(t) = \Big(A_\star + u(t)\sum_{n=1}^{6} B_n\Big)\boldsymbol{y}(t), \quad t \in [0,T], \quad \boldsymbol{y}(0) = \boldsymbol{y}^0,$$

where $A_\star = 2\pi\sum_{i=1}^{3}\omega_{\star,i}\widehat{A}_i + 2\pi\sum_{j=1}^{2} J_{\star,j}\widetilde{A}_j \in \mathbb{R}^{64\times 64}$, $B_n = 2\pi\widehat{B}_{\frac{n+1}{2}}$ for n odd and $B_n = 2\pi\widetilde{B}_{\frac{n}{2}}$ for n even. Here, the matrices $\widehat{A}_i$ and $\widetilde{A}_j$ correspond to the free evolution and the coupling of the spins, and the matrices $\widehat{B}_n$ and $\widetilde{B}_n$ correspond to the controlled evolution. The goal is to reconstruct $A_\star$, i.e. the coefficients $\omega_{\star,i}, J_{\star,j}$, which we assume to be equal to 1. Thus, we define the set $\mathcal{B} = \{A_1, \dots, A_5\}$ as the union of the matrices $\widehat{A}_i$ and $\widetilde{A}_j$. Experimentally, we assume that the final state $\boldsymbol{y}(T)$ is measured against a fixed state $\boldsymbol{y}^1$, meaning that is $C = (\boldsymbol{y}^1)^\top$. The vector $\boldsymbol{y}^1$ consists of zeros except for the second, fifth and 17th entries which are equal to one, while the initial state $\boldsymbol{y}^0$ is equal to one on the fourth, 13th and 49th entries.[2] For U_{ad} we use the box constraints $|u(t)| \leq 4$. We perform the same experiments as in Section 6.1 and 6.2 with the difference that GR and OGR are run with the same basis $\mathcal{B}$. The results are reported in Figure 6.3. All algorithms show essentially a similar performance.

[2] All matrices $\widehat{A}_i$, $\widetilde{A}_j$, $\widehat{B}_n$ and $\widetilde{B}_n$ and the vectors $\boldsymbol{y}^1$ and $\boldsymbol{y}^0$ are generated from the codes of [29].

Chapter 7

Reconstruction of parameter distributions in spin systems

In this chapter, we test our greedy algorithms in a well-recognized field of application: nuclear magnetic resonance (NMR) [42, 53, 55]. To control an ensemble of spin particles, it is crucial to have a good understanding of the involved Hamiltonian parameters, which may vary in a given range due to experimental limitations or uncertainties. In such cases, it is essential to identify the probability distribution of these parameters over the spin ensemble, i.e., the number of spins having a given value of the parameters. When the distribution is known or assumed to be of a simple form, such as Gaussian or Lorentzian, the identification is straightforward. However, it becomes significantly more challenging when the distribution is of a complex structure (e.g., with multiple peaks) or is completely unknown. We demonstrate that the control fields generated by our GR algorithms allow for an accurate reconstruction of the joint distribution of two Hamiltonian parameters, regardless of distribution's form.

This chapter is organized as follows. In Section 7.1, we describe the model system, while the GR algorithms for the setting of NMR are presented in Section 7.2. The analysis of the standard GR algorithm is provided in Section 7.3. Numerical results are presented in Section 7.4.

7.1 Model system

Consider a standard control problem in liquid-state NMR, i.e., a spin ensemble subjected to both inhomogeneous static and radio-frequency magnetic fields [52, 53, 66]. In a given rotating frame, each isochromat is characterized by a Bloch vector $\mathbf{M} = [M_x, M_y, M_z]^\top$, evolving in time according to the equations

$$\begin{cases} \dot{M}_x = -\omega M_y + (1+\alpha)\omega_y M_z, \\ \dot{M}_y = \omega M_x - (1+\alpha)\omega_x M_z, \\ \dot{M}_z = (1+\alpha)\omega_x M_y - (1+\alpha)\omega_y M_x. \end{cases}$$

Notice that the components of $\mathbf{M}$ satisfy $M_x^2 + M_y^2 + M_z^2 = M_0^2$, with M_0 the equilibrium magnetization. Here, ω_x and ω_y are time-dependent controls corresponding to the components of the radio-frequency magnetic field along the x- and y- directions. The parameters ω and α correspond to offset, i.e., to the inhomogeneity of the static magnetic field applied along the z- direction, and radio-frequency magnetic field inhomogeneities, respectively [55]. In standard experiments, we have $\frac{\omega}{2\pi} \in [-20, 20]$ Hz and $\alpha \in [-0.2, 0.2]$. For the purpose of this work, we assume that the probability densities of ω and α are unknown. The controls $\frac{\omega_x}{2\pi}$ and $\frac{\omega_y}{2\pi}$ are expressed in Hz. We consider a typical field amplitude ω_0 that can be fixed, for instance, to $\omega_0 = 2\pi \times 100$ Hz. We introduce normalized coordinates as

$$u_1 = 2\pi\frac{\omega_x}{\omega_0};\ u_2 = 2\pi\frac{\omega_y}{\omega_0};\ t' = \frac{\omega_0}{2\pi}t;\ \Delta = 2\pi\frac{\omega}{\omega_0}; \boldsymbol{y} = \frac{\mathbf{M}}{M_0}.$$

In what follows, we omit the prime to simplify the notations. We deduce that the differential system can be expressed in normalized units as

$$\begin{cases} \dot{y_1} = -\Delta y_2 + (1+\alpha)u_2 y_3 \\ \dot{y_2} = \Delta y_1 - (1+\alpha)u_1 y_3 \\ \dot{y_3} = (1+\alpha)u_1 y_2 - (1+\alpha)u_2 y_1 \end{cases} \tag{7.1}$$

with $y_1^2 + y_2^2 + y_3^2 = 1$. The initial state of the dynamics for each spin is the thermal equilibrium point, i.e., $\boldsymbol{y}^0 = [0, 0, 1]^\top$. Writing system (7.1) in our usual formulation, we have

$$\dot{\boldsymbol{y}}(t) = \Big[\Delta A + (1+\alpha)(u_1(t)B_1 + u_2(t)B_2)\Big]\boldsymbol{y}(t), \quad t \in (0, T], \quad \boldsymbol{y}(0) = \boldsymbol{y}^0, \tag{7.2}$$

where

$$A = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad B_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}, \qquad B_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}.$$

We consider a control time of the order of 100 ms, that corresponds to a normalized time T of the order of 10. The range of variation of the parameter Δ is $\Delta_0 + 2\pi[-0.2, 0.2]$, where Δ_0 is a frequency value that can be used to shift arbitrarily the interval. For the purpose of this work, we assume that $\Delta_0 \geq 0.4\pi$, meaning that $\Delta \geq 0$.

The goal is to estimate simultaneously the distributions for the parameters Δ and α by designing specific controls $\boldsymbol{u} = [u_1, u_2]^\top$. We consider an ensemble of N spins whose dynamics are governed by (7.2). We assume that the control amplitudes u_1, u_2 belong to the admissible set $U_{ad} = \{\boldsymbol{u} \in \mathbb{R}^2 \mid |u_1| \leq u_m, |u_2| \leq u_m\}$, where u_m is the maximum amplitude of each component. A simple way to proceed can be described as follows. We consider that the system of N spins is divided into K_Δ groups, and we associate with the ℓ-th subgroup a certain value Δ_ℓ and the corresponding probability $P_\star^\Delta(\ell) = \frac{N_{\Delta,\ell}}{N}$, $\ell = 1, \ldots, K_\Delta$, with $\sum_{\ell=1}^{K_\Delta} P_\star^\Delta(\ell) = 1$. The probability $P_\star^\Delta(\ell)$ is unknown, which means that the number of elements $N_{\Delta,\ell}$ of each group is to be found. Similarly, for the parameter α, we have K_α groups with the probabilities $P_\star^\alpha(\ell) = \frac{N_{\alpha,\ell}}{N}$, $\ell = 1, \ldots, K_\alpha$, with $\sum_{\ell=1}^{K_\alpha} P_\star^\alpha(\ell) = 1$ to estimate.

This problem leads to the reconstruction of two independent discrete distributions. However, this approach has two main drawbacks. First, the two random variables Δ and α are assumed to be independent. This is a limitation when trying to reconstruct the two unknown distributions, since any possible correlation is a priori neglected. Second, the online reconstruction problem is nonlinear, since the product of the two distributions would appear. For these reasons, rather than considering two independent distributions, we work directly with the joint distribution, i.e., the system of N spins is divided into K groups and we associate to each subgroup a pair $(\Delta, \alpha)_\ell$ and the corresponding joint probability $P_\star(\ell) = \frac{N_\ell}{N}$, $\ell = 1, \dots, K$, with $\sum_{\ell=1}^{K} P_\star(\ell) = 1$. Now, the joint probability $P_\star(\ell)$ is unknown, namely the number of elements N_ℓ affected by the pairs $(\Delta, \alpha)_\ell$. This approach has the advantage of taking into account correlation effects and the online reconstruction problem remains quadratic. It should be noted that these are acquired at the cost of an increase in the dimension of the unknown object(s), i.e., from two one-dimensional functions to a two-dimensional function. Finally, we point out that two independent distributions can also be treated as a specific case of joint distributions.

Since we are dealing with an inverse problem, we need to define what quantities can be observed in an experimental setting. In NMR, only the first two coordinates of the magnetization vector can be directly measured. We do not have accessed directly to the z component due to the strong constant magnetic field applied along this direction [55]. We denote by

$$\boldsymbol{z}((\Delta, \alpha), \boldsymbol{u}; t) = [y_1(t), y_2(t)]^\top$$

the projection of the Bloch vector onto the first two coordinates. Here, the dependence on (Δ, α) and $\boldsymbol{u}$ has been explicitly mentioned. The corresponding experimental realization of this controlled dynamic is obtained at $t = T$ and leads to

$$\boldsymbol{\varphi}_\star^{data}(\boldsymbol{u}, T) = [y_1^{\text{data}}(\boldsymbol{u}; T), y_2^{\text{data}}(\boldsymbol{u}; T)]^\top,$$

where $\boldsymbol{\varphi}_\star^{data}(\boldsymbol{u}, T)$ can be viewed as the average at time T of the experimental measures of all the spins of the set subjected to the control $\boldsymbol{u}$. The coordinates y_1^{data} and y_2^{data} are those of this measured magnetization vector.

The relation between the theoretical description of the dynamical system to the experimental outcome can be expressed as

$$\boldsymbol{\varphi}_\star^{data}(\boldsymbol{u}, T) = \sum_{\ell=1}^{K} P_\star(\ell) \boldsymbol{z}((\Delta, \alpha)_\ell, \boldsymbol{u}; T). \tag{7.3}$$

In general, one control protocol is not sufficient to obtain an appropriate reconstruction of the unknown $P_\star$, but a set of $\widetilde{K}$ control processes with $\widetilde{K}$ different control functions denoted $\boldsymbol{u}^m$, $m = 1, \cdots, \widetilde{K}$, needs to be used. On the basis of the experimental outputs, a straightforward way to determine $P_\star$ is to solve the minimization problem

$$\min_{P \in \mathbb{P}} \sum_{m=1}^{K} \| \sum_{\ell=1}^{K} P(\ell) \boldsymbol{z}((\Delta, \alpha)_\ell, \boldsymbol{u}^m; T) - \boldsymbol{\varphi}_\star^{data}(\boldsymbol{u}^m, T) \|_2^2, \tag{7.4}$$

where $\| \cdot \|$ denotes the standard Euclidean vector norm, and $\mathbb{P}$ is the convex and closed set of all probability distributions P that satisfy $P(\ell) \geq 0$ for $1 \leq \ell \leq K$ and $\sum_{\ell=1}^{K} P(\ell) = 1$.

At this point, it is clear that a key ingredient of the accuracy of the reconstruction process rests on the choice of a set of $\widetilde{K}$ controls $\boldsymbol{u}^k$. The identification of the number $\widetilde{K}$ of control functions is a difficult task. The theoretical analysis presented in Section 7.3 shows that the choice $\widetilde{K} = K$ is sufficient. GR computes exactly $\widetilde{K} = K$ control fields. However, we will show that OGR is capable of reducing (halving) the number $\widetilde{K}$ of control fields while guaranteeing an accurate reconstruction.

Let us now rewrite (7.4) in the form (2.3). We introduce a set $\mathcal{B} := \{\phi_j\}_{j=1}^K$ of linearly independent functions $\phi_j : \{1, \dots, K\} \to \mathbb{R}$ such that $\mathbb{P} \subset \operatorname{span}(\Phi)$, where span denotes the vector space generated by the functions. We also express P as $P(\ell) = \sum_{j=1}^K \beta_j \phi_j(\ell)$ and define

$$\boldsymbol{\varphi}(\phi(\boldsymbol{\beta}), \boldsymbol{u}; t) := \sum_{\ell=1}^{K} \sum_{j=1}^{K} \beta_j \phi_j(\ell) \boldsymbol{z}((\Delta, \alpha)_\ell, \boldsymbol{u}; t), \tag{7.5}$$

where the vector $\boldsymbol{\beta} = (\beta_j)_{j=1}^K$ is taken in $\widehat{\mathbb{R}}^K$, a subset of $\mathbb{R}^K$, so that $P = \sum_j \beta_j \phi_j$ is a probability distribution. Now, the minimization problem (7.4) becomes

$$\min_{\boldsymbol{\beta} \in \widehat{\mathbb{R}}^K} \sum_{m=1}^{K} \|\boldsymbol{\varphi}(\phi(\boldsymbol{\beta}), \boldsymbol{u}^m; T) - \boldsymbol{\varphi}_\star^{data}(\boldsymbol{u}^m, T)\|_2^2. \tag{7.6}$$

Notice that the main difference to the reconstruction problem (2.3) is that here, the unknown operator is not contained in the differential equation itself, but is rather part of the observable. In fact, by writing

$$\boldsymbol{Z}(\boldsymbol{u}; T) = \begin{bmatrix} \boldsymbol{z}((\Delta, \alpha)_1, \boldsymbol{u}; T) \\ \vdots \\ \boldsymbol{z}((\Delta, \alpha)_K, \boldsymbol{u}; T) \end{bmatrix} \in \mathbb{R}^{2K}$$

and

$$C_\star = \begin{bmatrix} P_\star(1) & 0 & \cdots & P_\star(K) & 0 \\ 0 & P_\star(1) & \cdots & 0 & P_\star(K) \end{bmatrix} \in \mathbb{R}^{2 \times 2K} \tag{7.7}$$

we obtain $\boldsymbol{\varphi}_\star^{data}(\boldsymbol{u}, T) = C_\star \boldsymbol{Z}(\boldsymbol{u}; T)$.

Finally, we point out that for the specific setting in this chapter, we consider the duration of each control pulse as a variable to be optimized together with its amplitude. In particular, we assume that the controls are constant in time, i.e., $\boldsymbol{u}(t) \equiv \boldsymbol{u} \in \mathbb{R}^2$, and that we can freely choose the control time up to a fixed maximum value T. Since the initial state is an equilibrium point, this is equivalent to turning on the control at a time $t \geq 0$. However, to be consistent with later notation, we rewrite the online reconstruction problem (7.6) as

$$\min_{\boldsymbol{\beta} \in \widehat{\mathbb{R}}^K} \sum_{m=1}^{K} \|\boldsymbol{\varphi}(\phi(\boldsymbol{\beta}), \boldsymbol{u}^m; t^m) - \boldsymbol{\varphi}_\star^{data}(\boldsymbol{u}^m, t^m)\|_2^2. \tag{7.8}$$

We show in Section 7.3 that the hypotheses are sufficient for the different examples to identify the probability distributions. The generality of GR allows one to tackle this situation in a straightforward manner.

Algorithm 5 Greedy Reconstruction Algorithm For Parameter Distributions (GRPD)

Require: A set of K linearly independent functions $\mathcal{B} = \{\phi_1, \dots, \phi_K\}$.

1: Compute the control $\boldsymbol{u}^1$ and the control time t^1 by solving

$$\max_{\substack{\boldsymbol{u}\in U_{ad}\\ t\in[0,T]}} \|\boldsymbol{\varphi}(\phi_1, \boldsymbol{u}, t)\|_2^2. \tag{7.9}$$

2: **while** $k \leq K-1$ **do**

3: Fitting step: Find $\boldsymbol{\beta}^k = (\beta_j^k)_{j=1,\dots,k}$ that solves

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^k} \sum_{m=1}^{k} \|\boldsymbol{\varphi}(\phi^{(k)}(\boldsymbol{\beta}), \boldsymbol{u}^m, t^m) - \boldsymbol{\varphi}(\phi_{k+1}, \boldsymbol{u}^m, t^m)\|_2^2, \tag{7.10}$$

where $\phi^{(k)}(\boldsymbol{\beta}) := \sum_{j=1}^{k} \beta_j \phi_j$.

4: Splitting step: Find $\boldsymbol{u}^{k+1}$ and t^{k+1} that solves

$$\max_{\substack{\boldsymbol{u}\in U_{ad}\\ t\in[0,T]}} \|\boldsymbol{\varphi}(\phi^{(k)}(\boldsymbol{\beta}^k), \boldsymbol{u}, t) - \boldsymbol{\varphi}(\phi_{k+1}, \boldsymbol{u}, t)\|_2^2. \tag{7.11}$$

5: Update $k \leftarrow k+1$.

6: **end while**

7.2 Greedy reconstruction algorithms for parameter distributions

We present in this section adapted versions of the GR Algorithm 1 and the OGR Algorithm 2 in order to simultaneously compute controls $\boldsymbol{u}^m$ and corresponding control times t^m.

Let us begin by stating the adapted version of GR for the reconstruction of parameter distributions (GRPD) in Algorithm 5. As we can see, the necessary adaptations are very straightforward. The initialization and splitting step simply maximize simultaneously over the control $\boldsymbol{u}$ and the control time t. The resulting control-time pairs $\{(\boldsymbol{u}^m, t^m)\}_{m=1}^k$ are then used together in the fitting step to compute the vectors $\boldsymbol{\varphi}(\phi^{(k)}(\boldsymbol{\beta}), \boldsymbol{u}^m, t^m)$ and $\boldsymbol{\varphi}(\phi_{k+1}, \boldsymbol{u}^m, t^m)$. Notice that the fitting step minimizes over the full space $\mathbb{R}^k$, meaning that $\sum_j \beta_j \phi_j$ does not have to be a probability distribution. However, this is a restrictive condition. On the contrary, it allows the algorithm to find and correct more nontrivial kernels than might be necessary.

Analogously, one can formulate an adapted version of OGR for the reconstruction of parameter distributions (OGRPD). However, we omit stating the full algorithm here for brevity.

7.3 Analysis

In this section, we prove that the controls generated by GRPD and OGRPD make possible the reconstruction of the unknown probability distributions of the parameters Δ and α,

i.e., they make the online reconstruction problem (7.4) uniquely solvable.

We start by recalling that problem (7.4) is equivalent to (7.8). Assuming that $P_\star$ can be written as $P_\star(\ell) = \sum_{j=1}^K \beta_{\star,j}\phi_j(\ell)$, we can write equation (7.8) in a compact form as follows:

$$\min_{\boldsymbol{\beta}\in\widehat{\mathbb{R}}^K} \langle \boldsymbol{\beta} - \boldsymbol{\beta}_\star, \widehat{W}(\boldsymbol{\beta} - \boldsymbol{\beta}_\star)\rangle, \tag{7.12}$$

where $\widehat{W} := \sum_m W(\boldsymbol{u}^m, t^m)$ is the sum of symmetric and PSD $K \times K$- matrices whose elements are defined as

$$[W(\boldsymbol{u}, t)]_{i,j} := \langle \boldsymbol{\gamma}_i(\boldsymbol{u}, t), \boldsymbol{\gamma}_j(\boldsymbol{u}, t)\rangle, \quad \text{for } i, j = 1, \dots, K \tag{7.13}$$

with

$$\boldsymbol{\gamma}_j(\boldsymbol{u}, t) := \boldsymbol{\varphi}(\phi_j, \boldsymbol{u}, ; t) = \sum_\ell \phi_j(\ell)\boldsymbol{z}((\Delta, \alpha)_\ell, \boldsymbol{u}; t), \quad \text{for } j = 1, \dots, K. \tag{7.14}$$

Since the set of vectors $\widehat{\mathbb{R}}^K$ is a convex subset of $\mathbb{R}^K$, we deduce that the problem is uniquely solvable if the matrix $\widehat{W}$ is PD. In the case $\widehat{W}$ has a non-trivial kernel, infinitely many solutions may exist which lead to wrong probability distributions different from the experimental one $P_\star$. We stress that the non-triviality of the kernel depends completely on the choice of the controls $\boldsymbol{u}^m$ and the corresponding control times t^m.

Using the notation (7.13)-(7.14), we can now also rewrite the sub-problems of GRPD in terms of the matrix $\widehat{W}$. The initialization problem (7.9) can be written as

$$\max_{\substack{\boldsymbol{u}\in U_{ad}\\ t\in[0,T]}} |[W(\boldsymbol{u}, t)]_{1,1}|^2. \tag{7.15}$$

The fitting step problem (7.10) is equivalent to

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^k} \langle \mathbf{v}_{\boldsymbol{\beta}}, [\widehat{W}^{(k)}]_{[1:k+1,1:k+1]}\mathbf{v}_{\boldsymbol{\beta}}\rangle, \tag{7.16}$$

where $\widehat{W}^{(k)} := \sum_{m=1}^k W(\boldsymbol{u}^m, t^m)$ and $\mathbf{v}_{\boldsymbol{\beta}} := [\boldsymbol{\beta}^\top, -1]^\top$. Finally, the splitting step problem (7.11) can be written as

$$\max_{\substack{\boldsymbol{u}\in U_{ad}\\ t\in[0,T]}} \langle \mathbf{v}_{\boldsymbol{\beta}^k}, [W(\boldsymbol{u}, t)]_{[1:k+1,1:k+1]}\mathbf{v}_{\boldsymbol{\beta}^k}\rangle. \tag{7.17}$$

For a detailed interpretation of these reformulated problems see Section 3.1.1. We summarize the relevant arguments in the following theorem.

Theorem 7.3.1 *Let $\{(\boldsymbol{u}^m, t^m)\}_{m=1}^K$ be a set of controls and corresponding control times generated by GRPD, such that $[W(\boldsymbol{u}_1, t_1)]_{1,1} > 0$. Let $\boldsymbol{\beta}^k$ be the solution to the fitting step problem (7.16) for $k = 1, \dots, K-1$, such that the vectors $\boldsymbol{v}_{\boldsymbol{\beta}^k} = [(\boldsymbol{\beta}^k)^\top, -1]^\top$ are not in the kernel of $[W(\boldsymbol{u}^{k+1}, t^{k+1})]_{[1:k+1,1:k+1]}$. Then the matrix $\widehat{W} = \sum_m W(\boldsymbol{u}^m, t^m)$ is PD.*

It remains to show that the splitting step can always find a control such that the vector $\mathbf{v}_{\boldsymbol{\beta}^k}$ is not in the kernel of $[W(\boldsymbol{u},t)]_{[1:k+1,1:k+1]}$. In fact, it is sufficient to show that for any $k \in \{1,\dots,K\}$ there exists a control $\boldsymbol{u} \in U_{ad}$ and a $t \in [0,T]$ such that

$$\langle \mathbf{v}_{\boldsymbol{\beta}^k}, [W(\boldsymbol{u},t)]_{[1:k+1,1:k+1]} \mathbf{v}_{\boldsymbol{\beta}^k} \rangle > 0.$$

We show in Theorem 7.3.2 that this is valid in the current context.

Theorem 7.3.2 *Let $k \in \{1,\dots,K-1\}$, $\widehat{W}^{(k)}_{[1:k,1:k]}$ be PD, $\boldsymbol{\beta}^k$ the solution to the fitting-step problem* (7.16)*, and $\boldsymbol{v}_{\boldsymbol{\beta}^k} = [(\boldsymbol{\beta}^k)^\top, -1]^\top$. Then any solution $(\boldsymbol{u},t)$ to the splitting-step problem* (7.17) *satisfies*

$$\langle \boldsymbol{v}_{\boldsymbol{\beta}^k}, W_{[1:k+1,1:k+1]}(\boldsymbol{u},t)\boldsymbol{v}_{\boldsymbol{\beta}^k} \rangle = \|\boldsymbol{\varphi}(\phi^{(k)}(\boldsymbol{\beta}^k),\boldsymbol{u},t) - \boldsymbol{\varphi}(\phi_{k+1},\boldsymbol{u},t)\|_2^2 > 0.$$

Proof. For brevity, we identify α with $1+\alpha$ for the remainder of this proof. We start by writing

$$\boldsymbol{\varphi}(\phi^{(k)}(\boldsymbol{\beta}^k),\boldsymbol{u},t) - \boldsymbol{\varphi}(\phi_{k+1},\boldsymbol{u},t) = \sum_{\ell=1}^{K} \left(\sum_{j=1}^{k} \beta_j^k \phi_j(\ell) - \phi_{k+1}(\ell) \right) \boldsymbol{z}((\Delta,\alpha)_\ell, \boldsymbol{u}; t).$$

Since the functions $\{\phi_1,\dots,\phi_K\}$ are linearly independent, it holds that

$$\exists \widetilde{\ell} \in \{1,\dots,K\}: \quad \phi_{diff}(\widetilde{\ell}) := \sum_{j=1}^{k} \beta_j^k \phi_j(\widetilde{\ell}) - \phi_{k+1}(\widetilde{\ell}) \neq 0. \tag{7.18}$$

Now, consider the control $\widetilde{\boldsymbol{u}} := [0,b]^\top$ and a corresponding control time $\widetilde{t} \in [0,T]$, where both $b \in \mathbb{R} \setminus \{0\}$ and $\widetilde{t}$ are to be chosen later. We have

$$\boldsymbol{z}((\Delta,\alpha),\boldsymbol{u};t) = C\boldsymbol{y}((\Delta,\alpha),\boldsymbol{u};t),$$

where $\boldsymbol{y}((\Delta,\alpha),\boldsymbol{u};t)$ is the solution to (7.2) and $C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$. Thus, we obtain

$$\boldsymbol{z}((\Delta,\alpha)_\ell, \widetilde{\boldsymbol{u}}; \widetilde{t}) = Ce^{\widetilde{t}(\Delta_\ell A + \alpha_\ell b B_2)}\boldsymbol{y}^0.$$

Since $\Delta_\ell A + \alpha_\ell b B_2$ is skew-symmetric, we can compute its exponential matrix explicitly. By setting $\widetilde{A} := \widetilde{t}(\Delta_\ell A + \alpha_\ell b B_2)$ and $x_\ell := \sqrt{\Delta_\ell^2 + \alpha_\ell^2 b^2}$, we have $e^{\widetilde{A}} = I_3 + \frac{\sin(\widetilde{t}x)}{\widetilde{t}x}\widetilde{A} + \frac{1-\cos(\widetilde{t}x_\ell)}{\widetilde{t}^2 x_\ell^2}\widetilde{A}^2$ (see, e.g., [63]). Since $CI_3\boldsymbol{y}^0 = 0$ and

$$\widetilde{A}^2 = \widetilde{t}^2 \begin{bmatrix} -\Delta_\ell^2 - \alpha_\ell b^2 & 0 & 0 \\ 0 & -\Delta_\ell^2 & \Delta_\ell \alpha_\ell b \\ 0 & \Delta_\ell \alpha_\ell b & -\alpha_\ell^2 b^2 \end{bmatrix},$$

we obtain

$$C\boldsymbol{y}((\Delta,\alpha)_\ell, \widetilde{\boldsymbol{u}}; \widetilde{t}) = C\left(\frac{\sin(\widetilde{t}x_\ell)}{x_\ell} \begin{bmatrix} \alpha_\ell b \\ 0 \\ 0 \end{bmatrix} + \frac{1-\cos(\widetilde{t}x_\ell)}{x_\ell^2} \begin{bmatrix} 0 \\ \Delta_\ell \alpha_\ell b \\ -\alpha_\ell^2 b^2 \end{bmatrix} \right) = \begin{bmatrix} \frac{\sin(\widetilde{t}x_\ell)}{x_\ell}\alpha_\ell b \\ \frac{1-\cos(\widetilde{t}x_\ell)}{x_\ell^2}\Delta_\ell \alpha_\ell b \end{bmatrix}.$$

Thus, we have

$$\boldsymbol{\varphi}(\phi^{(k)}(\boldsymbol{\beta}^k),\widetilde{\boldsymbol{u}},\widetilde{t})-\boldsymbol{\varphi}(\phi_{k+1},\widetilde{\boldsymbol{u}},\widetilde{t})=b\sum_{\ell=1}^{K}\phi_{diff}(\ell)\alpha_\ell\begin{bmatrix}\frac{\sin(\widetilde{t}x_\ell)}{x_\ell}\\ \Delta_\ell\frac{1-\cos(\widetilde{t}x_\ell)}{x_\ell^2}\end{bmatrix}=:F(\widetilde{t}).$$

Seeking a contradiction, we now assume that $\boldsymbol{\varphi}(\phi^{(k)}(\boldsymbol{\beta}^k),\widetilde{\boldsymbol{u}},\widetilde{t})-\boldsymbol{\varphi}(\phi_{k+1},\widetilde{\boldsymbol{u}},\widetilde{t})=0$ for all $\widetilde{t}\in[0,T]$ and all $b\in\mathbb{R}\setminus\{0\}$. Since F is analytic in $\widetilde{t}$, we obtain $F^{(k)}(\widetilde{t})=0$ for all $k\in\mathbb{N}$ and all $\widetilde{t}\in[0,T]$. For k odd, we have

$$F^{(k)}(\widetilde{t})=b\sum_{\ell=1}^{K}\phi_{diff}(\ell)\alpha_\ell(-1^{\frac{k-1}{2}})\begin{bmatrix}x_\ell^{k-1}\cos(\widetilde{t}x_\ell)\\ \Delta_\ell x_\ell^{k-2}\sin(\widetilde{t}x_\ell))\end{bmatrix}. \tag{7.19}$$

Since $F^{(k)}(T)=0$ for all k odd and $b\neq 0$, the first component of $F^{(k)}(T)$ in (7.19), for different k odd, implies that

$$\underbrace{\begin{bmatrix}1 & 1 & \cdots & 1\\ x_1^2 & x_2^2 & \cdots & x_K^2\\ x_1^4 & x_2^4 & \cdots & x_K^4\\ \vdots & \vdots & \vdots & \vdots\\ x_1^{\widetilde{K}} & x_2^{\widetilde{K}} & \cdots & x_K^{\widetilde{K}}\end{bmatrix}}_{=:D}\underbrace{\begin{bmatrix}\phi_{diff}(1)\alpha_1\cos(Tx_1)\\ \phi_{diff}(2)\alpha_2\cos(Tx_2)\\ \vdots\\ \phi_{diff}(K)\alpha_K\cos(Tx_K)\end{bmatrix}}_{=:\boldsymbol{\phi}_{\widetilde{t}}}=0.$$

Notice that D is a Vandermonde matrix (see, e.g., [56]). Now, let $\widetilde{K}=2(K-1)$, meaning that $D\in\mathbb{R}^{(\frac{\widetilde{K}}{2}+1)\times K}$ is a square matrix. Then, the determinant of D is given exactly by

$$\det(D)=\prod_{1\leq i<j\leq K}(x_j^2-x_i^2).$$

This implies that two rows of D are linearly independent if and only if $|x_i|\neq|x_j|$. Hence, $\det(D_x)=\det(D_y)\neq 0$ (and therefore $\boldsymbol{\phi}_{\widetilde{t}}=0$) if and only if $|x_i|\neq|x_j|$ for $i\neq j$. Recalling that $x_\ell=\sqrt{\Delta_\ell^2+\alpha_\ell^2b^2}$, $|x_i|\neq|x_j|$ is equivalent to $\Delta_i^2+\alpha_i^2b^2\neq\Delta_j^2+\alpha_j^2b^2$. For $i\neq j$ we also have $\alpha_i\neq\alpha_j$ and/or $\Delta_i\neq\Delta_j$ by definition. Since $\alpha_\ell\in[0.8,1.2]$ and $\Delta_\ell\in\Delta_0+2\pi[-0.2,0.2]$ with $\Delta_0\geq 0.4\pi$, we obtain $\alpha_i^2\neq\alpha_j^2$ and/or $\Delta_i^2\neq\Delta_j^2$ for $i\neq j$. Thus, there exists $b\in\mathbb{R}\setminus 0$ such that $\Delta_i^2+\alpha_i^2b^2\neq\Delta_j^2+\alpha_j^2b^2$ for all $i,j\in\{1,\ldots,K\}$ with $i\neq j$. In conclusion, we have $|x_i|\neq|x_j|$ for $i\neq j$, which implies that $\boldsymbol{\phi}_{\widetilde{t}}=0$ and therefore $\phi_{diff}(\ell)\alpha_\ell\cos(\widetilde{t}x_\ell)=0$ for all $\ell\in\{0,\ldots,K\}$ and all $\widetilde{t}\in[0,T]$. However, we also have $\phi_{diff}(\widetilde{\ell})\neq 0$ by (7.18), $\alpha_{\widetilde{\ell}}>0$ and $x_{\widetilde{\ell}}>0$. Thus, there exists $\widetilde{t}\in[0,T]$ such that $\phi_{diff}(\widetilde{\ell})\alpha_{\widetilde{\ell}}\cos(\widetilde{t}x_{\widetilde{\ell}})\neq 0$, which is a contradiction. □

Analogously to the proof of Theorem 7.3.2, one can show that any solution $(\boldsymbol{u}^1,t^1)$ to the initialization problem (7.15) satisfies $[W(\boldsymbol{u}^1,t^1)]_{1,1}>0$. We conclude our analysis by the following theorem.

Theorem 7.3.3 *Let $\{(\boldsymbol{u}^m,t^m)\}_{m=1}^K$ be a set of controls and corresponding control times generated by GRPD. Then problem* (7.8) *is uniquely solvable by $\boldsymbol{\beta}=\boldsymbol{\beta}_\star$.*

Proof. Let $\boldsymbol{\beta}^k$ be the solution to the fitting step problem (7.16) for $k = 1, \dots, K-1$. By Theorem 7.3.2, the vector $\mathbf{v}_{\boldsymbol{\beta}^k} = [(\boldsymbol{\beta}^k)^\top, -1]^\top$ is not in the kernel of $[W(\boldsymbol{u}^{k+1}, t^{k+1})]_{[1:k+1,1:k+1]}$ for all $k \in \{1, \dots, K-1\}$. Thus, we obtain by Theorem 7.3.1 that the matrix $\widehat{W} = \sum_m W(\boldsymbol{u}^m, t^m)$ is PD. Hence, problem (7.12) is uniquely solvable by $\boldsymbol{\beta} = \boldsymbol{\beta}_\star$. By equivalency of problems (7.12) and (7.8), we obtain the result. □

Notice that, in the notation above, OGRPD simply reorders rows and columns of the matrix $\widehat{W}^{(k)}$ while attempting to find and correct its kernel. In fact, the second improvement (lines 8-9 in OGR Algoithm 2) skips the splitting step only if there exists a row and column of $\widehat{W}^{(k)}$ with index ℓ_{k+1} such that, by swapping ϕ_{k+1} and $\phi_{\ell_{k+1}}$, the matrix $\widehat{W}^{(k)}_{[1:k+1,1:k+1]}$ is PD. Thus, if tol_{split} is sufficiently small, one can also prove convergence of OGRPD analogously to GRPD.

7.4 Numerical experiments

We test GRPD and OGRPD on two different variations of the setting described in Section 7.1. First, in Section 7.4.1, we consider the special case where the offset Δ is known, and only the distribution of the unknown parameter α is reconstructed. Afterwards, in Section 7.4.2, we reconstruct the full joint distributions for the parameters Δ and α.

All results in this section were produced using the `SPIRED` code.[1]

7.4.1 Reconstruction of single parameter distributions

Let us assume that the parameter Δ is known to be equal to $\pi/10$ for all particles. In this case, we only want to identify the distribution of the parameter α. We choose a maximum control time of $T = 16$. We consider two different probability distributions $P_\star$ for α, a symmetric double peak (see panel on the left in Figure 7.1) and a non-continuous step distribution (see panel on the left in Figure 7.2). They are discretized by a uniform grid of 30 points. Similarly, we discretize the set of linearly independent functions $\{\phi_j\}_{j=1}^K$ by setting $K = 30$ and $\phi_j = \boldsymbol{e}_j \in \mathbb{R}^{30}$ the j-th canonical vector in $\mathbb{R}^{30}$. Finally, we fix the tolerances for OGRPD to be $\mathrm{tol}_{split} = 10^{-14}$ and $\mathrm{tol}_{fit} = 10^{-4}$.

Now, let us briefly discuss how we solve the sub-steps of the algorithms numerically. The initialization and splitting step problems are solved by a second-order trust-region method. For the fitting step, we use the equivalent compact form (7.16). The corresponding first-order optimality system is given by

$$[\widehat{W}^{(k)}]_{[1:k,1:k]}\boldsymbol{\beta} = [\widehat{W}^{(k)}]_{[1:k,k+1]}. \tag{7.20}$$

Since the matrix $[\widehat{W}^{(k)}]_{[1:k,1:k]}$ is symmetric and PD, any solution to (7.20) is a global solution to (7.16). Hence, we solve the fitting step problem by solving the linear system (7.20) using a solver based on the SVD. This solver first computes the SVD of $[\widehat{W}^{(k)}]_{[1:k,1:k]}$,

[1] This code was published as part of [3] in the CPC Library: `https://doi.org/10.17632/6fsmzp6srg.1`

i.e., two orthogonal matrices $U, V \in \mathbb{R}^{k\times k}$ and a diagonal matrix $\Sigma \in \mathbb{R}^{k\times k}$ such that $U\Sigma V^\top = [\widehat{W}^{(k)}]_{[1:k,1:k]}$. To make the method more robust against numerical instabilities, it then removes all singular values that are smaller than a given tolerance, and the corresponding columns of U and V. Finally, it computes $\boldsymbol{\beta}$ by setting $\widetilde{\boldsymbol{\beta}} = V^\top \boldsymbol{\beta}$ and solving $\Sigma\widetilde{\boldsymbol{\beta}} = U^\top [\widehat{W}^{(k)}]_{[1:k,k+1]}$.

After running the algorithms, we reconstruct $P_\star$ by solving problem (7.8). Notice that, using the notation (7.13)-(7.14), the gradient of the cost function in (7.8) is given by $\widehat{W}\boldsymbol{\beta} - \sum_m \Gamma(\boldsymbol{u}^m, t^m)^\top \boldsymbol{\varphi}_\star^{data}(\boldsymbol{u}^m, t^m)$, where the columns of Γ are given by the $\boldsymbol{\gamma}_j(\boldsymbol{u}^m, t^m)$ defined in (7.14). We can also immediately see that the Hessian of the cost function in (7.8) is exactly $\widehat{W}$, which is guaranteed to be PD by our analysis in Section 7.3. Hence, the global solution to (7.8) is given by the (unique) solution to

$$\widehat{W}\boldsymbol{\beta} = \sum_m \Gamma(\boldsymbol{u}^m, t^m)^\top \boldsymbol{\varphi}_\star^{data}(\boldsymbol{u}^m, t^m). \tag{7.21}$$

However, in order to ensure that the coefficients of the computed solution correspond to a probability distribution (i.e., belong to $\widehat{\mathbb{R}}^K$), we add the necessary constraints and solve (7.8) with the second-order interior point algorithm of MATLAB's *fmincon*-solver. Nonetheless, the code includes an option to solve directly (7.21) using an SVD solver.

Now, we run both GRPD and OGRPD on the canonical set $\mathcal{B} = \{\phi_j\}_{j=1}^{30}$ of hat functions. In contrast to our experiments in Chapter 6, we do not include any additional random vectors in the canonical set for OGRPD and also do not remove any elements from the set during OGRPD (but still reorder them). The reason for this is that we experienced for the problem of this paper that additional random elements do not improve the results and removing elements from the canonical set does not reduce the number of controls, but is more likely to make the online reconstruction problem numerically unstable. While GRPD computes 30 controls, OGRPD only designs 15 by skipping 14 splitting steps. Recalling from (7.7) that $C_\star \in \mathbb{R}^{2\times 2K}$, we notice that OGRPD is able to reach the theoretical minimum number of controls established by Proposition 3.1.19.

We then choose $P_\star$ as the symmetric double peak distribution in Figure 7.1 (left) or the step distribution in Figure 7.2 (left) and compute the corresponding experimental realizations $\{\boldsymbol{\varphi}_\star^{data}(\boldsymbol{u}^m, t^m)\}_{m=1}^{\widetilde{K}}$ for the two resulting sets of control fields, with $\widetilde{K} = 30$ for GRPD and $\widetilde{K} = 15$ for OGRPD. Reconstructing $P_\star$ as described above, we obtain the coefficient vectors $\boldsymbol{\beta}_{rec}$ and thereby the distributions $P_{rec} = \sum_{j=1}^{30} \beta_{rec,j}\phi_j$ corresponding to GRPD and OGRPD.

The results are shown in Figures 7.1 and 7.2. The differences $P_\star - P_{rec}$ corresponding to the different distributions and control sets are plotted in Figures 7.3 and 7.4. Although OGRPD uses only half the amount of control functions (and therefore only half the amount of experimental data) compared to GRPD, both methods successfully reconstruct the two probability distributions of α.

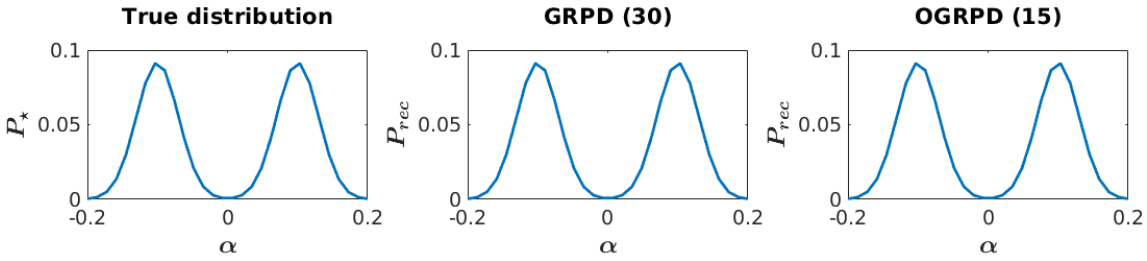

Figure 7.1: The plot on the left shows the true symmetric double peak probability distribution for $K = 30$ uniform mesh points. The plots in the middle and on the right contain the reconstructed distributions for the control sets generated by GRPD (containing 30 control fields) and OGRPD (containing 15 control fields).

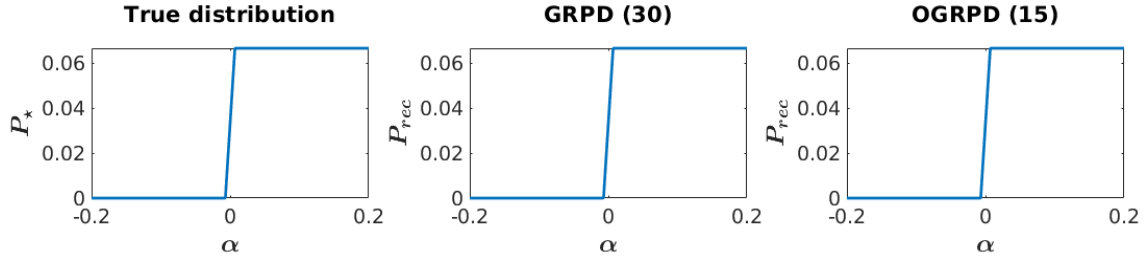

Figure 7.2: Same as Figure 7.1, but for a discontinuous step distribution.

7.4.2 Reconstruction of joint parameter distributions

Now, let us assume that the distributions of both parameters Δ and α are unknown. We choose again a maximum control time of 160 ms, which corresponds to a normalized time $T = 16$. The shift of the parameter Δ is set to $\Delta_0 = 4\pi$ and the width of its interval to $4\pi\Delta_1$, with $\Delta_1 = 0.2$. We consider two different joint probability distributions $P_\star$, a simple Gaussian one (see panel on the left in Figure 7.5) and a step distribution with three peaks (see panel on the left in Figure 7.7). They are discretized by a uniform mesh of 100 points (10 points in each direction). Similarly, we discretize the set of linearly independent functions $\{\phi_j\}_{j=1}^K$ by setting $K = 100$ and $\phi_j = \boldsymbol{e}_j \in \mathbb{R}^{100}$ the j-th canonical vector in $\mathbb{R}^{100}$. As in the previous section, we fix the tolerances for OGRPD to be $\text{tol}_{split} = 10^{-14}$ and $\text{tol}_{fit} = 10^{-4}$.

First, we choose $P_\star$ as the Gaussian distribution. Reconstructing $P_\star$ as described in Section 7.4.1, we obtain the coefficient vectors $\boldsymbol{\beta}_{rec}$ and thereby the distributions $P_{rec} = \sum_{j=1}^{100} \beta_{rec,j}\phi_j$ corresponding to GRPD and OGRPD, shown in Figure 7.5. Looking at the errors with respect to the true distribution $P_\star$ shown in Figure 7.6, we observe that OGRPD even outperforms GRPD by one order of magnitude. However, the difference is so small that it is not visible in the reconstructed distributions. Similar results are obtained for a step distribution with three peaks in Figure 7.7.

To investigate the dependence of the results on the choice of parameters, we repeat the

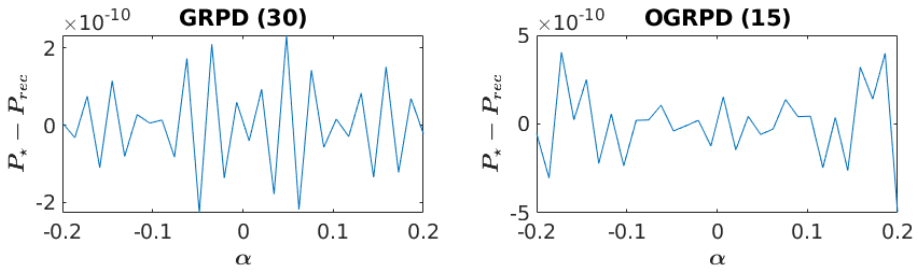

Figure 7.3: Difference between the true double peak probability distribution $P_\star$ (see left panel in Figure 7.1) and the distibutions P_{rec} reconstructed using the coontrol sets generated by GRPD (left) and OGRPD (right). In brackets are the number of control fields for each set.

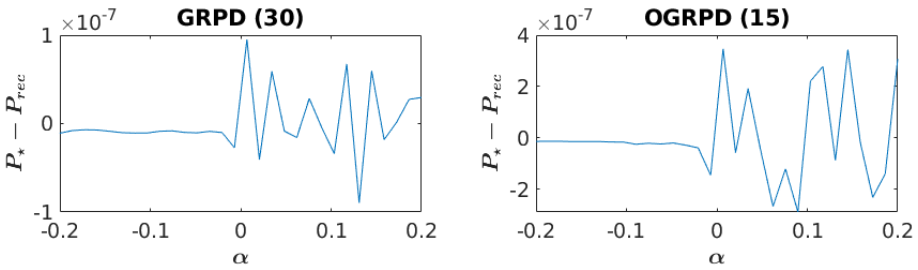

Figure 7.4: Same as Figure 7.3, but for the step distribution (see left panel in Figure 7.2).

experiment for different maximum control times, widths of the Δ-interval and $K = 400$ mesh points. First, we take a look at the number of control fields generated by OGRPD in Table 7.1. We observe that the number of generated control fields is increasing with decreasing maximum control time and decreasing width of the Δ-interval. We also observe that the ratio between the number of GRPD controls (which is equal to the number of mesh points K) and the number of OGRPD controls is decreasing with an increasing number of mesh points. To validate this point, we plot the number of controls for both algorithms, for different total numbers of mesh points in Figure 7.8.

An explanation of this behaviour is given by the condition number of the corresponding matrices W, defined in (7.13), representing the compact form (7.12) of the online reconstruction problem. The condition numbers corresponding to GRPD and OGRPD for the settings in Table 7.1 are shown in Tables 7.2 and 7.3. Based on our theoretical results for GRPD and OGRPD proving that $\widetilde{K} = K$ controls are sufficient, we also add a set of fully random controls (randomized within the given bounds u_m and T) that has the same number of controls as GRPD (i.e., $\widetilde{K} = 100$ and $\widetilde{K} = 400$, respectively). We observe that the condition number shows the same correlation with respect to the maximum control time, width of the Δ-interval and number of mesh points, as the number of OGRPD controls. In particular, the condition number of OGRPD is below $1e15$ for all settings where OGRPD computed less than 60% of the number of GRPD controls.

Regarding the condition numbers, GRPD and random controls show the same behaviour as OGRPD. The reason can be found by taking a closer look at the entries of the matrix

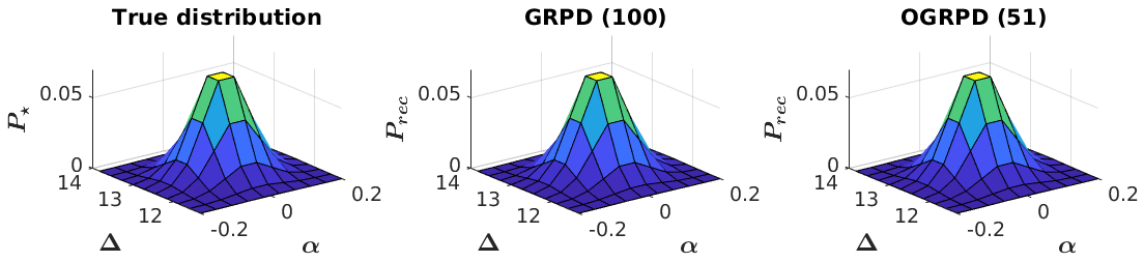

Figure 7.5: The plot on the left shows the true Gaussian probability distribution for $K = 100$ uniform mesh points. The plots in the middle and on the right contain the reconstructed distributions for the control sets generated by GRPD (containing 100 control fields) and OGRPD (containing 51 control fields).

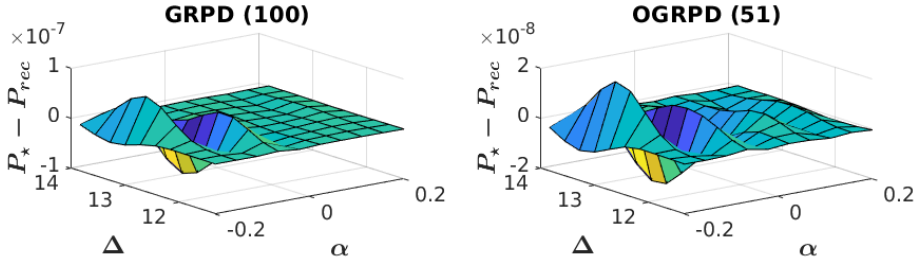

Figure 7.6: Same as Figure 7.3, but for the Gaussian distribution (see left panel in Figure 7.5).

W. It can be shown that the difference between two adjacent rows or columns of W is bounded in norm by u_m, T and the mesh size for the probability distribution, i.e., $\alpha_{\ell+1} - \alpha_\ell$ and $\Delta_{\ell+1} - \Delta_\ell$. We discuss this result in more detail in Section 7.5. We conclude that, if the control bound, the maximum control time, or the mesh size (or equivalently the width of the Δ-interval) is too small, the difference between two adjacent rows/columns of W can become numerically equal to zero, implying that W has a nontrivial kernel.

In order to investigate the impact of this numerical instability on the reconstructed results, we consider again the setting of the beginning of this section (i.e., $\Delta_1 = 0.2$ and $T = 16$), but for $K = 400$ mesh points. The results for a Gaussian and a step distribution with three peaks are plotted in Figures 7.9 and 7.10, respectively. We observe that all three control field sets are able to fully reconstruct the step distribution and, at least partially, the Gaussian distribution. This is because the admissible set of solutions for the online reconstruction problem (7.8) is restricted to $\widehat{\mathbb{R}}^K$. Thus, a bad condition number does not necessarily imply that it is impossible to (at least partially) reconstruct the true probability distribution. However, a good condition number guarantees stability of the numerical solver and improves the accuracy of the results. In this context, notice that if we would sufficiently increase either the control bound u_m, or the maximum control time T, both GRPD and OGRPD would show better condition numbers and be able to perfectly reconstruct also the Gaussian distribution in Figure 7.9.

We observe also that, if one knows the number of sufficient control functions $\widetilde{K} = K$,

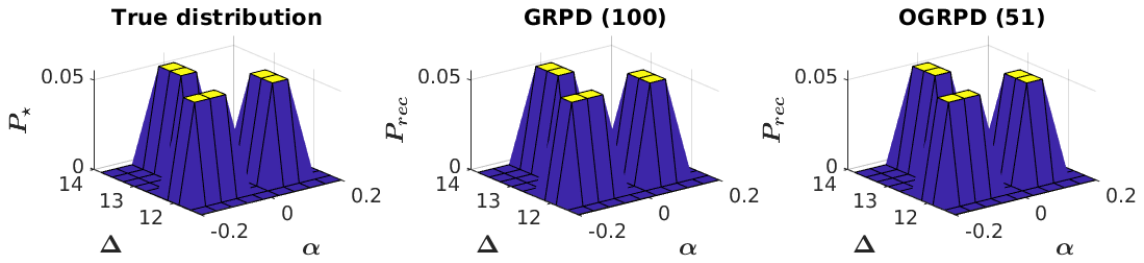

Figure 7.7: Same as Figure 7.5 but for a step distribution with three peaks. In brackets are the number of control fields for each set.

	$K = 100$				$K = 400$			
T \ Δ_1	8	16	24	32	8	16	24	32
0.1	70	**52**	**50**	**50**	259	299	276	240
0.2	**58**	**51**	**50**	**50**	305	293	**223**	**220**
0.4	**56**	**50**	**50**	**50**	326	256	**211**	**200**
0.8	**51**	**50**	**50**	**50**	292	**210**	**200**	**200**
1.6	**50**	**50**	**51**	**50**	275	**205**	**200**	**200**

Table 7.1: Number of controls computed by OGRPD for a control bound $u_m = 10$, and different numbers of discretization points K, maximum control times T and widths $4\pi\Delta_1$ of the Δ-interval. Bold numbers indicate that the number of OGRPD controls is less than 60% of the number of GRPD controls. Notice that GRPD always generates K controls.

then even completely random control fields can be able to perform similarly to GRPD and OGRPD controls. However, while OGRPD finds automatically $\widetilde{K}$ (reduces the number of control fields to a sufficient amount), there is no indicator for a sufficient amount of random controls in general. Additionally, the corresponding condition numbers are in many cases worse than for GRPD and OGRPD, as seen in Tables 7.2 and 7.3, meaning that they are more likely to show numerical instabilities. Thus, the recommended strategy is clearly OGRPD, since it is able to reduce the number of control fields by up to 50% (which, as we recall from Proposition 3.1.19, is the theoretical minimum), while accurately reconstructing the probability distributions.

Lastly, we remark that making the tolerance tol_{fit} smaller can generally lead to even fewer controls being computed by OGRPD. However, this in turn can lead to less accurate results in the reconstructed solution, meaning the user has to decide for themselves if such a trade-off is desirable.

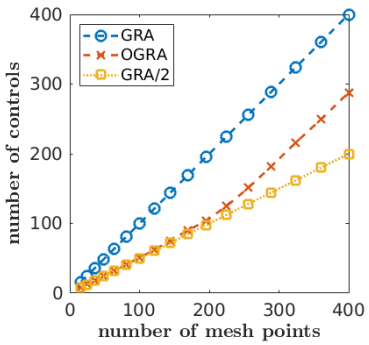

Figure 7.8: Number of controls for GRPD (dashed circles) and OGRPD (solid crosses) for different total numbers of mesh points. To highlight the ratio between the amount of controls, we also plot half the amount of GRPD controls (dotted squares).

	GRPD				OGRPD				random control set			
Δ_1 \ T	8	16	24	32	8	16	24	32	8	16	24	32
0.1	$3e16$	**5e07**	**7e03**	**6e03**	$3e15$	**3e08**	**7e05**	**7e06**	$1e18$	**1e09**	**1e06**	**3e06**
0.2	**7e09**	**4e06**	**3e03**	**1e03**	**5e09**	**1e08**	**1e08**	**1e06**	**7e12**	**1e09**	**4e04**	**9e03**
0.4	**1e11**	**5e03**	**1e03**	**1e03**	**2e11**	**3e06**	**1e07**	**1e07**	$1e16$	**7e04**	**9e03**	**1e03**
0.8	**1e06**	**2e03**	**1e03**	**1e03**	**3e07**	**8e05**	**2e05**	**2e05**	**7e10**	**3e03**	**2e03**	**1e03**
1.6	**2e04**	**9e02**	**1e03**	**8e02**	**2e06**	**4e07**	**1e05**	**7e07**	**5e09**	**1e04**	**2e03**	**1e03**

Table 7.2: Condition number of W for different control sets, maximum control times T and widths $4\pi\Delta_1$ of the Δ-interval. The total number of mesh points is $K = 100$ and the bound on the control is $u_m = 10$. Bold numbers indicate that the condition number is smaller than $1e15$.

7.5 Numerical stability

Let us take a closer look on the numerical stability of the matrix $\widehat{W}$. Notice that for constant controls the solution to the dynamical system (7.2) can be written as

$$\boldsymbol{y}((\Delta,\alpha),\boldsymbol{u};t) = e^{t(\Delta A+\alpha(u_1B_1+u_2B_2))}\boldsymbol{y}^0.$$

In order to estimate the difference between two solutions, we need the following technical lemma.

Lemma 7.5.1 *Let $N \in \mathbb{N}$ and $X, Y \in \mathbb{R}^{N\times N}$. Then*

$$\|e^{X+Y} - e^X\| \leq \|Y\|e^{\|X\|}e^{\|Y\|}. \tag{7.22}$$

Proof. For $n \in \mathbb{N}$ the binomial theorem (see, e.g., [34, Hilfssatz 2.12]) implies that

$$(X+Y)^n - X^n = \sum_{k=1}^{n} \binom{n}{k} X^k Y^{n-k}.$$

	GRPD				OGRPD				random control set			
Δ_1 \ T	8	16	24	32	8	16	24	32	8	16	24	32
0.1	2e20	1e19	1e20	2e15	3e19	2e19	9e19	3e15	4e19	2e19	2e19	3e18
0.2	1e19	1e19	**1e14**	**4e13**	6e19	5e19	**2e14**	**1e14**	2e19	2e19	6e19	3e18
0.4	8e19	2e19	**1e14**	**1e04**	5e19	8e18	**1e14**	**4e07**	6e19	4e19	8e15	**2e06**
0.8	3e19	**2e13**	**1e04**	**9e03**	3e19	**2e14**	**1e07**	**5e07**	5e19	4e18	**1e08**	**1e05**
1.6	6e20	**1e10**	**2e04**	**6e03**	1e20	**4e11**	**8e07**	**2e08**	3e19	6e18	**1e08**	**2e04**

Table 7.3: Same as Table 7.2 but for a total number of mesh points $K = 400$.

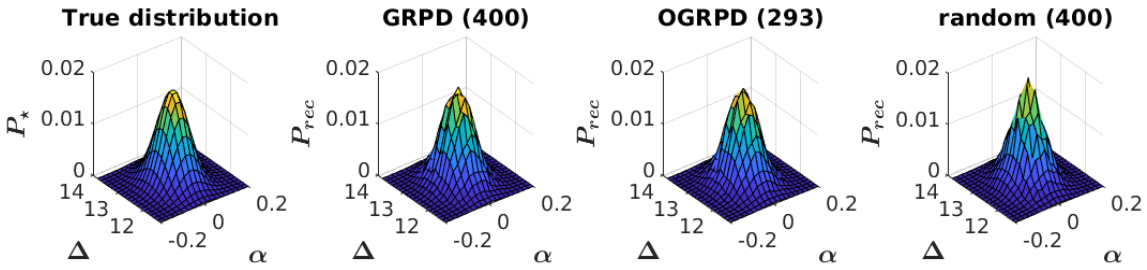

Figure 7.9: Same as Figure 7.5 but for $K = 400$ and including the reconstructed distribution for 400 random control fields.

Thus

$$\begin{aligned}\|(X+Y)^n - X^n\| &\leq \sum_{k=1}^{n} \binom{n}{k} \|X\|^k \|Y\|^{n-k} = (\|X\| + \|Y\|)^n - \|X\|^n \\ &\overset{(\star)}{=} (\|X\| + \|Y\| - \|X\|) \sum_{k=0}^{n-1} (\|X\| + \|Y\|)^k \|X\|^{n-k-1} \\ &\leq \|Y\| \sum_{k=0}^{n-1} (\|X\| + \|Y\|)^k (\|X\| + \|Y\|)^{n-k-1} \\ &= n\|Y\|(\|X\| + \|Y\|)^{n-1},\end{aligned}$$

where we used

$$\|X\|^n - \|Y\|^n = (\|X\| - \|Y\|) \sum_{k=0}^{n-1} \|X\|^k \|Y\|^{n-k-1}$$

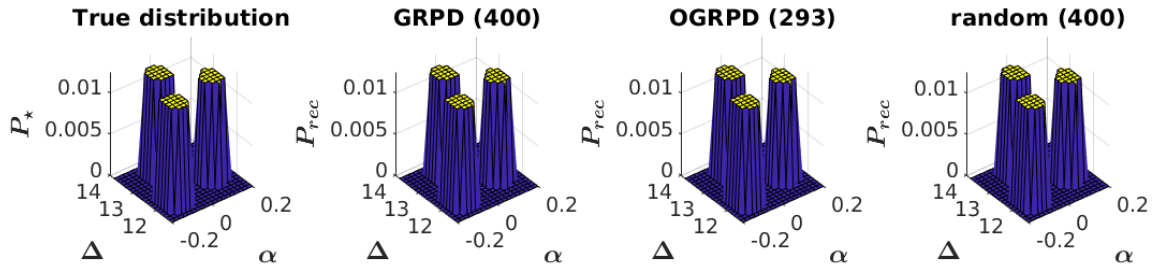

Figure 7.10: Same as Figure 7.9 but for a step distribution with three peaks.

to obtain $(\star)$. Hence, we get

$$\begin{aligned}\|e^{X+Y} - e^X\| &= \Big\| \sum_{n=0}^{\infty} \frac{(X+Y)^n}{n!} - \sum_{n=0}^{\infty} \frac{X^n}{n!} \Big\| \\ &\leq \sum_{n=1}^{\infty} \frac{\|(X+Y)^n - X^n\|}{n!} \\ &\leq \sum_{n=1}^{\infty} \frac{n\|Y\|(\|X\| + \|Y\|)^{n-1}}{n!} \\ &= \|Y\| \sum_{n=0}^{\infty} \frac{(\|X\| + \|Y\|)^n}{n!} \\ &= \|Y\| e^{\|X\|+\|Y\|} = \|Y\| e^{\|X\|} e^{\|Y\|}.\end{aligned}$$

□

Now, consider two parameter pairs $(\alpha, \Delta)_\ell$ and $(\alpha, \Delta)_{\ell+1}$, and define $D_\ell := t(\Delta_\ell A + \alpha_\ell(u_1 B_1 + u_2 B_2))$ and $D_{\ell+1} := t(\Delta_{\ell+1} A + \alpha_{\ell+1}(u_1 B_1 + u_2 B_2))$. Notice that

$$\begin{aligned}\|D_\ell - D_{\ell+1}\|_2 &= \|t(\Delta_\ell A + \alpha_\ell(u_1 B_1 + u_2 B_2)) - t(\Delta_{\ell+1} A + \alpha_{\ell+1}(u_1 B_1 + u_2 B_2))\|_2 \\ &\leq \underbrace{T\Big(|(\Delta_\ell - \Delta_{\ell+1})|\|A\|_2 + |\alpha_\ell - \alpha_{\ell+1}| u_m(\|B_1\|_2 + \|B_2\|_2)\Big)}_{=:E_\ell},\end{aligned}$$

where we used that $t \leq T$, $|u_1| \leq u_m$ and $|u_2| \leq u_m$. Clearly $E_\ell \to 0$ if $T \to 0$, or $u_m \to 0$, or $\Delta_{\ell+1} \to \Delta_\ell$ and $\alpha_{\ell+1} \to \alpha_\ell$. Setting $X = D_{\ell+1}$ and $Y = D_\ell - D_{\ell+1}$ in (7.22), we obtain

$$\begin{aligned}\|\boldsymbol{y}((\Delta,\alpha)_\ell, \boldsymbol{u}; t) - \boldsymbol{y}((\Delta,\alpha)_{\ell+1}, \boldsymbol{u}; t)\|_2 &= \|e^{D_\ell}\boldsymbol{y}^0 - e^{D_{\ell+1}}\boldsymbol{y}^0\|_2 \\ &\leq \|e^{D_\ell} - e^{D_{\ell+1}}\|_2 \underbrace{\|\boldsymbol{y}^0\|_2}_{=1} \\ &\leq \|D_\ell - D_{\ell+1}\|_2 e^{\|D_\ell\|_2} e^{\|D_\ell - D_{\ell+1}\|_2} \\ &\leq E_\ell e^{\|D_\ell\|_2} e^{E_\ell}.\end{aligned}$$

Recalling (7.14), $C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$ and that $\phi_j = \boldsymbol{e}_j$ in our example, the matrix entries of $\widehat{W}$ are given by

$$\widehat{W}_{i,j} = \sum_m \langle C\boldsymbol{y}((\Delta,\alpha)_i, \boldsymbol{u}^m; t^m), C\boldsymbol{y}((\Delta,\alpha)_j, \boldsymbol{u}^m; t^m)\rangle, \quad \text{for } i,j = 1,\ldots,K.$$

Thus, the difference between two rows (or analogously two columns) of $\widehat{W}$ is given by

$$[\widehat{W}]_{[\ell,:]} - [\widehat{W}]_{[\ell+1,:]} = \begin{bmatrix} \sum_m \left\langle \Big(C\boldsymbol{y}((\Delta,\alpha)_i, \boldsymbol{u}^m; t^m) - C\boldsymbol{y}((\Delta,\alpha)_j, \boldsymbol{u}^m; t^m) \Big), C\boldsymbol{y}((\Delta,\alpha)_1, \boldsymbol{u}^m; t^m) \right\rangle \\ \vdots \\ \sum_m \left\langle \Big(C\boldsymbol{y}((\Delta,\alpha)_i, \boldsymbol{u}^m; t^m) - C\boldsymbol{y}((\Delta,\alpha)_j, \boldsymbol{u}^m; t^m) \Big), C\boldsymbol{y}((\Delta,\alpha)_K, \boldsymbol{u}^m; t^m) \right\rangle \end{bmatrix}.$$

Since $\|\boldsymbol{y}((\Delta,\alpha), \boldsymbol{u}; t)\|_2 = 1$ for any $\Delta, \alpha, \boldsymbol{u}$ and t, we get

$$\|[\widehat{W}]_{[i,:]} - [\widehat{W}]_{[j,:]}\|_2^2 \leq K\|C\|_2^2 \sum_m \|\boldsymbol{y}((\Delta,\alpha)_i, \boldsymbol{u}^m; t^m) - \boldsymbol{y}((\Delta,\alpha)_j, \boldsymbol{u}^m; t^m)\|_2^2.$$

Hence, we obtain for two adjacent rows (or analogously two adjacent columns)

$$\|[\widehat{W}]_{[\ell,:]} - [\widehat{W}]_{[\ell+1,:]}\|_2 \leq \sqrt{K}\|C\|_2 n_{controls} E_\ell e^{\|D_\ell\|_2} e^{E_\ell},$$

where $\ell \in \{1, \ldots, K\}$ and $n_{controls} \in \mathbb{N}$ is the number of controls. Now, recall that $E_\ell \to 0$ if $T \to 0$, or $u_m \to 0$, or $\Delta_{\ell+1} \to \Delta_\ell$ and $\alpha_{\ell+1} \to \alpha_\ell$. Thus, the difference between two adjacent rows or columns of $\widehat{W}$ is bounded in norm by u_m, T and the mesh size for the probability distribution, i.e., $\alpha_{\ell+1} - \alpha_\ell$ and $\Delta_{\ell+1} - \Delta_\ell$.

Chapter 8

Greedy reconstruction algorithms for function approximation

In the previous chapters, we focused solely the reconstruction of objects associated with differential equations. However, the main idea of the GR algorithms is very versatile and does not inherently require the underlying model to be a differential equation. This allows us to extend our algorithms to general function approximation problems. In this context, they address two main aspects: the computation (or selection) of the data and the optimization of the structure of the approximation ansatz, within a given family of approximation functions. More specifically, given the task of finding an approximation (a surrogate) $g_\star$ of a function f, the goal of our methods is to construct a set of data points $\{(\widetilde{\boldsymbol{x}}_j, f(\widetilde{\boldsymbol{x}}_j))\}_{j=1}^K$ while optimizing the structure of the ansatz set $\mathcal{G}$, within a given family of approximation functions, in which one searches for $g_\star$. Given these two objects, one obtains $g_\star$ by computing an element in $\mathcal{G}$ that minimizes the distance from f when evaluated at the points $\{\widetilde{\boldsymbol{x}}_j\}_{j=1}^K$.

In the first part of this chapter, we consider the well-understood field of polynomial approximation, where one searches for a set of interpolation nodes that leads to the best approximation quality (see, e.g., [32, 60]). Here, while the structure of the approximation ansatz is fixed (a polynomial of a given order), the computation of the data (interpolating nodes) is a crucial aspect that strongly affects the approximation error. We prove that GR computes exactly a set of (tensor-product) Leja points (see, e.g, [13, 26, 36, 54] and references therein). Moreover, we show that OGR (for polynomial interpolation) is equivalent to the well-known Empirical Interpolation Method (EIM); see, e.g,. [10, 14, 20, 45, 58, 59] and references therein. Although the two methods can be equivalent, our greedy algorithms have one main advantage over EIM: EIM (and also other greedy algorithms used for function approximation, see, e.g., [15]) is limited to linear parametrizations of the ansatz function, while GR and OGR can also handle nonlinear parameterizations, such as weights and biases of neural networks.

In the second part, we introduce a new class of greedy algorithms that we call Data Greedy Reconstruction (DGR) algorithms. These procedures assume that a (very) large data set $\{(\widetilde{\boldsymbol{x}}_j, f(\widetilde{\boldsymbol{x}}_j))\}_{j=1}^J$, with $J \gg K$, is already available. Their goal is to select the smallest

subset of K elements from $\{(\widetilde{\boldsymbol{x}}_j, f(\widetilde{\boldsymbol{x}}_j))\}_{j=1}^J$ while simultaneously optimizing the structure of $\mathcal{G}$. In particular, we develop a combination of OGR and DGR tailored for neural network applications, called Network OGR (NOGR) algorithm. We provide a detailed discussion about the network structure optimization and illustrate how our algorithms can optimize the depth and number of neurons of a network.

8.1 Greedy reconstruction algorithms optimal interpolation points

For a (finite-dimensional) function space $\mathcal{G}$ over a compact set $X \subset \mathbb{R}^N$, the interpolant of a function $f : X \to \mathbb{R}^M$ for a set of points $\{\widetilde{\boldsymbol{x}}_j\}_{j=1}^K \subset X$ is the element $g_\star \in \mathcal{G}$ such that $g_\star(\widetilde{\boldsymbol{x}}_j) = f(\widetilde{\boldsymbol{x}}_j)$, $j = 1, \dots, K$. A general approach of finding this interpolant is to solve the least-squares problem

$$\min_{g \in \mathcal{G}} \sum_{j=1}^{K} \|f(\widetilde{\boldsymbol{x}}_j) - g(\widetilde{\boldsymbol{x}}_j)\|_2^2. \tag{8.1}$$

The overall goal is to find a $g_\star$ that approximates f well at all other points $\boldsymbol{x} \in X \setminus \{\widetilde{\boldsymbol{x}}_1, \dots, \widetilde{\boldsymbol{x}}_K\}$. In other words, the interpolant $g_\star$ must ideally solve

$$\min_{g \in \mathcal{G}} \max_{\boldsymbol{x} \in X} \|f(\boldsymbol{x}) - g(\boldsymbol{x})\|_2^2.$$

In that regard, the choice of the points $\{\widetilde{\boldsymbol{x}}_j\}_{j=1}^K$ plays a crucial role. One reasonable assumption is that for an increasing number of interpolation points K, the approximation error $\max_{\boldsymbol{x} \in X} \|f(\boldsymbol{x}) - g_\star(\boldsymbol{x})\|_2^2$ decreases. However, depending on the application, the evaluation of the function f at a large number of points might be very expensive or create an imbalance in the approximation properties of the interpolant $g_\star$ with respect to X. Thus, our goal is to find an efficient set of points $\{\widetilde{\boldsymbol{x}}_j\}_{j=1}^K$, i.e., the fewest number of points such that the approximation error is still reasonably small.

For this purpose, we introduce adapted versions of the GR Algorithm 1 and the OGR Algorithm 2. In the context of function approximation, they compute the inputs $\{\widetilde{\boldsymbol{x}}_j\}_{j=1}^K$ to generate the data $\{f(\widetilde{\boldsymbol{x}}_j)\}_{j=1}^K$ using a set (or basis) $\mathcal{B} := \{g_1, \dots, g_K\} \subset \mathcal{G}$.

In order to handle different function spaces $\mathcal{G}$, we assume that the set $\mathcal{B}$ can be parameterized, i.e., there exists $\widetilde{K} \leq K$, a closed and convex subspace $\mathcal{A} \subset \mathbb{R}^{\widetilde{K}}$, and an isomorphism $\widetilde{g} : \mathcal{A} \to \mathcal{G}, \boldsymbol{\beta} \mapsto \widetilde{g}(\boldsymbol{\beta}) := g_{\boldsymbol{\beta}}$. In this way, we can identify $\mathcal{B} = \{g_1, \dots, g_K\}$ with a set $\{\widetilde{\boldsymbol{\beta}}_1, \dots, \widetilde{\boldsymbol{\beta}}_K\} \subset \mathcal{A}$ such that $g_k := g_{\widetilde{\boldsymbol{\beta}}_k}$ for $k = 1, \dots, K$. Notice that this can be viewed as an extension of the previous chapters, where we considered the simple parameterization $g_{\boldsymbol{\beta}} = \sum_{k=1}^K \beta_k g_k$, for a linear basis $\mathcal{B}$, $\widetilde{K} = K$, $\mathcal{A} = \mathbb{R}^K$ and $\widetilde{\boldsymbol{\beta}}_k = \boldsymbol{e}_k$ the canonical vectors in $\mathbb{R}^K$.

GR and OGR applied to the setting of problem (8.1) is stated in Algorithms 6 and 7. Both GR and OGR are data-free methods, in the sense that they do not require evaluations of the function f (outputs). However, there are many applications, especially in the field of machine learning, where output data are already available. A common issue in these

Algorithm 6 GR for function approximation

Require: A set of functions $\mathcal{B} = \{g_1, \dots, g_K\} \subset \mathcal{G}$ with corresponding parameterizations $\{\widetilde{\boldsymbol{\beta}}_1, \dots, \widetilde{\boldsymbol{\beta}}_K\} \subset \mathbb{R}^{\widetilde{K}}$ with $K, \widetilde{K} \in \mathbb{N}$ with $\widetilde{K} \leq K$ and such that $g_k = g_{\widetilde{\boldsymbol{\beta}}_k}$ for $k = 1, \dots, K$.

1: Set $k = 1$ and find $\widetilde{\boldsymbol{x}}_1$ that solves

$$\max_{\boldsymbol{x} \in X} \|g_1(\boldsymbol{x})\|_2^2. \tag{8.2}$$

2: **while** $k \leq K - 1$ **do**

3: Fitting step: Find an $\boldsymbol{\beta}^k$ that solves

$$\min_{\boldsymbol{\beta} \in \operatorname{span}\{\widetilde{\boldsymbol{\beta}}_i\}_{i=1}^k} \sum_{j=1}^{k} \|g_{\boldsymbol{\beta}}(\widetilde{\boldsymbol{x}}_j) - g_{k+1}(\widetilde{\boldsymbol{x}}_j)\|_2^2. \tag{8.3}$$

4: Splitting step: Find an $\widetilde{\boldsymbol{x}}_{k+1}$ that solves

$$\max_{\boldsymbol{x} \in X} \|g_{\boldsymbol{\beta}^k}(\boldsymbol{x}) - g_{k+1}(\boldsymbol{x})\|_2^2. \tag{8.4}$$

5: Update $k \leftarrow k + 1$.

6: **end while**

data-driven settings is that the data is imbalanced, which can lead to biased models (see, e.g., [61, Section 5.3.1]) and thereby a lack of robustness. With the goal of selecting optimal data points that represent well the whole data set and avoid biased models, we introduce a data-driven version of GR (DGR) in Algorithm 8. Notice that the termination criterion, that is $\|g_{\boldsymbol{\beta}^k}(\widetilde{\boldsymbol{x}}_{k+1}) - \widetilde{\boldsymbol{y}}_{k+1}\|_2^2 < \text{tol}$, is similar to that of OGR where the evaluation $g_{k+1}(\widetilde{\boldsymbol{x}}_{k+1})$ is replaced by the data $\widetilde{\boldsymbol{y}}_{k+1}$. This ensures that DGR terminates when the function $g_{\boldsymbol{\beta}^k}$ is a sufficiently good approximation of the full data set $\{(\boldsymbol{x}_j, \boldsymbol{y}_j)\}_{j=1}^J$. In general, it is also possible to modify DGR by using other features of OGR to DGR like, for example, the automatic selection of the next element g_{k+1} from $\mathcal{B}$. However, we omit this discussion here, since the resulting method is very similar to the NOGR strategy introduced in Section 8.3.1.

8.2 GR algorithms for polynomial approximation

In this section, we apply our GR methods to the well-known case of function approximation by polynomials, i.e., $M = 1$ and $\mathcal{G} = \mathcal{P}_n(X)$, and study the corresponding behavior. For a set of linearly independent polynomials $\mathcal{B} = \{p_1, \dots, p_K\} \subset \mathcal{P}_n$, we consider the simple linear parameterization $p_{\boldsymbol{\beta}} = \sum_{k=1}^K \beta_k p_k$ for $\boldsymbol{\beta} \in \mathbb{R}^K$. This implies that, at the k-th GR iteration, the fitting step is interpolating the new polynomial p_{k+1} by a linear combination of the polynomials $p_1, \dots, p_k$. GR for polynomial interpolation is detailed in Algorithm 9. Now, we analyze the behavior of this algorithm in Sections 8.2.1 and 8.2.2. Finally, in Section 8.2.3 we discuss practical implementation aspects.

Algorithm 7 OGR for function approximation

Require: A set of functions $\mathcal{B} = \{g_1, \dots, g_K\} \subset \mathcal{G}$ with parameterizations $\mathcal{A} = \{\widetilde{\boldsymbol{\beta}}_1, \dots, \widetilde{\boldsymbol{\beta}}_K\} \subset \mathbb{R}_{\widetilde{K}}$ with $K, \widetilde{K} \in \mathbb{N}$, $\widetilde{K} \leq K$ and such that $g_k = g_{\widetilde{\boldsymbol{\beta}}_k}$, $k = 1, \dots, K$, and a tolerance $\text{tol} > 0$.

Find $\widetilde{\boldsymbol{x}}_1$ and ℓ_1 that solve

$$\max_{\ell \in \{1,\dots,K\}} \max_{\boldsymbol{x} \in X} \|g_\ell(\boldsymbol{x})\|_2^2. \tag{8.5}$$

Swap g_1 and g_{ℓ_1} in $\mathcal{B}$, $\widetilde{\boldsymbol{\beta}}_1$ and $\widetilde{\boldsymbol{\beta}}_{\ell_1}$ in $\mathcal{A}$, and set $f_{split} = \|g_1(\widetilde{\boldsymbol{x}}_1)\|_2^2$ and $k = 1$.
while $k \leq K - 1$ and $f_{split} \geq \text{tol}$ **do**
  for $\ell = k + 1, \dots, K$ **do**
    Fitting step: Find $\boldsymbol{\beta}^\ell$ that solves

$$\min_{\boldsymbol{\beta} \in \operatorname{span}\{\widetilde{\boldsymbol{\beta}}_i\}_{i=1}^k} \sum_{j=1}^{k} \|g_{\boldsymbol{\beta}}(\widetilde{\boldsymbol{x}}_j) - g_\ell(\widetilde{\boldsymbol{x}}_j)\|_2^2. \tag{8.6}$$

  end for
  Splitting step: Find $\widetilde{\boldsymbol{x}}_{k+1}$ and ℓ_{k+1} that solve

$$\max_{\ell \in \{k+1,\dots,K\}} \max_{\boldsymbol{x} \in X} \|g_{\boldsymbol{\beta}^\ell}(\boldsymbol{x}) - g_\ell(\boldsymbol{x})\|_2^2. \tag{8.7}$$

  Swap g_{k+1} and $g_{\ell_{k+1}}$ in $\mathcal{B}$, $\widetilde{\boldsymbol{\beta}}_{k+1}$ and $\widetilde{\boldsymbol{\beta}}_{\ell_{k+1}}$ in $\mathcal{A}$.
  Update $f_{split} \leftarrow \|g_{\boldsymbol{\beta}^k}(\widetilde{\boldsymbol{x}}_{k+1}) - g_{k+1}(\widetilde{\boldsymbol{x}}_{k+1})\|_2^2$ and $k \leftarrow k + 1$.
end while

8.2.1 Analysis of GR algorithms

In this section, we investigate the points selected by Algorithm 9 for one-dimensional and multivariate polynomial interpolation. First, we consider the standard case of one-dimensional polynomials, since it is well understood and the notation is simple. This allows us to formulate the main arguments of our analysis in a comprehensible way. Afterwards, we build the analysis for multivariate polynomials on top of the one-dimensional foundation.

One-dimensional polynomial interpolation

We begin with the one-dimensional setting $N = 1$ with $X \subset \mathbb{R}$ a closed interval. We also consider the standard assumption that the number of selected points matches the degree of the polynomials in $\mathcal{B}$, i.e., $K = n + 1$.

Recall (e.g., from [60, Theorem 8.2]) that for any set of points $\{\widetilde{x}_j\}_{j=1}^{n+1}$ and the polynomial $p_\star \in \mathcal{P}_n$ interpolating an $(n+1)$-times continuously differentiable function f at these points, there exists for each $x \in X$ a $\xi \in X$ such that

$$|f(x) - p_\star(x)| = \frac{\|f^{(n+1)}(\xi)\|_\infty}{(n+1)!} \prod_{j=1}^{n+1} |x - \widetilde{x}_j|. \tag{8.14}$$

Thus, to minimize the error $|f(x) - p_\star(x)|$, the points $\{\widetilde{x}_j\}_{j=1}^{n+1}$ should be chosen such that they minimize $\max_{x \in X} \prod_{j=1}^{n+1} |x - \widetilde{x}_j|$. One popular set of points that can be derived directly

Algorithm 8 DGR for function approximation

Require: A set of data points $\mathcal{Z} := \{(\boldsymbol{x}_j, \boldsymbol{y}_j)\}_{j=1}^{J} \subset X \times \mathbb{R}^M$ with $\boldsymbol{y}_j := f(\boldsymbol{x}_j)$ and $J \in \mathbb{N}$, a set of functions $\mathcal{B} = \{g_1, \dots, g_K\} \subset \mathcal{G}$ and a tolerance tol > 0.

1: Find $(\widetilde{\boldsymbol{x}}_1, \boldsymbol{y}_1)$ that solves

$$\max_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{Z}} \|\boldsymbol{y}\|_2^2. \tag{8.8}$$

2: Update $\mathcal{Z} \leftarrow \mathcal{Z} \setminus (\widetilde{\boldsymbol{x}}_1, \widetilde{\boldsymbol{y}}_1)$ and set $f_{split} = \|\widetilde{\boldsymbol{y}}_1\|_2^2$, $k = 1$.

3: **while** $k \leq J-1$ and $f_{split} \geq$ tol **do**

4: Fitting step: Find an $\boldsymbol{\beta}^k$ that solves

$$\min_{\boldsymbol{\beta}\in\operatorname{span}\{\widetilde{\boldsymbol{\beta}}_i\}_{i=1}^{k}} \sum_{j=1}^{k} \|g_{\boldsymbol{\beta}}(\widetilde{\boldsymbol{x}}_j) - \widetilde{\boldsymbol{y}}_j\|_2^2. \tag{8.9}$$

5: Splitting step: Find $(\widetilde{\boldsymbol{x}}_{k+1}, \widetilde{\boldsymbol{y}}_{k+1})$ that solves

$$\max_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{Z}} \|g_{\boldsymbol{\beta}^k}(\boldsymbol{x}) - \boldsymbol{y}\|_2^2. \tag{8.10}$$

6: Update $\mathcal{Z} \leftarrow \mathcal{Z} \setminus (\widetilde{\boldsymbol{x}}_{k+1}, \widetilde{\boldsymbol{y}}_{k+1})$, $f_{split} \leftarrow \|g_{\boldsymbol{\beta}^k}(\widetilde{\boldsymbol{x}}_{k+1}) - \widetilde{\boldsymbol{y}}_{k+1}\|_2^2$ and $k \leftarrow k+1$.

7: **end while**

from this goal, are the so-called Leja points. This class of interpolation points was first introduced by Edrei [36] and later studied by Leja [54]. Starting from any $\widetilde{x}_1 \in X$ the corresponding Leja sequence $(x_j)_{j=1}^{n+1}$ is generated iteratively by the update

$$\widetilde{x}_{k+1} = \arg\max_{x\in X} \prod_{j=1}^{k} |x - \widetilde{x}_j|, \qquad k = 1, \dots, n. \tag{8.15}$$

The update formula (8.15) has two main advantages. First, the update formula allows one to incorporate any preexisting set of points $\{\widetilde{x}_j\}_{j=1}^{k}$. This property is of particular interest in applications, where some preliminary data is already available. Second, the computation of the new point $\widetilde{x}_{k+1}$ allows one to evaluate the term $\max_{x\in X} \prod_{j=1}^{k} |x-\widetilde{x}_j| = \prod_{j=1}^{k} |\widetilde{x}_{k+1}-\widetilde{x}_j|$, which corresponds to the a-priori interpolation error given in (8.14).

As main result of this section, we show that GR computes exactly a sequence of Leja points if $\mathcal{B}$ is a polynomial basis of increasing order.

Theorem 8.2.1 (GR computes the Leja points) *Let $\mathcal{B} = \{p_1, \dots, p_{n+1}\} \subset \mathcal{P}_n$ be such that $\operatorname{span}\{p_1, \dots, p_k\} = \mathcal{P}_{k-1}$ for all $k = 1, \dots, n$. Then the points $\{\widetilde{x}_j\}_{j=1}^{n+1}$ computed by Algorithm 9 form a Leja sequence.*

Proof. We begin by constructing the coefficients $\boldsymbol{\beta}^k \in \mathbb{R}^k$ that solve the fitting step problem (8.12) at iteration k. Let us define $\widetilde{p}_k(x) := -a_{k+1} \prod_{j=1}^{k} (x - \widetilde{x}_j)$, where a_{k+1} is the leading coefficient of $p_{k+1} \in \mathcal{P}_k$. Then $\widetilde{p}_k + p_{k+1}$ is a polynomial in $\mathcal{P}_{k-1}$. Since $\{p_1, \dots, p_k\}$ form a basis of $\mathcal{P}_{k-1}$, there exists $\boldsymbol{\beta}^k \in \mathbb{R}^k$ such that $p_{\boldsymbol{\beta}^k} = \widetilde{p}_k + p_{k+1}$. Thus $p_{\boldsymbol{\beta}^k}(\widetilde{x}_j) - p_{k+1}(\widetilde{x}_j) = \widetilde{p}_k(\widetilde{x}_j) = 0$ for all $j = 1, \dots, k$, which implies that $\boldsymbol{\beta}^k = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^k} \sum_{j=1}^{k} |p_{\boldsymbol{\beta}}(\widetilde{x}_j) - p_{k+1}(\widetilde{x}_j)|$. The uniqueness follows by the fact that the points $\widetilde{x}_1, \dots, \widetilde{x}_k$ are distinct (which can be

Algorithm 9 GR for polynomial interpolation

Require: A set of linearly independent polynomials $\mathcal{B} = \{p_1, \dots, p_K\} \subset \mathcal{P}_n$.

1: Set $k = 1$ and find $\widetilde{\boldsymbol{x}}_1$ that solves

$$\max_{\boldsymbol{x} \in X} |p_1(\boldsymbol{x})|^2. \tag{8.11}$$

2: **while** $k \leq K - 1$ **do**

3: Fitting step: Find an $\boldsymbol{\beta}^k$ that solves

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^k} \sum_{j=1}^{k} |p_{\boldsymbol{\beta}}(\widetilde{\boldsymbol{x}}_j) - p_{k+1}(\widetilde{\boldsymbol{x}}_j)|^2, \tag{8.12}$$

where $p_{\boldsymbol{\beta}}(\boldsymbol{x}) = \sum_{i=1}^{k} \beta_i p_i(\boldsymbol{x})$.

4: Splitting step: Find an $\widetilde{\boldsymbol{x}}_{k+1}$ that solves

$$\max_{\boldsymbol{x} \in X} |p_{\boldsymbol{\beta}^k}(\boldsymbol{x}) - p_{k+1}(\boldsymbol{x})|^2. \tag{8.13}$$

5: Update $k \leftarrow k + 1$.

6: **end while**

shown by a simple iterative argument), meaning that the polynomial $p_{\boldsymbol{\beta}^k}$ and therefore also $\boldsymbol{\beta}^k$ are unique.

Now, using the polynomial $p_{\boldsymbol{\beta}^k}$ constructed above and solving (8.12), we obtain

$$|p_{\boldsymbol{\beta}^k}(x) - p_{k+1}(x)|^2 = |\widetilde{p}_k(x)|^2 = |a_{k+1}|^2 \prod_{j=1}^{k} |x - \widetilde{x}_j|^2.$$

Since a_{k+1} does not depend on x, the splitting-step problem (8.13) can be written as $\max_{x \in X} \prod_{j=1}^{k} |x - \widetilde{x}_j|^2$, which is equivalent to the Leja update (8.15). Thus, the splitting step of Algorithm 9 at iteration k computes exactly the point $\widetilde{x}_{k+1}$ in the Leja sequence. □

Notice that $p_1 \in \mathcal{P}_0$ implies that $p_1 \equiv c$ for some $c \in \mathbb{R}$. Thus, any $x \in X$ is a solution to the initialization problem (8.11). Together with the result from Theorem 8.2.1, this implies that one may choose any initial point $\widetilde{x}_1 \in X$ and GR will compute the corresponding Leja sequence $\{\widetilde{x}_j\}_{j=1}^{n+1}$.

Multivariate polynomial interpolation

Let us turn to the multivariate setting, i.e., $N > 1$ and $X = X_1 \times \cdots \times X_N$ with X_i a closed and bounded subinterval of $\mathbb{R}$ for $i = 1, \dots, N$. As in the one-dimensional case, we denote by $\mathcal{B}$ a basis of $\mathcal{P}_n = \mathcal{P}_n(X)$. Notice that $\dim(\mathcal{P}_n) = \binom{N+n}{N}$ (see, e.g., [70, Theorem 2.5]), which implies that GR will compute $K = \binom{N+n}{N}$ points.

To relate the points selected by GR to tensor products of Leja points, we introduce some additional notation. For an initial point $\widetilde{\boldsymbol{x}}_1 \in X$, we denote by $\{y_j^i\}_{j=1}^{n+1} \subset X_i$, y_j^i the Leja

sequence initialized by $\widetilde{x}_{1,i}$ for $i = 1, \dots, N$, i.e., $y_1^i = \widetilde{x}_{1,i}$ and

$$y_j^i = \operatorname*{arg\,max}_{x \in X_i} \prod_{\ell=1}^{j-1} |x - y_\ell^i|, \quad i = 1, \dots, N, \;\; j = 2, \dots, n+1. \tag{8.16}$$

Now, we define the m-first tensor-product Leja points for $m \in \mathbb{N}$ as

$$\mathcal{Y}_m := \Big\{ [y_{j_1}^1, \dots, y_{j_N}^N]^\top \mid j_1, \dots, j_N \in \mathbb{N}^+ \text{ s.t. } \sum_{i=1}^N j_i \leq N + m \Big\}, \tag{8.17}$$

with cardinality $\#\mathcal{Y}_m = \binom{N+m}{N}$. Let us give an example of $\mathcal{Y}_m$.

Example 8.2.2 (on the set $\mathcal{Y}_m$) *Let $N = n = 2$, $X = [-1,1]^2$ and $\widetilde{\boldsymbol{x}}_1 = [-1,-1]^\top$. Then $y_1^1 = y_1^2 = -1$, $y_2^1 = y_2^2 = \operatorname*{arg\,max}_{x \in [-1,1]} |x - (-1)| = 1$ and*

$$y_3^1 = y_3^2 = \operatorname*{arg\,max}_{x \in [-1,1]} |x+1||x-1| = 0.$$

Thus, the two Leja sequences $\{y_j^1\}_{j=1}^3$ and $\{y_j^2\}_{j=1}^3$ are given by $\{-1, 1, 0\}$. Since

$$\mathcal{Y}_m = \mathcal{Y}_{m-1} \cup \{[y_{j_1}^1, y_{j_2}^2]^\top \mid j_1, j_2 \in \mathbb{N}^+ \text{ s.t. } j_1 + j_2 = 2 + m\}$$

for $m \geq 1$, we obtain

$$\begin{aligned}
\mathcal{Y}_0 &= \{[y_1^1, y_1^2]^\top\} = \{[-1,-1]^\top\}, \\
\mathcal{Y}_1 &= \mathcal{Y}_0 \cup \{[y_1^1, y_2^2]^\top, [y_2^1, y_1^2]^\top\} = \mathcal{Y}_0 \cup \{[-1,1]^\top, [1,-1]^\top\}, \\
\mathcal{Y}_2 &= \mathcal{Y}_1 \cup \{[y_1^1, y_3^2]^\top, [y_2^1, y_2^2]^\top, [y_3^1, y_1^2]^\top\} = \mathcal{Y}_1 \cup \{[-1,0]^\top, [1,1]^\top, [0,-1]^\top\}, \\
\mathcal{Y}_3 &= \mathcal{Y}_2 \cup \{[y_2^1, y_3^2]^\top, [y_3^1, y_2^2]^\top\} = \mathcal{Y}_2 \cup \{[1,0]^\top, [0,1]^\top\}, \\
\mathcal{Y}_4 &= \mathcal{Y}_3 \cup \{[y_3^1, y_3^2]^\top\} = \mathcal{Y}_3 \cup \{[0,0]^\top\}.
\end{aligned}$$

These sets are represented in Figure 8.1. Notice that for $m \geq 5$ we have $\mathcal{Y}_m = \mathcal{Y}_4$.

Remark 8.2.3 (on the set $\mathcal{Y}_n$ and the full tensor product of Leja points) *Notice that the set $\mathcal{Y}_n = \mathcal{Y}_2$ in Example 8.2.2 still contains the full Leja tensor product $\{y_1^1, y_2^1\} \otimes \{y_1^2, y_2^2\} = \{-1,1\} \otimes \{-1,1\}$. However, this is not the case for general $n, N \in \mathbb{N}$ since a full Leja tensor product $\{y_1^1, \dots, y_m^1\} \otimes \cdots \otimes \{y_1^N, \dots, y_m^N\}$ of order $m \leq n$ contains all points $[y_{j_1}^1, \dots, y_{j_N}^N]^\top$ with $j_1, \dots, j_N \in \{1, \dots, m\}$. Thus, in order for the set $\mathcal{Y}_n$ to contain this point, we require n to satisfy $\sum_{i=1}^N m \leq N + n$, namely that $Nm \leq N + n$.*

The main result of this section (Theorem 8.2.8) is that the points computed by GR coincide with $\mathcal{Y}_n$. To prove it, two assumptions are needed. To clarify their meanings, we sketch the idea of the proof of Theorem 8.2.8 in what follows.

The main part of the result is proven by induction on the degree m of the polynomials in the basis $\mathcal{B} = \{p_1, \dots, p_K\}$. For the induction step, we follow the same idea of the proof for Theorem 8.2.1 and thus require that the basis elements are ordered according to their polynomial order and the appearing monomials. For this purpose, we introduce the following assumption on the order of $\mathcal{B}$.

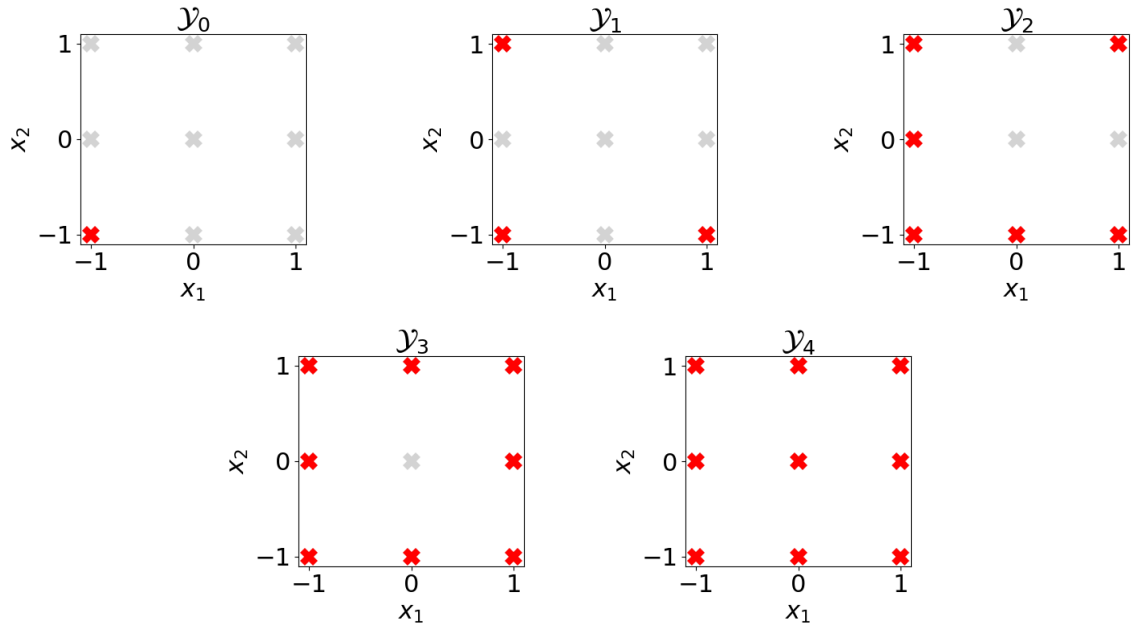

Figure 8.1: All tensor-product points (gray crosses) corresponding to the two Leja sequences $\{y_j^1\}_{j=1}^3 = \{y_j^2\}_{j=1}^3 = \{-1, 1, 0\}$ from Example 8.2.2, and points $\boldsymbol{x} = [x_1, x_2]^\top \in [-1, 1]^2$ (red crosses) contained in the sets $\mathcal{Y}_0$ (top left), $\mathcal{Y}_1$ (top middle), $\mathcal{Y}_2$ (top right), $\mathcal{Y}_3$ (bottom left) and $\mathcal{Y}_4$ (bottom right) as defined in (8.17).

Assumption 8.2.4 *Let $m \in \{0, \dots, n\}$, $k = \binom{N+m}{N}$ and $K_m = \binom{N+m+1}{N}$. Then the first k elements $p_1, \dots, p_k$ in $\mathcal{B}$ form a basis of $\mathcal{P}_m$. Additionally, for all $\ell \in \{k+1, \dots, K_m\}$ the polynomial $p_\ell \in \mathcal{B}$ introduces exactly one new monomial with respect to all monomials appearing in the polynomials $p_1, \dots, p_k$. In other words, there exists a (unique) polynomial $p \in \operatorname{span}(p_1, \dots, p_k)$ such that $p_\ell = a_{d_1^\ell, \dots, d_N^\ell} \boldsymbol{x}^{(d_1^\ell, \dots, d_N^\ell)} + p$, where $a_{d_1^\ell, \dots, d_N^\ell} \in \mathbb{R} \setminus \{0\}$ and $(d_1^\ell, \dots, d_N^\ell)$ denotes the componentwise degree of the new monomial $\boldsymbol{x}^{(d_1^\ell, \dots, d_N^\ell)}$ introduced by p_ℓ.*

Let us clarify Assumption 8.2.4 by the following example.

Example 8.2.5 *Let $N = n = 2$ and consider a basis $\mathcal{B} = \{p_1, \dots, p_K\}$ with $K = \binom{N+n}{N} = 6$ such that Assumption 8.2.4 holds. Then p_1 must be the constant polynomial, i.e., $p_1 \equiv a$ for some $a \in \mathbb{R}$. Now, there are two possible choices for p_2:*

$$p_2(\boldsymbol{x}) = a_1 x_1 + a_0 \qquad \text{or} \qquad p_2(\boldsymbol{x}) = a_1 x_2 + a_0,$$

for $a_1 \in \mathbb{R} \setminus \{0\}$, $a_0 \in \mathbb{R}$. Let us assume that $p_2(\boldsymbol{x}) = a_1 x_1 + a_0$. Then

$$p_3(\boldsymbol{x}) = b_1 x_2 + b_0,$$

for $b_1 \in \mathbb{R} \setminus \{0\}$, $b_0 \in \mathbb{R}$. Now, we have three choices for p_4:

$$p_4(\boldsymbol{x}) = \begin{cases} a_3x_1^2 + a_2x_2 + a_1x_1 + a_0, \\ a_3x_2^2 + a_2x_2 + a_1x_1 + a_0, \\ a_3x_1x_2 + a_2x_2 + a_1x_1 + a_0, \end{cases}$$

for $a_3 \in \mathbb{R} \setminus \{0\}$, $a_2, a_1, a_0 \in \mathbb{R}$. Depending on the choice of p_4, the polynomials p_5 and p_6 will be of the remaining two forms and thereby introduce the other monomials of degree 2.

Assumption 8.2.4 allows us to uniquely identify the basis elements in $\mathcal{B}$ by the componentwise degree of the monomials introduced by the basis elements. For example, if we choose $p_4 = a_3x_1x_2 + a_2x_2 + a_1x_1 + a_0$ in Example 8.2.5, then we can identify p_4 by the componentwise degree $(1, 1)$ that corresponds to the monomial $a_3x_1x_2$. Let us conclude the discussion on Assumption 8.2.4 with the following remark, relating the space of polynomials $\mathcal{P}_{m+1} \setminus \mathcal{P}_m$ with the set of Leja points $\mathcal{Y}_{m+1} \setminus \mathcal{Y}_m$, for $m = 0, \ldots, n$.

Remark 8.2.6 (on the sets $\mathcal{P}_m$ and $\mathcal{Y}_m$) *Let $m \in \{0, \ldots, n\}$, $k = \binom{N+m}{N}$ and $K_m = \binom{N+m+1}{N}$. Then Assumption 8.2.4 implies that the polynomials $\{p_\ell\}_{\ell=k+1}^{K_m}$ form a basis of $\mathcal{P}_{m+1} \setminus \mathcal{P}_m$. Thus, the componentwise degrees $(d_1^\ell, \ldots, d_N^\ell)$ of the monomials $\boldsymbol{x}^{(d_1^\ell, \ldots, d_N^\ell)}$ introduced by the polynomials p_ℓ, for $\ell = k+1, \ldots, K_m$, coincide with all possible combinations $(d_1, \ldots, d_N) \in \mathbb{N}^N$ that satisfy $\sum_{i=1}^N d_i = m + 1$. Moreover, we can write*

$$\begin{aligned} \mathcal{Y}_{m+1} \setminus \mathcal{Y}_m &= \Big\{ [y_{j_1}^1, \ldots, y_{j_N}^N]^\top \mid j_1, \ldots, j_N \in \mathbb{N}^+ \text{ s.t. } \sum_{i=1}^N j_i = N + m + 1 \Big\} \\ &= \Big\{ [y_{d_1+1}^1, \ldots, y_{d_N+1}^N]^\top \mid (d_1, \ldots, d_N) \in \mathbb{N}^N \text{ s.t. } \sum_{i=1}^N d_i = m + 1 \Big\}. \end{aligned}$$

Hence, there exists a one-to-one relation between the componentwise degrees of the monomials introduced by the polynomials $\{p_\ell\}_{\ell=k+1}^{K_m}$ and the Leja points $\mathcal{Y}_{m+1} \setminus \mathcal{Y}_m$.

The next step in the proof of our main result is to show that the splitting-step problem (8.13) is equivalent to a multivariate version of the Leja update (8.16). However, notice that maximization problems involving multivariate polynomials in general do not have a unique solution. Consider, for example, the maximization problem

$$\max_{\boldsymbol{x} \in X} |p(\boldsymbol{x})|^2, \; p(\boldsymbol{x}) = \left(\prod_{j=1}^{1} (x_1 - y_j^1) \right) \left(\prod_{j=1}^{0} (x_2 - y_j^2) \right) \cdots \left(\prod_{j=1}^{0} (x_N - y_j^N) \right) = x_1 - y_j^1. \quad (8.18)$$

Then any $\boldsymbol{x} = [x_1, \ldots, x_N]^\top \in X$ with $x_1 \in \arg\max_{x \in X_1} |x_1 - y_j^1|^2$ is a solution to (8.18). Notice that x_1 does not has to be unique. Thus, we introduce the following assumption that sets all components to the next entry of the corresponding Leja sequence.

Assumption 8.2.7 *At iteration $k \in \{1, \ldots, K-1\}$ of Algorithm 9, let $(d_1^{k+1}, \ldots, d_N^{k+1}) \in \mathbb{N}^N$ be the componentwise degree of the unique monomial that $p_{k+1} \in \mathcal{B}$ introduces, according to the second part of Assumption 8.2.4. If the splitting-step problem* (8.13) *is given by*

$\max_{\boldsymbol{x}\in X} |p_{\boldsymbol{\beta}^k}(\boldsymbol{x}) - p_{k+1}(\boldsymbol{x})|^2$ *with*

$$p_{\boldsymbol{\beta}^k}(\boldsymbol{x}) - p_{k+1}(\boldsymbol{x}) = a\left(\prod_{j=1}^{d_1^{k+1}}(x_1 - y_j^1)\right)\cdots\left(\prod_{j=1}^{d_N^{k+1}}(x_N - y_j^N)\right), \tag{8.19}$$

for some $a \in \mathbb{R}$*, then Algorithm 9 selects the point* $\widetilde{\boldsymbol{x}}_{k+1} = \left[y^1_{d_1^{k+1}+1}, \ldots, y^N_{d_N^{k+1}+1}\right]^\top$.

Notice that, if the polynomial $p_{\boldsymbol{\beta}^k}(\boldsymbol{x}) - p_{k+1}(\boldsymbol{x})$ is given as in (8.19), then

$$\max_{\boldsymbol{x}\in X} |p_{\boldsymbol{\beta}^k}(\boldsymbol{x}) - p_{k+1}(\boldsymbol{x})|^2 = a\left(\max_{x_1\in X_1}\prod_{j=1}^{d_1^{k+1}}|x_1 - y_j^1|^2\right)\cdots\left(\max_{x_N\in X_N}\prod_{j=1}^{d_N^{k+1}}|x_N - y_j^N|^2\right). \tag{8.20}$$

Thus, by the Leja update (8.16), $\left[y^1_{d_1^{k+1}+1}, \ldots, y^N_{d_N^{k+1}+1}\right]$ is always a solution to (8.20) and hence to the splitting-step problem (8.13). We can now state our main result.

Theorem 8.2.8 (multivariate GR computes tensor product Leja points) *Let* $n, N \in \mathbb{N}$*,* $K = \binom{N+n}{N}$*,* $\mathcal{B} = \{p_k\}_{k=1}^K$ *a basis of* $\mathcal{P}_n$ *and* $\widetilde{\boldsymbol{x}}_1 \in X$ *the solution to* (8.11)*. If Assumptions 8.2.4 and 8.2.7 hold, then the points computed by Algorithm 9 coincide with* $\mathcal{Y}_n$*.*

Proof. We prove the result by induction on the degree m of the polynomials in $\mathcal{B}$. The case $m = 0$ is trivial since $\mathcal{Y}_0 = \{[y_1^1, \ldots, y_1^N]^\top\} = \{\widetilde{\boldsymbol{x}}_1\}$. Let us now assume that for some $m \in \{1, \ldots, n\}$ and $k = \binom{N+m}{N}$ the set $\{\widetilde{\boldsymbol{x}}_j\}_{j=1}^k$ coincides with $\mathcal{Y}_m$. Then the result follows if we can show that the set $\{\widetilde{\boldsymbol{x}}_j\}_{j=1}^{K_m}$ for $K_m = \binom{N+m+1}{N}$ coincides with $\mathcal{Y}_{m+1}$. To do so, recalling Remark 8.2.6, it is sufficient to show that Algorithm 9 computes exactly the point $[y^1_{d_1^\ell+1}, \ldots, y^N_{d_N^\ell+1}]^\top$ at iteration $\ell \in \{k+1, \ldots, K_m\}$. We prove this result by a second induction argument on the index ℓ.

Let us begin with the base case $\ell = k+1$. By Assumption 8.2.4, we have

$$p_{k+1} = a_{d_1^{k+1},\ldots,d_N^{k+1}}\boldsymbol{x}^{(d_1^{k+1},\ldots,d_N^{k+1})} + p,$$

for some $p \in \operatorname{span}\{p_1, \ldots, p_K\}$. Now, we define the polynomial

$$\widetilde{p}(\boldsymbol{x}) := -a_{d_1^{k+1},\ldots,d_N^{k+1}}\left(\prod_{j=1}^{d_1^{k+1}}(x_1 - y_j^1)\right)\cdots\left(\prod_{j=1}^{d_N^{k+1}}(x_N - y_j^N)\right) \in \mathcal{P}_{m+1}, \tag{8.21}$$

where y_j^i are the Leja points defined in (8.16). By the definition of the polynomial degree given in Section 1.3, we can write $\widetilde{p}(\boldsymbol{x}) = \sum_{|\boldsymbol{\alpha}|=0}^m a_{\boldsymbol{\alpha}}\boldsymbol{x}^{\boldsymbol{\alpha}} - p_{k+1}(\boldsymbol{x})$, for some $a_{\boldsymbol{\alpha}} \in \mathbb{R}$. Thus, since the polynomials $\{p_1, \ldots, p_K\}$ form a basis of $\mathcal{P}_m$ (by Assumption 8.2.4), there exists a unique $\boldsymbol{\beta}^k \in \mathbb{R}^k$ such that

$$\widetilde{p}(\boldsymbol{x}) = p_{\boldsymbol{\beta}^k}(\boldsymbol{x}) - p_{k+1}(\boldsymbol{x}) = \sum_{i=1}^k \beta_i^k p_i(\boldsymbol{x}) - p_{k+1}(\boldsymbol{x}), \qquad \text{for all } \boldsymbol{x} \in X.$$

Recalling (8.17), we have for any $\boldsymbol{y} \in \mathcal{Y}_m$ that $\boldsymbol{y} = [y^1_{j_1}, \dots, y^N_{j_N}]^\top$ for some indices $j_1, \dots, j_N \in \mathbb{N}^+$ with $\sum_{i=1}^N j_i = N + m$. Since $\sum_{i=1}^N d_i^{k+1} = m + 1$, there exists at least one j_i such that $1 \le j_i \le d_i^{k+1}$.[1] Together with (8.21), this implies that $\widetilde{p}(\boldsymbol{y}) = 0$ for all $\boldsymbol{y} \in \mathcal{Y}_m$. Since we assume that the set $\{\widetilde{\boldsymbol{x}}_j\}_{j=1}^k$ coincides with $\mathcal{Y}_m$, we obtain $\widetilde{p}(\widetilde{\boldsymbol{x}}_j) = 0$ for all $j \in \{1, \dots, k\}$. Thus, $\boldsymbol{\beta}^k$ is the unique solution to the fitting-step problem (8.12), and the splitting-step problem (8.13) can be written as $\max_{\boldsymbol{x} \in X} |\widetilde{p}(\boldsymbol{x})|^2$. By Assumption 8.2.7, Algorithm 9 will now choose exactly $\widetilde{\boldsymbol{x}}_{k+1} = \left[y^1_{d_1^{k+1}+1}, \dots, y^N_{d_N^{k+1}+1}\right]^\top$.

Next, consider the induction step $\ell \to \ell + 1$. Let us denote by $\{\widetilde{\boldsymbol{x}}_j\}_{j=k+1}^\ell$ the set of points computed by Algorithm 9 from the base case until iteration ℓ. By the induction hypothesis, we have $\{\widetilde{\boldsymbol{x}}_j\}_{j=k+1}^\ell \in \mathcal{Y}_{m+1}$. Now, the induction step $\ell \to \ell + 1$ follows analogously to the base case above. We only have to show that the new polynomial

$$\widetilde{p}(\boldsymbol{x}) := -a_{d_1^{\ell+1}, \dots, d_N^{\ell+1}} \left(\prod_{j=1}^{d_1^{\ell+1}} (x_1 - y_j^1) \right) \cdots \left(\prod_{j=1}^{d_N^{\ell+1}} (x_N - y_j^N) \right) \in \mathcal{P}_{m+1} \tag{8.22}$$

satisfies $\widetilde{p}(\widetilde{\boldsymbol{x}}_j) = 0$ also for all $j \in \{k+1, \dots, \ell\}$. This can be shown by the following contradiction. Assume that $\widetilde{p}(\widetilde{\boldsymbol{x}}_j) \neq 0$ for some $j \in \{k+1, \dots, \ell\}$. By the assumption of the induction step, we have $\widetilde{\boldsymbol{x}}_j \in \mathcal{Y}_{m+1} \setminus \mathcal{Y}_m$ and $\widetilde{\boldsymbol{x}}_j = [y^1_{d_1^j+1}, \dots, y^N_{d_N^j+1}]^\top$ with $\sum_{i=1}^N d_i^j = m+1$. On the other hand, since $p_{\ell+1} \in \mathcal{P}_{m+1} \setminus \mathcal{P}_m$ we also have $\sum_{i=1}^N d_i^{\ell+1} = m + 1$. Thus, $\widetilde{p}(\widetilde{\boldsymbol{x}}_j) \neq 0$ together with (8.22) imply that $d_i^j + 1 > d_i^{\ell+1}$ for all $i \in \{1, \dots, N\}$. Since $\sum_{i=1}^N d_i^j = \sum_{i=1}^N d_i^{\ell+1}$, we obtain $d_i^j = d_i^{\ell+1}$ for all $i \in \{1, \dots, N\}$.[2] Since $j < \ell + 1$, this is a contradiction to the second part of Assumption 8.2.4. □

Notice that the full grid of tensor-product Leja points of order n is given by

$$\mathcal{Y} := \{y_j^1\}_{j=1}^{n+1} \times \cdots \times \{y_j^N\}_{j=1}^{n+1} = \{[y^1_{j_1}, \dots, y^N_{j_N}]^\top \mid j_1, \dots, j_N \in \{1, \dots, n+1\}\}.$$

Clearly, we have $\mathcal{Y}_n \subsetneq \mathcal{Y}$, meaning that GR applied to a basis $\mathcal{B}$ of $\mathcal{P}_n$ only computes a subset of the tensor-product Leja points. However, analogously to Theorem 8.2.8, one can also show that if we use a basis $\mathcal{B}$ that spans the same space as the monomials up to componentwise degree n^N, then GR computes the full tensor product $\mathcal{Y}$.

8.2.2 OGR and equivalence with EIM

As in Section 8.1, we can formulate an OGR for polynomial interpolation problems. The resulting method is stated in Algorithm 10. This algorithm looks similar to a popular greedy algorithm introduced in [14]: the empirical interpolation method (EIM). While EIM is mostly used in the context of reduced-basis applications for parameter-dependent

[1] This follows by a contradiction argument: assume that all $j_i > d_i^{k+1}$ for all $i \in \{1, \dots, N\}$, then $\sum_{i=1}^N j_i \ge \sum_i^N (d_i^{k+1} + 1) = N + m + 1$, contradicting $\sum_{i=1}^N j_i = N + m$.

[2] This follows by a contradiction argument: assume that there exists $i \in \{1, \dots, N\}$ with $d_i^j \neq d_i^{\ell+1}$. Then, since $\sum_{i=1}^N d_i^j = \sum_{i=1}^N d_i^{\ell+1}$, there exists $\widehat{i} \in \{1, \dots, N\}$ with $d_{\widehat{i}}^j < d_{\widehat{i}}^{\ell+1}$. This implies that $d_{\widehat{i}}^j + 1 \le d_{\widehat{i}}^{\ell+1}$, which contradicts $d_i^j + 1 > d_i^{\ell+1}$ for all $i \in \{1, \dots, N\}$.

Algorithm 10 OGR for polynomial interpolation

Require: A set $\mathcal{B}_{OGR} = \{p_1, \dots, p_K\} \subset \mathcal{P}_n$ for $K, n, N \in \mathbb{N}$ and a tolerance tol > 0.

1: Find $\widetilde{\boldsymbol{x}}_1$ and ℓ_1 that solve

$$\max_{\ell \in \{1,\dots,K\}} \max_{\boldsymbol{x} \in X} |p_\ell(\boldsymbol{x})|^2. \tag{8.23}$$

2: Swap p_1 and p_{ℓ_1} in $\mathcal{B}_{OGR}$, and set $f_{split} = |p_1(\widetilde{\boldsymbol{x}}_1)|$ and $k = 1$.
3: **while** $k \leq K - 1$ and $f_{split} \geq$ tol **do**
4: **for** $\ell = k + 1, \dots, K$ **do**
5: Fitting step: Find $\boldsymbol{\beta}^\ell$ that solves

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^k} \sum_{j=1}^{k} \Big| \sum_{i=1}^{k} \beta_i p_i(\widetilde{\boldsymbol{x}}_j) - p_\ell(\widetilde{\boldsymbol{x}}_j) \Big|^2. \tag{8.24}$$

6: **end for**
7: Splitting step: Find $\widetilde{\boldsymbol{x}}_{k+1}$ and ℓ_{k+1} that solve

$$\max_{\ell \in \{k+1,\dots,K\}} \max_{\boldsymbol{x} \in X} \Big| \sum_{i=1}^{k} \beta_i^\ell p_i(\boldsymbol{x}) - p_\ell(\boldsymbol{x}) \Big|^2. \tag{8.25}$$

8: Swap p_{k+1} and $p_{\ell_{k+1}}$ in $\mathcal{B}_{OGR}$, and set $p_{\boldsymbol{\beta}^k} = \sum_{i=1}^{k} \beta_i^{\ell_{k+1}} p_i$.
9: Update $f_{split} \leftarrow |p_{\boldsymbol{\beta}^k}(\widetilde{\boldsymbol{x}}_{k+1}) - p_{k+1}(\widetilde{\boldsymbol{x}}_{k+1})|$ and $k \leftarrow k + 1$.
10: **end while**

PDEs, it has also been studied for polynomial interpolation (see, e.g., [59, Section 3]). The version of EIM used for the setting of polynomial interpolation (compare [59, Section 2]), adapted to our notation, is stated in Algorithm 11.

Although the two methods are very similar, OGR has one main advantage over EIM: the parameterization discussed in Section 8.1. While EIM only considers linear combinations of the elements in $\mathcal{B}_{EIM}$, OGR can handle also nonlinear parameterizations, for example a set $\mathcal{B}_{OGR}$ of neural networks where the parameterizations are given by the weights and biases. We will discuss this example in more detail in Section 8.3.

Let us now compare OGR and EIM in more detail. The initializations of OGR (line 1 in Algorithm 10) and EIM (lines 1-2 in Algorithm 11) are obviously equivalent, since $\|p\|_{L^\infty(X)} = \max_{\boldsymbol{x} \in X} |p(\boldsymbol{x})|$ for any $p \in \mathcal{P}_n$. A similar analogy also holds for the maximization problems inside the while loops (line 7 in OGR and lines 8-9 in EIM). The difference here is that, while OGR selects elements p_k directly from the set $\mathcal{B}_{OGR}$, EIM generates a new basis of polynomials $\{q_k\}_{k=1}^{K_{max}}$ (see also line 10 in EIM). If we neglect this difference between p_i and q_i for a moment, we notice that the fitting step (line 5 in OGR) solves the strictly convex problem (8.24), whose optimality system coincides with the linear system (8.26). On the other hand, if at iteration k the polynomials $\{\widehat{p}_i\}_{i=1}^{k}$ computed by EIM coincide with the set $\{p_i\}_{i=1}^{k}$ selected by OGR, then we also have $\operatorname{span}\{p_i\}_{i=1}^{k} = \operatorname{span}\{q_i\}_{i=1}^{k}$. This is because q_1 is a rescaled version of $\widehat{p}_1$ (see line 3 in EIM) and the q_k for $k \geq 2$ are linear combinations of $\widehat{p}_1, \dots, \widehat{p}_k$ (see line 10 in EIM).

To prove the equivalence of OGR and EIM, we need the following results [59]:

Algorithm 11 EIM for polynomial interpolation

Require: A set of polynomials $\mathcal{B}_{EIM} = \{p_1, \dots, p_K\} \subset \mathcal{P}_n$ for $K, n, N \in \mathbb{N}$, and a maximum number of iterations $K_{max} \leq K$.

1: Compute $\widehat{p}_1 = \underset{p \in \mathcal{B}_{EIM}}{\arg\max} \|p\|_{L^\infty(X)}$.
2: Compute $\widetilde{\boldsymbol{y}}_1 = \underset{\boldsymbol{y} \in X}{\arg\max} |\widehat{p}_1(\boldsymbol{y})|$.
3: Set $q_1 = \frac{\widehat{p}_1}{\widehat{p}_1(\widetilde{\boldsymbol{y}}_1)}$ and $k = 1$.
4: **while** $k \leq K_{max} - 1$ **do**
5: **for** $p \in \mathcal{B}_{EIM}$ **do**
6: Find $\boldsymbol{\beta}(p) \in \mathbb{R}^k$ that solves the interpolation problem

$$\sum_{i=1}^{k} [\boldsymbol{\beta}(p)]_i q_i(\widetilde{\boldsymbol{y}}_j) = p(\widetilde{\boldsymbol{y}}_j), \quad j = 1, \dots, k. \tag{8.26}$$

7: **end for**
8: Compute $\widehat{p}_{k+1} = \underset{p \in \mathcal{B}_{EIM}}{\arg\max} \|p - \sum_{i=1}^{k} [\boldsymbol{\beta}(p)]_i q_i\|_{L^\infty(X)}$.
9: Compute $\widetilde{\boldsymbol{y}}_{k+1} = \underset{\boldsymbol{y} \in X}{\arg\max} |\widehat{p}_{k+1}(\boldsymbol{y}) - \sum_{i=1}^{k} [\boldsymbol{\beta}(\widehat{p}_{k+1})]_i q_i(\boldsymbol{y})|$.
10: Set $q_{k+1} = \frac{r_{k+1}}{r_{k+1}(\widetilde{\boldsymbol{y}}_{k+1})}$, where $r_{k+1}(\boldsymbol{y}) = \widehat{p}_{k+1}(\boldsymbol{y}) - \sum_{i=1}^{k} [\boldsymbol{\beta}(\widehat{p}_{k+1})]_i q_i(\boldsymbol{y})$.
11: Update $k \leftarrow k + 1$.
12: **end while**

P1 If $K_{max} \leq \binom{N+n}{N}$ then problem (8.26) is uniquely solvable (see [59, Theorem 1]).

P2 For all $k \in \{1, \dots, K_{max}\}$ and all $p \in \operatorname{span}\{q_i\}_{i=1}^k$ we have $p - \sum_{i=1}^{k} [\boldsymbol{\beta}(p)]_i q_i = 0$ (see [59, Lemma 1]).

Theorem 8.2.9 (Equivalence of OGR and EIM) *Assume that $\mathcal{B}_{OGR} = \mathcal{B}_{EIM} =: \mathcal{B}$. Let $\{\widetilde{\boldsymbol{x}}_j\}_{j=1}^{K_{OGR}}$, $\{p_i\}_{i=1}^{K_{OGR}}$ and $\{\widetilde{\boldsymbol{y}}_j\}_{j=1}^{K_{max}}$, $\{q_i\}_{i=1}^{K_{max}}$ be the points and polynomials selected by OGR and EIM, respectively. Then for any $\widehat{K} \in \mathbb{N}$, $\widehat{K} \leq \min\{K_{OGR}, K_{max}\}$, it holds that $\widetilde{\boldsymbol{x}}_j = \widetilde{\boldsymbol{y}}_j$ for $j = 1, \dots, \widehat{K}$ and $\operatorname{span}\{p_i\}_{i=1}^{\widehat{K}} = \operatorname{span}\{q_i\}_{i=1}^{\widehat{K}}$.*

Proof. Notice that EIM considers all elements of $\mathcal{B}$ in each iteration and does not reorder the set $\mathcal{B}$. Thus, we may assume that the polynomials in the set $\mathcal{B}$ are ordered as OGR selects them, i.e. the $\{p_i\}_{i=1}^{K_{OGR}}$ selected by OGR are exactly the first K_{OGR} polynomials in $\mathcal{B}$.

We now prove the result by induction. Since $\|p\|_{L^\infty(X)} = \max_{\boldsymbol{x} \in X} |p(\boldsymbol{x})|$ for any $p \in \mathcal{P}_n$, OGR and EIM choose the same first point $\widetilde{\boldsymbol{x}}_1 = \widetilde{\boldsymbol{y}}_1$ and corresponding polynomial $p_{\ell_1} = \widehat{p}_1$. By the definition of q_1 in line 3, we obtain $\operatorname{span}(p_{\ell_1}) = \operatorname{span}(q_1)$.

Next, we assume that at iteration $k \leq \widehat{K}$ we have $\widetilde{\boldsymbol{x}}_j = \widetilde{\boldsymbol{y}}_j$ for $j = 1, \dots, k$, and $\widehat{p}_i = p_i$ in EIM for $i = 1, \dots, k$, implying that $\operatorname{span}\{p_i\}_{i=1}^k = \operatorname{span}\{q_i\}_{i=1}^k$. From **P1** we obtain that (8.26) has a unique solution $\boldsymbol{\beta}_{EIM}^\ell$ for $\ell = 0, \dots, K$. Since $\operatorname{span}\{p_i\}_{i=1}^k = \operatorname{span}\{q_i\}_{i=1}^k$, the optimality system of (8.25) coincides with (8.26). Thus, also (8.25) has a unique solution $\boldsymbol{\beta}_{OGR}^\ell$ for $\ell = k+1, \dots, K$ with $\sum_{i=1}^{k} [\boldsymbol{\beta}_{OGR}^\ell]_{i+1} p_i = \sum_{i=1}^{k} [\boldsymbol{\beta}_{EIM}^\ell]_{i+1} q_i$. From **P2** we know that $p_\ell(\boldsymbol{x}) - \sum_{i=1}^{k} \beta_{i+1}^\ell q_i(\boldsymbol{x}) = 0$ for $\ell = 1, \dots, k$. On the other hand, the polynomials

$p_{k+1}, \ldots, p_K$ are linearly independent from $\{p_i\}_{i=1}^{k}$. Thus, the splitting step (8.25) of OGR is equivalent to the two maximization problems of EIM in lines 8-9, meaning that we obtain $\widetilde{\boldsymbol{x}}_{k+1} = \widetilde{y}_{k+1}$ and $p_{\ell_{k+1}} = \widehat{p}_{k+1}$. Finally, by the definition of q_{k+1} we obtain $\operatorname{span}\{p_i\}_{i=1}^{\widehat{K}} = \operatorname{span}\{q_i\}_{i=1}^{\widehat{K}}$. □

8.2.3 Practical implementation aspects

To solve the initialization problems (8.11) and (8.23), and the splitting-step problems (8.13) and (8.25) numerically, we consider a strategy used for both the Leja points (see, e.g., [62]) and EIM (see, e.g., [14, 59]). The idea is to replace the input space X by a finite approximation (e.g., a grid), which we denote by X_h, compute the value of the cost function at every $\boldsymbol{x} \in X_h$, and select the point $\widetilde{\boldsymbol{x}}$ corresponding to the maximum value.

Notice that the computational effort of this approach grows exponentially with respect to the dimension of X (increasing with N and n). A possible solution, based on the so-called Fast Leja points (see [12]), is to adaptively enlarge the discrete set X_h in each iteration. A similar strategy can also be applied to our GR algorithms. However, the discussion of these implementation details goes beyond the scope of this work.

8.2.4 Data approximation GR for polynomial interpolation

Following the idea described in Section 8.2.3 and replacing the space X by $X_h = \{\boldsymbol{x}_j\}_{j=1}^{J}$, $J \in \mathbb{N}$, the evaluations of the polynomials $p_1, \ldots, p_K$ on X_h can be precomputed, resulting in a discrete set

$$\mathcal{Z} := \{(\boldsymbol{x}_j, p_i(\boldsymbol{x}_j))\}_{\substack{i\in\{1,\ldots,N\},\\ j\in\{1,\ldots,J\}}}.$$

In this way, we can provide GR directly with $\mathcal{Z}$, instead of the set of polynomials $\mathcal{B}$. On the other hand, this approach can also be interpreted as GR trying to approximate the "data" $\mathcal{Z}$ by polynomial functions.

This motivates the application of DGR (Algorithm 8 in Section 8.1) to the setting of polynomial interpolation. The idea is to consider a data set $\mathcal{Z} = \{(\boldsymbol{x}_j, f(\boldsymbol{x}_j))\}_{j=1}^{J} \subset X \times \mathbb{R}$ and find the smallest number of data points $\widetilde{J} \in \mathbb{N}$, $\widetilde{J} \leq J$, such that the polynomial of degree $\widetilde{J} - 1$ interpolating $\{(\widetilde{\boldsymbol{x}}_j, f(\widetilde{\boldsymbol{x}}_j))\}_{j=1}^{\widetilde{J}}$ also approximates well the remaining data $\mathcal{Z} \backslash \{(\widetilde{\boldsymbol{x}}_j, f(\widetilde{\boldsymbol{x}}_j))\}_{j=1}^{\widetilde{J}}$. For brevity, let us denote $y_j := f(\boldsymbol{x}_j)$, $j = 1, \ldots, J$. The new version of DGR is stated in Algorithm 12. Notice that the fitting step in line 4 is no longer formulated as a least squares problem, since the interpolation of k points by a polynomial of degree $k - 1$ is exact.

Let us now test the performance of DGR to interpolate the famous Runge function

$$f(x) = \frac{1}{1 + 25x^2}, \qquad x \in [-1, 1]. \tag{8.29}$$

We provide DGR with a tolerance $\text{tol} = 10^{-10}$ and a data set $\{(x_j, f(x_j))\}_{j=1}^{J}$ with equidistant points $x_j = -1 + 2\frac{j-1}{J-1}$, $j = 1, \ldots, J$. To solve the interpolation problem in the fitting

Algorithm 12 DGR for approximation by polynomials

Require: A data set $\mathcal{Z} = \{(\boldsymbol{x}_j, y_j)\}_{j=1}^J$ for $J \in \mathbb{N}$ and a tolerance tol > 0.

1: Find $(\widetilde{\boldsymbol{x}}_1, \widetilde{y}_1)$ that solves

$$\max_{(\boldsymbol{x},y)\in\mathcal{Z}} |y|^2. \tag{8.27}$$

2: Update $\mathcal{Z} \leftarrow \mathcal{Z} \setminus (\widetilde{\boldsymbol{x}}_1, \widetilde{y}_1)$ and set $f_{split} = |\widetilde{y}_1|$, $k = 1$.

3: **while** $k \leq J-1$ and $f_{split} \geq$ tol **do**

4: Fitting step: Find $p^k \in \mathcal{P}_{k-1}$ that interpolates $\{(\widetilde{\boldsymbol{x}}_j, \widetilde{y}_j)\}_{j=1}^k$.

5: Splitting step: Find $(\widetilde{\boldsymbol{x}}_{k+1}, \widetilde{y}_{k+1})$ that solves

$$\max_{(\boldsymbol{x},y)\in\mathcal{Z}} |p^k(\boldsymbol{x}) - y|^2. \tag{8.28}$$

6: Update $\mathcal{Z} \leftarrow \mathcal{Z} \setminus (\widetilde{\boldsymbol{x}}_{k+1}, \widetilde{y}_{k+1})$, $f_{split} \leftarrow |p^k(\widetilde{\boldsymbol{x}}_{k+1}) - \widetilde{y}_{k+1}|$ and $k \leftarrow k+1$.

7: **end while**

step (line 4 of DGR), we use the barycentric Lagrange interpolation (see [23]). To test the performance of DGR, we evaluate the interpolation error

$$\max_{x\in X_h} |f(x) - p^k(x)| \tag{8.30}$$

at each iteration k of DGR on a fine grid $X_h \subset [-1,1]$ of 100000 equidistant points. We then compare the result to the interpolation error for the same amount of Chebyshev nodes (compare, e.g., [60, (10.20)]) and Fast Leja points (see [12]). The results for data sets of sizes $J = 100$ and $J = 1000$ are shown in Figure 8.2.

We observe that DGR for a data set of size $J = 1000$ terminates after $k = 124$ iterations (by reaching the required tolerance tol $= 10^{-10}$) and is able to match the convergence rate of the Chebyshev nodes and Fast Leja points. If we only provide DGR with $J = 100$ data points, it begins to diverge after 40 iterations. However, this is to be expected since the data $\{(x_j, f(x_j))\}_{j=1}^J$ provided to DGR are produced using equidistant points x_j, which are known to have a divergent interpolation error for the Runge function (compare, e.g., [60, Example 8.1]). However, the interpolation error for pure equidistant points in Figure 8.2 already starts to diverge at 10 interpolation points.

Obviously, the computational effort of DGR is larger than that in computing the Chebyshev nodes or Fast Leja points. Nevertheless, our investigations serve as a proof-of-concept that this data GR framework can perform well for polynomials, with the advantage that it can be extended to other frameworks.

8.3 A greedy algorithm for function approximation by residual neural networks

The advantage of our class of greedy algorithms is that the approximation ansatz does not have to be linear with respect to the coefficients $\boldsymbol{\beta}$. This allows us to extend them to the field of deep learning, where a neural network depends nonlinearly on parameters, like weights and biases as in the following example.

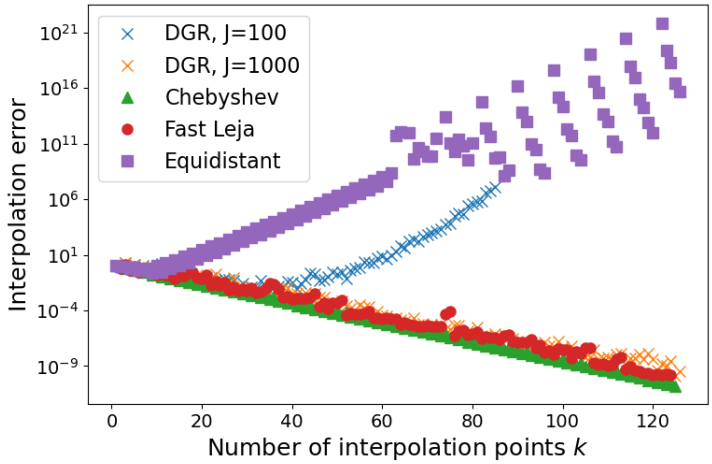

Figure 8.2: Interpolation error (8.30) for the Runge function (8.29), using the barycentric Lagrange interpolation for the points computed by DGR at iteration k for a data set of size $J = 100$ (blue crosses) and $J = 1000$ (orange crosses), and for Chebyshev nodes (green triangles), Fast Leja points (red circles) and equidistant points (purple squares) of degree k.

We consider the following class of residual neural networks. For a network $F_{W,b}$ of depth m with activation function σ and input $\boldsymbol{x} =: \boldsymbol{z}^0$, we let

$$\boldsymbol{z}^i = \begin{cases} \sigma(W^{[i-1]}\boldsymbol{z}^{i-1} + \boldsymbol{b}^{[i-1]}) & \text{for } i \in \{1, m\}, \\ \boldsymbol{z}^{i-1} + \sigma(W^{[i-1]}\boldsymbol{z}^{i-1} + \boldsymbol{b}^{[i-1]}) & \text{otherwise.} \end{cases}$$

In other words, the input of each hidden layer is summed directly to its output. That automatically implies that the widths of all hidden layers are equal. Thus, the architecture of any such network $F_{W,b}$ is determined by its depth m and the widths of all hidden layers d.

For inputs $(\widetilde{\boldsymbol{x}}_j)_{j=1}^J$, the problem of approximating a function f by a neural network can be formulated as

$$\min_{(W,b)\in\mathcal{W}(d)} L\Big((F_{W,b}(\widetilde{\boldsymbol{x}}_j))_{j=1}^J, (f(\widetilde{\boldsymbol{x}}_j))_{j=1}^J\Big), \tag{8.31}$$

where L is a loss function, for example, the mean squared error

$$L\Big((F_{W,b}(\widetilde{\boldsymbol{x}}_j))_{j=1}^J, (f(\widetilde{\boldsymbol{x}}_j))_{j=1}^J\Big) = \frac{1}{J}\sum_{j=1}^{J} \|f(\widetilde{\boldsymbol{x}}_j) - F_{W,b}(\widetilde{\boldsymbol{x}}_j)\|_2^2. \tag{8.32}$$

In Section 8.3.1, we formulate a combination of OGR and DGR that is able to simultaneously select a minimal network architecture (namely m and d) and an optimal subset of training data for a given data set. Afterwards, in Section 8.3.3, we test the performance of our new method, using residual neural networks to learn the Runge function and a spiral data set.

Algorithm 13 NOGR for deep neural networks

Require: A set of networks $\{F^1_{W_i,b_i}\}_{i=1}^K$ with depths $\{m_i\}_{i=1}^K$ and widths $\{d_i\}_{i=1}^K$ for $K \in \mathbb{N}$, maximum network depth m_{max} and maximum hidden layer width d_{max}, a set of training data $\mathcal{Z} = \{(\boldsymbol{x}_j, \boldsymbol{y}_j)\}_{j=1}^J$ for $J \in \mathbb{N}$, a maximum number of iterations $K_{it} \leq J$ and two tolerances $\mathrm{tol}_{fit}, \mathrm{tol}_{split} > 0$.

Set $\mathcal{L} = \{1, \dots, K\}$.
$((\widetilde{\boldsymbol{x}}_1, \widetilde{\boldsymbol{y}}_1), f_{split}, F^1_{NOGR}) \leftarrow$ splitting_step$(\mathcal{L}, \{F^1_{W_\ell,b_\ell}\}_{\ell\in\mathcal{L}}, \mathcal{Z})$.
Update $\mathcal{Z} \leftarrow \mathcal{Z} \setminus \{(\widetilde{\boldsymbol{x}}_1, \widetilde{\boldsymbol{y}}_1)\}$ and set $k = 1$.
while $k \leq K_{it}$ and $f_{split} \geq \mathrm{tol}_{split}$ **do**
    $(\{F^{k+1}_{W_i,b_i}\}_{i=1}^K, \mathcal{L}) \leftarrow$ fitting_step$(\{F^k_{W_i,b_i}\}_{i=1}^K, (\widetilde{\boldsymbol{x}}_j, \widetilde{\boldsymbol{y}}_j)_{j=1}^k, \mathrm{tol}_{fit})$
    $((\widetilde{\boldsymbol{x}}_{k+1}, \widetilde{\boldsymbol{y}}_{k+1}), f_{split}, F^{k+1}_{NOGR}) \leftarrow$ splitting_step$(\mathcal{L}, \{F^{k+1}_{W_\ell,b_\ell}\}_{\ell\in\mathcal{L}}, \mathcal{Z})$.
    Update $\mathcal{Z} \leftarrow \mathcal{Z} \setminus \{(\widetilde{\boldsymbol{x}}_{k+1}, \widetilde{\boldsymbol{y}}_{k+1})\}$ and $k \leftarrow k + 1$
end while
return Training points $\{(\widetilde{\boldsymbol{x}}_j, \widetilde{\boldsymbol{y}}_j)\}_{j=1}^k$ and the (trained) network $F_{NOGR} = F^k_{NOGR}$ with smallest validation loss f_{split}.

Algorithm 14 splitting_step

Require: A set of indices $\mathcal{L}$, the corresponding networks $\{F_{W_\ell,b_\ell}\}_{\ell\in\mathcal{L}}$, and a set of points $\mathcal{Z} = \{(\boldsymbol{x}_j, \boldsymbol{y}_j)\}_{j=1}^{\widetilde{J}}$, with $\widetilde{J} \in \mathbb{N}$.

for $\ell \in \mathcal{L}$ **do**
    Splitting step: Find $(\widehat{\boldsymbol{x}}_\ell, \widehat{\boldsymbol{y}}_\ell)$ that solve

$$\max_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{Z}} L(F_{W_\ell,b_\ell}(\boldsymbol{x}), \boldsymbol{y}). \tag{8.33}$$

    Set $f_\ell = L(F_{W_\ell,b_\ell}(\widehat{\boldsymbol{x}}_\ell), \widehat{\boldsymbol{y}}_\ell)$.
end for
Compute $\widetilde{\ell} = \arg\min_{\ell\in\mathcal{L}} f_\ell$.
Set $(\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{y}}) = (\widehat{\boldsymbol{x}}_{\widetilde{\ell}}, \widehat{\boldsymbol{y}}_{\widetilde{\ell}})$, $f_{split} = f_{\widetilde{\ell}}$, and $F_{NOGR} = F_{W_{\widetilde{\ell}},b_{\widetilde{\ell}}}$.
return The training point $(\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{y}})$ with validation loss f_{split} and corresponding network F_{NOGR}.

8.3.1 OGR for minimal neural network structures

In many machine learning applications the training data is already available. However, the choice of an appropriate/optimal neural network structure for the specific data is often only based on heuristics, experience, or determined by trial and error. This leaves open the possibility of choosing a suboptimal network size, which can lead to under/overfitting (see, e.g., [44, Section 5.2]). Moreover, even if a "good" network structure is found, it is a common issue that the data is imbalanced, leading to biased models (see, e.g., [61, Section 5.3.1]) and thereby lack of robustness.

To address these issues, we introduce a new method, called Network OGR (NOGR), which combines the ideas of the OGR Algorithm 7 and the DGR Algorithm 8. The main idea of this new greedy algorithm is to select simultaneously a subset of optimal training data points and the smallest network (trained on the selected data) capable of well representing the non-selected data. NOGR is stated in Algorithm 13, while its splitting and fitting steps are detailed in Algorithms 14 and 15. For a better overview, we summarize the role of some variables in Table 8.1.

Algorithm 15 fitting_step

Require: A set of networks $\{F_{W_i,b_i}\}_{i=1}^K$ with depths $\{m_i\}_{i=1}^K$ and widths $\{d_i\}_{i=1}^K$ for $K \in \mathbb{N}$, a set of points $\{(\widetilde{\boldsymbol{x}}_j, \widetilde{\boldsymbol{y}}_j)\}_{j=1}^k$ for $k \in \mathbb{N}$, and tolerances $\mathrm{tol}_{fit} \in \mathbb{R}$, m_{max} and d_{max}.

Set $\mathcal{K} = \emptyset$.
while $\mathcal{K} = \emptyset$ **do**
  for $\ell = 1, \dots, K$ **do**
    Fitting step: Train F_{W_ℓ,b_ℓ} on $\{(\widetilde{\boldsymbol{x}}_j, \widetilde{\boldsymbol{y}}_j)\}_{j=1}^k$, i.e., find $(\widetilde{W}_\ell, \widetilde{b}_\ell)$ that solve

$$\min_{(W,b)\in\mathcal{W}(d^\ell)} L\Big(\{F_{W,b}(\widetilde{\boldsymbol{x}}_j)\}_{j=1}^k, \{\widetilde{\boldsymbol{y}}_j\}_{j=1}^k\Big), \tag{8.34}$$

    and set $F_{W_\ell,b_\ell} = F_{\widetilde{W}_\ell,\widetilde{b}_\ell}$.
  end for
  Set $\mathcal{K} = \{\ell \in \{1, \dots, K\} \mid L\Big(\{F_{W_\ell,b_\ell}(\widetilde{\boldsymbol{x}}_j)\}_{j=1}^k, \{\widetilde{\boldsymbol{y}}_j\}_{j=1}^k\Big) < \mathrm{tol}_{fit}\}$.
  for $i \in \{1, \dots, K\} \setminus \mathcal{K}$ **do**
    Increase m_i and/or d_i (not beyond m_{max}, d_{max}), and update F_{W_i,b_i} accordingly.
  end for
  if $\min_{i\in\{1,\dots,K\}}(m_i, d_i) = (m_{max}, d_{max})$ **then**
    STOP with message: *"Was not able to fit the data, stopping NOGR!"*
  end if
end while
return A set of trained networks $\{F_{W_j,b_j}\}_{j=1}^K$ and a set of indices $\mathcal{K}$.

Let us describe the first iteration of NOGR in more detail. First, NOGR applies the splitting step (Algorithm 14) to the full set of networks $\{F^1_{W_i,b_i}\}_{i=1}^K$. In this step, for each network a training point $(\widehat{\boldsymbol{x}}, \widehat{\boldsymbol{y}}) \in \mathcal{Z}$ is computed by maximizing the validation error in (8.33). Among all these training points, the one that corresponds to the network F^1_{NOGR} with the smallest validation error f_{split} is selected and denoted by $(\widetilde{\boldsymbol{x}}_1, \widetilde{\boldsymbol{y}}_1)$. Next, NOGR checks whether the network F^1_{NOGR} already represents well enough all the training data, namely the validation error f_{split} is below the tolerance tol_{split}. If that is not the case, NOGR continues with the fitting step (Algorithm 15). Here, all networks are trained on $(\widetilde{\boldsymbol{x}}_1, \widetilde{\boldsymbol{y}}_1)$. If the training of some networks is unsuccessful, i.e., the final loss of the corresponding training problems is larger than the tolerance tol_{fit}, then the depths and/or widths of these networks are increased. The fitting step is repeated until one training was successful or the maximum network depth m_{max} and maximum width of the hidden layers d_{max} are reached. In the latter case, the process is stopped because none of the admissible network architecture was able to represent the selected data. Otherwise, if the training was successful for one or more of the networks, these (trained) networks are registered in the set $\mathcal{L}$ for the next splitting step.

8.3.2 Practical implementation aspects

The main advantage of the particular class of residual neural networks that we introduced at the beginning of Section 8.3 is that it limits the amount of choices when upgrading the network. One can either add a new hidden layer with the same width as all the other hidden layers or increase the width of all existing hidden layers by 1.

Variable name	Description/Role
m_{max}, d_{max}	Bounds on the depth and hidden layer width of the networks $\{F_{W_j,b_j}\}_{j=1}^K$.
tol_{fit}	Tolerance used in the fitting step to decide whether the training of a network was successful or not.
tol_{split}	Tolerance used to terminate NOGR if the validation error of a network is sufficiently small.
$\mathcal{K}$	Set of indices computed during the fitting step, indicating which networks were successfully trained.
$\mathcal{L}$	Set of indices deciding which networks are considered in the splitting step. At iterations $k \geq 2$ we have $\mathcal{L} = \mathcal{K}$.
$\mathcal{Z}$	Set of training points not selected by NOGR at a certain iteration.

Table 8.1: Description for some variables used in Algorithms 13, 14, and 15.

Let us explain this upgrade strategy in more detail. We initialize NOGR with a set of networks $\{F_{W_i,b_i}\}_{i=1}^K$ that have no hidden layers (i.e., $m_i = d_i = 0$), but different weights and biases that are randomly chosen in $[-1, 1]$. In the first architecture update in lines 7-9 of the fitting step (Algorithm 15), we replace all $\{F_{W_i,b_i}\}_{i\in\{1,\dots,K\}\setminus\mathcal{K}}$ with random networks that have exactly one hidden layer of width 1, i.e., $m_i = d_i = 1$. For all further updates, we use the smallest (successfully trained) network as reference size and add either a new hidden layer, or increase the width of all hidden layers by 1. More precisely, by defining $\#F_{W,b}$ as the total number of weights and biases in the network $F_{W,b}$, we compute the index of the smallest (successfully trained) network $\widehat{\ell}$ as

$$\widehat{\ell} = \begin{cases} \underset{\ell\in\mathcal{K}}{\arg\min}\, \#F_{W_\ell,b_\ell} & \text{if } \mathcal{K} \neq \emptyset, \\ \underset{\ell\in\{1,\dots,K\}}{\arg\min}\, \#F_{W_\ell,b_\ell} & \text{otherwise.} \end{cases}$$

Then we set $(\widetilde{m}_1, \widetilde{d}_1) = (m_{\widehat{\ell}} + 1, d_{\widehat{\ell}})$ and $(\widetilde{m}_2, \widetilde{d}_2) = (m_{\widehat{\ell}}, d_{\widehat{\ell}} + 1)$, and replace the set $\{F_{W_i,b_i}\}_{i\in\{1,\dots,K\}\setminus\mathcal{K}}$ with random networks such that half of them have depth $\widetilde{m}_1$ and hidden layers of width $\widetilde{d}_1$, and the other half have depth $\widetilde{m}_2$ and hidden layers of width $\widetilde{d}_2$.

Remark 8.3.1 *NOGR can also be extended to other classes of neural networks. The main challenge here is to design an appropriate strategy to update the network architecture (lines 7-9 in Algorithm 15). For general feedforward neural networks, for example, the number of possible incremental upgrades (by one node) grows proportionally to the number of hidden layers.*

8.3.3 Numerical experiments

We test the performance of NOGR, using the residual network structure and upgrade strategy discussed in Section 8.3.2. For our first test we consider again the Runge function.

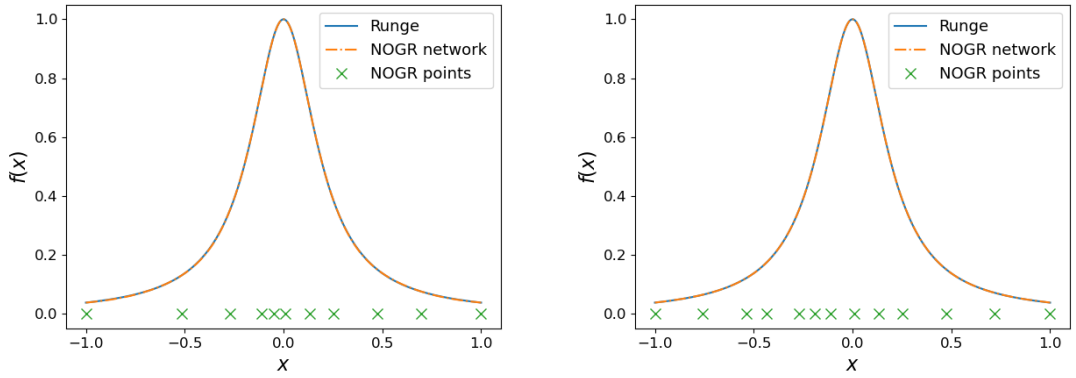

Figure 8.3: **Left**: Runge function (blue line), inputs $(\widetilde{x}_j)_{j=1}^{11}$ of the data points selected by NOGR (green crosses), for tolerances $\text{tol}_{fit} = \text{tol}_{split} = 10^{-5}$, and output of the network F_{NOGR} on an equidistant grid X_h of 100000 points (orange dash-dotted line). **Right**: Same as left, but for smaller tolerances $\text{tol}_{fit} = \text{tol}_{split} = 10^{-7}$.

Afterwards, we apply NOGR to a two-dimensional spiral data set. Our implementations use PyTorch [8].

An optimal residual network for the Runge function

Our goal is to find the smallest residual network that is able to learn the Runge function (8.29). Similar to Section 8.2.4, we consider a data set $\{(x_j, f(x_j))\}_{j=1}^{J}$ for $J = 100$ equidistant points $x_j \in [-1, 1]$.

For all networks in this section, we fix input and output size to $d_0 = d_m = 1$ and use the hyperbolic tangent as activation function, i.e., $\sigma(x) = \tanh(x)$. Since we want to learn the outputs of a function, we consider the mean squared error loss (8.32). We solve the fitting-step problem (8.34) using a BFGS method with the full gradient to have good convergence properties. Notice that there is no need to apply any stochastic optimization approach, since the fitting step at iteration k trains the networks only on the selected training points $\{(\widetilde{\boldsymbol{x}}_j, \widetilde{\boldsymbol{y}}_j)\}_{j=1}^{k}$.

We run NOGR for a set of $K = 100$ networks, a maximum network width and depth of 20 and $\text{tol}_{fit} = \text{tol}_{split} = 10^{-5}$. The algorithm stops after 11 iterations, selecting data points $(\widetilde{x}_j, f(\widetilde{x}_j))_{j=1}^{11}$ and a network F_{NOGR} of size $(1, 1, 1, 1, 1, 1, 1)$, i.e., 5 hidden layers of width 1, accumulating to 11 trainable parameters. The loss of F_{NOGR} on the full training data set of 100 points is $5.32 \cdot 10^{-7}$, the loss on a fine grid $X_h \subset [-1, 1]$ of 100000 equidistant points is $5.38 \cdot 10^{-7}$. The results are plotted in Figures 8.3 and 8.4. For a fixed network $F_{W,b}$ the interpolation error in Figure 8.4 is defined as

$$\max_{x \in X_h} |f(x) - F_{W,b}(x)|. \tag{8.35}$$

To obtain the plots in Figure 8.4, we train the networks $\{F^k_{W_i,b_i}\}_{i=1}^{100}$ used in the NOGR fitting step in the iterations $k = 3, \dots, 11$ on the data points $\{(x_j, f(x_j))\}_{j=1}^{k}$, where

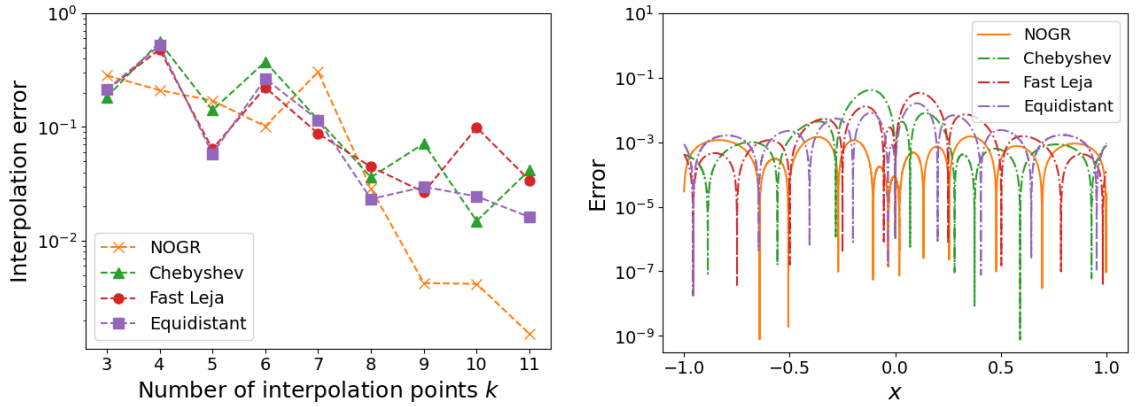

Figure 8.4: **Left**: Interpolation error (8.35) of F^k_{NOGR} at iteration $k \in \{3, \dots, 11\}$ (orange crosses) for tolerances $\text{tol}_{fit} = \text{tol}_{split} = 10^{-5}$, compared to the smallest interpolation error achieved by one of the networks $\{F^k_{W_i,b_i}\}_{i=1}^{100}$ trained on k Chebyshev nodes (green triangles), Fast Leja points (red circles) or equidistant points (purple squares), respectively. **Right**: Absolute difference (8.36) for the final network $F_{NOGR} = F^{11}_{NOGR}$ (orange line) evaluated on a fine grid of 100000 points in $[-1, 1]$, compared to the error for the one network from the set $\{F^{11}_{W_i,b_i}\}_{i=1}^{100}$ that achieved the smallest interpolation error when trained on 11 Chebyshev nodes (green dash-dotted line), Fast Leja points (red dash-dotted line) or equidistant points (purple dash-dotted line), respectively.

the inputs x_j correspond to k Chebyshev nodes, Fast Leja points or equidistant points, respectively. We then select, for each iteration and each input set, the network $F^k_{W,b}$ with the smallest interpolation error (8.35), and plot this error in the left-hand side plot of Figure 8.4. For the final iteration $k = 11$, we additionally plot the absolute difference

$$|f(x) - F_{W,b}(x)|, \qquad x \in X_h, \tag{8.36}$$

for the networks $F^{11}_{W,b}$, selected for the different sets of inputs, in the right-hand side plot of Figure 8.4. We observe that all input sets perform similarly until iteration $k = 8$. Afterwards, NOGR outperforms the other input sets by one order of magnitude. One possible reason is the fact that NOGR places more inputs towards the center of the interval, where the Runge function exhibits more rapid changes. Indeed, as we can see from the right-hand side plot in Figure 8.4, the networks corresponding to Chebyshev nodes, Fast Leja points and equidistant points have the largest error in the center area of the interval. On the other hand, the error peaks for the network F_{NOGR} seem to be the smallest in the center. This could also explain why the equidistant points perform better than Chebyshev and Leja points, which tend to accumulate more points towards the bounds of the interval. However, we also notice that the points, where the error is dropping very low (i.e., the intersection points between the smooth network functions $F_{W,b}(x)$ and the Runge function), do not necessarily coincide with the training points. The reason for this is that there is no guarantee that a trained network will interpolate the training data exactly, especially since we are dealing with rather small networks.

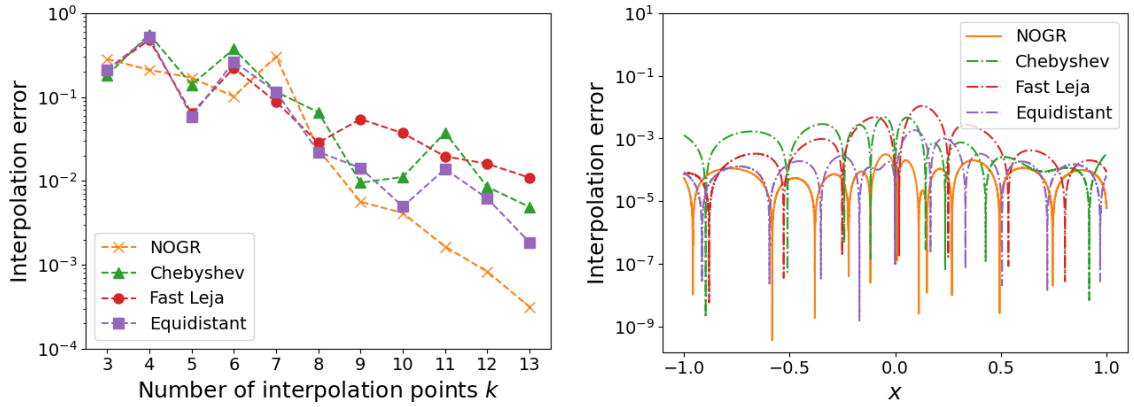

Figure 8.5: Same as Figure 8.4, but for smaller tolerances $\text{tol}_{fit} = \text{tol}_{split} = 10^{-7}$.

To better understand the role of the tolerances in NOGR, we repeat our test with $\text{tol}_{fit} = \text{tol}_{split} = 10^{-7}$. As expected, the algorithm requires more iterations (13) and a larger network F_{NOGR} of size $(1, 2, 2, 2, 2, 2, 2, 1)$ in order to fit the data in accordance with the smaller tolerances. This means that F_{NOGR} has one more layer than for the larger tolerances, and the width of all hidden layers is increased by 1. Although this corresponds to "only" two upgrades of the network structure, the number of trainable parameters more than triples to a total of 36. On the other hand, we also obtain a smaller loss on the full training data set ($9.48 \cdot 10^{-9}$) and on the fine grid X_h ($9.56 \cdot 10^{-9}$). The results are plotted in Figures 8.3 and 8.5. As before, the networks trained on the data selected by NOGR outperform the ones trained on data corresponding to other input sets.

Optimal learning of a two-dimensional spiral data set

In this numerical experiment, we apply our NOGR to find the smallest residual neural network that separates two spirals in $\mathbb{R}^2$. The data set for this example is adapted from [47], and consists of 600 input vectors $\boldsymbol{x} \in \mathbb{R}^2$ and corresponding labels $\boldsymbol{y} \in \{\binom{0}{1}, \binom{1}{0}\} \subset \mathbb{R}^2$ that indicate whether $\boldsymbol{x}$ belongs to the red or blue spiral. We randomly select $J = 450$ data points for NOGR and set aside the remaining 150 points to test the performance of NOGR a posteriori. For all networks in this section, we set input and output size to $d_0 = d_m = 2$. As activation function, we use again the hyperbolic tangent, i.e., $\sigma(x) = \tanh(x)$. Since we consider a classification problem, we use the cross entropy loss in both sub-problems of NOGR

$$L(\boldsymbol{z}, \boldsymbol{y}) := -\sum_{i=1}^{2} \log \frac{\exp([\boldsymbol{z}]_i)}{\sum_{j=1}^{2} \exp([\boldsymbol{z}]_j)} [\boldsymbol{y}]_i.$$

We solve the fitting-step problem (8.34) with a (full gradient) BFGS method.

We run NOGR for a set of $K = 100$ networks, a maximum network width and depth of 20 and $\text{tol}_{fit} = \text{tol}_{split} = 10^{-7}$. The algorithm stops after 58 iteration with a network F_{NOGR} of size $(2, 4, 2)$, i.e., one hidden layer of width 4, accumulating to 20 trainable parameters. The loss of F_{NOGR} on the full training data set of 450 points is $1.06 \cdot 10^{-9}$, the loss on the

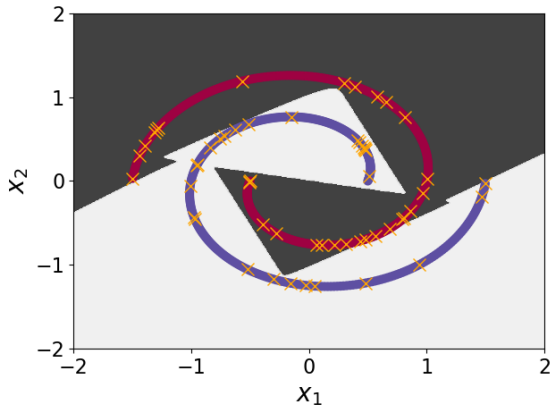

Figure 8.6: Full spiral data set (red and blue curve), inputs $\{\widetilde{\boldsymbol{x}}_j\}_{j=1}^{58}$ of the data points selected by NOGR (orange crosses) and decision output of the network F_{NOGR} for all points on the domain (dark and light gray areas).

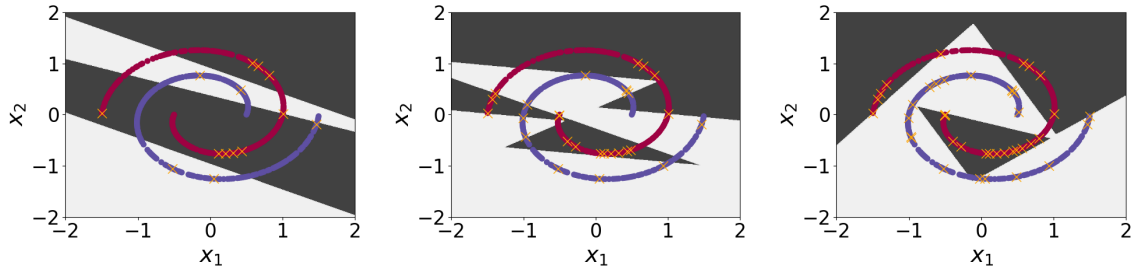

Figure 8.7: Spiral training data (red and blue curve), inputs $(\widetilde{\boldsymbol{x}}_j)_{j=1}^{k}$ of the data points selected by NOGR at iterations $k = 14, 28, 42$ (from left to right) and decision output of the network F_{NOGR}^{k} for all points on the domain (dark and light gray areas).

150 test points is $6.36 \cdot 10^{-9}$ with a classification accuracy of 100%. The spiral training data, the outputs of the trained network F_{NOGR} and the selected data points $\{(\widetilde{\boldsymbol{x}}_j, \widetilde{\boldsymbol{y}}_j)\}_{i=1}^{58}$ are plotted in Figure 8.6. As we can see, the relatively small network F_{NOGR} selected by NOGR is able to separate the two spirals, while using only $\approx 13\%$ of the training data. In fact, the interface drawn by F_{NOGR} between the dark and light area is almost the best possible linear separation for the fewest number of vertices. The placement of $\{(\widetilde{\boldsymbol{x}}_j, \widetilde{\boldsymbol{y}}_j)\}_{j=1}^{58}$ is also very interesting. Many of them seem to be chosen close to vertices and edges of the separation interface. To better understand the interplay of selecting points and training the network during NOGR, we show in Figure 8.7 the inputs $\{\widetilde{\boldsymbol{x}}_j\}_{j=1}^{k}$ of the data points selected by NOGR and the output of the intermediate network F_{NOGR}^{k}, at iterations $k = 14, 28, 42$.

Chapter 9

Conclusions and Outlook

In this thesis, we developed, analyzed and tested greedy reconstruction algorithms. In particular, we tackled the reconstruction of operators in dynamical systems. Our analysis for the reconstruction of control operators in linear ODEs revealed the strong dependence of the performance of the standard GR algorithm on the observability properties of the system. The OGR algorithm, on the other hand, was able to adjust the basis elements used to reconstruct the unknown operator, ensuring a precise and efficient reconstruction.

To extend our results to nonlinear problems for the reconstruction of drift operators in linear and bilinear systems, we considered a linearized approach inspired by GN. We proved that the controls obtained by LGR lead to local convergence of GN applied to the online nonlinear identification problem. In addition to observability, this convergence depends also on the controllability properties of the underlying system. Our results also extended to the controls obtained by the standard GR algorithm on the original system (without linearization). Numerical experiments demonstrated the efficiency of our algorithms.

The effectiveness of our algorithms was further validated by an application to spin dynamics in NMR, where GR and OGR successfully reconstructed different joint probability distributions of two Hamiltonian parameters.

Finally, we introduced a new class of GR algorithms designed to improve the data selection process for general function approximation problems while simultaneously minimizing the structure of the ansatz within a given family of approximation functions. Theoretical and numerical studies on polynomial interpoltion and neural network approximations highlighted the efficiency of our methods.

Overall, our greedy reconstruction algorithms offer significant potential for future research. A first extension of OGR to partial differential equations in [16], focusing on the reconstruction of unknown monotone nonlinear operators in semilinear elliptic models, has already proven to be very effective. The structure optimization features of our NOGR algorithm also appear highly promising, with a wide range of possible applications, for example minimizing the kernel sizes of convolutional neural networks for image classification.

Appendix A

Local convergence of the Gauss-Newton method

We provide a detailed proof for the local convergence of GN, inspired by [51, Theorem 2.4.1].

Theorem A.0.1 (local convergence of GN) *Let $\boldsymbol{\beta}_\star$ be a minimizer of (2.15) such that for all $m \in \{1, \dots, K\}$ the function R_m is Lipschitz continuously differentiable near $\boldsymbol{\beta}_\star$. Then there exist $\widehat{c}, \delta > 0$ such that if $\boldsymbol{\beta}_c \in \mathcal{B}_\delta(\boldsymbol{\beta}_\star)$, then $\widehat{W}_c = \sum_{m=1}^K R_m'(\boldsymbol{\beta}_c)^\top R_m'(\boldsymbol{\beta}_c)$ is nonsingular and the error in the GN iteration satisfies*

$$\|\boldsymbol{\beta}_+ - \boldsymbol{\beta}_\star\| \leq \widehat{c}\,\Big(\|\boldsymbol{\beta}_c - \boldsymbol{\beta}_\star\|^2 + \|\boldsymbol{\beta}_c - \boldsymbol{\beta}_\star\| \sum_{m=1}^K \|R_m(\boldsymbol{\beta}_\star)\|\Big), \tag{A.1}$$

where $\boldsymbol{\beta}_+$ denotes the next GN iterate:

$$\boldsymbol{\beta}_+ = \boldsymbol{\beta}_c - \Big(\sum_{m=1}^K R_m'(\boldsymbol{\beta}_c)^\top R_m'(\boldsymbol{\beta}_c)\Big)^{-1}\Big(\sum_{m=1}^K R_m'(\boldsymbol{\beta}_c)^\top R_m(\boldsymbol{\beta}_c)\Big). \tag{A.2}$$

Proof. Let L_m be the Lipschitz constant of R_m' for $m = 1, \dots, K$ and $\delta > 0$ be sufficiently small such that the matrix $\widehat{W}(\boldsymbol{\beta}) := \sum_{m=1}^K R_m'(\boldsymbol{\beta})^\top R_m'(\boldsymbol{\beta})$ is nonsingular for all $\boldsymbol{\beta} \in \mathcal{B}_\delta(\boldsymbol{\beta}_\star)$. The existence of such a δ follows by Lemma 5.7. Let $\boldsymbol{\beta}_c \in \mathcal{B}_\delta(\boldsymbol{\beta}_\star)$ and define $\boldsymbol{e}_c := \boldsymbol{\beta}_c - \boldsymbol{\beta}_\star$ and $\boldsymbol{e}_+ = \boldsymbol{\beta}_+ - \boldsymbol{\beta}_\star$. We have

$$\begin{aligned} R_m(\boldsymbol{\beta}_c) &= R_m(\boldsymbol{\beta}_\star) + \int_0^1 R_m'(\boldsymbol{\beta}_c + \lambda(\boldsymbol{\beta}_\star - \boldsymbol{\beta}_c))\boldsymbol{e}_c d\lambda \\ &= R_m(\boldsymbol{\beta}_\star) + R_m'(\boldsymbol{\beta}_c)\boldsymbol{e}_c + \int_0^1 \Big(R_m'(\boldsymbol{\beta}_c + \lambda(\boldsymbol{\beta}_\star - \boldsymbol{\beta}_c)) - R_m'(\boldsymbol{\beta}_c)\Big)\boldsymbol{e}_c d\lambda \end{aligned}$$

and hence

$$\begin{aligned} \|R_m'(\boldsymbol{\beta}_c)\boldsymbol{e}_c + R_m(\boldsymbol{\beta}_\star) - R_m(\boldsymbol{\beta}_c)\| &= \|\int_0^1 \Big(R_m'(\boldsymbol{\beta}_c + \lambda(\boldsymbol{\beta}_\star - \boldsymbol{\beta}_c)) - R_m'(\boldsymbol{\beta}_c)\Big)\boldsymbol{e}_c d\lambda\| \\ &\leq \int_0^1 \|R_m'(\boldsymbol{\beta}_c + \lambda(\boldsymbol{\beta}_\star - \boldsymbol{\beta}_c)) - R_m'(\boldsymbol{\beta}_c)\|\|\boldsymbol{e}_c\| d\lambda \end{aligned}$$

Using the Lipschitz continuity of R'_m with Lipschitz constant L_m, we obtain

$$\|R'_m(\boldsymbol{\beta}_c)\boldsymbol{e}_c + R_m(\boldsymbol{\beta}_\star) - R_m(\boldsymbol{\beta}_c)\| \leq \int_0^1 \lambda L_m \|\boldsymbol{\beta}_c - \boldsymbol{\beta}_\star\| \|\boldsymbol{e}_c\| d\lambda = \frac{L_m}{2}\|\boldsymbol{e}_c\|^2. \tag{A.3}$$

Since $\boldsymbol{\beta}_\star$ is a minimizer of (2.15), we also have

$$0 = \nabla\Big(\sum_{m=1}^K \frac{1}{2} R_m(\boldsymbol{\beta})^\top R_m(\boldsymbol{\beta})\Big)\Big|_{\boldsymbol{\beta}=\boldsymbol{\beta}_\star} = \sum_{m=1}^K R'_m(\boldsymbol{\beta}_\star)^\top R_m(\boldsymbol{\beta}_\star)$$

and thus

$$-\sum_{m=1}^K R'_m(\boldsymbol{\beta}_c)^\top R_m(\boldsymbol{\beta}_\star) = \sum_{m=1}^K \langle R'_m(\boldsymbol{\beta}_\star) - R'_m(\boldsymbol{\beta}_c), R_m(\boldsymbol{\beta}_\star)\rangle. \tag{A.4}$$

Using (A.2) and $\Big(\sum_{m=1}^K R'_m(\boldsymbol{\beta}_c)^\top R'_m(\boldsymbol{\beta}_c)\Big)^{-1}\Big(\sum_{m=1}^K R'_m(\boldsymbol{\beta}_c)^\top R'_m(\boldsymbol{\beta}_c)\Big) = \mathcal{I}_K$, we obtain

$$\begin{aligned}
\boldsymbol{e}_+ &= \boldsymbol{e}_c - \Big(\sum_{m=1}^K R'_m(\boldsymbol{\beta}_c)^\top R'_m(\boldsymbol{\beta}_c)\Big)^{-1}\Big(\sum_{m=1}^K R'_m(\boldsymbol{\beta}_c)^\top R_m(\boldsymbol{\beta}_c)\Big) \\
&= \Big(\sum_{m=1}^K R'_m(\boldsymbol{\beta}_c)^\top R'_m(\boldsymbol{\beta}_c)\Big)^{-1}\Big(\sum_{m=1}^K R'_m(\boldsymbol{\beta}_c)^\top \big(R'_m(\boldsymbol{\beta}_c)\boldsymbol{e}_c - R_m(\boldsymbol{\beta}_c)\big)\Big).
\end{aligned}$$

In summary, we have

$$\begin{aligned}
\|\boldsymbol{e}_+\| &= \Big\|\Big(\sum_{m=1}^K R'_m(\boldsymbol{\beta}_c)^\top R'_m(\boldsymbol{\beta}_c)\Big)^{-1}\Big(\sum_{m=1}^K R'_m(\boldsymbol{\beta}_c)^\top \big(R'_m(\boldsymbol{\beta}_c)\boldsymbol{e}_c - R_m(\boldsymbol{\beta}_c)\big)\Big)\Big\| \\
&\leq \underbrace{\Big\|\Big(\sum_{m=1}^K R'_m(\boldsymbol{\beta}_c)^\top R'_m(\boldsymbol{\beta}_c)\Big)^{-1}\Big\|}_{=:c_{\boldsymbol{\beta}_c}} \Big\|\sum_{m=1}^K R'_m(\boldsymbol{\beta}_c)^\top \big(R'_m(\boldsymbol{\beta}_c)\boldsymbol{e}_c - R_m(\boldsymbol{\beta}_c)\big)\Big\| \\
&= c_{\boldsymbol{\beta}_c}\Big\|\sum_{m=1}^K R'_m(\boldsymbol{\beta}_c)^\top\big(-R_m(\boldsymbol{\beta}_\star) + R'_m(\boldsymbol{\beta}_c)\boldsymbol{e}_c + R_m(\boldsymbol{\beta}_\star) - R_m(\boldsymbol{\beta}_c)\Big\| \\
&\leq c_{\boldsymbol{\beta}_c}\Bigg[\Big\|\sum_{m=1}^K R'_m(\boldsymbol{\beta}_c)^\top R_m(\boldsymbol{\beta}_\star)\Big\| + \sum_{m=1}^K \|R'_m(\boldsymbol{\beta}_c)\| \|R'_m(\boldsymbol{\beta}_c)\boldsymbol{e}_c + R_m(\boldsymbol{\beta}_\star) - R_m(\boldsymbol{\beta}_c)\|\Bigg] \\
&\overset{(A.3)}{\leq} c_{\boldsymbol{\beta}_c}\Bigg[\Big\|\sum_{m=1}^K R'_m(\boldsymbol{\beta}_c)^\top R_m(\boldsymbol{\beta}_\star)\Big\| + \sum_{m=1}^K \|R'_m(\boldsymbol{\beta}_c)\| \frac{L_m}{2}\|\boldsymbol{e}_c\|^2\Bigg]
\end{aligned}$$

$$
\begin{aligned}
&\overset{(A.4)}{\leq} c_{\boldsymbol{\beta}_c}\left[\sum_{m=1}^{K}\|\langle R'_m(\boldsymbol{\beta}_\star)-R'_m(\boldsymbol{\beta}_c),R_m(\boldsymbol{\beta}_\star)\rangle\|+\sum_{m=1}^{K}\|R'_m(\boldsymbol{\beta}_c)\|\frac{L_m}{2}\|\boldsymbol{e}_c\|^2\right]\\
&\leq c_{\boldsymbol{\beta}_c}\sum_{m=1}^{K}\Big(L_m\|\boldsymbol{\beta}_\star-\boldsymbol{\beta}_c\|\|R_m(\boldsymbol{\beta}_\star)\|+\|R'_m(\boldsymbol{\beta}_c)\|\frac{L_m}{2}\|\boldsymbol{e}_c\|^2\Big)\\
&\leq \underbrace{\Big(\max_{m\in\{1,\dots,K\}}L_m\Big)}_{=:L_{max}} c_{\boldsymbol{\beta}_c}\sum_{m=1}^{K}\frac{2\|\boldsymbol{e}_c\|\|R_m(\boldsymbol{\beta}_\star)\|+\|R'_m(\boldsymbol{\beta}_c)\|\|\boldsymbol{e}_c\|^2}{2}\\
&\leq L_{max}c_{\boldsymbol{\beta}_c}\Bigg[\sum_{m=1}^{K}\frac{2\|\boldsymbol{e}_c\|\|R_m(\boldsymbol{\beta}_\star)\|+\|R'_m(\boldsymbol{\beta}_c)\|\|\boldsymbol{e}_c\|^2}{2}+\|\boldsymbol{e}_c\|^2\\
&\qquad\qquad+\frac{\|R'_m(\boldsymbol{\beta}_c)\|\|R_m(\boldsymbol{\beta}_\star)\|\|\boldsymbol{e}_c\|}{2}\Bigg]\\
&= L_{max}c_{\boldsymbol{\beta}_c}\sum_{m=1}^{K}\frac{2+\|R'_m(\boldsymbol{\beta}_c)\|}{2}\Big(\|\boldsymbol{e}_c\|^2+\|\boldsymbol{e}_c\|\|R_m(\boldsymbol{\beta}_\star)\|\Big)\\
&\leq L_{max}c_{\boldsymbol{\beta}_c}\max_{i\in\{1,\dots,K\}}\left(1+\frac{\|R'_i(\boldsymbol{\beta}_c)\|}{2}\right)\Big(\|\boldsymbol{e}_c\|^2+\|\boldsymbol{e}_c\|\sum_{m=1}^{K}\|R_m(\boldsymbol{\beta}_\star)\|\Big)
\end{aligned}
$$

Setting

$$
\begin{aligned}
\widehat{c} &= L_{max}\max_{\substack{\boldsymbol{\beta}\in\mathcal{B}_\delta(\boldsymbol{\beta}_\star),\\ i\in\{1,\dots,K\}}} c_{\boldsymbol{\beta}}\left(1+\frac{\|R'_i(\boldsymbol{\beta})\|}{2}\right)\\
&= L_{max}\max_{\substack{\boldsymbol{\beta}\in\mathcal{B}_\delta(\boldsymbol{\beta}_\star),\\ i\in\{1,\dots,K\}}}\Big\|\Big(\sum_{m=1}^{K}R'_m(\boldsymbol{\beta})^\top R'_m(\boldsymbol{\beta})\Big)^{-1}\Big\|\left(1+\frac{\|R'_i(\boldsymbol{\beta})\|}{2}\right)
\end{aligned}
$$

completes the proof. □

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

DGR	Data greedy reconstruction algorithm
EIM	Empirical interpolation method
GN	Gauss-Newton method
GR	Greedy reconstruction algorithm
GRPD	Greedy reconstruction algorithm for probability distributions
LGR	Linearized greedy reconstruction algorithm
NMR	Nuclear magnetic resonance
NOGR	Network optimized greedy reconstruction algorithm
ODE	Ordinary differential equation
OGR	Optimized greedy reconstruction algorithm
OGRPD	Optimized greedy reconstruction algorithm for probability distributions
PD	Positive definite
PSD	positive semi-definite

Bibliography

[6] A. Alexanderian et al. "A fast and scalable method for A-optimal design of experiments for infinite-dimensional Bayesian nonlinear inverse problems". In: *SIAM J. Sci. Comput.* 38.1 (2016), A243–A272.

[7] C. D. Aliprantis and K. C. Border. *Infinite Dimensional Analysis: a Hitchhiker's Guide*. Springer, 2006.

[8] J. Ansel et al. "PyTorch 2: Faster machine learning through dynamic Python bytecode transformation and graph compilation". In: *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*. ASPLOS '24. New York, NY, USA: Association for Computing Machinery, 2024, pp. 929–947. ISBN: 9798400703850. DOI: 10.1145/3620665.3640366.

[9] P. J. Antsaklis and A. N. Michel. *Linear Systems*. Birkhäuser Basel, 2006.

[10] J.-P. Argaud, B. Bouriquet, F. de Caso, H. Gong, Y. Maday, and O. Mula. "Sensor placement in nuclear reactors based on the generalized empirical interpolation method". In: *Journal of Computational Physics* 363 (2018), pp. 354–370. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2018.02.050.

[11] A. C. Atkinson. *Optimum Experimental Design*. Springer Berlin Heidelberg, 2011.

[12] J. Baglama, D. Calvetti, and L. Reichel. "Fast Leja points." eng. In: *ETNA. Electronic Transactions on Numerical Analysis [electronic only]* 7 (1998), pp. 124–140. URL: http://eudml.org/doc/119747.

[13] J. Baglama, D. Calvetti, and L. Reichel. "Iterative methods for the computation of a few eigenvalues of a large symmetric matrix". In: *BIT Numerical Mathematics* 36 (1996), pp. 400–421. URL: https://doi.org/10.1007/BF01731924.

[14] M. Barrault, Y. Maday, N. Nguyen, and A. Patera. "An 'empirical interpolation' method: Application to efficient reduced-basis discretization of partial differential equations". In: *Comptes Rendus Mathematique* 339 (Nov. 2004), pp. 667–672. DOI: 10.1016/j.crma.2004.08.006.

[15] A. R. Barron, A. Cohen, W. Dahmen, and R. A. DeVore. "Approximation and learning by greedy algorithms". In: *The Annals of Statistics* 36.1 (2008), pp. 64 –94. DOI: 10.1214/009053607000000631.

[16] J. Bartsch, S. Buchwald, G. Ciaramella, and S. Volkwein. *Reconstruction of unknown nonlinear operators in semilinear elliptic models using optimal inputs*. 2024. DOI: 10.48550/arXiv.2405.12153.

[17] L. Baudouin and A. Mercado. "An inverse problem for Schrödinger equations with discontinuous main coefficient". In: *Appl. Anal.* 87.10-11 (2008), pp. 1145–1165.

[18] F. Bauer and C. Fike. "Norms and exclusion theorems". In: *Numer. Math.* 2 (1960), pp. 137–141.

[19] I. Bauer, H. G. Bock, S. Körkel, and J. P. Schlöder. "Numerical methods for optimum experimental design in DAE systems". In: *J. Comput. Appl. Math.* 120.1 (2000), pp. 1 –25.

[20] M. Bebendorf, Y. Maday, and B. Stamm. "Comparison of some reduced representation approximations". In: *Reduced Order Methods for Modeling and Computational Reduction*. Ed. by A. Quarteroni and G. Rozza. Cham: Springer International Publishing, 2014, pp. 67–100. ISBN: 978-3-319-02090-7. DOI: 10.1007/978-3-319-02090-7_3.

[21] C. Berge. *Topological spaces: including a treatment of multi-valued functions, vector spaces and convexity; Translated by E.M. Patterson*. Oliver & Boyd Edinburgh, 1963.

[22] D. Bernstein and W. So. "Some explicit formulas for the matrix exponential". In: *IEEE Transactions on Automatic Control* 38.8 (1993), pp. 1228–1232. DOI: 10.1109/9.233156.

[23] J.-P. Berrut and L. N. Trefethen. "Barycentric Lagrange interpolation". In: *SIAM Review* 46.3 (2004), pp. 501–517. DOI: 10.1137/S0036144502417715.

[24] S. Bonnabel, M. Mirrahimi, and P. Rouchon. "Observer-based Hamiltonian identification for quantum systems". In: *Automatica* 45.5 (2009), pp. 1144 –1155.

[25] A. Borzì, G. Ciaramella, and M. Sprengel. *Formulation and numerical solution of quantum control problems*. Philadelphia, PA: SIAM, 2017.

[26] L. Bos, S. De Marchi, A. Sommariva, and M. Vianello. "Computing multivariate Fekete and Leja points by numerical linear algebra". In: *SIAM Journal on Numerical Analysis* 48.5 (2010), pp. 1984–1999. DOI: 10.1137/090779024.

[27] L. Bris et al. "Hamiltonian identification for quantum systems: Well posedness and numerical approaches". In: *ESAIM: Contr. Optim. Ca.* 13.2 (2007), pp. 378–395.

[28] C. G. Broyden. "The convergence of a class of double-rank minimization algorithms 1. General considerations". In: *IMA Journal of Applied Mathematics* 6.1 (Mar. 1970), pp. 76–90. ISSN: 0272-4960. DOI: 10.1093/imamat/6.1.76.

[29] G. Ciaramella and A. Borzì. "SKRYN: A fast semismooth-Krylov–Newton method for controlling Ising spin systems". In: *Computer Physics Communications* 190 (2015), pp. 213–223.

[30] P. G. Ciarlet. *Linear and Nonlinear Functional Analysis with Applications*. Applied mathematics. SIAM, Philadelphia, PA, 2013.

[31] J. Coron. *Control and Nonlinearity*. Math. Surveys Monogr. AMS, 2007.

[32] P. Davis. *Interpolation and Approximation*. Massachussetts: Blaisdell Publishing Company, 1963.

[33] B. De Schutter. "Minimal state-space realization in linear system theory: An overview". In: *J. Comput. Appl. Math.* 121.1-2 (2000), pp. 331–354. ISSN: 0377-0427.

[34] R. Denk and R. Racke. *Kompendium der ANALYSIS - Ein kompletter Bachelor-Kurs von Reellen Zahlen zu Partiellen Differentialgleichungen*. Vieweg+Teubner Verlag Wiesbaden, 2011. DOI: 10.1007/978-3-8348-8184-7.

[35] A. Donovan and H. Rabitz. "Exploring the Hamiltonian inversion landscape". In: *Phys. Chem.* 16 (29 2014), pp. 15615–15622.

[36] A. Edrei. "Sur les déterminants récurrents et les singularités d'une fonction donnée par son développement de Taylor". fre. In: *Compositio Mathematica* 7 (1940), pp. 20–88. URL: http://eudml.org/doc/88739.

[37] R. Fletcher. "A new approach to variable metric algorithms". In: *The Computer Journal* 13.3 (Jan. 1970), pp. 317–322. ISSN: 0010-4620. DOI: 10.1093/comjnl/13.3.317.

[38] Y. Fu and G. Turinici. "Quantum Hamiltonian and dipole moment identification in presence of large control perturbations". In: *ESAIM: Contr. Optim. Ca.* 23.3 (2017), pp. 1129–1143.

[39] J. M. Geremia and H. Rabitz. "Global, nonlinear algorithm for inverting quantum-mechanical observations". In: *Phys. Rev. A* 64 (2 2001), p. 022710.

[40] J. M. Geremia and H. Rabitz. "Optimal Hamiltonian identification: The synthesis of quantum optimal control and quantum inversion". In: *J. Chem. Phys.* 118.12 (2003), pp. 5369–5382.

[41] J. M. Geremia, W. Zhu, and H. Rabitz. "Incorporating physical implementation concerns into closed loop quantum control experiments". In: *J. Chem. Phys.* 113.24 (2000), pp. 10841–10848.

[42] S. Glaser et al. "Training Schrödinger's cat: quantum optimal control". In: *Eur. Phys. J. D* 69.12 (2015), p. 279.

[43] D. Goldfarb. "A family of variable-metric methods derived by variational means". In: *Mathematics of Computing* 24 (1970), pp. 23–26. DOI: 10.1090/S0025-5718-1970-0258249-6.

[44] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. Cambridge, MA: MIT Press, 2016.

[45] M. A. Grepl, Y. Maday, N. C. Nguyen, and A. T. Patera. "Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations". In: *ESAIM: M2AN* 41.3 (2007), pp. 575–605. DOI: 10.1051/m2an:2007031.

[46] K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[47] E. Herberg, R. Herzog, F. Köhne, L. Kreis, and A. Schiela. *Sensitivity-based layer insertion for residual and feedforward neural networks*. 2023. arXiv: 2311.15995 [cs.LG].

[48] R. A. Horn and C. R. Johnson. *Matrix Analysis*. 2nd. USA: Cambridge University Press, 2012. ISBN: 0521548233.

[49] R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis*. Cambridge Univ. Press, 1991.

[50] J. N. Juang and R. S. Pappa. "An eigensystem realization algorithm for modal parameter identification and model reduction". In: *J. Guid. Control Dynam.* 8.5 (1985), pp. 620–627.

[51] C. T. Kelley. *Iterative Methods for Optimization*. Philadelphia: SIAM, 1999.

[52] K. Kobzar, T. E. Skinner, N. Khaneja, S. J. Glaser, and B. Luy. "Exploring the limits of broadband excitation and inversion: II. Rf-power optimized pulses". In: *J. Magn. Reson.* 194.1 (2008), pp. 58–66. ISSN: 1090-7807.

[53] M. Lapert, Y. Zhang, M. A. Janich, S. J. Glaser, and D. Sugny. "Exploring the physical limits of saturation contrast in magnetic resonance imaging". In: *Sci. Rep.* 2 (2012), p. 589.

[54] F. Leja. "Sur certaines suites liées aux ensembles plan et leur application à la représentation conforme". In: *Ann. Polon. Math.* 4 (1957), pp. 8–13.

[55] M. Levitt. *Spin dynamics: Basics of nuclear magnetic resonance*. Wiley, 2013. ISBN: 9781118681848.

[56] K. Lundengård. "Generalized Vandermonde matrices and determinants in electromagnetic compatibility". PhD thesis. 2017.

[57] Y. Maday and J. Salomon. "A greedy algorithm for the identification of quantum systems". In: *Proceedings of the 48th IEEE Conference on Decision and Control, 2009, Held jointly whit the 28th Chinese Control Conference (CDC/CCC 2009)*. IEEE Conference on Decision and Control. 2009, pp. 375–379.

[58] Y. Maday and O. Mula. "A generalized empirical interpolation method: Application of reduced basis techniques to data assimilation". In: *Analysis and Numerics of Partial Differential Equations*. Ed. by F. Brezzi, P. Colli Franzone, U. Gianazza, and G. Gilardi. Milano: Springer Milan, 2013, pp. 221–235. ISBN: 978-88-470-2592-9. DOI: 10.1007/978-88-470-2592-9_13.

[59] Y. Maday, N. C. Nguyen, A. T. Patera, and S. H. Pau. "A general multipurpose interpolation procedure: the magic points". In: *Communications on Pure and Applied Analysis* 8.1 (2009), pp. 383–404. ISSN: 1534-0392. DOI: 10.3934/cpaa.2009.8.383.

[60] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical Mathematics (Texts in Applied Mathematics)*. Berlin, Heidelberg: Springer-Verlag, 2006. ISBN: 3540346589.

[61] K. Rasheed, A. Qayyum, M. Ghaly, A. Al-Fuqaha, A. Razi, and J. Qadir. "Explainable, trustworthy, and ethical machine learning for healthcare: A survey". In: *Computers in Biology and Medicine* 149 (2022), p. 106043. ISSN: 0010-4825. DOI: https://doi.org/10.1016/j.compbiomed.2022.106043.

[62] L. Reichel. "Newton interpolation at Leja points". In: *BIT* 30 (1990), pp. 332–346.

[63] Rodrigues. "Des lois géométriques qui régissent les déplacements d'un système solide dans l'espace, et de la variation des coordonnées provenant de ces déplacements considérés indépendamment des causes qui peuvent les produire." fre. In: *Journal de Mathématiques Pures et Appliquées* (1840), pp. 380–440.

[64] W. Rudin. *Real and Complex Analysis, 3rd Ed.* USA: McGraw-Hill, Inc., 1987. ISBN: 0070542341.

[65] D. Shanno. "Conditioning of quasi-Newton methods for function minimization". In: *Mathematics of Computing* 24 (1970), pp. 647–656. DOI: 10.1090/S0025-5718-1970-0274029-X.

[66] T. E. Skinner, T. O. Reiss, B. Luy, N. Khaneja, and S. J. Glaser. "Tailoring the optimal control cost function to a desired output: application to minimizing phase errors in short broadband excitation pulses". In: *J. Magn. Reson.* 172.1 (2005), pp. 17–23. ISSN: 1090-7807.

[67] E. D. Sontag. *Mathematical Control Theory: Deterministic Finite Dimensional Systems (2Nd Ed.)* Berlin, Heidelberg: Springer-Verlag, 1998.

[68] L. N. Trefethen and D. Bau. *Numerical Linear Algebra.* SIAM, Philadelphia, PA, 1997.

[69] Y. Wang et al. "A quantum Hamiltonian identification algorithm: Computational complexity and error analysis". In: *IEEE Trans. Autom. Control* 63.5 (2018), pp. 1388–1403.

[70] H. Wendland. "Haar spaces and multivariate polynomials". In: *Scattered Data Approximation.* Cambridge Monographs on Applied and Computational Mathematics. Cambridge: Cambridge University Press, 2004, pp. 18–23.

[71] E. F. Whittlesey. "Analytic functions in Banach spaces". In: *Proceedings of the American Mathematical Society* 16.5 (1965), pp. 1077–1083. (Visited on 01/16/2023).

[72] S. Xue, R. Wu, D. Li, and M. Jiang. "A gradient algorithm for Hamiltonian identification of open quantum systems". In: *Phys. Rev. A* 103 (2021), p. 022604.

[73] J. Zhang and M. Sarovar. "Quantum Hamiltonian identification from measurement time traces". In: *Phys. Rev. Lett.* 113 (8 2014), p. 080401.

[74] W. Zhou, S. Schirmer, E. Gong, H. Xie, and M. Zhang. "Identification of Markovian open system dynamics for qubit systems". In: *Chinese Sci. Bull.* 57.18 (2012), pp. 2242–2246.