UNIVERSITY OF CALIFORNIA
RIVERSIDE

Closed-Form and Robust Expressions for the Data-Driven Control of Centralized and Distributed Systems

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Mechanical Engineering

by

Federico Celi

March 2024

Dissertation Committee:

Dr. Fabio Pasqualetti, Chairperson
Dr. Wei Ren
Dr. Erfan Nozari

The Dissertation of Federico Celi is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

> *"Beate le marionette,* sospirai*, su le cui teste di legno il finto cielo si conserva senza strappi! Non perplessità angosciose, né ritegni, né intoppi, né ombre, né pietà; nulla! E possono attendere bravamente e prender gusto alla loro commedia e amare e tener se stesse in considerazione e in pregio, senza soffrir mai vertigini o capogiri, poiché per la loro statura e per le loro azioni quel cielo è un tetto proporzionato."*[1]
>
> — Luigi Pirandello, *Il fu Mattia Pascal* (1904)

I extend my deepest gratitude to my advisor, Prof. Fabio Pasqualetti, whose guidance has been an invaluable contribution in this journey. Fabio embodies daily the multifaceted roles of a graduate advisor: teacher, leader, collaborator. His ability to challenge and refine ideas, provide students with support during moments of struggle, and acknowledge their achievements has positively shaped my growth as a researcher.

I am also indebted to Prof. Giacomo Baggio, whose commitment to the highest quality in every step of the research process has set an exemplary role for me and my work. Giacomo's leading-by-example approach has been a guiding light. Giacomo sat down with me (albeit virtually) every time a proof would not click with me, patiently showing me the way and motivating me to always go a step further. The quality of this work, and of my research going forward, owes much to you.

To Prof. Jorge Cortés, Dr. Ahmed Allihboy, and Dr. Jaimie Swartz, my heartfelt appreciation for enduring with me in the intricate and non-linear process that is research. Your dedication, collaborative spirit, and the constructive environment you fostered have

[1] Blessed be the puppets, *I sighed*, on whose wooden heads the fake sky is preserved without tears! No perplexing anxieties, nor hesitations, nor hindrances, nor shadows, nor pitty; nothing! And they can patiently wait, enjoy their comedy, love, esteem themselves, without ever suffering dizziness or giddiness, as such a sky is a proportionate roof for their stature and actions.

been instrumental in the success of this work. I am equally grateful to every remarkable researcher and scientist I have had the privilege to collaborate with over the years, including friends and mentors back at the University of Pisa, Georgia Tech, Amazon and NATO's CMRE. Each contribution has played a role in shaping my academic journey. Thank you all for your enduring support and shared commitment to advancing knowledge in our fields.

I express my sincere gratitude to my esteemed committee members, Prof. Wei Ren and Prof. Erfan Noziari, who have been guides throughout my graduate school journey. It has been a genuine honor to engage with them, benefiting from their insights and feedback. Their support and constructive input have positively shaped the development of my research questions, enriching the overall depth and quality of my work. I extend my thanks to Joe Patterson for the insights and effort in drafting and improving this work.

I cannot thank enough the people that made my time at the University of California, Riverside, so enjoyable every day. Thank you in particular to my lab mates throughout the years: Abed, Darshan, Taosha, Shivanshu, Manuel, Alberto, Silvia, Umberto, Sarbendu and Shiqi. We never had enough hikes, camping trips, Thanksgiving dinners together. A special thought goes to Tommaso and Lasata, and to Yuzhen and Shen-Shen for the deep friendship we cultivated throughout the years and for the late-night/long-drive conversations about Life, the Universe and Everything.

I am grateful to my friends and family back in Italy. Your love, support and trust made this possible. I will avoid listing the names of all the influential people in this category, but if you happen to read these lines one day, know you are among them.

Finally, I wish to thank, immensely, Boyu. Thank you for enduring with me

throughout the years, navigating through my grad-school-induced-mood-swings, and for always having the right thing to say in any occasion. I always felt your support, whether we were working in the same room, or waking up half a world apart. Thank you for believing in me, especially in those moments when I had lost hope. I cannot wait to see what is next.

To my friends, my family, and to Boyu, my bridge over troubled waters.

"E respirando brezze che dilagano su terre senza limiti e confini

Ci allontaniamo e poi ci ritroviamo più vicini,

E più in alto e più in là,

Ora figli dell'immensità."[2]

— Lucio Battisti & Mogol, *La collina dei ciliegi* (1973)

[2]And breathing through the zephyrs, spreading borderless and free, // we drift away, then in reunion find our plea, // and higher, farther, you shall see // forever progeny of immensity.

ABSTRACT OF THE DISSERTATION

Closed-Form and Robust Expressions for the Data-Driven Control
of Centralized and Distributed Systems

by

Federico Celi

Doctor of Philosophy, Graduate Program in Mechanical Engineering
University of California, Riverside, March 2024
Dr. Fabio Pasqualetti, Chairperson

The traditional approach for the control of dynamical systems relies on the availability of a model describing the system to be controlled. Typically, a model is derived from first principles; however the feasibility of this strategy is threatened as the scope of controller design is expanding to increasingly complex domains. Alternatively, data can be used to identify a system's model when this is unavailable and challenging to derive from first principles. A pipeline for control design then becomes a two-step process: (i) use data to identify a model of the system, (ii) synthesize a controller for the identified model.

Recently, there has been a growing interest in exploring control strategies for unmodelled systems that directly leverage data in one-step approach, with the expectation that this will improve the efficiency and performance of the overall process. The interest for this direct approach to data-driven control was a consequence of the streak of successes of machine learning for tasks such as classification, image generation and token prediction. Such a remarkable progress by the machine learning community came at the price of formality: in fact, a true understanding of the drawbacks and limitations of this modeling

approach is missing. While failures might pose a minor risk in some consumer applications, these become unacceptable and potentially life-threatening in control applications. The challenge for modern data-driven control is therefore to capitalize on these advancements while grounding any result in a formal framework, where performance metrics, robustness concerns and system-theoretic properties are all first-class citizens.

This thesis introduces a comprehensive approach for designing controllers from data. Specifically, it formalizes a methodology that yields closed-form and robust solutions for a variety of control problems. Traditional control problems are re-examined through a data-driven lens, from establishing a system's observability, to designing optimal controllers for network systems. In practical terms, given a desired control objective and (possibly) noisy data from the system, the desired control action can be determined by feeding the data directly into the proposed expression, bypassing system identification. The result of this approach is a straightforward controller design technique with formally characterizable robustness bounds.

Contents

List of Figures **xii**

1 Introduction **1**
1.1 Data, modeling and control: a short long history 1
1.2 Statement of contributions and thesis overview 6
1.3 Notation 10

2 Background **12**
2.1 Linear, discrete, time-invariant systems and data 12
2.2 Data-based free and forced responses 15
2.3 Data-based system-theoretic properties 18

3 Expressions for optimal open-loop input-output data-driven control **21**
3.1 Introduction 21
3.2 Data-driven formulas for open-loop LQ control 23
3.3 Minimum energy control 28
3.4 Datasets with heterogeneous control horizons 32
3.5 Robustness of data-driven open-loop LQ control 39
3.5.1 Data-driven LQ control with known noise statistics 40
3.5.2 Data-driven LQ control with unknown noise statistics 43
3.6 Conclusions 45
3.7 Appendix 46

4 Estimation of the optimal state-feedback control from data **56**
4.1 Introduction 57
4.2 Related work 58
4.3 Closed-form state-output data-driven controls 59
4.4 Convergence to the LQR gain 62
4.5 Numerical analysis 68
4.6 Conclusions 72
4.7 Appendix 72
4.7.1 Convergence to static optimal controller 73

4.7.2 Exponential convergence of the state 74

5 Data-driven eigenstructure assignment & sparse pole placement 76
5.1 Introduction 77
5.2 Related work 78
5.3 Pole placement and eigenstructure assignment 79
5.4 Data-driven pole placement and eigenstructure assignment 82
5.5 Data-driven pole placement with sparse feedback matrices 86
5.5.1 Data-driven minimum-gain pole placement with sparse static feedback 87
5.5.2 Data-driven maximally sparse feedback 90
5.6 Conclusions 91

6 A data-driven approach to geometric control 93
6.1 Introduction 93
6.2 Related work 95
6.3 Controlled and conditioned invariant subspaces 96
6.4 Data-driven geometric control 98
6.5 Data-driven feedbacks for invariant subspaces and invariant zeros 103
6.6 Malicious attacks: an illustrative example 107
6.7 Conclusions 109

7 Distributed data-driven control for network systems 110
7.1 Introduction 110
7.2 Related work 111
7.3 Setup for multiagent learning 113
7.4 Multiagent learning of optiamal controls 116
7.5 Numerical results and illustrative examples 128
7.5.1 An application to scaling ring networks 129
7.5.2 An application to Watts-Strogatz networks 130
7.5.3 Receding horizon implementation 130
7.5.4 Comparison with splitting methods 131
7.5.5 Comparison with [Alonso et al., 2022] with noisy data 132
7.6 Conclusions 134

8 Conclusions 137

Bibliography 140

List of Figures

3.1 This figure shows the output trajectory of the system in Example 8 with $\mathbf{u}_T$ computed through (3.5). Data is collected as described in (2.2), with X_0 and U random matrices with i.i.d. entries. The experiment is run with $Q_t = \mathbf{blkdiag}(10, 10, 10, 100, 100, 100)$, $R_t = I_m$, for all $t = \{1, \dots, T\}$, and $T = 150$. Panel (a) shows the trajectories of the position (solid lines) and the orientation angles (dashed lines). Panel (b) shows the trajectory of the position in 3-D space (circle: initial position, cross: final position). 29

3.2 Panels (a)-(b) show the norm of the minimum energy control computed through: (i) the model-based formula $\mathbf{u}_T = C_T^\dagger(x_\mathrm{f} - A^T x_0)$, where C_T is the controllability matrix defined in Lemma 3, (ii) the controllability Gramian with $u(t) = B^\mathsf{T}(A^\mathsf{T})^{T-t-1} W_T^\dagger(x_\mathrm{f} - A^T x_0)$, (iii) the exact data-driven expression (3.13), and (iv) the asymptotically correct expression (3.16). The underlying system is generated randomly, with $n = 20$, $m = 2$, $T = 40$, $x_\mathrm{f} = [1\, 1 \cdots 1]^\mathsf{T}$ and N is specified on the x-axis. The curves represent the average over 100 experiments with input data generated as random i.i.d. normal entries. Notice that, in accordance with the conditions imposed by Lemma 2, the expression (3.12) becomes exact once $N = n + mT = 80$ linearly independent experimental trajectories have been collected. Further, notice that (3.16) returns a feasible (the control satisfies the constraint) yet suboptimal (the cost achieved by the control is not optimal) solution for finite values of N. Panels (c)-(d) show the norm of the inputs $\mathbf{u}_T$ computed as above, and the corresponding errors in the final state, as a function of the system dimension n. We note that while the Gramian-based solution is technically exact, it is effectively less robust than the exact data-driven formulas in (3.13) and (3.16) for systems with a large number of states. This loss of numerical precision is a consequence of the computational process involved in computing and manipulating the controllability Gramian. We let $m = 2$, $T = n$, $N = mT + 20$, and $x_\mathrm{f} = [1\, 1 \cdots 1]^\mathsf{T}$. The matrices A and B are generated randomly. 30

3.3 This figure conceptually shows the data collection phase with heterogeneous datasets. 33

3.4 This figure shows an example of decomposition of a input-state trajectory in sub-trajectories: a trajectory of length $T = 10$ is divided in three sub-trajectories of lengths $T_1 = 4$, $T_2 = 2$, $T_3 = 4$. 35
3.5 This figure shows the asymptotic consistency of equation (3.20) for problem (3.2). The left panel shows how the controller computed with a noisy dataset with noise compensation in (3.20) (solid lines) asymptotically converges to the optimal controller $\mathbf{u}_T^*$ for (3.2), as opposed to the non-compensated expression in (3.6) (dashed lines), for varying noise statistics. The experiment is performed over a randomly generated system with $n = 5$, $m = 3$, $p = 2$ and $T = 15$. Finally, $\sigma_0 = \sigma_Y = 0.1$, while different values of σ_U are shown in the legend. The right panel further supports this result by showing how the difference between P and P_c evolves as the number of experiment increases and the noise in the data is appropriately accounted for as shown in Theorem 13 (here, $\sigma_U = 0.5$). 53
3.6 This figure shows the convergence results of Theorem 15. As implied by the theorem, although the noise statistics remain unknown, the difference between the desired final output and that computed with a noisy dataset approaches zero as the number of experiments increases. The experiment is performed over a randomly generated system with $n = 5$, $m = 3$, $p = 2$ and $T = 15$. The noise on the output is additive, gaussian, and with distribution $\mathcal{N}(1, 0.5)$. 54

4.1 This figure shows the convergence of K_{CF} in (4.21) to K^* in (4.7). The underlying system is that of Example 22. For each choice of time horizon T we plot $\|K^* - K_{\text{CF}}\|$ on a logarithmic scale. 65
4.2 This figure supports the results of this chapter. In particular, we compute K_{LQR}^0 using Theorem 20 for increasing T. As expected from (4.26) the distance between $\mathbf{U}_0$ and K_{LQR}, computed as the 2-norm of their difference, decreases as T grows. The experiment is performed over a randomly generated system with $n = 5$ and $m = 2$. 67
4.3 In this figure we show the performance of (4.21) when concatenating multiple trajectories as proposed in Section 4.5. The underlying system is generated randomly, with $n = 16$, $m = 4$. For each combination of time horizon T and number of concatenated trajectories r we plot $\|K^* - K_{\text{CF}}\|$ on a logarithmic scale, where K^* is the LQR gain (4.7). 70

4.4 This figure shows the performance of the approach described in Theorem 19 (direct) to estimate the LRQ gain K^* for systems of increasing size. In the top panel we plot the error of the estimated K_{CF} (4.21) as $\|K^* - K_{\text{CF}}\|$ when one optimal trajectory is used (upwards triangles) and when five optimal trajectories are used (downwards triangles), see Section 4.5. In the bottom panel we plot the computational time to run the algorithm (in seconds). We compare this to the strategy proposed in [De Persis and Tesi, 2020] (dots). We notice how our approach scales comparatively when estimating K^*, at least when multiple trajectories are used. At the same time (4.21) offers a more computationally-efficient strategy, especially for larger values of n. The simulation parameters are $m = 5$, $T = 150$, $N = mT + n$. The system matrices are generated randomly for each n. 71

6.1 This figure shows a comparison for computing $\boldsymbol{\mathcal{V}}^*$ with two different data driven approaches for minimal system [Katayama, 2005, Example 6.6]. Outputs Y are collected with noise, i.e., $Y = \hat{Y} + \Delta_Y$, where Y is the measured data and Δ_Y is the noise matrix with i.i.d. entries, zero mean, and variance σ_Y. In blue (dashed) we show the result obtained trough the MOESP algorithm (see [Katayama, 2005]). In red (solid) we show the result obtained trough the approach proposed in this chapter (DDA). State trajectories X are assumed to be known only by MOESP, while DDA requires only X_0. For every approach and for every value of σ_Y we perform a total of 100 Montecarlo simulations and plot the mean value of the angle between the estimated and the model based $\boldsymbol{\mathcal{V}}^*$ (true subspace). $T = 50$ for MOESP, and $T = 3$, $N = 20$ for DDA. 103

6.2 An example of consensus network. On the left, agents are numbered from 1 through 14, where nodes $\{12, 13, 14\}$ (in black) are the leaders and nodes $\{4, 11\}$ (square) are the network monitors. On the right, the weighted adjacency matrix for the follower nodes $\{1, \dots, 11\}$ is shown, together with the input and output matrices (the numerical values of the entries of the matrices are color coded and belong to the set $\{0, 0.2, 0.4, 0.6, 0.8\}$). 107

6.3 In this figure we show an attack on the network of Fig. 6.2. The systems initial condition is chosen randomly and the leaders impose $u = [-2\ 2\ 4]^\top$. The attacker waits for the system to reach its equilibrium and then, at time $t = 10s$, injects an attack A_T as proposed in Sec 6.6. We notice how the system state its perturbed from the equilibrium, while the output of the system remains unaffected by the attack, rendering the attack action effectively invisible at the output. We use the following parameters: time horizon $T = 49 > n$, and number of experimental trajectories $N = n + mT = 158$, with X_0 and U satisfying assumption in (2.4). 108

7.1 In this figure we show a visual depiction of the data collection phase. N experiments of length T are carried out over a network system. Each agent $i \in \{1, \dots, M\}$ collects input-state trajectories of a subset of the network nodes, i.e., the initial state $x^j_{0,i}$, the state trajectory $\mathbf{x}^j_{T,i}$, and the input trajectory $\mathbf{u}^j_{T,i}$, for each experiment j. 115
7.2 This figure shows the results associated with the experiment of Section 7.5.1. In panel (a) we show the communication graph of an example network, with $M = 6$ and $d = 3$. All other networks in this example are similar in structure (ring network), with varying diameters $d \in \{3, \dots, 8\}$. All experiments shown in panel (b) are performed with $T = 5$, $\varepsilon = 10^{-3}$ and the tolerance for the pseudoinverse operation is set to $\texttt{tol} = 10^{-8}$. 129
7.3 This figure shows the results associated with the experiment of Section 7.5.2. In panel (a) we show a randomly generated Watts-Strogats network with $M = 30$ agents. In panel (b) we show the convergence of Algorithm 1, comparing it to the model-based centralized solution of (7.2). Additional simulation parameters are $T = 5$, $\varepsilon = 10^{-3}$ and the tolerance for the pseudoinverse operation is set to $\texttt{tol} = 10^{-8}$. 130
7.4 This figure shows a comparison between the approach in Algorithm 2 and a distributed data-driven solver based on [Allibhoy and Cortés, 2020] (splitting method). The experiments are performed with noiseless data from the same system, a randomly generated network with $M = 3$ agents, and diverse state size. This comparison highlights the *computational time to accuracy trade-off* that the approaches have with respect to a particular design parameter, namely the number of iterations of the splitting method, and the parameter ε for Algorithm 2. The runtime is reported in seconds (lower is better), and the distance from the optimal input is computed as the norm of the difference of the solution and the exact optimal control computed through a model-based LQR solver (lower is better). Additional simulation parameters are $T = 5$, $\texttt{tol} = 10^{-8}$. 133
7.5 Comparison of the robustness of two different distributed data driven controllers with noisy data. The underlying model is a chain of four interconnected bi-dimensional oscillators, as discussed in [Alonso et al., 2022, Sec. VI]. Problem (7.2) is considered with $Q = 1$, $R = 1$, and random initial condition. The top panel shows the difference between the reference state evolution and the solution of Algorithm 2 (left) and of [Alonso et al., 2022] (right) when data are collected without noise. As expected, both methods converge to the equilibrium ($T = 10$). In the middle panel the same comparison is shown when data are collected with noise (each input and state trajectory is perturbed with an additive i.i.d. disturbance with zero mean and variance $\sigma^2 = 0.1$). In the bottom panel we run the same simulation for data collected with variance $\sigma^2 = 5$. $\varepsilon = 0.01$ and $\texttt{tol} = 10^{-8}$. 135

Chapter 1

Introduction

"Anything that is in the world when you're born is normal and ordinary and is just a natural part of the way the world works. Anything that's invented between when you're fifteen and thirty-five is new and exciting and revolutionary and you can probably get a career in it. Anything invented after you're thirty-five is against the natural order of things."

— Douglas Adams, *The Salmon of Doubt* (2002)

1.1 Data, modeling and control: a short long history

Leveraging data in science and technology is not a novel concept. In fact, data collection and analysis are at the very core of the scientific process as testified, for example, by Galileo Galilei and his legendary experiments in the late 1500s in Pisa, Italy. Galileo sought to identify a mathematical model to explain the space-time evolution of a free-falling object. To do so, he dropped objects of different masses from his hometown's belltower, the Leaning Tower of Pisa, and recorded the time it took them to reach the ground. His goal was

to demonstrate that the dynamic of a free-falling object is independent of the object's mass. Performing experiments and collecting an appropriate data-set was then a crucial step of the broader process aimed at validating or disproving some theory. In recent years, a notable shift has occurred, wherein data has assumed a more central role, largely attributed to the unprecedented abundance of data and a widespread democratization of data processing capabilities. Galileo (or more probably his grad students) had to climb up a tower multiple times to perform a sufficiently large number of experiments, and consequently process the collected data for hours on end to come up with a model of the observed phenomenon. Meanwhile, today data can be collected semiautomatically, and are automatically processed by cluster (or even personal) computers (while a grad student is probably comfortably scrolling their favorite social media app).

Whereas modeling a system is useful to predict how a system will evolve, control theory goes a step beyond: not only is a control scientist interested in the evolution of a system that is left free to interact with its environment (e.g., the falling of a body from 50 meters above ground) but on how the system will react as a consequence of some exogenous stimulation, known as control input. This is how, through human ingenuity, a free falling body becomes an aircraft. Clearly, this introduces numerous new challenges. In order for the control scientist to confidently characterize the degree by which an aileron needs to move to ensure that a plane can navigate through a certain turbulence, a near-prefect model of the input (aileron angle) to output (aircraft configuration) description is needed. In this context, this description can be made available to the control scientist by an aerospace engineer, through first principles. Modeling from first principles involves

developing a mathematical model of a system based on fundamental physical principles, laws, and relationships governing the system's behavior. Instead of relying on empirical data or experimental observations, this approach aims to derive the model directly from the underlying physics. In the context of aircraft configuration, modeling from first principles would involve understanding and applying the fundamental principles of aerodynamics, mechanics, and other relevant physical laws.

A common way to describe the model of a system is the so-called state-space representation. The main idea behind this approach it to represent the system's dynamics through a set of variables, namely the internal state (x), input (u), and output (y), whose evolution in time is described as a set of first-order differential or difference equations, e.g., $\dot{x}(t) = Ax(t)+Bu(t)$ and $y(t) = Cx(t)+Du(t)$, where we refer to the state-space model of the input-output dynamics as $\Sigma = (A, B, C, D)$. The end goal for a control scientist is to design a function f_{control}, usually referred to as control law, with $u(t) = f_{\text{control}}(\Sigma, \hat{y}, t)$ and such that the output $y(t)$ of the system is as close as possible to some desired output $\hat{y}(t)$.[1] The state-space representation was developed starting from the 1960s, following the groundbreaking works of Rudolf E. Kalman [Kalman, 1960]. The state-space approach became popular as it provided a unified framework for modeling linear time-invariant systems, making it easier to analyze and design controllers through formal tools such as linear algebra and convex optimization. The result of this is a streamlined theory that allows to study a system's properties, and that provides the tools to predict and control its behavior while formally characterizing the convergence and robustness properties of a given control law f_{control}.[2]

[1] Here, $t \geq 0$ is time, x, u and y are real or complex valued vectors, and A, B, C, D are real or complex valued matrices. Notice that here we are restricting the class of systems under investigation to the linear and time invariant. This will be introduced formally in Chapter 2.

[2] In the context of control theory, convergence and robustness are key concepts. At a high level, a control

First-principles modeling, i.e., finding $\Sigma = (A, B, C, D)$ for a given system from fundamental physical laws, is the focus of an extensive cross-fields scientific effort, from engineering to the life sciences, and integrating data in this process has always been contemplated and necessary. However, as mentioned, lately data have been gaining an ever increasing role in system modeling. Maybe not surprisingly, Kalman himself, together with his student Bin-Lun Ho, started formalizing an approach for finding Σ when successfully rich experiments $\mathcal{D}$ could be collected for the system under investigation. Briefly, the Ho-Kalman approach to system identification and control [Ho and Kalman, 1966, Katayama, 2005] is a two step approach: (i) identify a model (Σ) of the system through the available data ($\mathcal{D}$), then (ii) base the design of the control law (f_{control}) on the model identified in step one. This was an important contribution, at a time where data collection and elaboration was expensive compared to today's standards, and where a model would ensure "backwards compatibility" with those approaches already developed for a model-based-first world. Ultimately, however, the goal of a control engineer is to control a system, so a model is not technically needed if enough information could be extracted from data. The question then naturally arises: why not basing the design of the control strategy f_{control} directly on $\mathcal{D}$?

The arguments in favor of this mind-shift are multiple, but are importantly motivated by the progress achieved by computer scientists over the better part of the last decade in machine learning and related artificial intelligence applications.[3] The promise of machine

law f_{control} is said to converge if $y(t)$ gets closer to the desired output $\hat{y}(t)$ as time t increases. The study of the robustness of f_{control}, instead, aims at characterizing how this convergence holds (or breaks) if some uncertainty is present in the process, e.g., when the model Σ is not perfect.

[3]The taxonomy of machine and deep learning, and broadly of artificial intelligence, is diverse and in constant evolution. The reader is referred to monographs [Russell and Norvig, 2021, Goodfellow et al., 2016] and the references therein for a more accurate description of this fascinating filed than the one offered in this work.

learning is that, given a problem to solve (e.g., binary classification) a proper architecture (e.g., a deep neural network) and enough data samples, a solution can be *automatically* found by minimizing a loss function. Of course, this is not a one-size-fits all solution and many hours of human ingenuity go into the process of finding the right architecture and data labeling strategy for the problem of interest. Once the right approach is found, however, this results in a fairly automated process, with more data generally implying better results. Unfortunately, this solution-driven strategy comes with some downsides for control applications. In fact, while there is plenty of anecdotal evidence for the successes of machine learning, there is little understanding by the scientific community on why and how this approach actually achieves this remarkable performance. Most crucially, its is by now common knowledge that these architectures, including neural networks, are brittle: perturbing the inputs might (and will) affect the outcome of a neural network, with potentially catastrophic consequences [Szegedy et al., 2014]. A famous example of this phenomenon is how a neural network used to classify road signs can be easy fooled to classify a "STOP" sign as a "45 mph" sign when this is just partially covered by some tape, in a way that would fool no human [Eykholt et al., 2018].

Therefore, while machine learning solutions offer a promising way to automate the design of a controller, the cost of loosing the structured framework typical of control theory and the possibility to formally prove results, for example, about the convergence and robustness of the approach is a downside that a control scientist cannot afford. This is the context in which, at the end of the 2010s, the momentum for a modern approach to *direct data-driven* control suddenly arose [Baggio et al., 2019, Coulson et al., 2019, De Persis

and Tesi, 2020] (these works will be discussed in detail throughout this work, together with other important papers from the community). The end goal of the pioneers that kickstarted this sub-field of control theory is to capitalize on the successes and advances of the machine learning community while preserving the formal sandbox of control theory. Great progress has ben made since then. The design of a control law directly from data has, in less than a decade, graduated to a filed of its own. This is suggested for example by the fact that at the 62th IEEE Conference on Decision and Control, held in Singapore in December of 2023, four out of the eighty sessions were specifically dedicated to data-driven control, and about 70 papers mentioned the word "data-driven" in their title (or 5% of the venue's total). This thesis sits comfortably in this category and offers a framework to approach this problem from a control-theoretic data-first point of view.

1.2 Statement of contributions and thesis overview

Throughout this thesis we develop strategies for the design of direct data-driven controllers for discrete-time, linear, and time-invariant systems. Whenever possible we focus on solutions resulting in closed-form expressions, as opposed to optimization-based expressions. In practice, we begin by developing the framework at the core of the approach, which is based on a data-driven formulation of the state-space approach to modeling and control (Chapter 2). Building on this framework we develop closed-form expressions for both open-loop (Chapter 3) and closed-loop (Chapter 4) optimal controllers, and include formal bounds for the robustness properties of these controllers to noise in the data. To show the versatility and applicability of the proposed framework to a broad spectrum of problems, we

suggest a strategy to design feedback controllers for (possibly sparse) pole placement and eigenstructure assignment problems (Chapter 5). Interestingly, this framework naturally leads to a geometric interpretation of control theory based on data (Chapter 6). Finally, we discuss how to distribute this framework in a networked scenario (Chapter 7).

Our main premise is that, especially for systems with linear time-invariant dynamics, system-theoretic properties and control methods are well-understood and have been developed over the years, ranging from tests and algorithms in the frequency domain [Åström and Murray, 2010] to methods using state-space [Kailath, 1980] and geometric computations [Basile and Marro, 1991], among others. The analysis, design and synthesis of control methods based on the system *behaviors* [Willems and Polderman, 1997] is certainly possible and interesting, but perhaps most useful only if complemented with an understanding of when a specific problem should be solved within a specific domain (e.g., frequency, state space, behaviors) and using a specific algorithm. This type of questions, which has been present, for instance in the machine learning research to characterize the tradeoffs between generative and discriminative models [Ng and Jordan, 2001], has instead received only scarce attention in the context of data-driven control. As we shall see, while data-driven and model-based methods are theoretically equivalent in the absence of noise and assuming perfect computations, in practice, the methods can differ considerably in the way they propagate uncertainties thus leading unexpectedly to different results. Furthermore, even within the data-driven framework, different formalisms can lead to expressions with different complexity, interpretability and performance, thus motivating careful analysis and comparisons. We refer the interested reader to [Markovsky and Dörfler, 2021] for a more

comprehensive survey of data-driven control methods and the behavioral framework.

The main contributions for each chapter of this thesis are detailed below.

Chapter 2 In this chapter we introduce the problem setting for data-driven control and review the existing literature on the topic. We also introduce some preliminary definitions and discuss the foundational results of this work, which will be used throughout the thesis.

Chapter 3 In this chapter we develop upon the results presented in Chapter 2 and show how these can be used to solve a series of optimal control problems. In particular, we discuss the solutions to the open-loop linear-quadratic control problem, as well as simplified formulas for the particular case of minimum energy control. We further show how these results can be extended to deal with datasets of heterogeneous length, and discuss how this approach compares to alternative methods in the literature.

Chapter 4 The contribution of this chapter is twofold. First, we give a closed-form solution to design quadratically optimal controllers for an unknown linear system. The solution is based on the assumption that sufficient data can be collected and used to directly compute the controller of interest. No further assumption is imposed on the data, namely, it does not have to be optimal. Differently from other direct data-driven solutions, the approach presented in this work is in closed-form. Second, we leverage the data-driven controls from above to compute a feedback of the state which asymptotically converges to the Linear Quadratic Regulator (LQR) gain. We show through simulations how our approach offers a good tradeoff between numerical accuracy and computational efficiency.

Chapter 5 This chapter presents novel results on pole placement and eigenstructure assignment with sparse feedback achieved by using (possibly direct) data-driven formulas.

Specifically, we characterize (i) the allowable eigenvector subspace and (ii) the set of feedback gains which solve the pole placement problem, both as a function of data. From these, we derive (iii) a closed-form expression of the gain that solves the eigenstructure assignment problem. Additionally, we (iv) discuss strategies for computing sparse feedback controllers for the pole placement problem, by incorporating our data-driven expressions into nonlinear optimization problems. Finally, numerical simulations demonstrate the effectiveness of the proposed approach.

Chapter 6 This chapter aims at linking the geometric approach to control design with data-driven methods, with numerous contributions. First, for the linear, discrete, time-invariant systems described by the triple (A, B, C), we derive explicit, closed-form data-driven expressions of (i) $\boldsymbol{\mathcal{V}}^*$, the largest $(A, \mathbf{Im}(B))$-controlled invariant subspace contained in $\mathbf{Ker}(C)$, (ii) $\boldsymbol{\mathcal{S}}^*$, the smallest $(A, \mathbf{Ker}(C))$-conditioned invariant subspace containing $\mathbf{Im}(B)$, (iii) the feedback gain F such that $(A+BF)\boldsymbol{\mathcal{V}}^* \subseteq \boldsymbol{\mathcal{V}}^*$, and (iv) the invariant zeros of (A, B, C). Since $\boldsymbol{\mathcal{V}}^*$ and $\boldsymbol{\mathcal{S}}^*$ are the basis of the geometric approach developed in [Basile and Marro, 1991], our data-driven formulas constitute the basis of a data-driven and model-free theory of geometric control. Second, our results show that the fundamental invariant subspaces of the geometric approach, which are often computed recursively when operating in the state space, have a simple and direct interpretation in the higher-dimensional data space, where they can be computed by solving appropriately defined sets of linear equations. Third, we demonstrate the utility of our formulas to design undetectable data-driven attacks.

Chapter 7 In this chapter we leverage the tools developed thought this thesis to develop a data-driven and distributed algorithm to learn optimal controls in a multi-agent environ-

ment. This approach is data-driven as the control design relies exclusively on prerecorded input-state trajectories of an unknown linear system, and it is distributed as it assumes that the recorded trajectories are not available to any single agent, but are partitioned throughout the network. The goal of the agents is to compute the control action which globally minimizes a given quadratic cost function of the states and inputs, from a given initial condition. To do this, we leverage an algorithm based on iterative projections in order to distribute the computation of the control in a network of agents with partial access to data. This work builds upon and significantly expands the approach presented in previous chapters of this thesis and departs from the cited literature in a number of ways. First, it does not require data from past recordings to be shared (at least directly) among the networks' nodes, offering an implicit layer of privacy. Moreover, it relies on a closed-form expression that provably converges to a solution, with a prescribed distance from optimality, after a finite number of iterations of the algorithm bounded by the diameter of the network.

1.3 Notation

The mathematical notation used in this thesis is summarized below.

(i) We let $\mathbb{R}$ and $\mathbb{N}$ denote the set of real and non-negative integer numbers, respectively.

(ii) Given a matrix $A \in \mathbb{R}^{n\times m}$, we let $\mathbf{Rank}(A)$, $\mathbf{Basis}(A)$, $\mathbf{Ker}(A)$, A^{T}, $\sigma_{\min}(A)$, $\sigma_{\max}(A)$ denote the rank, a basis of the column space, the kernel, the transpose, and the smallest and largest singular values of A, respectively.

(iii) We use $\{0\}$ to denote the trivial subspace containing only the origin. Given a matrix

A and a subspace $\mathcal{V}$ of appropriate dimensions, $A^{-1}\mathcal{V}$ denotes the pre-image of $\mathcal{V}$ by the, possibly singular, matrix A.

(iv) $V = \mathbf{Basis}(\mathcal{V})$ denotes any full-column rank matrix such that $\mathbf{Im}(V) = \mathcal{V}$.

(v) We let $\mathbf{blkdiag}(A_1, \ldots, A_n)$ be the block diagonal matrix with blocks $A_i \in \mathbb{R}^{n_i \times m_i}$.

(vi) We denote the Moore-Penrose pseudoinverse of matrix A with $A^\dagger$.

(vii) The Kronecker product between matrices A and B is denoted by $A \otimes B$ [Bernstein, 2009].

(viii) We indicate the 2-norm of a matrix or vector with $\|\cdot\|_2$.

(ix) We let $A \succ 0$ ($A \succeq 0$) denote a positive definite (positive semidefinite) matrix.

(x) We let $\mathbf{vec}(A)$ be the vectorization of matrix A.

(xi) For a positive semidefinite matrix $W \in \mathbb{R}^{n \times n}$ and vector $x \in \mathbb{R}^n$, we let $\|x\|_W = \sqrt{x^\mathsf{T} W x}$. I_n and $0_{n,m}$ denote the $n \times n$ identity matrix and $n \times m$ zero matrix, respectively (subscripts will be omitted when clear from the context).

(xii) For a random vector $x \colon \Omega \to \mathbb{R}^n$, we let $\mathbb{P}[x \in S]$ and $\mathbb{E}[x]$ be the probability that x takes on a value in a set $S \subseteq \mathbb{R}^n$ and the expected value of x, respectively. We let a.s. denote almost surely, and $\xrightarrow{\text{a.s.}}$ almost sure convergence.

Chapter 2

Background

In this chapter we introduce the technical framework upon which lies the entirety of this thesis. In particular, we introduce the class of system under investigation and the assumptions we make for the data collection phase. We further introduce some of the foundational results of our approach, namely how to compute a data-based representation of the free and the forced response of a system, from data. We then briefly discuss how our framework compares with alternative approaches in the literature for data-driven control design.

2.1 Linear, discrete, time-invariant systems and data

Throughout this work, we study the problem of designing control inputs for linear time-invariant systems with the aim to solve a variety of optimal control and robustness problems. We do this without knowing the system dynamics and by, instead, leveraging a set of pre-recorded input-output trajectories, together with state trajectories when needed.

In particular, we consider systems with linear, discrete-time, time-invariant dynamics of the form

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t), \end{aligned} \tag{2.1}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$ and $y \in \mathbb{R}^p$ are the state, input and output vectors at time $t \in \mathbb{N}$, respectively, and the matrices A, B, C and D are unknown. We assume that inputs U, states X, and outputs Y are available from a set of $N \in \mathbb{N}$ control experiments with finite horizon $T \in \mathbb{N}$:

$$U = \begin{bmatrix} \mathbf{u}_T^1 & \cdots & \mathbf{u}_T^N \end{bmatrix} \in \mathbb{R}^{mT \times N}, \tag{2.2a}$$

$$X_0 = \begin{bmatrix} x^1(0) & \cdots & x^N(0) \end{bmatrix} \in \mathbb{R}^{n \times N}, \tag{2.2b}$$

$$X = \begin{bmatrix} \mathbf{x}_T^1 & \cdots & \mathbf{x}_T^N \end{bmatrix} \in \mathbb{R}^{nT \times N}, \tag{2.2c}$$

$$Y = \begin{bmatrix} \mathbf{y}_T^1 & \cdots & \mathbf{y}_T^N \end{bmatrix} \in \mathbb{R}^{pT \times N}. \tag{2.2d}$$

In (2.2), $\mathbf{u}_T^i$, $\mathbf{x}_T^i$, and $\mathbf{y}_T^i$ are the vectors containing the inputs, states, and outputs of the i-th experiment:

$$\mathbf{u}_T^i = \mathbf{vec}(u^i(0), \ldots, u^i(T-1)), \qquad (i\text{-th input trajectory})$$

$$\mathbf{x}_T^i = \mathbf{vec}(x^i(1), x^i(2), \ldots, x^i(T)), \qquad (i\text{-th state trajectory})$$

$$\mathbf{y}_T^i = \mathbf{vec}(y^i(0), \ldots, y^i(T-1)). \qquad (i\text{-th output trajectory})$$

We remark that the full input-state-output dataset (2.2) is not always needed to solve the problems described throughout this work; the subset of required data will be specified based on the problem at hand. Further, when convenient, we will use the following matrices, which

can be extracted from (2.2):

$$X_F = \begin{bmatrix} x^1(T) & \cdots & x^N(T) \end{bmatrix} \in \mathbb{R}^{n \times N}, \tag{2.3a}$$

$$Y_F = \begin{bmatrix} y^1(T-1) & \cdots & y^N(T-1) \end{bmatrix} \in \mathbb{R}^{p \times N}. \tag{2.3b}$$

For the noiseless system (2.1), knowledge of the system matrices A, B, C and D is equivalent to the availability of a (sufficiently rich) dataset (2.2). In fact, any dataset (2.2) can be generated with the matrices A, B, C and D using (2.1), and, in turn, such matrices can be reconstructed uniquely (under mild data rank conditions) using the dataset (2.2):[1]

$$X_m^+ = \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} X_m^- \\ U_m \end{bmatrix}, \text{ and } Y_m = \begin{bmatrix} C & D \end{bmatrix} \begin{bmatrix} X_m^- \\ U_m \end{bmatrix},$$

where

$$\begin{aligned}
U_m &= \begin{bmatrix} u^1(0), \ldots, u^1(T-1), \ldots, u^N(0), \ldots, u^N(T-1) \end{bmatrix}, \\
X_m &= \begin{bmatrix} x^1(0), \ldots, x^1(T), \ldots, x^N(0), \ldots, x^N(T) \end{bmatrix}, \\
X_m^- &= \begin{bmatrix} x^1(0), \ldots, x^1(T-1), \ldots, x^N(0), \ldots, x^N(T-1) \end{bmatrix}, \\
X_m^+ &= \begin{bmatrix} x^1(1), \ldots, x^1(T), \ldots, x^N(1), \ldots, x^N(T) \end{bmatrix}, \\
Y_m &= \begin{bmatrix} y^1(0), \ldots, y^1(T), \ldots, y^N(0), \ldots, y^N(T-1) \end{bmatrix}.
\end{aligned}$$

Clearly, the system matrices can be computed uniquely whenever the matrix $\begin{bmatrix} X_m^- \\ U_m \end{bmatrix}$ is full row rank. While this analysis seems to discourage the pursuit of data-driven methods, since sufficiently-rich datasets are effectively a model of the system dynamics, in this thesis we will show that, instead, data-driven computations allow for alternative and sometimes more

[1] Similar formulas are contained also in [De Persis and Tesi, 2020], among others.

insightful, direct, and computationally-favorable solutions to classic control problems, thus contributing to the theory of systems and enriching our control tools. We remark that a detailed analysis of the tradeoffs between direct data-driven methods and classic identification-based approaches in noisy settings is much more nuanced [Krishnan and Pasqualetti, 2021], deserves a dedicated treatment, and will not be addressed here. Rather, we will show how our direct solutions obtained with noiseless data can be used to study, and modified to counteract, the effect of noise and perturbations on the collected datasets.

2.2 Data-based free and forced responses

We begin by stating and proving a technical result.

Lemma 1 (Rank of block matrices). *Let $A \in \mathbb{R}^{n_1 \times m}$ and $B \in \mathbb{R}^{n_2 \times m}$, with $m \geq n_1 + n_2$. Let $K_A = \mathbf{Basis}(\mathbf{Ker}(A))$ and $K_B = \mathbf{Basis}(\mathbf{Ker}(B))$. The following statements are equivalent:*

(i) $\mathbf{Rank}\left(\begin{bmatrix} A \\ B \end{bmatrix}\right) = n_1 + n_2$;

(ii) $\mathbf{Rank}(AK_B) = n_1$ *and* $\mathbf{Rank}(BK_A) = n_2$.

Proof.

((ii) implies (i).) We will show that statement (ii) is violated when statement (i) is violated. Let vectors v_A and v_B satisfy $v_A A + v_B B = 0$, with $[v_A \ v_B] \neq 0$. Then, $v_A A K_B + v_B B K_B = v_A A K_B = 0$, which implies that either $v_A = 0$ or $\mathbf{Rank}(AK_B) < n_1$. Similarly, $v_A A K_A + v_B B K_A = v_B B K_A = 0$, which implies that either $v_B = 0$ or

$\mathbf{Rank}(BK_A) < n_2$. Since v_A and v_B cannot be simultaneously zero, we conclude that statement (ii) implies (i).

((i) implies (ii).) Notice that

$$\mathbf{Rank}\left(\begin{bmatrix} A \\ B \end{bmatrix}\right) = \mathbf{Rank}\left(\begin{bmatrix} A \\ B \end{bmatrix} C\right),$$

with C any invertible matrix of appropriate dimension. Let $C = [K_B \; B^{\mathsf{T}}]$. Then,

$$\begin{aligned} n_1 + n_2 = \mathbf{Rank}\left(\begin{bmatrix} A \\ B \end{bmatrix}\right) &= \mathbf{Rank}\left(\begin{bmatrix} A \\ B \end{bmatrix} \begin{bmatrix} K_B & B^{\mathsf{T}} \end{bmatrix}\right) \\ &= \mathbf{Rank}\left(\begin{bmatrix} AK_B & AB^{\mathsf{T}} \\ 0 & BB^{\mathsf{T}} \end{bmatrix}\right), \end{aligned}$$

which implies that $\mathbf{Rank}(AK_B) = n_1$. Repeating the reasoning with $C = [K_A \; A^{\mathsf{T}}]$ concludes the proof. ■

Going back to the problem at hand, a powerful result underlying most data-driven approaches is Willems' Fundamental Lemma [Willems et al., 2005], which, loosely speaking, gives a sufficient conditions under which any T-long trajectory of the system (2.1) can be constructed as linear combinations of those in an appropriately constructed input-output dataset [Markovsky and Dörfler, 2021]. We next state a reformulation of Willems' Fundamental Lemma that uses the dataset (2.2) and that allows us to distinguish between the trajectories of (2.1) obtained with and without a control input. This result will be instrumental for our derivations.

Lemma 2 (Data-based free and forced representation) *Let* (2.2)-(2.3) *be the data*

generated by the system (2.1). *Assume that*

$$\mathbf{Rank}\left(\begin{bmatrix} X_0 \\ \\ U \end{bmatrix}\right) = mT + n. \tag{2.4}$$

Let $K_U = \mathbf{Basis}(\mathbf{Ker}(U))$ *and* $K_0 = \mathbf{Basis}(\mathbf{Ker}(X_0))$. *Then, for any initial state* x_0 *and input* $\mathbf{u}_T$, *there exist vectors* α *and* β *such that* $x_0 = X_0 K_U \alpha$ *and* $\mathbf{u}_T = U K_0 \beta$. *Moreover,*

$$\begin{bmatrix} \mathbf{x}_T \\ \\ \mathbf{y}_T \end{bmatrix} = \begin{bmatrix} XK_U & XK_0 \\ \\ YK_U & YK_0 \end{bmatrix} \begin{bmatrix} \alpha \\ \\ \beta \end{bmatrix} \tag{2.5}$$

are the state and output trajectories of length T *of* (2.1) *generated by* x_0 *and* $\mathbf{u}_T$.

Proof. Using (2.4) and Lemma 1, the matrices $X_0 K_U$ and $U K_0$ are full-row rank. Hence, for every x_0 and $\mathbf{u}_T$ there exist α and β such that $x_0 = X_0 K_U \alpha$ and $\mathbf{u}_T = U K_0 \beta$. Notice that the data matrices satisfy the relations

$$\begin{bmatrix} X \\ \\ Y \end{bmatrix} = \begin{bmatrix} O_T^X & F_T^X \\ \\ O_T^Y & F_T^Y \end{bmatrix} \begin{bmatrix} X_0 \\ \\ U \end{bmatrix}, \tag{2.6}$$

where

$$O_T^X = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^T \end{bmatrix}, \qquad F_T^X = \begin{bmatrix} B & \cdots & 0 & 0 \\ AB & \cdots & 0 & 0 \\ & \ddots & & \\ A^{T-1}B & \cdots & AB & B \end{bmatrix},$$

$$O_T^Y = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{T-1} \end{bmatrix}, \qquad F_T^Y = \begin{bmatrix} D & \cdots & 0 & 0 \\ CB & \cdots & 0 & 0 \\ & \ddots & & \\ CA^{T-2}B & \cdots & CB & D \end{bmatrix}.$$

Notice that

$$XK_U = O_T^X X_0 K_U + F_T^X U K_U = O_T^X X_0 K_U,$$

$$XK_0 = O_T^X X_0 K_0 + F_T^X U K_0 = F_T^X U K_0.$$

Then, the state trajectory x_T of (2.1) with input $\mathbf{u}_T = UK_0\beta$ and initial state $x_0 = X_0 K_U \alpha$ can be written as

$$\mathbf{x}_T = O_T^X x_0 + F_T^X \mathbf{u}_T = O_T^X X_0 K_U \alpha + F_T^X U K_0 \beta$$

$$= XK_U\alpha + XK_0\beta.$$

The claimed expression for $\mathbf{y}_T$ is obtained similarly using the matrices O_T^Y and F_T^Y, thus concluding the proof. ■

Lemma 2.5 states that any trajectory of the system (2.1) can be written as a linear combination of a collection of previously-recorded trajectories. Yet, Lemma 2.5 provides a more granular decomposition of the trajectories of the system (2.1) given data, as it identifies the *free response* of the system from the initial condition x_0, namely $XK_U\alpha$, and the *forced response* of the system from the input $\mathbf{u}_T$, namely $XK_0\beta$.

2.3 Data-based system-theoretic properties

In addition to being of general interest, the formulas that we discussed thus far allow for the recorded data to be used to predict the system trajectories given the initial condition and input sequence, rather than just as a description of the system dynamics, and to analyze certain system-theoretic properties in a purely data-driven manner without requiring the identification of the system dynamics (see also [Celi and Pasqualetti, 2022b]).

Lemma 3 (Data-driven reachability and observability). *Let the data matrices X_0 and U satisfy* (2.4)*. Then*

$$\mathbf{Rank}(O_T^Y) = \mathbf{Rank}(YK_U), \tag{2.7}$$

where $O_T^Y = \begin{bmatrix} C^\mathsf{T} & (CA)^\mathsf{T} & \dots & (CA^{T-1})^\mathsf{T} \end{bmatrix}^\mathsf{T}$*, is the T-steps observability matrix of* (2.1)*. Similarly,*

$$\mathbf{Rank}(C_T) = \mathbf{Rank}(X_F K_0), \tag{2.8}$$

where $C_T = \begin{bmatrix} A^{T-1}B & \dots & AB & B \end{bmatrix}$ *is the T-steps controllability matrix of* (2.1) *and X_F is as in* (2.3)*.*

Proof. Notice from (2.6) that $YK_U = O_T^Y X_0 K_U$. From [Bernstein, 2009, Fact 2.10.2], $\mathbf{Im}(O_T^Y X_0 K_U) = \mathbf{Im}(O_T^Y)$ since $X_0 K_U$ is full row-rank (cif. (2.4) and Lemma 1). Hence, $\mathbf{Rank}(O_T^Y) = \mathbf{Rank}(YK_U)$. Similarly, $X_F K_0 = C_T U K_0$ and UK_0 is full row-rank, thus implying that $\mathbf{Rank}(C_T) = \mathbf{Rank}(X_F K_0)$, which concludes the proof. ■

Lemma 3 relates the rank of the T-step observability matrix O_T^Y to the data-driven matrix YK_U. Clearly, when $T \geq n$, the system (2.1) is observable if and only if $\mathbf{Rank}(YK_U) = n$. Similar comments hold in (2.8) for the reachability subspace of (2.1).

Remark 4 (Single vs multiple data trajectories) *A single experimental trajectory may be sufficient to obtain a data-driven representation of the system dynamics. A single trajectory, in fact, is used in Willems' Fundamental Lemma and in several reformulations of this result, e.g., [Willems et al., 2005, Yu et al., 2021, Verhoek et al., 2021, Schmitz et al., 2022, Lopez and Müller, 2022]. We remark that the use of multiple trajectories, as we do in Lemma 2, carries some advantages. First, the formulas in Lemma 2 remain valid if only a*

single, long, trajectory is available. In fact, the case of a single, long trajectory organized as a Hankel matrix is a special case of our formalism. To see this, using the notation in [van Waarde et al., 2020b], the data collected from a single trajectory of length τ is organized as

$$\mathcal{H}_1(x) = \begin{bmatrix} x(0) & x(1) & \cdots & x(\tau - T) \end{bmatrix},$$

$$\mathcal{H}_T(u) = \begin{bmatrix} u(0) & u(1) & \cdots & u(\tau - T) \\ \vdots & \vdots & & \vdots \\ u(T-1) & u(T) & \cdots & u(\tau - 1) \end{bmatrix}.$$

Clearly, one can set $X_0 = \mathcal{H}_1(x)$ and $\mathbf{u}_T^i$ equal to the i-th column of $\mathcal{H}_T(u)$ to equivalently express the data as in our framework. Thus, considering multiple, short trajectories as in (2.2) *effectively generalizes and includes the case of a single, long trajectory. Second, the use of multiple trajectories is convenient when the dynamics are unstable, since the system needs to be simulated for a shorter time horizon compared to the case of a single trajectory. More generally, using multiple trajectories produces data matrices that are numerically better conditioned and yield more reliable computations. Finally, the use of multiple trajectories is convenient from a statistical perspective when the collected data is corrupted by noise [Tu et al., 2022].* □

Remark 5 (State vs output measurements) *We assume here that the state of the system* (2.1) *can be directly measured. Notice, however, that this is not a restrictive assumption since a state measurement can be replaced with a finite window of inputs and outputs to solve appropriate control problems. In fact, the dynamics of* (2.1) *can be equivalently written using only inputs and outputs as done, e.g., in [Al Makdah et al., 2022] for the data-driven LQG control problem.* □

Chapter 3

Expressions for optimal open-loop input-output data-driven control

The possibility of expressing the free and forced responses of a dynamical system as a direct function of data, as shown in the previous chapter, unlocks the possibility to design control algorithms based entirely on this data-centric representation. We begin our journey by solving a typical problem in control, that is: "given a dynamical system in some initial condition, which is the most *efficient* way to take it to a different desired condition?". As should be clear by now, the novelty of our approach stems from solving the above question when data should be leveraged in place of a system's model.

3.1 Introduction

Optimal control is among the most successful and diverse sub-fields at the intersection of control theory and mathematics, and has been credited with enabling numerous

societal breakthroughs, e.g., the Moon Landing [Taylor, 2019, Miller et al., 2022]. While keeping our eyes on the stars, in this chapter we will humbly keep our feet on the ground by introducing a general framework for solving optimal control problems through the lenses of a data-driven approach. In particular, we will focus on a minimization problem of the form

$$\begin{aligned} &\text{minimize} && \text{some cost function over input } u \text{ and output } y \\ &\text{subject to} && u \text{ and } y \text{ trajectories being compatible with observed data.} \end{aligned} \tag{3.1}$$

We will restrict the above cost function to a quadratic function of the inputs and outputs, weighted according to some desired metric. For example, one could decide to minimize the energy consumed by a system, in which case the cost function could simply be $\|u\|_2^2$ (cf. Section 3.3). We remark that minimizing $\Sigma_{t=0}^{T-1} u(t)$ would also be technically correct, and the two problems have indeed the same solution. However the quadratic nature of the cost function offers some mathematical advantages by making the above problem convex. Naturally, one could consider a maximization problem when substituting the cost function with some performance metric. In fact, the two are effectively the same problem, except for a minus sign in front of the cost function / performance metric. Given the quadratic nature of the cost function and the linear relationship between the system's inputs and outputs, this problem is usually referred to as a Linear-Quadratic optimization problem. Of course, optimal control is this and much much more. We refer the interested reader to the references in [Zhou et al., 1996, Boyd and Vandenberghe, 2004] for a more exhaustive list of problems and relevant work at the intersection of control and optimization.

3.2 Data-driven formulas for open-loop LQ control

Following [Celi et al., 2023a], we start by studying the LQ control problem

$$\begin{aligned} \underset{u,\,x,\,y}{\text{minimize}} \quad & \sum_{t=0}^{T-1} \left(\|y(t)\|_{Q_t}^2 + \|u(t)\|_{R_t}^2 \right) \\ \text{subject to} \quad & x(t+1) = Ax(t) + Bu(t), \\ & y(t) = Cx(t) + Du(t), \\ & x(0) = x_0,\ y(T-1) = y_{\mathrm{f}}, \end{aligned} \tag{3.2}$$

where $Q_t \succeq 0$ and $R_t \succ 0$ are the (time-varying) output and input weighting matrices, x_0 is the initial state, and y_{f} the desired value of the output at time $T-1$. Throughout this chapter, we assume that (2.1) is output controllable in $T-1$ steps[1] to guarantee feasibility of (3.2) for any choice of $x_0 \in \mathbb{R}^n$ and $y_f \in \mathbb{R}^p$, and that the rank condition (2.4) holds. Problem (3.2) generalizes the classic (open-loop) linear-quadratic control framework by including the possibility of minimizing a linear function of the state (as opposed to the whole state) in addition to the control input. Let $\mathbf{u}_T = \mathbf{vec}(u(0), \dots, u(T-1))$ be the vector of inputs, and

$$\begin{aligned} Q &= \mathbf{blkdiag}(Q_0, \dots, Q_{T-1}), \\ R &= \mathbf{blkdiag}(R_0, \dots, R_{T-1}). \end{aligned} \tag{3.3}$$

Using the notation from Chapter 1.3, we now present a closed-form solution to the LQ control problem (3.2) that relies only on the data collected in (2.2). First, we notice that the cost in (3.2) can be written in vector form as

$$\sum_{t=0}^{T-1} \left(\|y(t)\|_{Q_t}^2 + \|u(t)\|_{R_t}^2 \right) = \mathbf{y}_T^{\mathsf{T}} Q \mathbf{y}_T + \mathbf{u}_T^{\mathsf{T}} R \mathbf{u}_T.$$

[1] System (2.1) is output controllable in T steps if $\begin{bmatrix} CA^{T-1}B & \cdots & CAB & CB & D \end{bmatrix}$ has full row rank. We remark that Lemma 3 can be adapted to verify whether a system is output controllable directly from data, substituting X_F with Y_F.

Second, from Lemma 2 we have $\mathbf{y}_T = YK_U\alpha + YK_0\beta$ and $\mathbf{u}_T = UK_0\beta$, for some vectors α and β. Then, by letting $K = \begin{bmatrix} K_U & K_0 \end{bmatrix}$, the input and output trajectories can be equivalently written as

$$\mathbf{y}_T = YK \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \text{ and } \mathbf{u}_T = UK \begin{bmatrix} \alpha \\ \beta \end{bmatrix},$$

respectively, and the cost in (3.2) becomes

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix}^{\mathsf{T}} \Big((YK)^{\mathsf{T}} Q (YK) + (UK)^{\mathsf{T}} R (UK) \Big) \begin{bmatrix} \alpha \\ \beta \end{bmatrix}.$$

Similarly, the equality constraints in (3.2) can be written as

$$\begin{bmatrix} X_0 \\ Y_F \end{bmatrix} K \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} x_0 \\ y_{\mathrm{f}} \end{bmatrix}.$$

The above reasoning allows us to reformulate the LQ control problem (3.2) as the data-based problem

$$\begin{aligned} \underset{\gamma}{\text{minimize}} \quad & \|L\gamma\|_2^2 \\ \text{subject to} \quad & W\gamma = z, \end{aligned} \tag{3.4}$$

where

$$\gamma = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \; L = \begin{bmatrix} Q^{1/2} YK \\ R^{1/2} UK \end{bmatrix}, \; W = \begin{bmatrix} X_0 \\ Y_F \end{bmatrix} K, \; z = \begin{bmatrix} x_0 \\ y_{\mathrm{f}} \end{bmatrix},$$

which admits the solution

$$\gamma^\star = (I - K_W (LK_W)^\dagger L) W^\dagger z,$$

with $K_W = \mathbf{Basis}(\mathbf{Ker}(W))$. This leads to the following data-driven solution to Problem (3.2).

Theorem 6 (Data-driven LQ control). *The input* $\mathbf{u}_T^*$ *that solves the LQ control problem* (3.2) *is*

$$\mathbf{u}_T^* = UK \underbrace{(I - K_W(LK_W)^\dagger L)W^\dagger \begin{bmatrix} x_0 \\ y_{\mathrm{f}} \end{bmatrix}}_{\gamma^*}. \tag{3.5}$$

With the additional assumption that $\mathbf{Rank}(Q^{1/2}O_T^Y) = n$,[2] an alternative expression for $\mathbf{u}_T^*$ that does not require the computation of kernel matrices is extracted from

$$\begin{bmatrix} x_0 \\ \mathbf{u}_T^* \end{bmatrix} = P^{-\frac{1}{2}} \left(\begin{bmatrix} X_0 \\ Y_F \end{bmatrix} \begin{bmatrix} X_0 \\ U \end{bmatrix}^\dagger P^{-\frac{1}{2}} \right)^\dagger \begin{bmatrix} x_0 \\ y_{\mathrm{f}} \end{bmatrix}, \tag{3.6}$$

where

$$P = \left(Y \begin{bmatrix} X_0 \\ U \end{bmatrix}^\dagger \right)^\mathsf{T} Q \left(Y \begin{bmatrix} X_0 \\ U \end{bmatrix}^\dagger \right) + \begin{bmatrix} 0 & 0 \\ 0 & R \end{bmatrix}. \tag{3.7}$$

Proof of Equation (3.6)*:* Problem (3.2) can be equivalently rewritten as

$$\begin{aligned} \underset{\mathbf{u}_T}{\text{minimize}} \quad & \left\| P^{\frac{1}{2}} \begin{bmatrix} x(0) \\ \mathbf{u}_T \end{bmatrix} \right\|_2^2 \\ \text{subject to} \quad & \\ & \begin{bmatrix} I & 0 \\ O_F^Y & F_F^Y \end{bmatrix} \begin{bmatrix} x(0) \\ \mathbf{u}_T \end{bmatrix} = \begin{bmatrix} x_0 \\ y_{\mathrm{f}} \end{bmatrix}, \end{aligned} \tag{3.8}$$

where O_F^Y and F_F^Y denotes the last p rows of the matrices O_T^Y and F_T^Y, and

$$P = \begin{bmatrix} O_T^Y & F_T^Y \end{bmatrix}^\mathsf{T} Q \begin{bmatrix} O_T^Y & F_T^Y \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & R \end{bmatrix} \tag{3.9}$$

[2]This is a mild condition that is satisfied, for instance, when (2.1) is observable and $Q \succ 0$.

From the assumptions $R \succ 0$ and $\mathbf{Rank}(Q^{1/2}O_T^Y) = n$, it follows that $P \succ 0$ (see [Celi et al., 2023c, Theorem III.1] for a proof of this fact). Then, by defining

$$v = P^{\frac{1}{2}} \begin{bmatrix} x(0) \\ \mathbf{u}_T \end{bmatrix},$$

we can rewrite (3.8) as

$$\begin{aligned} &\underset{v}{\text{minimize}} \quad \|v\|_2^2 \\ &\text{subject to} \\ &\begin{bmatrix} I & 0 \\ O_F^Y & F_F^Y \end{bmatrix} P^{-\frac{1}{2}} v = \begin{bmatrix} x_0 \\ y_{\text{f}} \end{bmatrix}. \end{aligned} \tag{3.10}$$

The minimizer of (3.10) is

$$v^* = \left(\begin{bmatrix} I & 0 \\ O_F^Y & F_F^Y \end{bmatrix} P^{-\frac{1}{2}} \right)^{\dagger} \begin{bmatrix} x_0 \\ y_{\text{f}} \end{bmatrix}.$$

Thus, the solution $\mathbf{u}_T^*$ to (3.8) satisfies

$$\begin{bmatrix} x_0 \\ \mathbf{u}_T^* \end{bmatrix} = P^{-\frac{1}{2}} v^* = P^{-\frac{1}{2}} \left(\begin{bmatrix} I & 0 \\ O_F^Y & F_F^Y \end{bmatrix} P^{-\frac{1}{2}} \right)^{\dagger} \begin{bmatrix} x_0 \\ y_{\text{f}} \end{bmatrix}.$$

To conclude, note that, under assumption (2.4), P in (3.9) equals (3.7) and

$$\begin{bmatrix} I_n & 0 \\ O_F^Y & F_F^Y \end{bmatrix} = \begin{bmatrix} X_0 \\ Y_F \end{bmatrix} \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\dagger},$$

from which (3.6) follows.

We remark that (3.5) and (3.6) rely on assumption (2.4). In practice, this imposes a lower bound on the number N of trajectories recorded in (2.2), i.e., $N \geq mT + n$. In fact, when this condition is not satisfied, then the Problem (3.4) may be infeasible or not

return the desired optimal control for some choices of the initial condition x_0 and target output y_f.

Remark 7 (Closed-form solutions vs iterative and optimization-based solutions). *The distinctive aspect of the above results lies in the utilization of a closed-form formula, as shown in Theorem 6 and in most of the results discussed in this thesis. While exceptions do exist, such as those in [da Silva et al., 2018, Pellegrino et al., 2023a, Pellegrino et al., 2023b], it is noteworthy that much of the existing literature on data-driven control predominantly revolves around iterative and optimization-based methodologies. It is essential to acknowledge that certain problems necessitate optimization-based approaches, often those involving nonlinear systems or input-output constraints. Nevertheless, when the opportunity for closed-form solutions arises, closed-form expressions provide not only valuable insights into problem solutions but also confer numerical advantages over alternative methods, see e.g., [Celi et al., 2022] for a discussion on the computational advantages of closed-form formulas over optimization-based solutions. Further, closed-form formulas have been recently used in the context of data-driven control to solve a set of diverse problems, such as the computation of the Kalman Filter and of the Linear Quadratic Gaussian regulator [Al Makdah and Pasqualetti, 2023], the identification of geometric invariant subspaces [Celi and Pasqualetti, 2022b], and the solution of the eigenstructure assignment problem [Celi et al., 2023b].*

Example 8 *We now apply the result from Theorem 6 to control a simplified model of a quadcopter. The quadcopter is dimensioned after a Crazyflie drone, with data* (2.2) *collected using Matlab simulations. For more details on the model of the system, we refer the reader*

to [Wang et al., 2016]. The state of the system is

$$x = \begin{bmatrix} x & \dot{x} & \theta & \dot{\theta} & y & \dot{y} & \phi & \dot{\phi} & z & \dot{z} & \psi & \dot{\psi} \end{bmatrix}^{\mathsf{T}},$$

where (x, y, z) *and* $(\dot{x}, \dot{y}, \dot{z})$ *are the coordinates and linear velocities, respectively, and* (θ, ϕ, ψ) *and* $(\dot{\theta}, \dot{\phi}, \dot{\psi})$ *are the roll, pitch and yaw angles and their respective velocities. The measured outputs are limited to the coordinates* (x, y, z) *and the asset angles* (θ, ϕ, ψ)*. The system is simulated in Fig. 3.1, where* $\mathbf{u}_T$ *is obtained from* (3.5). □

3.3 Minimum energy control

The minimum-energy control problem (ME) is obtained from (3.2) by letting $Q_t = 0$ and $R_t = I$. It reads as [Kailath, 1980, Pasqualetti et al., 2014]

$$\begin{aligned} \min_{u} \ & \sum_{t=0}^{T-1} \|u(t)\|_2^2 \\ \text{s.t. } & x(t+1) = Ax(t) + Bu(t), \\ & y(t) = Cx(t) + Du(t), \\ & x(0) = x_0, \quad y(T-1) = y_{\mathrm{f}}. \end{aligned} \tag{3.11}$$

Clearly, a solution to this problem can be obtained from Theorem 6 by simply letting $L = UK$. An insightful instance of the minimum energy control problem (3.11) is when $x_0 = 0$ and $X_0 = 0$, that is, the initial conditions of the experimental data and of the control problem are all equal to zero. In this case, although Lemma 2 cannot be used for a data-based representation of the system trajectories since X_0 is not full-row rank, a direct expression for the minimum energy input can readily be obtained.

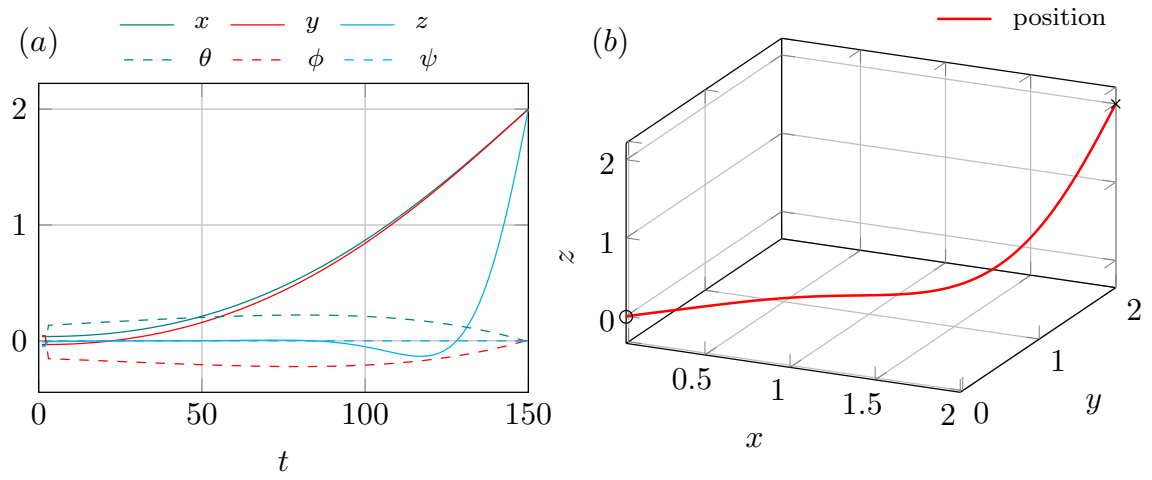

Figure 3.1: This figure shows the output trajectory of the system in Example 8 with $\mathbf{u}_T$ computed through (3.5). Data is collected as described in (2.2), with X_0 and U random matrices with i.i.d. entries. The experiment is run with $Q_t = \mathbf{blkdiag}(10, 10, 10, 100, 100, 100)$, $R_t = I_m$, for all $t = \{1, \dots, T\}$, and $T = 150$. Panel (a) shows the trajectories of the position (solid lines) and the orientation angles (dashed lines). Panel (b) shows the trajectory of the position in 3-D space (circle: initial position, cross: final position).

Theorem 9 (Data-driven ME control [Baggio et al., 2019]). *Let $X_0 = 0$ and $\mathbf{Rank}(U) = mT$. Then, the input $\mathbf{u}_T^*$ that solves the minimum energy control problem (3.11) is*

$$\mathbf{u}_T^* = U(I - K_{Y_F}(UK_{Y_F})^\dagger U)Y_F^\dagger y_\mathrm{f}, \tag{3.12}$$

where Y_F are the final states of the recorded trajectories (2.3) and $K_{Y_F} = \mathbf{Basis}(\mathbf{Ker}(Y_F))$.

Equation (3.12) is obtained from (3.5) leveraging the simplifications due to $X_0 = 0$. Along the lines of the derivation of (3.6), the following alternative minimum-energy control expression holds:

$$\mathbf{u}_T^* = (Y_F U^\dagger)^\dagger y_\mathrm{f}. \tag{3.13}$$

When $C = I$ and $D = 0$ this formula offers a data-driven way to compute the T-step

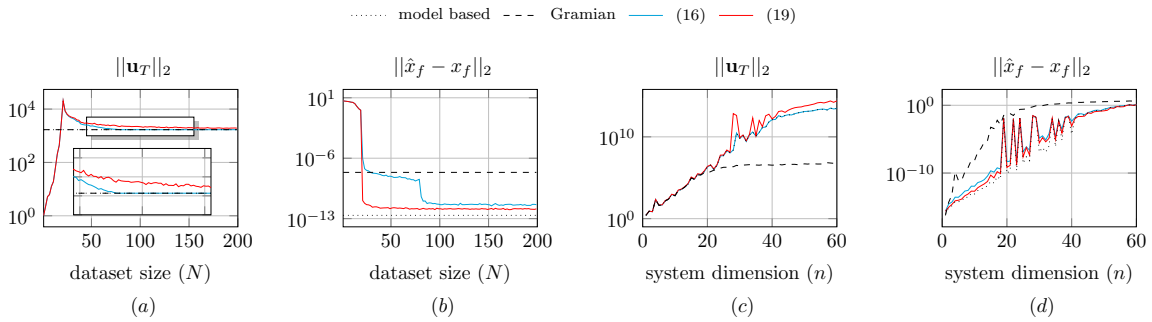

Figure 3.2: Panels (a)-(b) show the norm of the minimum energy control computed through: (i) the model-based formula $\mathbf{u}_T = C_T^{\dagger}(x_{\mathrm{f}} - A^T x_0)$, where C_T is the controllability matrix defined in Lemma 3, (ii) the controllability Gramian with $u(t) = B^{\mathsf{T}}(A^{\mathsf{T}})^{T-t-1}W_T^{\dagger}(x_{\mathrm{f}} - A^T x_0)$, (iii) the exact data-driven expression (3.13), and (iv) the asymptotically correct expression (3.16). The underlying system is generated randomly, with $n = 20$, $m = 2$, $T = 40$, $x_{\mathrm{f}} = [1\,1\,\cdots\,1]^{\mathsf{T}}$ and N is specified on the x-axis. The curves represent the average over 100 experiments with input data generated as random i.i.d. normal entries. Notice that, in accordance with the conditions imposed by Lemma 2, the expression (3.12) becomes exact once $N = n + mT = 80$ linearly independent experimental trajectories have been collected. Further, notice that (3.16) returns a feasible (the control satisfies the constraint) yet suboptimal (the cost achieved by the control is not optimal) solution for finite values of N. Panels (c)-(d) show the norm of the inputs $\mathbf{u}_T$ computed as above, and the corresponding errors in the final state, as a function of the system dimension n. We note that while the Gramian-based solution is technically exact, it is effectively less robust than the exact data-driven formulas in (3.13) and (3.16) for systems with a large number of states. This loss of numerical precision is a consequence of the computational process involved in computing and manipulating the controllability Gramian. We let $m = 2$, $T = n$, $N = mT + 20$, and $x_{\mathrm{f}} = [1\,1\,\cdots\,1]^{\mathsf{T}}$. The matrices A and B are generated randomly.

Gramian and its eigenvalues. In fact, it can be seen that in this case equation (3.13) can be written as a function of X_F and[3] the controllability matrix in (2.8) equals $C_T = X_F U^\dagger$. Consequently, the Gramian satisfies

$$W_T = C_T C_T^\mathsf{T} = X_F U^\dagger {U^\dagger}^\mathsf{T} X_F^\mathsf{T}. \tag{3.14}$$

Similarly, since the smallest (resp. largest) eigenvalues of the Gramian identify the states that require largest (resp. smallest) input energy, these can also be computed as

$$\begin{aligned} \sigma_{min}^{-1}(W_T) &= \max_{\|x_\mathrm{f}\|_2=1} \|(X_F U^\dagger)^\dagger x_\mathrm{f}\|_2^2 = \sigma_{max}^2((X_F U^\dagger)^\dagger), \\ \sigma_{max}^{-1}(W_T) &= \min_{\|x_\mathrm{f}\|_2=1} \|(X_F U^\dagger)^\dagger x_\mathrm{f}\|_2^2 = \sigma_{min}^2((X_F U^\dagger)^\dagger). \end{aligned} \tag{3.15}$$

Finally, when the entries of U are i.i.d. random variables with zero mean and nonzero finite variance, one can obtain simplified expressions of the minimum energy control input. For instance, the simplified expression

$$\hat{\mathbf{u}}_T = U X_F^\dagger x_\mathrm{f}, \tag{3.16}$$

converges to the minimum-energy input $\mathbf{u}_T^*$ almost surely, as the number of trajectories N increases [Baggio et al., 2019]. Since the minimum energy input is unique and for any finite value of N it generally holds $\hat{\mathbf{u}}_T \neq \mathbf{u}_T^*$, it follows that (3.16) is a suboptimal input for the minimum energy problem (3.11). In particular, the input (3.16) drives the system to the desired final state x_f with non-minimum energy, while requiring fewer numerical operations for its computation when compared with (3.12), (3.13). This property can be useful when dealing with large (network) systems, for which these calculations are generally ill-conditioned [Baggio et al., 2019, Baggio et al., 2021, Pasqualetti et al., 2014]. In support

[3] In this case the terminal constraint for (3.11) becomes $x(T) = x_\mathrm{f}$.

of these claims, in Fig. 3.2 we perform a series of numerical experiments, where we asses the numerical robustness and accuracy of direct data-driven controls. We notice that the accuracy in computing the minimum-energy control input using the data-driven expression (3.13) is comparable to that achieved when using the model-based counterpart, yet numerically more accurate than the model-based Gramian formula. Further, in Fig. 3.2(c)-(d) we notice that the accuracy of the Gramian-based control input decreases as n increases, while the data-driven expressions of the minimum-energy control inputs remain accurate for systems of considerably larger dimension.

As we progress through this work, the closed-loop expressions outlined in this section will form the foundational building blocks for deriving novel closed-form solutions to a variety of problems. Among these, throught this thesis we will cover the infinite-horizon linear quadratic regulator problem [Celi et al., 2022], the linear quadratic Gaussian control problem [Al Makdah and Pasqualetti, 2023], the distributed data-driven linear quadratic control problem [Celi et al., 2023c]. Further, we will answer several robustness questions related data-driven control [Celi et al., 2023c, Anguluri et al., 2020].

3.4 Datasets with heterogeneous control horizons

The results presented thus far assume that the experimental trajectories have the same length. In practice, however, it may be more convenient to collect data from heterogeneous experiments with different control horizons. The first question that we answer is whether the trajectories of the system can be represented as a linear combination of trajectories of different lengths, and for which control horizon. Is it possible to represent

trajectories of length greater than those of the control experiments? Ultimately, this analysis leads to an extension of Lemma 2.

To formalize the discussion, assume that the control experiments are performed using M distinct horizons $T_i \in \mathbb{N}, i \in \{1, \dots, M\}$, and that the available data is organized as $(U_i, X_{0,i}, X_i, Y_i)\,, i \in \{1, \dots, M\}$, where the i-th set contains N_i experiments, and $X_{0,i} \in \mathbb{R}^{n \times N_i}$, $U_i \in \mathbb{R}^{mT_i \times N_i}$, $X_i \in \mathbb{R}^{nT_i \times N_i}$, and $Y_i \in \mathbb{R}^{pT_i \times N_i}$ denote the matrices containing the initial states of the experiments, and the input, state, and output sequences with horizon T_i. Finally, let $X_{\mathrm{F},i}$ contain the last n rows of X_i and $\mathcal{D}_H = \{(U_i, X_{0,i}, X_i, Y_i)\}_{i=1}^{M}$ be the set of heterogeneous data. See Fig. 3.3 for an illustration of the heterogeneous dataset.

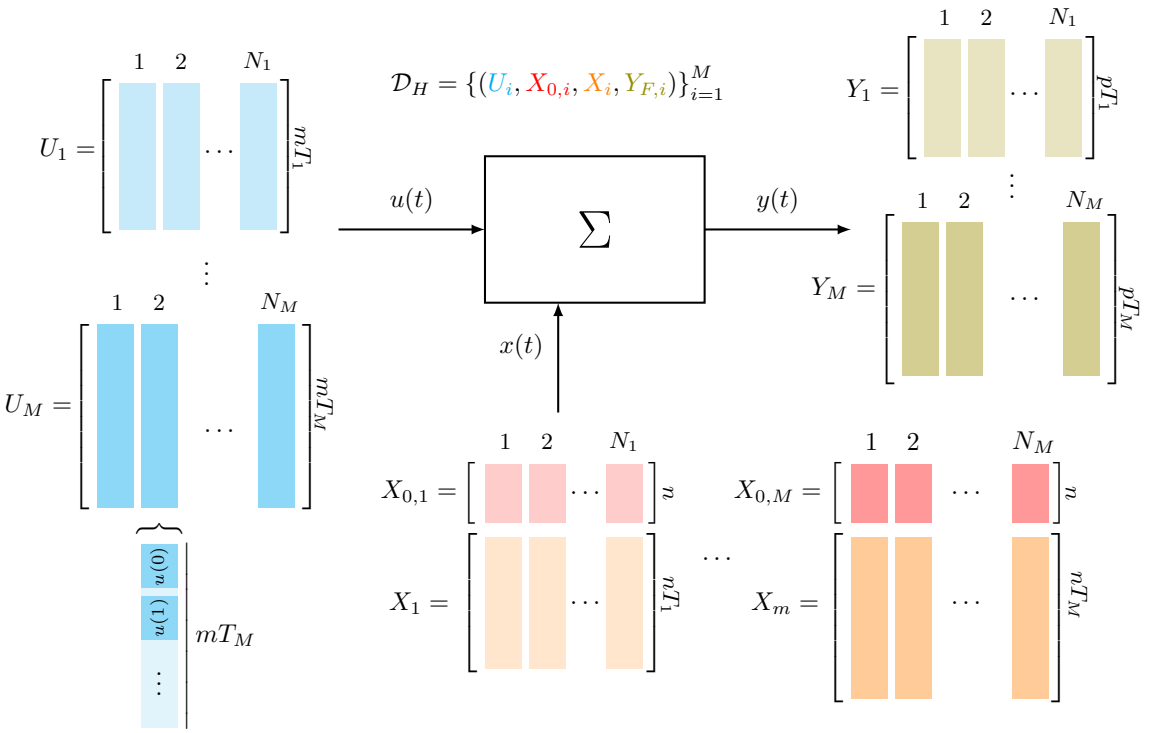

Figure 3.3: This figure conceptually shows the data collection phase with heterogeneous datasets.

We now present an extension of Lemma 2 that allows us to represent trajectories

of (2.1) of length T using the heterogeneous data $\mathcal{D}_H$, where the horizon T is an integer-weighted combination of the experimental horizons T_i (e.g., if $T_1 = 2$, $T_2 = 3$, and $T_3 = 8$, we could take $T = T_2 + 2T_3 = 19$). The main ideas behind this result are that a trajectory of length T can be broken up in multiple sub-trajectories, where the initial state of each sub-trajectory equals the final state of the previous sub-trajectory, and each sub-trajectory admits a data-based representation as in Lemma 2 (see also Fig. 3.4). We remark that the decomposition of a trajectory into multiple parts of a certain length may not be unique, thus leading to potentially multiple data-driven representations of the same trajectory when using heterogeneous data (for instance, if $T_1 = 2$, $T_2 = 3$, and $T_3 = 8$, then we could take $T = 19 = 8T_1 + T_3 = T_2 + 2T_3 = T_3 + T_2 + T_3$).

Lemma 10 (Data-based free and forced response representation with heterogeneous data [Baggio and Pasqualetti, 2020]) *Let $\mathcal{D}_H$ be the set of heterogeneous data and assume that*

$$\mathbf{Rank}\left(\begin{bmatrix} X_{0,i} \\ U_i \end{bmatrix}\right) = mT_i + n. \tag{3.17}$$

Let $\ell_1, \dots, \ell_p$ be a sequence of indices such that $T = \sum_{i=1}^{p} T_{\ell_i}$. Further, let $K_{0,i} = \mathbf{Basis}(\mathbf{Ker}(X_{0,i}))$, $K_{U,i} = \mathbf{Basis}(\mathbf{Ker}(U_i))$,

$$V_i = (X_{0,\ell_{i+1}} K_{U,\ell_{i+1}})^\dagger X_{F,\ell_i} K_{U,\ell_i}$$

$$Z_i = (X_{0,\ell_{i+1}} K_{U,\ell_{i+1}})^\dagger X_{F,\ell_i} K_{0,\ell_i}.$$

Then, for any initial state x_0 and input u_T, there exist vectors α and β such that

$$x_0 = X_{0,\ell_1} K_{U,\ell_1} \alpha, \text{ and}$$

$$u_T = \mathbf{blkdiag}(U_{\ell_1} K_{0,\ell_1}, \dots, U_{\ell_p} K_{0,\ell_p})\beta.$$

Moreover, $\mathbf{x}_T$ and $\mathbf{y}_T$ in (3.18) are the state and output trajectories of length T of (2.1) generated by x_0 and u_T.

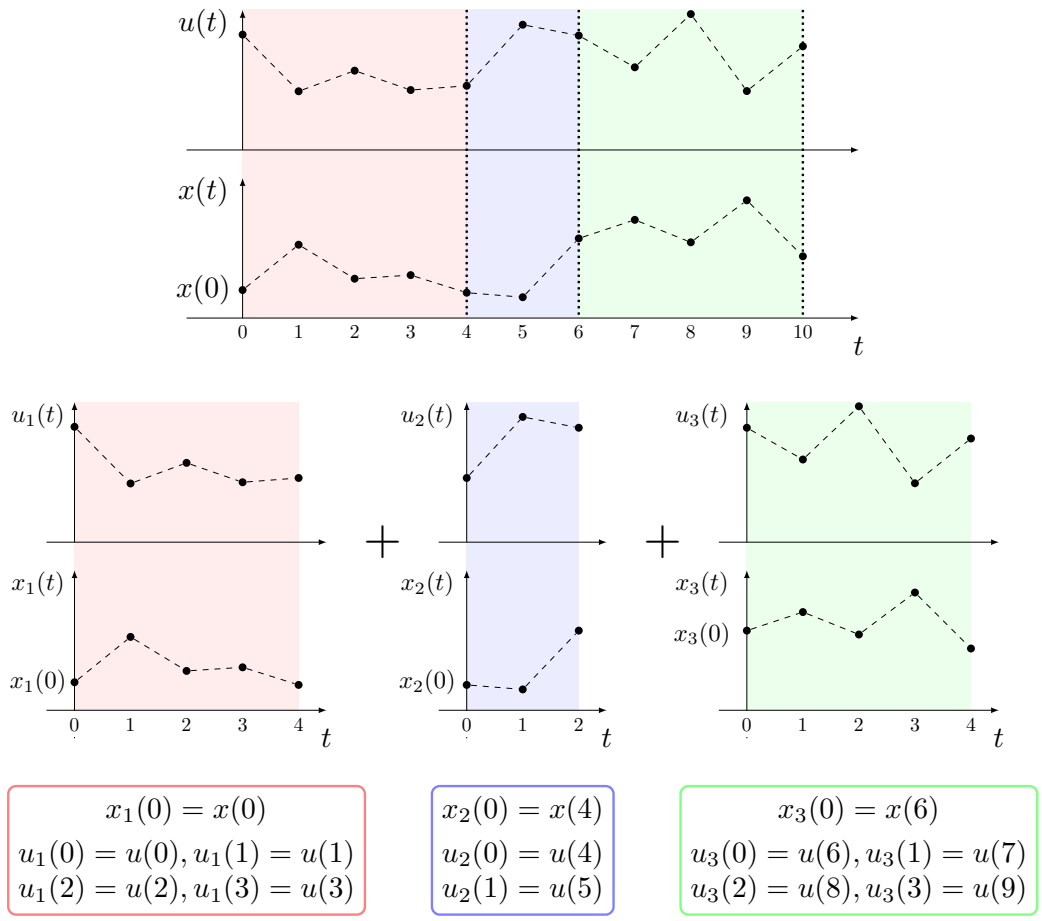

Figure 3.4: This figure shows an example of decomposition of a input-state trajectory in sub-trajectories: a trajectory of length $T = 10$ is divided in three sub-trajectories of lengths $T_1 = 4$, $T_2 = 2$, $T_3 = 4$.

Proof. From Lemma 2, there exist vectors α and β_1 such that $x_0 = X_{0,\ell_1} K_{U,\ell_1} \alpha$ and $\mathbf{u}_{T_1} = U_{\ell_1} K_{0,\ell_1} \beta_1$, and the corresponding state trajectory in the interval $[1, T_{\ell_1}]$ is

$$\mathbf{x}_{T_{\ell_1}} = X_{\ell_1} K_{U,\ell_1} \alpha + X_{\ell_1} K_{0,\ell_1} \beta_1.$$

In particular, it holds

$$x(T_{\ell_1}) = X_{F,\ell_1} K_{U,\ell_1} \alpha + X_{F,\ell_1} K_{0,\ell_1} \beta_1,$$

Next, since $X_{0,\ell_2}K_{U,\ell_2}$ has full row rank, we have

$$x(T_{\ell_1}) = X_{0,\ell_2}K_{U,\ell_2}\gamma,$$

where

$$\begin{aligned}\gamma &= (X_{0,\ell_2}K_{U,\ell_2})^{\dagger}(X_{F,\ell_1}K_{U,\ell_1}\alpha + X_{F,\ell_1}K_{0,\ell_1}\beta_1)\\ &= V_1\alpha + Z_1\beta_1.\end{aligned}$$

Thus, using Lemma 2 again, there exist a vector β_2 such that the input in the interval $[0, T_{\ell_1} + T_{\ell_2} - 1]$ can be written as

$$\mathbf{u}_{T_{\ell_1}+T_{\ell_2}} = \begin{bmatrix} U_{\ell_1}K_{0,\ell_1} & 0 \\ 0 & U_{\ell_2}K_{0,\ell_2} \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix},$$

and the corresponding state trajectory in $[1, T_{\ell_1} + T_{\ell_2}]$ as

$$\begin{aligned}\mathbf{x}_{T_{\ell_1}+T_{\ell_2}} &= \begin{bmatrix} X_{\ell_1}K_{U,\ell_1}\alpha + X_{\ell_1}K_{0,\ell_1}\beta_1 \\ X_{\ell_2}K_{U,\ell_2}\gamma + X_{\ell_2}K_{0,\ell_2}\beta_2 \end{bmatrix}\\ &= \begin{bmatrix} X_{\ell_1}K_{U,\ell_1}\alpha + X_{\ell_1}K_{0,\ell_1}\beta_1 \\ X_{\ell_2}K_{U,\ell_2}V_1\alpha + X_{\ell_2}K_{U,\ell_2}Z_1\beta_1 + X_{\ell_2}K_{0,\ell_2}\beta_2 \end{bmatrix}\\ &= \begin{bmatrix} X_{\ell_1}K_{U,\ell_1} & X_{\ell_1}K_{0,\ell_1} & 0 \\ X_{\ell_2}K_{U,\ell_2}V_1 & X_{\ell_2}K_{U,\ell_2}Z_1 & X_{\ell_2}K_{0,\ell_2} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta_1 \\ \beta_2 \end{bmatrix}.\end{aligned}$$

The expression of $\mathbf{x}_T$ in (3.18) follows by iterating the previous argument p times and collecting all β_i as $\beta = [\beta_1^{\mathsf{T}} \cdots \beta_p^{\mathsf{T}}]^{\mathsf{T}}$. A similar reasoning holds for the output trajectory $\mathbf{y}_T$.

■

Intuitively, the data-based representation of the T-steps state and output trajectories of Lemma 10 is obtained by suitably "gluing" together the data-based representations

of system trajectories of lengths $\{T_1, \ldots, T_M\}$. Note in particular that the horizon T can be longer than the horizons of the experimental trajectories, thus allowing for the representation of trajectories that have never been observed during the experiments. As a special case, when the experimental trajectories have the same horizon, that is $M = 1$, it is possible to reconstruct from data system trajectories with horizons equal to any multiple integer of T_1. This is illustrated in the next example.

Example 11 (Data-based representation of trajectories longer than the experimental data as in Lemma 10) *Consider the scalar system*

$$x(t+1) = ax(t) + u(t), \quad a \in \mathbb{R},$$

and the dataset with horizon $T = 2$

$$U_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, X_{0,1} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, X_1 = \begin{bmatrix} a & 1 & 0 \\ a^2 & a & 1 \end{bmatrix}.$$

It holds

$$K_{U,1} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \; K_{0,1} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad V_1 = a^2, \quad Z_1 = \begin{bmatrix} a & 1 \end{bmatrix}.$$

From Lemma 10, for any x_0 *and* $\mathbf{u}_4$*, there exist* α *and* β *such that*

$$x_0 = X_{0,1} K_{U,1} \alpha = \alpha, \; \textit{and}$$

$$\mathbf{u}_4 = \mathbf{blkdiag}(U_1 K_{0,1}, U_1 K_{0,1}) \beta = \beta,$$

and the state trajectory with initial condition x_0 *and input sequence* $\mathbf{u}_4$ *in the interval* $[1, 4]$

is given by

$$\mathbf{x}_4 = \begin{bmatrix} X_1K_{U,1} & X_1K_{0,1} & 0 \\ X_1K_{U,1}V_1 & X_1K_{U,1}Z_1 & X_1K_{0,1} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

$$= \left[\begin{array}{c|cc|cc} a & 1 & 0 & 0 & 0 \\ a^2 & a & 1 & 0 & 0 \\ \hline a^3 & a^2 & a & 1 & 0 \\ a^4 & a^3 & a^2 & a & 1 \end{array}\right] \begin{bmatrix} \alpha \\ \beta \end{bmatrix}.$$

□

In addition to providing a new data-based representation of the trajectories of (2.1) that allows for multiple heterogeneous experiments, Lemma 10 can also be used to derive data-driven formulas that solve LQ control problems, among others. For instance, under the assumption of Lemma 10, the data-driven LQ control expression in (6) remains valid when the dataset $\mathcal{D}_H$ is used by redefining L and W as follows:

$$L = \begin{bmatrix} Q^{1/2}Y_{\ell_{1:p}} \\ R^{1/2}U_{\ell_{1:p}} \end{bmatrix}, \qquad W = \begin{bmatrix} X_{0,\ell_1}K_{U,\ell_1} \\ Y_{\ell_{1:p},\mathrm{F}} \end{bmatrix},$$

where $U_{\ell_{1:p}} = \mathbf{blkdiag}(U_{\ell_1}K_{0,\ell_1}, \ldots, U_{\ell_M}K_{0,\ell_p})$, and $Y_{\ell_{1:p}}$ and $Y_{\ell_{1:p},\mathrm{F}}$ denote the matrices consisting of the last pT and p rows, respectively, of the matrix in (3.18).

Remark 12 (Alternative approaches to using heterogeneous datasets) *Alternative data-based representations of system trajectories with heterogenous datasets have been proposed in [Baggio and Pasqualetti, 2020, van Waarde et al., 2020b]. In [Baggio and Pasqualetti, 2020], the experimental data consist of inputs, initial and final state recordings with different time horizons T_i. The resulting data-based representation of state trajectories*

is given in sampled form and exploits a data-based reconstruction of A^{T_i}. In [van Waarde et al., 2020b] a generalization of Willems' Fundamental Lemma to the case of multiple trajectories is presented. Differently from the approach proposed in this chapter, the one in [van Waarde et al., 2020b] does not seem to allow for the computation of trajectories longer than the experiments. □

3.5 Robustness of data-driven open-loop LQ control

One of the main advantages of the direct data-driven approach with explicit formulas presented thus far is the possibility to analytically explore the impact of perturbations on these formulas and easily compute their sensitivity to parameter variations. In fact, such an analysis yields a way to quantify and improve the robustness of the direct data-driven approach to noisy and corrupted datasets. We consider two cases, which differ in the available knowledge of the statistics of the perturbations affecting the data. Let the perturbed dataset be

$$\tilde{U} = U + \Delta_U, \tag{3.19a}$$

$$\tilde{X}_0 = X_0 + \Delta_0, \tag{3.19b}$$

$$\tilde{X} = X + \Delta_X, \tag{3.19c}$$

$$\tilde{Y} = Y + \Delta_Y, \tag{3.19d}$$

where U, X_0, X, Y denote the ground truth values as in (2.2) and Δ_U, Δ_0, Δ_X, Δ_Y contain stochastic perturbations. For the purpose of this discussion, we focus on the data-driven control in (3.6). However, the analysis that follows can be adapted to other data-driven control expressions.

3.5.1 Data-driven LQ control with known noise statistics

We start with the standard scenario of i.i.d. perturbations characterized by known second-order statistics. In particular, we assume that Δ_U, Δ_0, Δ_Y are random matrices consisting of i.i.d. entries with zero mean and variance σ_U^2, σ_0^2, and σ_Y^2, respectively. In this setting, the data-driven control (3.6) is not consistent; that is, it does not converge to the true optimal control input even when the amount of available data grows to infinity. This lack of consistency is due to the pseudoinverse operation

$$\begin{bmatrix} \tilde{X}_0 \\ \tilde{U} \end{bmatrix}^\dagger = \begin{bmatrix} \tilde{X}_0 \\ \tilde{U} \end{bmatrix}^\mathsf{T} \left(\begin{bmatrix} \tilde{X}_0 \\ \tilde{U} \end{bmatrix} \begin{bmatrix} \tilde{X}_0 \\ \tilde{U} \end{bmatrix}^\mathsf{T} \right)^\dagger,$$

which contains quadratic terms (namely, $\tilde{X}_0\tilde{X}_0^\mathsf{T}$, $\tilde{U}\tilde{U}^\mathsf{T}$) that, by the law of large numbers, introduce variance-dependent biases (namely, $\sigma_0^2 NI$ and $\sigma_U^2 NI$) as N grows. However, since the noise variances are known, we can include correction terms in (3.6) that compensate for these biases and achieve asymptotically accurate data-driven formulas. Specifically, (3.6) can be modified as follows:

$$\begin{bmatrix} x_0^c \\ \\ \mathbf{u}_T^c \end{bmatrix} = (P_c^\dagger)^{\frac{1}{2}} \left(\begin{bmatrix} I_n & 0 \\ \\ \tilde{Y}_F \begin{bmatrix} \tilde{X}_0 \\ \\ \tilde{U} \end{bmatrix}_c^\dagger \end{bmatrix} (P_c^\dagger)^{\frac{1}{2}} \right)^\dagger \begin{bmatrix} x_0 \\ \\ y_\mathrm{f} \end{bmatrix}, \tag{3.20}$$

where

$$P_c = \left(\tilde{Y} \begin{bmatrix} \tilde{X}_0 \\ \tilde{U} \end{bmatrix}_c^{\dagger} \right)^{\mathsf{T}} Q \left(\tilde{Y} \begin{bmatrix} \tilde{X}_0 \\ \tilde{U} \end{bmatrix}_c^{\dagger} \right) + \begin{bmatrix} 0 & 0 \\ 0 & R \end{bmatrix},$$

$$\begin{bmatrix} \tilde{X}_0 \\ \tilde{U} \end{bmatrix}_c^{\dagger} = \begin{bmatrix} \tilde{X}_0 \\ \tilde{U} \end{bmatrix}^{\mathsf{T}} \left(\begin{bmatrix} \tilde{X}_0 \\ \tilde{U} \end{bmatrix} \begin{bmatrix} \tilde{X}_0 \\ \tilde{U} \end{bmatrix}^{\mathsf{T}} - \begin{bmatrix} \sigma_0^2 N I_n & 0 \\ 0 & \sigma_U^2 N I_{mT} \end{bmatrix} \right)^{\dagger},$$

and the following consistency result holds.

Theorem 13 (Asymptotic consistency of (3.20)**).** *Assume that*

(i) the columns of X_0 and U are i.i.d. and satisfy the condition (2.4) *almost surely as $N \to \infty$, and*

(ii) the entries of Δ_U, Δ_0, and Δ_Y are i.i.d. with zero mean and variances σ_U^2, σ_0^2, σ_Y^2.

Then, $\mathbf{u}_T^c$ in (3.20) *converges almost surely to $\mathbf{u}_T^*$ as $N \to \infty$.*

Proof. By the strong law of large numbers [Van der Vaart, 2000] and the assumptions (i)-(ii) on the experiments and noise, as N grows, the entries of $\frac{1}{N} X_0 \Delta_0^{\mathsf{T}}$, $\frac{1}{N} X_0 \Delta_U^{\mathsf{T}}$, $\frac{1}{N} U \Delta_0^{\mathsf{T}}$, $\frac{1}{N} U \Delta_U^{\mathsf{T}}$, $\frac{1}{N} \Delta_0 \Delta_U^{\mathsf{T}}$ tend to zero almost surely, while $\frac{1}{N} \Delta_0 \Delta_0^{\mathsf{T}}$ and $\frac{1}{N} \Delta_U \Delta_U^{\mathsf{T}}$ tend to $\sigma_0^2 I$, $\sigma_U^2 I$ almost surely. This implies that, as $N \to \infty$,

$$\frac{1}{N} \begin{bmatrix} \tilde{X}_0 \\ \tilde{U} \end{bmatrix} \begin{bmatrix} \tilde{X}_0 \\ \tilde{U} \end{bmatrix}^{\mathsf{T}} \xrightarrow{\text{a.s.}} \frac{1}{N} \begin{bmatrix} X_0 \\ U \end{bmatrix} \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\mathsf{T}} + \begin{bmatrix} \sigma_0^2 I_n & 0 \\ 0 & \sigma_U^2 I_{mT} \end{bmatrix}, \tag{3.21}$$

Similarly, by the strong law of large numbers, the entries of $\frac{1}{N} Y \Delta_0^{\mathsf{T}}$, $\frac{1}{N} Y \Delta_U^{\mathsf{T}}$, $\frac{1}{N} U \Delta_Y^{\mathsf{T}}$, $\frac{1}{N} X_0 \Delta_Y^{\mathsf{T}}$, $\frac{1}{N} \Delta_U \Delta_Y^{\mathsf{T}}$, $\frac{1}{N} \Delta_0 \Delta_Y^{\mathsf{T}}$ tend to zero almost surely with N, so that, as $N \to \infty$,

$$\frac{1}{N} \tilde{Y} \begin{bmatrix} \tilde{X}_0 \\ \tilde{U} \end{bmatrix}^{\mathsf{T}} \xrightarrow{\text{a.s.}} \frac{1}{N} Y \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\mathsf{T}}. \tag{3.22}$$

Equations (3.21) and (3.22) imply that, as $N \to \infty$,

$$\tilde{Y} \begin{bmatrix} \tilde{X}_0 \\ \tilde{U} \end{bmatrix}_c^{\dagger} = \frac{1}{N} \tilde{Y} \begin{bmatrix} \tilde{X}_0 \\ \tilde{U} \end{bmatrix}^{\mathsf{T}} \left(\frac{1}{N} \begin{bmatrix} \tilde{X}_0 \\ \tilde{U} \end{bmatrix} \begin{bmatrix} \tilde{X}_0 \\ \tilde{U} \end{bmatrix}^{\mathsf{T}} - \begin{bmatrix} \sigma_0^2 I_n & 0 \\ 0 & \sigma_U^2 I_{mT} \end{bmatrix} \right)^{\dagger}$$

$$\xrightarrow{\text{a.s.}} Y \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\mathsf{T}} \left(\begin{bmatrix} X_0 \\ U \end{bmatrix} \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\mathsf{T}} \right)^{\dagger} = Y \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\dagger}.$$

Since the map in (3.20) is a continuous function of $\tilde{U}$, $\tilde{X}_0$, $\tilde{Y}$ at $\Delta_U = \Delta_0 = \Delta_Y = 0$, from the previous equation and the continuous mapping theorem [Van der Vaart, 2000], as $N \to \infty$,

$$\begin{bmatrix} x_0^c \\ \mathbf{u}_T^c \end{bmatrix} \xrightarrow{\text{a.s.}} P^{-\frac{1}{2}} \left(\begin{bmatrix} I_n & 0 \\ Y_F \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\dagger} \end{bmatrix} P^{-\frac{1}{2}} \right)^{\dagger} \begin{bmatrix} x_0 \\ y_{\mathrm{f}} \end{bmatrix}$$

$$= P^{-\frac{1}{2}} \left(\begin{bmatrix} X_0 \\ Y_F \end{bmatrix} \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\dagger} P^{-\frac{1}{2}} \right)^{\dagger} \begin{bmatrix} x_0 \\ y_{\mathrm{f}} \end{bmatrix} = \begin{bmatrix} x_0 \\ \mathbf{u}_T^* \end{bmatrix},$$

where we used the identity $X_0 \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\dagger} = \begin{bmatrix} I_n & 0 \end{bmatrix}$, which holds when $\begin{bmatrix} X_0 \\ U \end{bmatrix}$ has full row rank, and the fact that (2.4) is satisfied almost surely as $N \to \infty$. ■

Fig. 3.5 shows the asymptotic consistency of (3.20), as predicted by Theorem 13, for a randomly generated system.

3.5.2 Data-driven LQ control with unknown noise statistics

In the more general case of (possibly correlated) perturbations with unknown second-order statistics, the robustness of (3.6) can be assessed through a local sensitivity analysis. Here, we assume for simplicity that the noise acts on the output measurements only, that is $\Delta_0 = \Delta_U = 0$. However, an analysis similar to the one that follows can be carried out also for noisy inputs and states. Let

$$F(U, X_0, Y) = (P^\dagger)^{\frac{1}{2}} \left(\begin{bmatrix} X_0 \\ Y_F \end{bmatrix} \begin{bmatrix} X_0 \\ U \end{bmatrix}^\dagger (P^\dagger)^{\frac{1}{2}} \right)^\dagger \begin{bmatrix} x_0 \\ y_{\mathrm{f}} \end{bmatrix} \tag{3.23}$$

denote the data-driven control map (3.6) with P as in (3.7), and $\mathbf{supp}(\Delta_Y) = \{i : \delta_{Y,i} \neq 0\}$, with $\delta_{Y,i} = \mathbf{vec}(\Delta_Y)_i$, denote the set of corrupted entries of Y. Since F is Fréchet-differentiable with respect to Y at the ground truth data, we can write it through its Taylor expansion as

$$\begin{aligned} F(U, X_0, \tilde{Y}) = F(U, X_0, Y) + \nabla F_Y(U, X_0, Y)\mathbf{vec}(\Delta_Y) \\ + r(U, X_0, Y, \Delta_Y), \end{aligned} \tag{3.24}$$

with $\lim_{\|\Delta_Y\|_2 \to 0} \|r(U, X_0, Y, \Delta_Y)\|_2 / \|\Delta_Y\|_2 = 0$ and where $\nabla F_Y(U, X_0, Y)$ is the Jacobian matrix of F with respect to Y calculated at the ground truth data. If the expected norms of the perturbations are sufficiently small, then (3.24) can be well approximated as (see the Appendix)

$$F(U, X_0, \tilde{Y}) \approx F(U, X_0, Y) + \nabla F_Y(U, X_0, Y)\mathbf{vec}(\Delta_Y). \tag{3.25}$$

For notational convenience, let $\nabla F_{Y,i}$ be the i-th column of $\nabla F_Y(U, X_0, Y)$, and let

$$\Delta y_{\mathrm{f}} = \|\tilde{y}_{\mathrm{f}} - y_{\mathrm{f}}\|_2$$

measure the error induced by the noisy data on the final output $\tilde{y}_\mathrm{f}$ when using the data-driven control input in (3.25). We next investigate how the sensitivity of the data-driven map, as quantified by the norm of the Jacobian matrix ∇F_Y, is related to the data size N.

Lemma 14 (Properties of $\|\nabla F_Y\|_2$ as a function of N). *Assume that the entries of X_0, U are independent[4] of N and that $\sigma^2_{\min}([X_0^\mathsf{T}\ U^\mathsf{T}]^\mathsf{T}) \geq cN$ where $c > 0$ is a constant independent of N. Then, for all $i \in \mathbf{supp}(\Delta_Y)$, $\|\nabla F_{Y,i}\|_2 \leq k_{Y,i}/N$, where $k_{Y,i} > 0$ are constants independent of N.*

The proof of the above lemma is rather involved and is deferred to the appendix. The condition $\sigma^2_{\min}([X_0^\mathsf{T}\ U^\mathsf{T}]^\mathsf{T}) \geq cN$ is typically satisfied for random i.i.d. initial conditions and inputs.[5] Thus, Lemma 14 shows that all $\|\nabla F_{Y,i}\|_2$ typically converge to zero as the number of experiments N increases. Under this scenario, as additional data become available, the map F becomes increasingly more robust against corrupted data. This conclusion is instrumental for the following result, whose proof is postponed to the Appendix.

Theorem 15 (Asymptotic robustness for sublinear number of perturbations). *In addition to the assumptions in Lemma 14, assume also that the entries of Δ_Y are independent of N. Then, if the cardinality of $\mathbf{supp}(\Delta_Y)$ grows sublinearly[6] with N, for any $\tau > 0$,*

$$\lim_{N\to\infty} \mathbb{P}\left[\Delta y_\mathrm{f} \geq \tau\right] = 0. \tag{3.26}$$

[4] We say that a random variable x is independent of a deterministic parameter α if the distribution of x is not a function of α.

[5] If U and X_0 have i.i.d. entries with zero mean and variance σ^2, $\frac{1}{N}\sigma^2_{\min}([X_0^\mathsf{T}\ U^\mathsf{T}]^\mathsf{T})$ tends almost surely to σ^2 as N tends to infinity by the law of large numbers.

[6] A sequence $\{x_n\}$ grows sublinearly with n if $\lim_{n\to\infty} x_n/n = 0$.

Theorem 15 guarantees that the error in the final output decreases to zero when N increases, regardless of Δ_Y (see Fig. 3.6 for a numerical example). This ensures the robustness of the data-driven control action for small, possibly adversarial, perturbations.

Remark 16 (Comparison with existing approaches). *Numerous studies have focused on developing data-driven controllers that can effectively handle disturbances generated by worst-case or stochastic noise models. Most of the existing approaches rely on robustified versions of data-based optimization problems, typically achieved through suitable regularizations (e.g., see [Coulson et al., 2021, De Persis and Tesi, 2021, Berberich et al., 2020b, Dörfler et al., 2023, Breschi et al., 2023]) or by leveraging classic robust control tools (e.g, see [van Waarde et al., 2020a, Bisoffi et al., 2021, Bisoffi et al., 2022, Berberich et al., 2022]). The distinctive feature of the approach of this chapter lies in the use of closed-form expressions which allows to asymptotically compensate the influence of noise and to explicitly characterize the sensitivity of data-driven controls.* □

3.6 Conclusions

In this chapter we presented a framework to solve a variety of linear quadratic control problems for linear systems using a dataset of input, state, and output trajectories collected offline, and we analyzed the robustness of these solutions to noise and arbitrary perturbations of the data. Differently from approaches relying on system identification, optimization, and policy iteration, in this chapter we proposed a set of closed-form expressions for optimal and sub-optimal control sequences, which are computationally efficient, insightful, and enable a direct sensitivity analysis of data-driven control. In fact, in some cases,

these formulas display favorable computational properties even when compared to classic model-based solutions, while also avoiding the solution of typically implicit and recursive Riccati equations. This chapter, together with the remaining of this thesis, shows that the data-driven approach to control may offer solutions that are computationally advantageous with respect to classic methods in the state-space, frequency, or geometric approaches. However, a detailed analysis of when data-driven methods should be preferred to classic ones, a comprehensive comparison of direct and indirect methods, the extension beyond linear quadratic control, among others, remain outstanding timely questions and potentially interesting research avenues.

3.7 Appendix

The following result, whose proof follows from [Anguluri et al., 2020, Lemma 1], confirms that when the expected norm of the perturbation Δ_Y is small so is the residual r of the expansion (3.24).

Lemma 17 (First-order approximation of $F(U,X_0,\tilde{Y})$) *Let $\nabla F_{Y,i}$ be the i-th column of ∇F_Y and*

$$\Sigma = F(U, X_0, \tilde{Y}) - F(U, X_0, Y) - \sum_{i \in \mathbf{supp}(\Delta_Y)} \delta_{Y,i} \nabla F_{Y,i}(U, X_0, Y).$$

Then, for any $\tau > 0$,

$$\lim_{\mathbb{E}[\|\mathbf{vec}(\Delta_Y)\|_2] \to 0} \mathbb{P}\left[\|\Sigma\|_2 \geq \tau \sqrt{\mathbb{E}[\|\mathbf{vec}(\Delta_Y)\|_2]} \right] = 0.$$

Proof of Lemma 14: Let

$$G = \begin{bmatrix} X_0 \\ Y_F \end{bmatrix} \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\dagger}$$

and rewrite (3.23) as

$$\begin{aligned} F(U, X_0, Y) &= P^{-\frac{1}{2}} \left(G P^{-\frac{1}{2}} \right)^{\dagger} \begin{bmatrix} x_0 \\ y_{\mathrm{f}} \end{bmatrix} \\ &= P^{-1} G^{\mathsf{T}} \left(G P^{-1} G^{\mathsf{T}} \right)^{-1} \begin{bmatrix} x_0 \\ y_{\mathrm{f}} \end{bmatrix}. \end{aligned}$$

Let y_i denote the i-th element of $\mathbf{vec}(Y)$, it holds

$$\begin{aligned} \nabla F_{Y,i} = & \frac{\partial F(U, X_0, Y)}{\partial y_i} \\ = & \frac{\partial P^{-1}}{\partial y_i} G^{\mathsf{T}} \left(G P^{-1} G^{\mathsf{T}} \right)^{-1} \begin{bmatrix} x_0 \\ y_{\mathrm{f}} \end{bmatrix} && (3.27) \\ & + P^{-1} \frac{\partial G^{\mathsf{T}}}{\partial y_i} \left(G P^{-1} G^{\mathsf{T}} \right)^{-1} \begin{bmatrix} x_0 \\ y_{\mathrm{f}} \end{bmatrix} && (3.28) \\ & + P^{-1} G^{\mathsf{T}} \frac{\partial \left(G P^{-1} G^{\mathsf{T}} \right)^{-1}}{\partial y_i} \begin{bmatrix} x_0 \\ y_{\mathrm{f}} \end{bmatrix}. && (3.29) \end{aligned}$$

Notice that

$$P = \left(Y \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\dagger} \right)^{\mathsf{T}} Q \left(Y \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\dagger} \right) + \begin{bmatrix} 0 & 0 \\ 0 & R \end{bmatrix}$$

$$= \begin{bmatrix} O_T^Y & F_T^Y \end{bmatrix}^{\mathsf{T}} Q \begin{bmatrix} O_T^Y & F_T^Y \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & R \end{bmatrix}, \tag{3.30}$$

$$G = \begin{bmatrix} I_n & 0 \\ O_F^Y & F_F^Y \end{bmatrix} \tag{3.31}$$

since (2.4) holds by assumption.

Let $\nabla F_{Y,i}^{(1)}$ denote the matrix in (3.27). This matrix can be written as in (3.32), where Γ_i is a $nT \times N$ matrix with one entry (corresponding to the element y_i) equal to one and zeros otherwise, and where we used that $\frac{\partial P^{-1}}{\partial y_i} = P^{-1} \frac{\partial P}{\partial y_i} P^{-1}$ (e.g., see [Bernstein, 2009]). From (3.32),

$$\begin{aligned} \|\nabla F_{Y,i}^{(1)}\|_2 &\leq \ell_{Y,i}^{(1)} \left\| \Gamma_i \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\dagger} \right\|_2 \\ &\leq \ell_{Y,i}^{(1)} \left\| \Gamma_i \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\mathsf{T}} \left(\begin{bmatrix} X_0 \\ U \end{bmatrix} \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\mathsf{T}} \right)^{-1} \right\|_2 \\ &\leq \ell_{Y,i}^{(1)} \left\| \Gamma_i \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\mathsf{T}} \right\|_2 \sigma_{\min}^{-2} \left(\begin{bmatrix} X_0 \\ U \end{bmatrix} \right) \\ &\leq \ell_{Y,i}^{(1)} \left\| \Gamma_i \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\mathsf{T}} \right\|_2 \frac{1}{cN}, \end{aligned} \tag{3.35}$$

where

$$\ell_{Y,1}^{(1)} = 2\|P^{-1}\|_2^2\|Q\|_2\| \begin{bmatrix} O_T^Y & F_T^Y \end{bmatrix} \|_2 \cdot \cdot \left\| P^{-1}G^\mathsf{T}\left(GP^{-1}G^\mathsf{T}\right)^{-1} \begin{bmatrix} x_0 \\ y_\mathrm{f} \end{bmatrix} \right\|_2$$

does not depend on N because of (3.30), (3.31). In the first step of (3.35) we used the submultiplicativity of matrix 2-norm, in the second step the fact that $A^\dagger = A^\mathsf{T}(AA^\mathsf{T})^{-1}$ when A is full-row rank, the third step follows from the fact that $\left\|A^{-1}\right\|_2$ equals the reciprocal of the minimum eigenvalue of A if A is positive definite, and the fourth step from the assumption on $\sigma_{\min}^2([X_0^\mathsf{T}\ U^\mathsf{T}]^\mathsf{T}) \geq cN$. Finally, since the matrix $\Gamma_i \begin{bmatrix} X_0^\mathsf{T} & U^\mathsf{T} \end{bmatrix}$ has only one row different from zero and the entries of such row are independent of N by assumption, (3.35) implies that $\|\nabla F_{Y,i}^{(1)}\|_2 \leq k_{Y,i}^{(1)}/N$, where $k_{Y,i}^{(1)} > 0$ is a constant independent of N.

Next, let $\nabla F_{Y,i}^{(2)}$ denote the matrix in (3.28). We can write $\nabla F_{Y,i}^{(2)}$ as in (3.33), where Ξ_i is a matrix with one entry set to one and all other entries set to zero, if y_i corresponds to an entry of Y_F, and the zero matrix, otherwise. Similarly as before, we have

$$\|\nabla F_{Y,i}^{(2)}\|_2 \leq \ell_{Y,i}^{(2)} \left\| \Xi_i \begin{bmatrix} X_0 \\ U \end{bmatrix}^\mathsf{T} \right\|_2 \frac{1}{cN}, \tag{3.36}$$

where

$$\ell_{Y,2}^{(1)} = \|P^{-1}\|_2^2 \left\| \left(GP^{-1}G^\mathsf{T}\right)^{-1} \begin{bmatrix} x_0 \\ y_\mathrm{f} \end{bmatrix} \right\|_2$$

does not depend on N. Since the matrix $\Xi_i \begin{bmatrix} X_0^\mathsf{T} & U^\mathsf{T} \end{bmatrix}$ is either the zero matrix or has only one row different from zero and the entries of such row are independent of N by assumption, (3.36) implies that $\|\nabla F_{Y,i}^{(2)}\|_2 \leq k_{Y,i}^{(2)}/N$, where $k_{Y,i}^{(2)} > 0$ is a constant independent of N.

Finally, let $\nabla F_{Y,i}^{(3)}$ denote the matrix in (3.29). We can write $\nabla F_{Y,i}^{(3)}$ as in (3.34). From the triangle inequality of the 2-norm and along the same lines that led to the upper bounds on $\|\nabla F_{Y,i}^{(1)}\|_2$ and $\|\nabla F_{Y,i}^{(2)}\|_2$,

$$\begin{aligned}\|\nabla F_{Y,i}^{(3)}\|_2 &\leq \ell_{Y,i}^{(3,1)} \left\| \Gamma_i \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\mathsf{T}} \right\|_2 \frac{1}{cN} + \ell_{Y,i}^{(3,2)} \left\| \Xi_i \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\mathsf{T}} \right\|_2 \frac{1}{cN} \\ &\leq k_{Y,i}^{(3)}/N \end{aligned} \tag{3.37}$$

for suitable positive constants $\ell_{Y,i}^{(3,1)}$, $\ell_{Y,i}^{(3,2)}$, $k_{Y,i}^{(3)}$ independent of N.

To conclude, from the triangle inequality and the above upper bounds on $\|\nabla F_{Y,i}^{(1)}\|_2$, $\|\nabla F_{Y,i}^{(2)}\|_2$, $\|\nabla F_{Y,i}^{(3)}\|_2$,

$$\|\nabla F_{y,i}\|_2 \leq \|\nabla F_{Y,i}^{(1)}\|_2 + \|\nabla F_{Y,i}^{(2)}\|_2 + \|\nabla F_{Y,i}^{(3)}\|_2 \leq \frac{k_{Y,i}}{N}$$

where $k_{Y,i} > 0$ is independent of N.

Proof of Theorem 15: By definition of Δy_{f},

$$\begin{aligned}\Delta y_{\mathrm{f}} = \|\tilde{y}_{\mathrm{f}} - y_{\mathrm{f}}\|_2 &= \left\| \sum_{i \in \mathbf{supp}(\Delta_Y)} \delta_{Y,i} \begin{bmatrix} O_T^Y & F_T^Y \end{bmatrix} \nabla F_{Y,i} \right\|_2, \\ &\leq \sum_{i \in \mathbf{supp}(\Delta_Y)} |\delta_{Y,i}| \left\| \begin{bmatrix} O_T^Y & F_T^Y \end{bmatrix} \nabla F_{Y,i} \right\|_2 = \overline{\Delta y_{\mathrm{f}}}. \end{aligned} \tag{3.38}$$

By the monotonicity of probability measures, for any $\tau > 0$, the set inclusion $\{\Delta y_{\mathrm{f}} \geq \tau\} \subseteq \{\overline{\Delta y_{\mathrm{f}}} \geq \tau\}$ holds, which implies $\mathbb{P}\left[\Delta y_{\mathrm{f}} \geq \tau\right] \leq \mathbb{P}\left[\overline{\Delta y_{\mathrm{f}}} \geq \tau\right]$. Note that $|\delta_{Y,i}|$ are non-negative random variables. Thus, by Markov's inequality [Van der Vaart, 2000] and the linearity of

the expected value, for any $\tau > 0$,

$$\begin{aligned}
\mathbb{P}\left[\Delta y_{\mathrm{f}} \geq \tau\right] &\leq \mathbb{P}\left[\overline{\Delta y_{\mathrm{f}}} \geq \tau\right] \\
&\leq \frac{1}{\tau}\left(\sum_{i\in\mathbf{supp}(\Delta_Y)}\left\|\begin{bmatrix} O_T^Y & F_T^Y \end{bmatrix}\nabla F_{Y,i}\right\|_2 \mathbb{E}[|\delta_{Y,i}|]\right) \\
&\leq \frac{c}{\tau}\left(\sum_{i\in\mathbf{supp}(\Delta_Y)} \|\nabla F_{Y,i}\|_2 \,\mathbb{E}[|\delta_{Y,i}|]\right) \\
&\leq \frac{c}{\tau}|\mathbf{supp}(\Delta_Y)| \max_i\{\|\nabla F_{Y,i}\|_2\} \max_i\{\mathbb{E}[|\delta_{Y,i}|]\} \qquad (3.39)
\end{aligned}$$

where $c = \| \begin{bmatrix} O_T^Y & F_T^Y \end{bmatrix} \|_2$ and $|\mathbf{supp}(\Delta_Y)|$ stands for the cardinality of $\mathbf{supp}(\Delta_Y)$. Since the distributions of $\delta_{X,i}$ are independent of N so are $\mathbb{E}[|\delta_{X,i}|]$. Hence, by (3.39) and Lemma 14, it follows that

$$\mathbb{P}\left[\Delta y_{\mathrm{f}} \geq \tau\right] \leq \frac{k_Y}{\tau}\frac{|\mathbf{supp}(\Delta_Y)|}{N}$$

where $k_Y > 0$ is a constant independent of N. This concludes the proof.

$$\begin{bmatrix} \mathbf{x}_T \\ \mathbf{y}_T \end{bmatrix} = \begin{bmatrix} X_{\ell_1}K_{U,\ell_1} & X_{\ell_1}K_{0,\ell_1} & 0 & \cdots & \cdots & 0 \\ X_{\ell_2}K_{U,\ell_2}V_1 & X_{\ell_2}K_{U,\ell_2}Z_1 & X_{\ell_2}K_{0,\ell_2} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_{\ell_p}K_{U,\ell_p}V_{p-1}\cdots V_1 & X_{\ell_p}K_{U,\ell_p}Z_{p-1}\cdots Z_1 & X_{\ell_p}K_{U,\ell_p}Z_{M-2}\cdots Z_1 & \cdots & X_{\ell_p}K_{0,\ell_p}Z_1 & X_{\ell_p}K_{0,\ell_p} \\ Y_{\ell_1}K_{U,\ell_1} & Y_{\ell_1}K_{0,\ell_1} & 0 & \cdots & \cdots & 0 \\ Y_{\ell_2}K_{U,\ell_2}V_1 & Y_{\ell_2}K_{U,\ell_2}Z_1 & Y_{\ell_2}K_{0,\ell_2} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ Y_{\ell_p}K_{U,\ell_p}V_{M-1}\cdots V_1 & Y_{\ell_p}K_{U,\ell_p}Z_{M-1}\cdots Z_1 & Y_{\ell_p}K_{U,\ell_p}Z_{M-2}\cdots Z_1 & \cdots & Y_{\ell_p}K_{0,\ell_p}Z_1 & Y_{\ell_p}K_{0,\ell_p} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \tag{3.18}$$

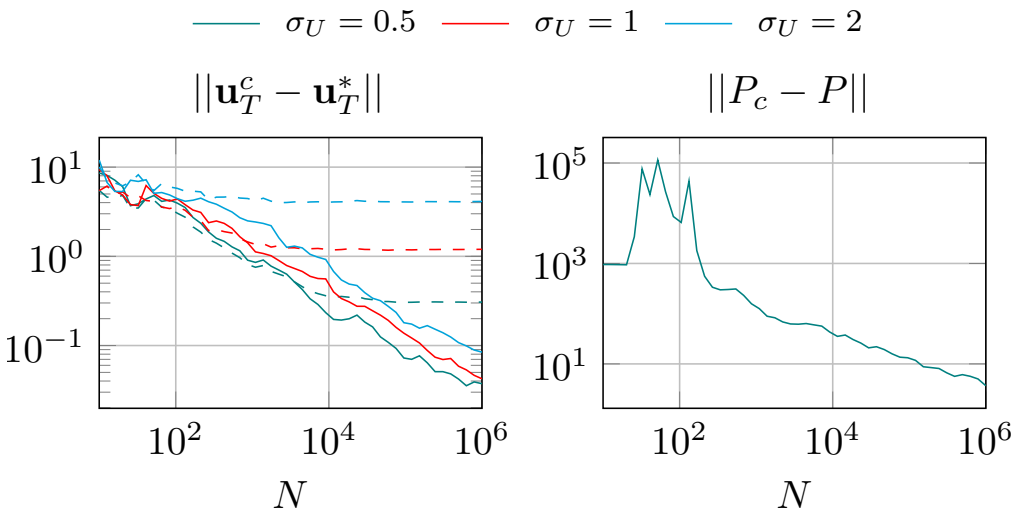

Figure 3.5: This figure shows the asymptotic consistency of equation (3.20) for problem (3.2). The left panel shows how the controller computed with a noisy dataset with noise compensation in (3.20) (solid lines) asymptotically converges to the optimal controller $\mathbf{u}_T^*$ for (3.2), as opposed to the non-compensated expression in (3.6) (dashed lines), for varying noise statistics. The experiment is performed over a randomly generated system with $n = 5$, $m = 3$, $p = 2$ and $T = 15$. Finally, $\sigma_0 = \sigma_Y = 0.1$, while different values of σ_U are shown in the legend. The right panel further supports this result by showing how the difference between P and P_c evolves as the number of experiment increases and the noise in the data is appropriately accounted for as shown in Theorem 13 (here, $\sigma_U = 0.5$).

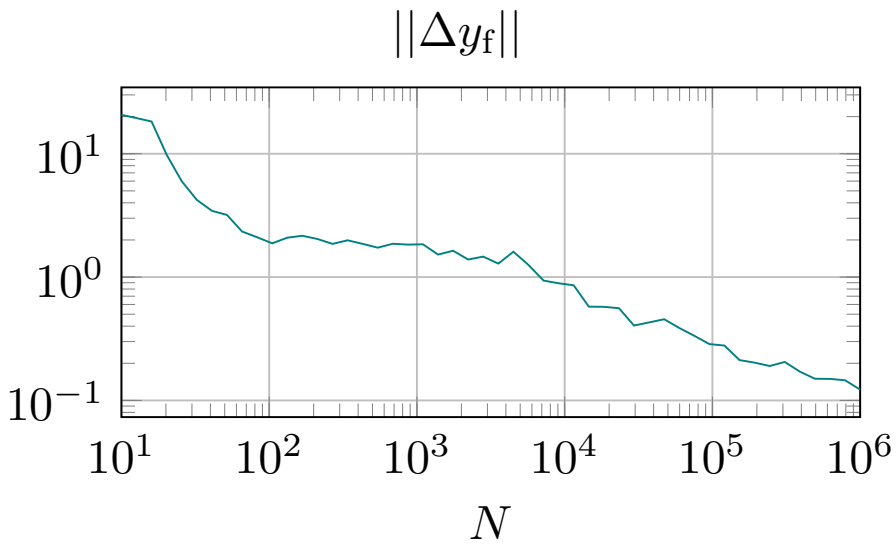

Figure 3.6: This figure shows the convergence results of Theorem 15. As implied by the theorem, although the noise statistics remain unknown, the difference between the desired final output and that computed with a noisy dataset approaches zero as the number of experiments increases. The experiment is performed over a randomly generated system with $n = 5$, $m = 3$, $p = 2$ and $T = 15$. The noise on the output is additive, gaussian, and with distribution $\mathcal{N}(1, 0.5)$.

$$\nabla F^{(1)}_{Y,i} = \frac{\partial P^{-1}}{\partial y_i} G^{\mathsf{T}} \left(GP^{-1}G^{\mathsf{T}}\right)^{-1} \begin{bmatrix} x_0 \\ \\ y_{\mathrm{f}} \end{bmatrix} = P^{-1} \frac{\partial P}{\partial y_i} P^{-1} G^{\mathsf{T}} \left(GP^{-1}G^{\mathsf{T}}\right)^{-1} \begin{bmatrix} x_0 \\ \\ y_{\mathrm{f}} \end{bmatrix}$$

$$= P^{-1} \left(\left(\Gamma_i \begin{bmatrix} X_0 \\ \\ U \end{bmatrix}^{\dagger} \right)^{\mathsf{T}} QY \begin{bmatrix} X_0 \\ \\ U \end{bmatrix}^{\dagger} + \left(Y \begin{bmatrix} X_0 \\ \\ U \end{bmatrix}^{\dagger} \right)^{\mathsf{T}} Q\Gamma_i \begin{bmatrix} X_0 \\ \\ U \end{bmatrix}^{\dagger} \right) P^{-1} G^{\mathsf{T}} \left(GP^{-1}G^{\mathsf{T}}\right)^{-1} \begin{bmatrix} x_0 \\ \\ y_{\mathrm{f}} \end{bmatrix}$$

$$= P^{-1} \left(\left(\Gamma_i \begin{bmatrix} X_0 \\ \\ U \end{bmatrix}^{\dagger} \right)^{\mathsf{T}} Q \begin{bmatrix} O^Y_T & F^Y_T \end{bmatrix} + \begin{bmatrix} O^Y_T & F^Y_T \end{bmatrix}^{\mathsf{T}} Q\Gamma_i \begin{bmatrix} X_0 \\ \\ U \end{bmatrix}^{\dagger} \right) P^{-1} G^{\mathsf{T}} \left(GP^{-1}G^{\mathsf{T}}\right)^{-1} \begin{bmatrix} x_0 \\ \\ y_{\mathrm{f}} \end{bmatrix}, \tag{3.32}$$

$$\nabla F^{(2)}_{Y,i} = P^{-1} \frac{\partial G^{\mathsf{T}}}{\partial y_i} \left(GP^{-1}G^{\mathsf{T}}\right)^{-1} \begin{bmatrix} x_0 \\ \\ y_{\mathrm{f}} \end{bmatrix} = P^{-1} \Xi_i \begin{bmatrix} X_0 \\ \\ U \end{bmatrix}^{\dagger} \left(GP^{-1}G^{\mathsf{T}}\right)^{-1} \begin{bmatrix} x_0 \\ \\ y_{\mathrm{f}} \end{bmatrix}, \tag{3.33}$$

$$\nabla F^{(3)}_{Y,i} = P^{-1} G^{\mathsf{T}} \frac{\partial \left(GP^{-1}G^{\mathsf{T}}\right)^{-1}}{\partial y_i} \begin{bmatrix} x_0 \\ \\ y_{\mathrm{f}} \end{bmatrix} = P^{-1} G^{\mathsf{T}} \left(GP^{-1}G^{\mathsf{T}}\right)^{-1} \frac{\partial (GP^{-1}G^{\mathsf{T}})}{\partial y_i} \left(GP^{-1}G^{\mathsf{T}}\right)^{-1} \begin{bmatrix} x_0 \\ \\ y_{\mathrm{f}} \end{bmatrix}$$

$$+ P^{-1} G^{\mathsf{T}} \left(GP^{-1}G^{\mathsf{T}}\right)^{-1} GP^{-1} \frac{\partial G^{\mathsf{T}}}{\partial y_i} \left(GP^{-1}G^{\mathsf{T}}\right)^{-1} \begin{bmatrix} x_0 \\ \\ y_{\mathrm{f}} \end{bmatrix}. \tag{3.34}$$

Chapter 4

Estimation of the optimal state-feedback control from data

When dealing with unknown systems, data can be used to directly learn controllers with desirable features, thus bypassing system identification, as previous chapters of this thesis have shown. In this chapter we go a step further and we present strategies to identify optimal feedback gains of the state for unknown linear systems directly trough closed-form functions of the data. In particular, when data is sufficiently informative, (i) we find the control input that minimizes a finite-horizon quadratic function of the states and inputs and (ii) we show how these inputs enable the estimate of the static feedback controller that minimizes an infinite-horizon quadratic function, i.e., the Linear Quadratic Regulator. Our formulas are closed-form, making them computationally efficient and of straightforward interpretation.

4.1 Introduction

Consider once again the dynamical system in (2.1). In this chapter we will focus on its state evolution, i.e.,

$$x(t+1) = Ax(t) + Bu(t), \qquad x(0) = x_0, \tag{4.1}$$

together with the quadratic cost function

$$J = x(T)^\top Q x(T) + \sum_{t=0}^{T-1} x(t)^\top Q x(t) + u(t)^\top R u(t), \tag{4.2}$$

where $Q \succeq 0$ and $R \succ 0$ are the cost matrices associated with the states and controls, respectively. We assume that (A, B) is controllable and $(A, Q^{\frac{1}{2}})$ observable. A standard problem in optimal control is to compute the control sequence $\mathbf{u}_T^*$ that minimizes J subject to (4.1). Notice that this problem is a particular instance of the one in discussed in Chapter 3. However, how we shall see, in this chapter we develop an alternative solution which is better suited for the problem of estimating the LQR gain.

Solutions to the above problem are well known in the model-based setting [Lancaster and Rodman, 1995]. For instance, for any finite horizon ($T < \infty$), the control sequence minimizing (4.2) can be found through the dynamic feedback of the state

$$u(t) = K(t)x(t), \tag{4.3}$$

with

$$K(t) = -(R + B^\top P(t+1)B)^{-1} B^\top P(t+1)A, \tag{4.4}$$

and where $P(t)$ is computed backwards in time as

$$P(t) = Q + A^\top P(t+1)A - A^\top P(t+1)B(R + B^\top P(t+1)B)^{-1} B^\top P(t+1)A, \tag{4.5}$$

with terminal condition $P(T) = Q$.

When dealing with infinite horizon ($T = \infty$), the time varying controller (4.3) reduces to a static feedback

$$u(t) = K^* x(t), \tag{4.6}$$

where the LQR gain K^* is defined as

$$K^* = -(R + B^\top P^* B)^{-1} B^\top P^* A, \tag{4.7}$$

and where P^* is the (unique) positive definite solution to the discrete time algebraic Riccati equation

$$P^* = Q + A^\top P^* A - A^\top P^* B (R + B^\top P^* B)^{-1} B^\top P^* A. \tag{4.8}$$

In this chapter we seek to compute $\mathbf{u}_T = \mathbf{vec}(u(0), \ldots, u(T-1))$, with $u(t)$ as in (4.6) for every time t, for an unknown system with data collected as detailed in Chapter 2.

4.2 Related work

The design of optimal controllers for linear systems has been the focus of extensive research [Lancaster and Rodman, 1995, Zhou et al., 1996]. From the early days, strategies to compute quadratically optimal controllers for a known linear system have usually relied on the iterative computation of Riccati Equations [Kleinman, 1968, Vit, 1972]. Connections between the open-loop and closed-loop solutions of quadratic optimal controls appeared in [Diem and Anderson, 1976, Lewis, 1981, Furuta and Wongsaisuwan, 1993].

Recently, data-driven controls have experienced an increasing popularity. With some notable exceptions [Ziegler and Nichols, 1942], early data-driven controls relied on a

two-step approach: (i) use data to identify the underlying system's model [Gevers, 2005] and (ii) design a controller for the identified system with traditional model-based tools. Lately, however, this two-steps approach has been challenged by the direct data-driven approach, in which the system identification step is bypassed by using data to directly design a controller with desired properties [Willems et al., 2005, Brunton and Kutz, 2019]. A number of works in this area have proposed data-based solutions to several control problems for linear systems, including open-loop optimal control [Baggio et al., 2019, Baggio et al., 2021, Baggio and Pasqualetti, 2020, Monshizadeh, 2020], closed-loop and robust control [da Silva et al., 2018, van Waarde and Mesbahi, 2020, Markovsky and Rapisarda, 2008, De Persis and Tesi, 2020, van Waarde et al., 2020, Berberich et al., 2020a, Rotulo et al., 2020], distributed [Allibhoy and Cortés, 2020, Jiao et al., 2021, Celi et al., 2021], and predictive control [Coulson et al., 2019, Berberich et al., 2020b]. Differently from most of the cited literature, e.g., [De Persis and Tesi, 2020, Coulson et al., 2019], the results proposed in this chapter offer closed-form solutions which have some benefits, for example with respect to the computational time needed to obtain the solution. Recently, a closed-form data-driven strategy to find optimal controls appeared in [da Silva et al., 2018]. Differently from [da Silva et al., 2018] we use data to compute a closed-form solution without identifying the system's Markov parameters.

4.3 Closed-form state-output data-driven controls

We rewrite J in (4.2) in compact form as

$$J = \mathbf{x}_T^\top \hat{Q} \mathbf{x}_T + \mathbf{u}_T^\top \hat{R} \mathbf{u}_T, \tag{4.9}$$

for $\mathbf{u}_T = \mathbf{vec}(u(0), \ldots, u(T-1))$, $\mathbf{x}_T = \mathbf{vec}(x(0), \ldots, x(T-1))$, and with $\hat{Q} = I_{T+1} \otimes Q$, $\hat{R} = I_T \otimes R$. We refer to the unique, optimal, control input minimizing J as $\mathbf{u}_T^*$ for given initial condition $x(0) = x_0$.

Theorem 18 (Data-driven optimal control) *Let X_0, X and U be as in (2.2). Let $K_0 = \mathbf{Basis}(\mathbf{Ker}(X_0))$ and $K_U = \mathbf{Basis}(\mathbf{Ker}(U))$. Then, given x_0, $Q \succeq 0$ and $R \succ 0$, the control input $\mathbf{u}_T^*$ minimizing (4.2) is*

$$\mathbf{u}_T^* = -UK_0 S^\dagger (XK_0)^\top \hat{Q} XK_U (X_0 K_U)^\dagger x_0, \tag{4.10}$$

with

$$S = (XK_0)^\top \hat{Q}(XK_0) + (UK_0)^\top \hat{R}(UK_0). \tag{4.11}$$

Moreover, the corresponding optimal state trajectory is

$$\mathbf{x}_T^* = X(K_U - K_0 S^\dagger (XK_0)^\top \hat{Q} XK_U)(X_0 K_U)^\dagger x_0. \tag{4.12}$$

Proof. Notice that, by virtue of Lemma 2, there always exist α and β such that

$$J = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}^\top \begin{bmatrix} (XK_U)^\top \hat{Q}(XK_U) & (XK_U)^\top \hat{Q}(XK_0) \\ (XK_0)^\top \hat{Q}(XK_U) & S \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix},$$

where S is as in (4.11). We now show that the following hold

$$S \succeq 0 \tag{4.13}$$

$$\mathbf{Ker}(S) \subseteq \mathbf{Ker}\left((XK_U)^\top \hat{Q}(XK_0) \right). \tag{4.14}$$

Clearly, $S \succeq 0$ holds by construction. Then, rewrite $S = \begin{bmatrix} XK_0 \\ UK_0 \end{bmatrix}^\top \begin{bmatrix} \hat{Q} & 0 \\ 0 & \hat{R} \end{bmatrix} \begin{bmatrix} XK_0 \\ UK_0 \end{bmatrix}$ and notice that if $v \in \mathbf{Ker}(S)$ then $v^\top S v = \left\| \begin{bmatrix} \hat{Q}^{1/2} XK_0 \\ \hat{R}^{1/2} UK_0 \end{bmatrix} v \right\|_2^2 = 0$. Therefore it must hold that $\hat{Q}^{1/2}(XK_0)v = 0$ and that $(XK_U)^\top \hat{Q}(XK_0)v = 0$, concluding that v also belongs to $\mathbf{Ker}\left((XK_U)^\top \hat{Q}(XK_0)\right)$. When (4.13)-(4.14) hold, by Lemma [Furuta and Wongsaisuwan, 1993, Lemma A.2], J admits a minimum in β and J can be further decomposed as

$$J = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}^\top \begin{bmatrix} I & \Gamma^\top \\ 0 & I \end{bmatrix} \begin{bmatrix} \Pi & 0 \\ 0 & S \end{bmatrix} \begin{bmatrix} I & 0 \\ \Gamma & I \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \tag{4.15}$$

where

$$\begin{aligned} \Gamma &= S^\dagger (XK_0)^\top \hat{Q}(XK_U), \\ \Pi &= (XK_U)^\top \hat{Q}(XK_U) - (XK_U)^\top \hat{Q}(XK_0)\Gamma. \end{aligned} \tag{4.16}$$

From (4.15), $J = \alpha^\top \Pi \alpha + (\beta + \Gamma\alpha)^\top S(\beta + \Gamma\alpha)$, which yields $\min_\beta J = \alpha^\top \Pi \alpha$ and hence the β minimizing J is

$$\beta = -\Gamma\alpha = S^\dagger (XK_0)^\top \hat{Q}(XK_U)\alpha. \tag{4.17}$$

By substituting $\alpha = (X_0 K_U)^\dagger x_0$ and $\mathbf{u}_T = UK_0\beta$ (cf. Lemma 2) back into (4.17) one obtains equation (4.10). Finding $\mathbf{x}_T^*$ in (4.12) follows an equivalent reasoning. ■

Theorem 18 shows how to compute the optimal control $\mathbf{u}_T^*$ through data alone. We remark that (4.10) is a closed-form solution which relies solely on data gathered from independent and non-optimal experiments of (4.1). This is an extension of the results in [Baggio et al., 2021] as here we do not assume $x_0 = 0$ and we do not impose a desired final condition in (4.2). For the sake of simplicity, we assumed that data (2.2) is collected

without noise. It is natural to ask how (4.10) would perform with noisy data. If some prior knowledge about the noise is available, such as its distribution, expression (4.10) can be modified to obtain asymptotically correct data-driven formulas. A strategy to achieve this has been shown in [Baggio et al., 2021] and its extension to (4.10) is straightforward. A closed-form data-driven procedure for $\mathbf{u}_T^*$ appeared recently in [da Silva et al., 2018]. However, the solution proposed in [da Silva et al., 2018] requires the identification of the Markov parameters of (4.1) (i.e., of matrices O_T^X and F_T^X). On the contrary, the approach proposed in Theorem 18 bypasses the identification step altogether thanks to the formulation of Lemma 2.

4.4 Convergence to the LQR gain

We now show how to estimate the LQR gain from a set of finite experiments. We do this by designing a gain thanks to the results of Theorem 18, and by showing that it converges exponentially fast to the LQR gain (4.7).

Theorem 19 (Data-driven estimate of the LQR gain I) *Let X_0, X and U be as in (2.2). Given $x_0 \neq 0$, $Q \succeq 0$ and $R \succ 0$, let $\mathbf{u}_T^*$ and $\mathbf{x}_T^*$ be as in (4.10) and (4.12), respectively. Let $U_T = \begin{bmatrix} u(0) & \cdots & u(T-1) \end{bmatrix}$ and $X_T = \begin{bmatrix} x(1) & \cdots & x(T) \end{bmatrix}$, and assume that X_T is full-row rank for some $\bar{t} \leq T$. Then*

$$\lim_{T \to \infty} U_T X_T^\dagger = K^*. \tag{4.18}$$

where K^* is in (4.7). Moreover, there exist $c > 0$ and $0 < \rho < 1$ independent of T such

that

$$\left\| U_T X_T^\dagger - K^* \right\| \leq c\rho^T. \tag{4.19}$$

Proof. From $u_t = K_t x_t$ in (4.3) (were for simplicity we let $u(t) = u_t$) we can rewrite $U_T X_T^\dagger = \begin{bmatrix} K_0 x_0^* & \cdots & K_{T-1} x_{T-1}^* \end{bmatrix} X_T^\dagger$. Then, for $\Delta K_t = K_t - K^*$ we can write, $\forall\ T \geq \bar{t}$

$$\begin{aligned} U_T X_T^\dagger - K^* &= [K_0 x_0^* \ \cdots \ K_{T-1} x_{T-1}^*] X_T^\dagger - K^* \\ &= K^* X_T X_T^\dagger + \Delta_T X_T^\dagger - K^* = \Delta_T X_T^\dagger, \end{aligned} \tag{4.20}$$

where $\Delta_T = [\Delta K_0 x_0^* \ \cdots \ \Delta K_{T-1} x_{T-1}^*]$ and where we used the fact that $X_T X_T^\dagger = I$ since X_T is full-row rank. We now study the convergence of the above two terms Δ_T and $X_T^\dagger$. We now show that norm of the first term of (4.20) converges exponentially fast

$$\begin{aligned} \|\Delta_T\|^2 &\leq \sum_{i=0}^{T-1} \|\Delta K_i\|^2 \|x_i^*\|^2 \\ &\leq \sum_{i=0}^{T-1} (c_1^2 \mu^{2(T-i-1)})(c_2^2 \lambda^{2i}) \\ &= (c_1 c_2)^2 \mu^{2(T-1)} \sum_{i=0}^{T-1} \left(\frac{\lambda}{\mu}\right)^{2i} \\ &= (c_1 c_2)^2 \mu^{2(T-1)} \frac{1 - (\lambda/\mu)^{2T}}{1 - (\lambda/\mu)^2} \\ &= \frac{(c_1 c_2)^2}{1 - (\lambda/\mu)^2} \left[\mu^{2(T-1)} - \frac{\lambda^{2(T-1)}}{\mu^2} \right] \\ &\leq \bar{c}_1 (\max\{\mu, \lambda\})^{2T} \end{aligned}$$

for a suitable $\bar{c}_1 \in \mathbb{R}^+$ independent of T and for λ, μ such that $0 < \lambda < 1$, $0 < \mu < 1$.[1] In the second inequality we used the fact that $\|\Delta K_t\|$ (resp. $\|x_i^*\|$) converges exponentially fast according to Lemma 24 (resp. Lemma 25). Further, the norm of the second term in

[1] We implicitly assumed that $\lambda \neq \mu$. When $\lambda = \mu$ it can be shown that $\|\Delta_T\|^2 \leq \bar{c}_1 (\lambda + \varepsilon)^{2T}$, for $\bar{c}_1$ independent of T and $\varepsilon > 0$ arbitrarily small.

(4.20) is bounded by

$$\begin{aligned}\left\|X_T^\dagger\right\|^2 &= \left\|X_T^\top (X_T X_T^\top)^{-1}\right\|^2 \\ &\leq \left\|X_T^\top\right\|^2 \left\|(X_T X_T^\top)^{-1}\right\|^2 \leq \bar{c}_2\end{aligned}$$

with $\bar{c}_2 \in \mathbb{R}^+$ is independent of T, and where we used the fact that $\|X_T\|^2 \leq \sum_{i=0}^{T} \|x_i^*\|^2 \leq \sum_{i=0}^{T} c_2^2 \mu^{2i} \leq c_2^2 \frac{1}{1-\mu^2}$ and $\left\|(X_T X_T^\top)^{-1}\right\|^2 \leq \frac{1}{\lambda_{\min}(X_T X_T^\top)^2} \leq \frac{1}{\lambda_{\min}(X_{\bar{t}} X_{\bar{t}}^\top)^2}$, which are both constant and not time-dependent.

Therefore

$$\left\|U_T X_T^\dagger - K^*\right\|^2 = \left\|\Delta_T X_T^\dagger\right\|^2 \leq \bar{c}_2 \bar{c}_1 (\max\{\mu, \lambda\})^{2T},$$

which concludes the proof. ■

Theorem 19 shows how the gain

$$K_{\text{CF}} = U_T X_T^\dagger, \tag{4.21}$$

which is computed through a closed-form direct-data driven formula, converges exponentially fast to the LQR gain.

Next, we propose an alternative approach to the same problem. The next results allows further to explicitly compute the any element of the sequence of dynamic feedback controller (4.4).

Theorem 20 (Data-driven estimate of the LQR gain II). *Let U_t and X_t be the submatrices of U and X in* (2.2) *obtained by selecting only the inputs and states at time t,*

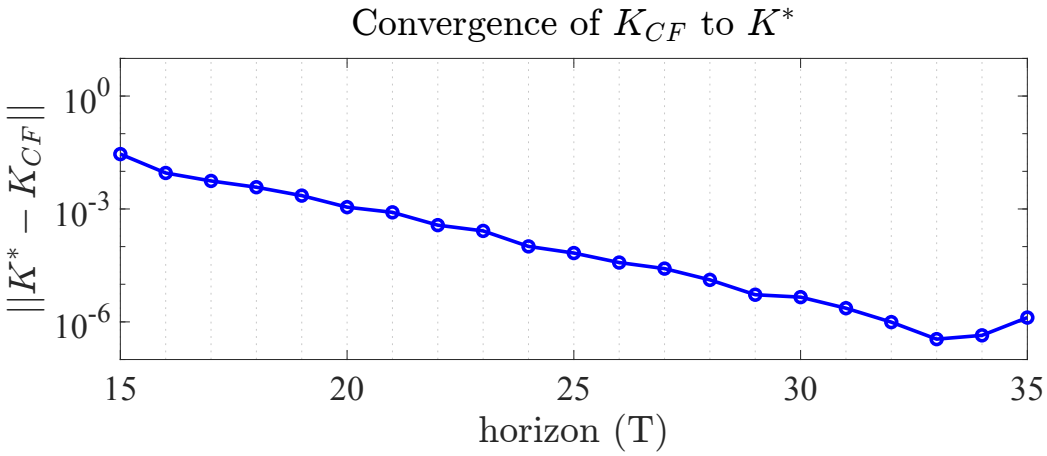

Figure 4.1: This figure shows the convergence of K_{CF} in (4.21) to K^* in (4.7). The underlying system is that of Example 22. For each choice of time horizon T we plot $\|K^* - K_{\text{CF}}\|$ on a logarithmic scale.

and define the matrices $L = \begin{bmatrix} (Q^{1/2}XK)^{\mathsf{T}} & (R^{1/2}UK)^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}$, *and*

$$\mathbf{U}_t = U_t K(I_N - K_W(LK_W)^{\dagger}L)W^{\dagger}, \tag{4.22}$$

$$\mathbf{X}_t = \begin{cases} I, & t = 0, \\ X_t K(I_N - K_W(LK_W)^{\dagger}L)W^{\dagger}, & t > 0, \end{cases} \tag{4.23}$$

with $W = X_0 K$ *and* $K_W = \mathbf{Basis}(\mathbf{Ker}(W))$. *Then,*[2]

$$K_{LQR}^t = \mathbf{U}_t(\mathbf{X}_t)^{-1}. \tag{4.24}$$

Proof. First, we provide a data-driven solution to the problem in (4.2). Following a procedure similar to the proof of Theorem 6, the above can be written as in (3.4), with

$$\gamma = \beta,\ L = \begin{bmatrix} Q^{1/2}X \\ R^{1/2}U \end{bmatrix},\ W = X_0 K,\ \text{and}\ z = x_0.$$

[2]We assume here that the matrix A is invertible, which guarantees that $\mathbf{X}_t$ is also invertible. When $\mathbf{X}_t$ is not invertible, the gain $\mathbf{U}_t(\mathbf{X}_t)^{\dagger}$ generates the input sequence that solves the LQ problem at hand, but it may differ from the gain K_{LQR}^t.

Then, the optimal input and state trajectories of for a given x_0 can then be computed as

$$\mathbf{u}_T^* = UK(I_N - K_W(LK_W)^\dagger L)W^\dagger x_0,$$

$$\mathbf{x}_T^* = XK(I_N - K_W(LK_W)^\dagger L)W^\dagger x_0.$$

Let $\mathbf{U}_t$ and $\mathbf{X}_t$ be as in (4.22). Notice that the i-th column of $\mathbf{X}_t$ equals the state at time t of the optimal state trajectory for (4.2) with initial state given by the i-th column of the identity matrix. Then, $\mathbf{U}_t = K_{LQR}^t \mathbf{X}_t$. To conclude, we show that if A is invertible then $\mathbf{X}_t$ is invertible. To this end, let $A_t = A + BK_{LQR}^t$ denote the (time-varying) state matrix of the closed-loop system and notice that

$$A_t = \underbrace{(I - B(R_t + B^\mathsf{T} P_{t+1} B)^{-1} B^\mathsf{T} P_{t+1})}_{H_t} A, \tag{4.25}$$

where $P_t \succeq 0$ satisfies the Riccati equation (4.5). Using the push-through identity $(I + XY)^{-1}X = Y(I + YX)^{-1}$, which holds for any matrices X, Y such that $(I + XY)$ is invertible [Bernstein, 2009, Fact 2.16.16], H_t can be written as

$$\begin{aligned} H_t &= I - B(R_t + B^\mathsf{T} P_{t+1} B)^{-1} B^\mathsf{T} P_{t+1} \\ &= I - BR_t^{-1}(I + B^\mathsf{T} P_{t+1} B R_t^{-1})^{-1} B^\mathsf{T} P_{t+1} \\ &= I - BR_t^{-1} B^\mathsf{T} P_{t+1}(I + BR_t^{-1} B^\mathsf{T} P_{t+1})^{-1}. \end{aligned}$$

From the last identity it follows that

$$H_t(I + BR_t^{-1} B^\mathsf{T} P_{t+1}) = I,$$

which implies that H_t is invertible with inverse $H_t^{-1} = I + BR_t^{-1} B^\mathsf{T} P_{t+1}$. Finally, observe that, for $t \geq 1$,

$$\mathbf{X}_t = A_{t-1} A_t \cdots A_0 = H_{t-1} A H_{t-2} A \cdots H_0 A.$$

Since the product of invertibile matrices is invertible, it follows that if A is invertible then $\mathbf{X}_t$ is invertible. ■

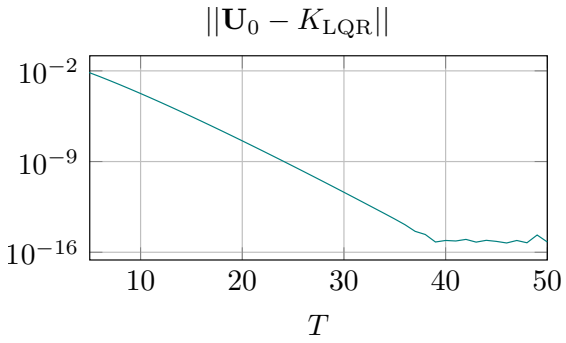

Figure 4.2: This figure supports the results of this chapter. In particular, we compute K^0_{LQR} using Theorem 20 for increasing T. As expected from (4.26) the distance between $\mathbf{U}_0$ and K_{LQR}, computed as the 2-norm of their difference, decreases as T grows. The experiment is performed over a randomly generated system with $n = 5$ and $m = 2$.

The above result allows us to compute any element of the sequence of time-varying controllers K^t_{LQR} as long as $t \leq T$, where T is the horizon of the available dataset (2.2). Interestingly, Theorem 20 only uses forward trajectories of the system and provides a closed-form, explicit expression of the LQR gains, thus avoiding the use of recursive, implicit Riccati equations or backward-in-time dynamic programming. We highlight that K^0_{LQR} converges to the steady state gain K_{LQR} as T increases (see also Fig. 4.2). In particular, we have

$$\|\mathbf{U}_0 - K_{LQR}\|_2 \leq c\rho^T, \tag{4.26}$$

where $c > 0$ and $0 < \rho < 1$ are suitable constants independent of t [Lancaster and Rodman, 1995].

Remark 21 (Related work on data-driven LQ control) *Linear quadratic control has received the most attention in the recent data-driven control literature. As opposed to the direct and closed-formulas discussed in this chapter and, e.g., in [Pellegrino et al., 2023a, Pellegrino et al., 2023b], most approaches in the literature rely on indirect schemes*

[da Silva et al., 2018, Aangenent et al., 2005], where a model of the system is first identified, optimization-based schemes [De Persis and Tesi, 2020, Coulson et al., 2019, Rotulo et al., 2020, Dörfler et al., 2022], or iterative schemes inspired by reinforcement learning approaches [Recht, 2019, Abbasi-Yadkori et al., 2019, Bradtke et al., 1994, Fazel et al., 2018, Gravell et al., 2020, Dean et al., 2020, Mohammadi et al., 2019], among others. Alternative data-driven approaches to Theorem 20 to compute feedback gains can be found, among others, in [De Persis and Tesi, 2020, Celi and Pasqualetti, 2022b, Al Makdah et al., 2022] and [Celi et al., 2023b, Bianchin, 2023], which solve the general eigenstructure assignment problem via static feedback in a purely data-driven manner. □

4.5 Numerical analysis

We now analyze the results from this chapter by means of numerical simulations. We begin by implementing Theorem 18 for a reference system.

Example 22 (Finite-time optimal control) *We consider the discretized version of a batch reactor system (with sampling time of* $0.1s$*), as in [De Persis and Tesi, 2020], where*

$$A = \begin{bmatrix} 1.178 & 0.001 & 0.511 & -0.403 \\ -0.051 & 0.661 & -0.011 & 0.061 \\ 0.076 & 0.335 & 0.560 & 0.382 \\ 0 & 0.335 & 0.089 & 0.849 \end{bmatrix}, B = \begin{bmatrix} 0.004 & -0.087 \\ 0.467 & 0.001 \\ 0.213 & -0.235 \\ 0.213 & -0.016 \end{bmatrix},$$

which is open-loop unstable. We set $T = 15$ *and perform a series of* $N = n + mT = 34$ *experiments such that* X_0 *and* U *satisfy 2.4. For* $x_0 = \mathbf{1}^\top$, $Q = I$, $R = I$, *we let* J^*_{CF} *be the cost obtained by letting the system evolve according to the optimal control* (4.10)*, and* J^*_{MB}

be the cost obtained by solving (4.2) *numerically (model based). We find that* $J^*_{CF} = 22.593$*, with* $\|J^*_{MB} - J^*_{CF}\| \leq 10^{-12}$. □

Next, we implement the result from Theorem 19 and compare it to the recent data driven LQR solution proposed in [De Persis and Tesi, 2020]. We remark once more the key conceptual difference between the two approaches: while both methods are based on data, the approach presented here is closed-form, whereas the one in [De Persis and Tesi, 2020] is based on an optimization problem solved through LMIs.

Example 23 (Estimating the LQR gain) *Consider the system of Example 22, now with the goal of estimating the optimal LQR gain* (4.7)*. We let* K_{CF} *be as in* (4.21)*,* K_{DD} *be the result of [De Persis and Tesi, 2020, Theorem 4], and* K^* *be* (4.7) *obtained through the model (we use* `idare` *in* `MATLAB`*). For* $T = 15$ *we find* $\|K^* - K_{CF}\| \approx 10^{-2}$ *(resp* $\|K^* - K_{DD}\| \approx 10^{-7}$*) obtained with a computational time in the order of* $10^{-2}s$ *(resp* $1s$*). All computations were performed on an* `Apple M1 - 3.2 GHz - 8 cores` *processor.* □

We remark two notable outcomes: (i) solution (4.21) is faster to compute, since it is explicit and it does not rely on the solution of an optimization problem. However, (ii) it is less precise, since it only holds asymptotically. This can be improved by increasing the time horizon T. Repeating Example 23 for $T = 35$ we obtain $\|K^* - K_{\mathrm{CF}}\| \approx 10^{-6}$ (see Fig. 4.1) and $\|K^* - K_{\mathrm{DD}}\| \approx 10^{-7}$, with computational times in the order of 10^{-2}s and 1s, respectively. This shows that for big enough values of T the two methods achieve a comparable precision in estimating the LQR feedback K^*. At the same time, the method proposed in this chapter is computationally more efficient.

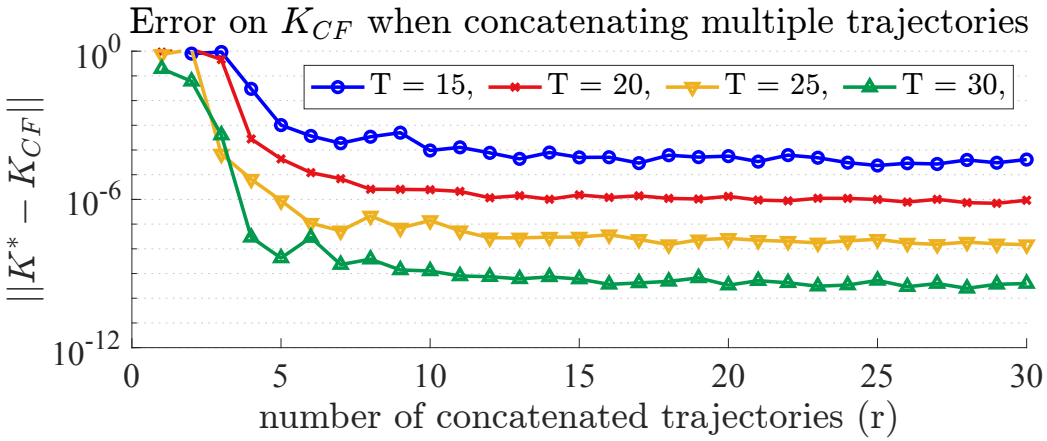

Figure 4.3: In this figure we show the performance of (4.21) when concatenating multiple trajectories as proposed in Section 4.5. The underlying system is generated randomly, with $n = 16$, $m = 4$. For each combination of time horizon T and number of concatenated trajectories r we plot $\|K^* - K_{\mathrm{CF}}\|$ on a logarithmic scale, where K^* is the LQR gain (4.7).

We now discuss how Theorem 19 performs when estimating the optimal feedback K^* for systems of increasing size n. Although Theorem 19 holds for any n, there are practical implementation issues that need addressing. Intuitively, larger systems will require a longer horizon T to obtain an accurate estimate of K^*. At the same time, the optimal state trajectory x_t^* vanishes exponentially fast, causing X_T in (4.21) to become ill-conditioned. This, in practice, renders the estimation of K^* imprecise for large n. One could mitigate this numerical issue by using higher precision when computing the optimal controller. However, this is not always practical. To overcome this issue, we propose a strategy to overcome these numerical issues arising for large values of n (see Fig. 4.3 for a numerical validation of this approach).

In particular, we propose the following strategy: (i) collect data , (ii) choose a family of initial conditions x_0^i with $i = \{1, \dots, r\}$, and find $(\{u(t)^i\}_{t=0}^{T-1}) = \{\mathbf{u}_T^* \text{ in (4.10)} \mid x_0 = x_0^i\}$ (resp. $(\{x(t)^i\}_{t=0}^{T-1}) = \{\mathbf{x}_T^* \text{ in (4.12)} \mid x_0 = x_0^i\}$), (iii) let $U_i = \begin{bmatrix} u(0)^\mathsf{T} & \dots & u(T-1)^\mathsf{T} \end{bmatrix}^\mathsf{T}$

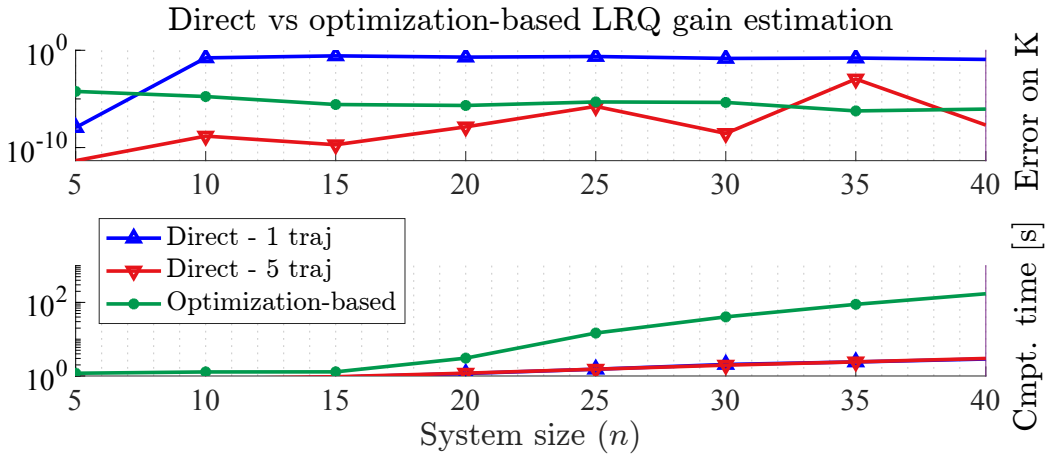

Figure 4.4: This figure shows the performance of the approach described in Theorem 19 (direct) to estimate the LRQ gain K^* for systems of increasing size. In the top panel we plot the error of the estimated K_{CF} (4.21) as $\|K^* - K_{\text{CF}}\|$ when one optimal trajectory is used (upwards triangles) and when five optimal trajectories are used (downwards triangles), see Section 4.5. In the bottom panel we plot the computational time to run the algorithm (in seconds). We compare this to the strategy proposed in [De Persis and Tesi, 2020] (dots). We notice how our approach scales comparatively when estimating K^*, at least when multiple trajectories are used. At the same time (4.21) offers a more computationally-efficient strategy, especially for larger values of n. The simulation parameters are $m = 5$, $T = 150$, $N = mT + n$. The system matrices are generated randomly for each n.

(resp. $X_i = \begin{bmatrix} x(0)^\mathsf{T} & \dots & x(T-1)^\mathsf{T} \end{bmatrix}^\mathsf{T}$), for all i, and $U_T = [U_1 \ \cdots \ U_r]$ (resp. $X_T = [X_1 \ \cdots \ X_r]$), and finally, (iv) compute K^* as in (4.21), through the concatenated values of U_T and X_T.

We notice that the above strategy does not require the collection of more experimental data, but simply to compute a collection of optimal trajectories through the already available data. Moreover, inputs x_0^i do not have to satisfy any particular requirement and can be chosen randomly. Fig. 4.4 shows a numerical implementation of this idea and compares it to [De Persis and Tesi, 2020]. We show how (4.21) works for small systems, but quickly starts underperforming with respect to [De Persis and Tesi, 2020] when n increases.

However, when 5 trajectories are concatenated as discussed above, our approach achieves a level of precision comparable to that of [De Persis and Tesi, 2020], while maintaining efficient computational times.

4.6 Conclusions

In this chapter we discussed a novel solution to the problem of computing optimal controllers directly from sampled data of an unknown linear system. When sufficient data are available, we show that this approach finds the exact solution minimizing a quadratic control objective of the states and inputs. This result can also be used to estimate a static feedback controller that converges to the LQR gain exponentially fast. Finally, we validate these findings through a series of numerical simulations and highlight the computational efficiency of the approach. In the next chapter we will study another classical problem in control theory, i.e., eigenstructure assignment. We will also propose an approach for pole placement with sparsity constraints on the feedback gain.

4.7 Appendix

With a slight abuse of notation, in this appendix we refer to the entry at time t of a sequence $\mathbf{x}_T$ as x_t instead of $x(t)$.

4.7.1 Convergence to static optimal controller

Lemma 24 (Convergence of (4.4) to (4.7)) *For given T, let K_t and K^* be as in (4.4) and (4.7), respectively. Then,*

$$\|K_{T-t} - K^*\| \leq c_1 \mu^t, \tag{4.27}$$

for $0 < \mu < 1$ and finite $c_1 \in \mathbb{R}^+$.

Proof. For fixed T, let P_t be as in (4.5) with $P_T = Q$, and let P^* be as in (4.8). It is known [Kailath et al., 2000, Theorem 14.5.1] that

$$\|P_{T-t} - P^*\| \leq \eta^{2t} p, \tag{4.28}$$

$0 < \eta < 1$ and finite $p \in \mathbb{R}^+$. Let $F_{T-t} = B^\top P_{T-t} B + R$ (resp. $F = B^\top P^* B + R$) and $G_{T-t} = B^\top P_{T-t} A$ (resp. $G = B^\top P^* A$), and from (4.4) notice that $K_{T-t} = F_{T-t}^{-1} G_{T-t}$. It follows that

$$\|G_{T-t} - G\| \leq \eta^{2t} p \, \|B\| \, \|A\| = \eta^{2t} p_1$$

$$\begin{aligned} \left\|F_{T-t}^{-1} - F^{-1}\right\| &\leq \left\|F_{T-t}^{-1}\right\| \|F - F_{T-t}\| \left\|F^{-1}\right\| \\ &\leq \left\|F_{T-t}^{-1}\right\| \left\|B^\top\right\| \eta^{2t} p \, \|B\| \left\|F^{-1}\right\| = \eta^{2t} p_2, \end{aligned}$$

for some positive, finite, scalars p_1, p_2. Now consider $H = (F_{T-t}^{-1} - F^{-1})(G + G_{T-t})$ and notice that $\|H\| \leq \eta^{2t} p_3$, for some positive finite scalar p_3. We find the following

$$\begin{aligned} F_{T-t}^{-1} G_{T-t} - F^{-1} G &= F^{-1} G_{T-t} - F_{T-t}^{-1} G + H \\ &= F^{-1} G_{T-t} - F_{T-t}^{-1} G + H - \\ &\quad - F^{-1} G + F^{-1} G \\ &= F^{-1}(G_{T-t} - G) - \\ &\quad (F_{T-t}^{-1} - F^{-1}) G + H. \end{aligned}$$

Finally, we can derive

$$\begin{aligned}\|K_{T-t} - K^*\| &= \left\|F_{T-t}^{-1}G_{T-t} - F^{-1}G\right\| \\ &\leq \|H\| + \left\|F_{T-t}^{-1} - F^{-1}\right\| \|G\| + \\ &\quad + \left\|F^{-1}\right\| \|G_{T-t} - G\| \\ &\leq c_1\mu^t,\end{aligned}$$

where $0 < \mu < 1$, and $c_1 \in \mathbb{R}^+ < \infty$. ■

4.7.2 Exponential convergence of the state

It is easy to see that for the optimal feedback K^* in (4.7), and for any initial condition, $x_{t+1} = (A+BK^*)x_t$ converges to zero exponentially fast [Lancaster and Rodman, 1995]. We now show an equivalent result for $x_{t+1} = (A + BK_t)x_t$, controlled through the time varying K_t in (4.4).

Lemma 25 (Exponential convergence of $x(t)$ with dynamic feedback) *Let the initial condition be x_0 and let $x_{t+1} = (A + BK_t)x_t$, with K_t in (4.4). Then*

$$\|x_t\| \leq c_2\lambda^t, \tag{4.29}$$

for $0 < \lambda < 1$ and finite $c_2 \in \mathbb{R}^+$.

Proof. We write

$$\begin{aligned}x_{t+1} &= (A + BK_t)x_t \\ &= \underbrace{(A + BK^*)}_{F} x_t + \underbrace{B(K_t - K^*)}_{G_t} x_t,\end{aligned}$$

where F is stable and G_t is a vanishing perturbation. In fact, from Lemma 24, $\|B(K_t - K^*)\| \leq \|B\| \|K_t - K^*\| \leq c_1\mu^{T-t}$. Then, for any arbitrarily small $\varepsilon > 0$ there exist sufficiently large

T^* and $\bar{t} < T^*$ such that $\|G_k\| \leq \varepsilon$ for $k = \{0, \ldots, T - \bar{t}\}$, $\forall T \geq T^*$. We can then write the state at any time $t = \{\bar{t}, \ldots, T\}$ as

$$\begin{aligned} x_t &= \prod_{k=0}^{t-1} (F + G_k) x_0 \\ &= \prod_{k=T-\bar{t}}^{t} (F + G_k) \prod_{k=0}^{T-\bar{t}-1} (F + G_k) x_0, \end{aligned}$$

and hence

$$\|x_t\| \leq q \|x_{T-\bar{t}}\|,$$

where $q \in \mathbb{R}^+$, and where we used the fact that $\prod_{k=T-\bar{t}}^{t}(F + G_k)$ is bounded in norm since it is a product of finite terms. To show the exponential convergence of $\|x_{t-\bar{t}}\|$ consider the quadratic Lyapunov function $V(x) = x^\top P x$, with $P \succ 0$. Then,

$$\begin{aligned} \Delta V(x) &= V(x_{t+1}) - V(x_t) \\ &= x_{t+1}^\top P x_{t+1} - x_t^\top P x_t \\ &= x_t^\top (F^\top P F - P) x_t + 2 x_t^\top F P G_t x_t \\ &\quad + x_t^\top G_t P G_t x_t \\ &\leq \lambda_{\max}(F^\top P F - P) \|x_t\|^2 + \\ &\quad 2 \|F\| \|P\| \|G_t\| \|x_t\|^2 + \|G_t\|^2 \|P\| \|x_t\|^2 \\ &\leq (-q + 2\varepsilon \|F\| \|P\| + \varepsilon^2 \|P\|) \|x\|^2 \end{aligned}$$

where we used the fact that $\left\|F^\top P F - P\right\| \leq \lambda_{\max}(F^\top P F - P) < -q < 0$ and from which we can conclude that $(-q + 2\varepsilon \|F\| \|P\| + \varepsilon^2 \|P\|) \|x\|^2 < 0$ for ε small enough, which always exists for large enough T. When ε is small enough, therefore, $\Delta V(x) < 0$, for all $x \neq 0$, ensuring the exponential stability of x_t, i.e., $\|x_t\| \leq c_2 \lambda^t$ [Khalil, 2002]. ■

Chapter 5

Data-driven eigenstructure assignment & sparse pole placement

In this chapter we present a novel approach for solving the pole placement and eigenstructure assignment problems through data-driven methods. By using open-loop data alone, we show that it is possible to characterize the allowable eigenvector subspaces, as well as the set of feedback gains that solve the pole placement problem. Additionally, we propose a closed-form expression for the feedback gain that solves the eigenstructure assignment problem. Finally, we discusses a series of optimization problems aimed at finding sparse feedback gains for the pole placement problem.

5.1 Introduction

As shown so far in this work, data-driven control has been applied extensively in optimal control. However, despite the popularity of data-driven control, the problems of data-driven pole placement and eigenstructure assignment have not enjoyed the same degree of attention, and have remain unexplored until recently [Celi et al., 2023b, Mukherjee and Hossain, 2022, Bianchin, 2023]. The traditional (i.e., model-based, non-sparse) pole placement and eigenstructure assignment problems have a rich history, including in practical applications [Innocenti and Stanziola, 1990, Andry et al., 1983]. The pole placement (eigenstructure assignment) problem consists in finding a static feedback gain that produces a closed-loop system where the state matrix has a pre-specified set of eigenvalues (eigenvalues and eigenvectors) [Liu and Patton, 1998]. We refer to the seminal works [Wonham, 1967, Wonham and Morse, 1970, Moore, 1976, Klein and Moore, 1977, Kautsky et al., 1985] and to the recent papers [Padula et al., 2021, Teoh et al., 2022], which highlight the ongoing interest in these topics.

In general, the feedback gain which solves the pole placement problem is not unique, adding a certain degree of freedom on its choice. This can be leveraged to enforce further control objectives, for example, by imposing a sparsity pattern on the feedback gain itself. By using a feedback with predefined sparsity patterns, or by maximizing the overall number of zero entries of the feedback gain, the number of feedback signals can be reduced while still achieving the desired closed-loop behavior. This can be advantageous in applications where the number of sensors or feedback signals is limited as, for example, in complex network systems [Lin et al., 2011].

In this chapter, we address these problems and show that it is possible to place the closed-loop eigenvalues exactly at any desired location by designing a static feedback gain through open-loop data alone, i.e., without explicit knowledge of the system matrices. Further, we show that the static gain for eigenstructure assignment can be found through a closed-form data-driven expression. Finally, we apply these results to the design of sparse feedback gains.

5.2 Related work

Despite the recent advancements in data-driven control theory [Baggio et al., 2019, Coulson et al., 2019, De Persis and Tesi, 2020], to the best of our knowledge, the only works discussing data-driven strategies for pole placement and eigenstructure assignment are [Mukherjee and Hossain, 2022] and [Bianchin, 2023]. Both [Mukherjee and Hossain, 2022, Bianchin, 2023] are based on the behavioral approach [Polderman and Willems, 1997] and rely on the *Fundamental Lemma* [Willems et al., 2005] to characterize the behavior of a linear system from a single, long, experimental trajectory. In contrast, our approach collects data from multiple trajectories, which has proven advantageous, e.g., when dealing with unstable systems, since shorter trajectories can be leveraged. A detailed analysis of the benefits of using multiple (shorter) trajectories over a single trajectory in control and reinforcement learning can be found in [Tu et al., 2022]. It is worth noting that [Mukherjee and Hossain, 2022] uses Linear Matrix Inequalities to solve the problem of pole placement and it does not offer any closed-form solution, while [Bianchin, 2023] does not discuss a characterization of the set of static feedbacks for pole placement. Additionally, neither

[Mukherjee and Hossain, 2022] nor [Bianchin, 2023] provide any insights on designing sparse feedback gains, which is a relatively unexplored topic even in the model-based framework [Lin et al., 2011, Katewa and Pasqualetti, 2021]. This knowledge gap further motivates our interest in this problem. In [Eising and Cortés, 2022], the authors propose a data-driven approach to designing sparse stabilizing feedback gains, however this method does not assign specific eigenvalues/eigenvectors as we do in this chapter. Recently, the System Level Approach to Controller Synthesis has proposed a set of tools for designing constrained robust, sparse, and optimal controllers, see [Wang et al., 2019]. However, the System Level Approach is based on designing a dynamic compensator, while the problems of pole placement and eigenstructure assignment are based on static feedback gains [Liu and Patton, 1998].

5.3 Pole placement and eigenstructure assignment

We focus on the dynamical system in (2.1), and in particular, on its state evolution

$$x(t+1) = Ax(t) + Bu(t), \qquad x(0) = x_0, \tag{5.1}$$

where $x(t) \in \mathbb{R}^n$ and $u(t) \in \mathbb{R}^m$ are the state and input vectors, respectively, at time $t \in \mathbb{N}$, with $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$, and where we assume that $\mathbf{Rank}(B) = m$. In this chapter we study the problem of computing a controller $K \in \mathbb{R}^{m\times n}$ which shapes the closed loop trajectory

$$x(t+1) = (A - BK)x(t) \tag{5.2}$$

according to some design objectives. We assume that the model of the dynamical system, i.e., matrices A and B, is not available and, instead, we leverage a series of offline open-loop trajectories as described in Chapter 2.

In the following we let $\mathcal{L} = \{\lambda_1, \ldots, \lambda_n\}$ be the set of desired closed-loop eigenvalues and $\mathcal{V} = \{v_1, \ldots, v_n\}$ be the set of desired closed-loop eigenvectors, with $\lambda_i \in \mathbb{C}$ and $v_i \in \mathbb{C}^n$, for all $i \in \{1, \ldots, n\}$.[1]

Assumption 26 (Properties of closed-loop eigenvalues and eigenvectors) *We assume that the set of desired eigenvalues $\mathcal{L}$ of $A - BK$ is closed under complex conjugation, and that, for each eigenvalue, the geometric multiplicity matches the algebraic multiplicity. Eigenvectors corresponding to complex conjugate eigenvalues are complex conjugate.* □

We leave the problem of generalizing the results of this chapter to eigenvalues with different geometric and algebraic multiplicity as the focus for future research, and note that the condition on the multiplicity of the eigenvalues in Assumption 26 is met when the elements of $\mathcal{L}$ are distinct.

A well-known limitation of the eigenstructure assignment problem is that it does not allow for arbitrary selection of eigenvalue/eigenvector pairs through feedback gain K. Instead, the choice of each eigenvector is constrained to a specific subspace within the system's state space. This restriction is formally expressed in the following theorem.

Theorem 27 (Feasibility of eigenstructure assignment [Liu and Patton, 1998]) *Using the control law $u(t) = -Kx(t)$ for system (5.1) n eigenvalues of $A - BK$ may be assigned and m entries of each corresponding eigenvector can be chosen freely.* □

[1] Although both left and right eigenvectors can be considered, in this chapter we shall refer to the right eigenvectors.

Theorem 27 ensures that the eigenvalues in $\mathcal{L} = \{\lambda_1, \ldots, \lambda_n\}$ can be freely assigned, but restricts the choice of each eigenvector $v_i \in \mathbb{R}^n$ in $\mathcal{V} = \{v_1, \ldots, v_n\}$ to a subspace $\mathcal{P}_i \subseteq \mathbb{R}^n$, termed the *allowable eigenvector subspace* [Liu and Patton, 1998]. We recall a second fundamental result which ensures the existence and uniqueness of K under certain conditions.

Theorem 28 (Uniqueness of feedback for eigenstructure assignment [Moore, 1976]) *Let $\mathcal{L} = \{\lambda_1, \ldots, \lambda_n\}$ and $\mathcal{V} = \{v_1, \ldots, v_n\}$, with $v_i \in \mathcal{P}_i$, be the set of desired closed-loop eigenvalues and eigenvectors, with* $\mathbf{Rank}(\begin{bmatrix} v_1 & \ldots & v_n \end{bmatrix}) = n$. *Let Assumption 26 hold, and let* $\mathbf{Rank}(B) = m$. *Then, the matrix K such that $A - BK$ has eigenvalues in $\mathcal{L}$ and eigenvectors in $\mathcal{V}$ exists and is unique.* □

In the pole placement problem, the set of desired closed-loop eigenvectors is not specified. This leaves some degrees of freedom in the selection of the columns of $\mathcal{V}$, as long as $v_i \in \mathcal{P}_i$ (cf. Theorem 29). In general, the matrix K in the pole placement problem is, therefore, not unique. In the next section we give a data-driven expression for $\mathcal{P}_i$, as well as a data-driven characterization of the set of matrices K such that $\rho(A - BK) = \mathcal{L}$. Further, we show that the unique K can be found as a closed-form function of the data for the eigenstructure assignment problem. In Section 5.5 we leverage the flexibility on K for the pole placement problem, by introducing additional design goals, i.e., by enforcing sparsity constraints on K.

5.4 Data-driven pole placement and eigenstructure assignment

We begin with a data-driven expression to compute $\mathcal{P}_i$, the allowable eigenvector subspace associated with eigenvalue λ_i for system (5.1). That is, we wish to find the subspace $\mathcal{P}_i$ such that $(A-BK)\upsilon_i = \lambda_i \upsilon_i$, for all $\upsilon_i \in \mathcal{P}_i$, and some K. We remark that $\mathcal{P}_i$ is independent of K [Liu and Patton, 1998], and in this work we compute $\mathcal{P}_i$ without any explicit knowledge of system matrices A and B, but by leveraging only offline data (2.2). Throughout this chapter, given $\lambda_i \in \mathcal{L}$, we define

$$\Lambda_i = \begin{bmatrix} I & \lambda_i I & \lambda_i^2 I & \cdots & \lambda_i^{T-1} I & \lambda_i^T I \end{bmatrix}^\top \in \mathbb{R}^{n(T+1)\times n}, \tag{5.3}$$

and matrices $Z = \begin{bmatrix} I_{nT} & 0_{nT\times n} \end{bmatrix}$ (i.e., the matrix that extracts the first nT rows from Λ_i) and $W = \begin{bmatrix} 0_{nT\times n} & I_{nT} \end{bmatrix}$ (i.e., the matrix that extracts the last nT rows from Λ_i).

Theorem 29 (Data-driven allowable eigenvector subspace) *Let λ_i be a desired closed-loop eigenvalue. The eigenvector associated with λ_i belongs to the following subspace:*

$$\mathcal{P}_i = X_0 K_U \begin{bmatrix} I & 0 \end{bmatrix} \mathbf{Ker}\left(\begin{bmatrix} XK_U - W\Lambda_i X_0 K_U & XK_0 \end{bmatrix} \right). \tag{5.4}$$

Proof. First, we notice that if λ_i and υ_i are an eigenvalue and the corresponding eigenvector of $A-BK$, then the following must hold

$$(A-BK)\upsilon_i = \lambda_i \upsilon_i. \tag{5.5}$$

Further, only a trajectory $\mathbf{x}_T$ starting in υ_i will always remain in υ_i when evolving according to (5.2). That is, if and only if $x(0) = \upsilon_i \in \mathcal{P}_i$, then $x(t) \in \mathbf{Im}(\upsilon_i)$ for all times t, and

$x(t+1) = \lambda_i x(t)$. For a trajectory of length T, this can be written as

$$\mathbf{x}_T = \begin{bmatrix} \lambda_i I \\ \lambda_i^2 I \\ \vdots \\ \lambda_i^T I \end{bmatrix} x(0) = \begin{bmatrix} XK_U & XK_0 \end{bmatrix} \begin{bmatrix} \alpha \\ \\ \beta \end{bmatrix} \tag{5.6}$$

if and only if $x(0) \in \mathcal{P}_i$. From the rightmost equality in (5.6) and by noticing that $x(0) = X_0 K_U \alpha$ (cf. Lemma 2) one can conclude that $x(0) \in \mathcal{P}_i$ if and only if α and β verify

$$\begin{bmatrix} \alpha \\ \\ \beta \end{bmatrix} \in \mathbf{Ker} \left(\begin{bmatrix} XK_U - W\Lambda_i X_0 K_U & XK_0 \end{bmatrix} \right). \tag{5.7}$$

Extracting α from (5.7) and recalling that

$$x(0) = v_i = X_0 K_U \alpha, \tag{5.8}$$

concludes the proof. ■

Through Theorem 29 one can write the allowable eigenvector subspace associated to eigenvalue λ_i for a closed-loop dynamics (5.2) as a function of the open-loop data (2.2). However, Theorem 29 cannot be used to compute a static feedback controller K. In fact, simply imposing $\mathbf{u}_T = UK_0\beta$ with β as in (5.7) might result in an input generated by a non-static feedback. Next, we give a condition that restricts the choice of β so that $\mathbf{u}_T$ is the result of a static feedback of the state, i.e., $u(t) = -Kx(t)$. Specifically, in Theorem 30 we characterize the set of all static feedback K that precisely place the eigenvalues of $A - BK$ to the desired set $\mathcal{L}$.

Theorem 30 (Data-driven pole placement) *Let $\mathcal{L} = \{\lambda_1, \ldots, \lambda_n\}$ be the set of desired*

closed-loop eigenvalues. Then, $\rho(A-BK)=\mathcal{L}$ if and only if $K\in\mathcal{K}$, where

$$\mathcal{K}=\left\{K:\bigcap_{i=1}^{n}\mathbf{Ker}\left(\begin{bmatrix}(I\otimes K)Z\Lambda_i X_0K_U & UK_0\end{bmatrix}\right)\neq 0\right\}. \tag{5.9}$$

Proof. Let λ_i be a desired closed loop eigenvalue and let $x(0)\in\mathcal{P}_i$. Then $x(t+1)=\lambda_i x(t)=\lambda_i^{t+1}x(0)$. From $u(t)=-Kx(t)=-K\lambda_i^t x(0)$ we write

$$\mathbf{u}_T=-\begin{bmatrix}K & & & & \\ & K & & & \\ & & K & & \\ & & & \ddots & \\ & & & & K\end{bmatrix}\begin{bmatrix}I\\ \lambda_i I\\ \lambda_i^2 I\\ \vdots\\ \lambda_i^{T-1}I\end{bmatrix}x(0). \tag{5.10}$$

We recall that $\mathbf{u}_T=UK_0\beta$ and $x(0)=X_0K_U\alpha$ in (5.10) (cf. Lemma 2) and therefore we can write

$$UK_0\beta=(I_T\otimes -K)Z\Lambda_i X_0K_U\alpha, \tag{5.11}$$

for all $i\in\{1,\ldots,n\}$. Equation (5.11) needs to be verified for every Λ_i and therefore the vectors α and β need to satisfy

$$\begin{bmatrix}\alpha\\ \beta\end{bmatrix}\in\bigcap_{i=1}^{n}\mathbf{Ker}\left(\begin{bmatrix}(I\otimes K)Z\Lambda_i X_0K_U & UK_0\end{bmatrix}\right). \tag{5.12}$$

That is, K is a static feedback (cf. (5.10)) such that $\rho(A-BK)=\mathcal{L}$ if and only if there exist a non-zero vector $[\alpha^\top\ \beta^\top]^\top$ satisfying (5.12). From this conclusion, the condition (5.9) on $\mathcal{K}$ is directly derived. ■

Through Theorem 30 we can characterize the set of feedback gains such that $\rho(A-BK)=\mathcal{L}$, that is, *all* the feedback gains which solve the pole placement problem for

a given set $\mathcal{L}$. This condition will be used in Section 5.5 with the aim of extracting matrices K from $\mathcal{K}$ which satisfy some desired sparsity pattern.

We conclude this section by leveraging Theorem 29 and 30 to find a closed-form expression for the eigenstructure assignment problem, i.e., when both $\mathcal{L}$ and $\mathcal{V}$ are given. For simplicity, and without affecting the generality of the approach, we limit the data collection phase in (2.2) to $T = 1$.

Theorem 31 (Closed-form expression of the feedback gain for eigenstructure assignment) *Let $\mathcal{L} = \{\lambda_1, \ldots, \lambda_n\}$ and $\mathcal{V} = \{v_1, \ldots, v_n\}$, with $v_i \in \mathcal{P}_i$, be the set of desired closed loop eigenvalues and eigenvectors. Let*

$$\begin{bmatrix} \alpha_i \\ \\ \beta_i \end{bmatrix} = \mathbf{Basis}\left(\mathbf{Ker}\left(\begin{bmatrix} XK_U - \lambda_i X_0 K_U & XK_0 \end{bmatrix}\right)\right)\gamma_i, \tag{5.13}$$

where X, X_0, and U are as in (2.2) with $T = 1$, and γ_i satisfies $\mathbf{Basis}(\mathcal{P}_i)\gamma_i = v_i$. The closed-loop matrix $A - BK$, with

$$K = -UK_0 \begin{bmatrix} \beta_1 & \ldots & \beta_n \end{bmatrix} \left(X_0 K_U \begin{bmatrix} \alpha_1 & \ldots & \alpha_n \end{bmatrix}\right)^{-1}, \tag{5.14}$$

has eigenvalues $\mathcal{L}$ and eigenvectors $\mathcal{V}$. □

Proof. From condition (5.12), we seek the K such that

$$\begin{bmatrix} \alpha_i \\ \\ \beta_i \end{bmatrix} \in \mathbf{Ker}\left(\begin{bmatrix} KX_0U & UK_0 \end{bmatrix}\right), \ \forall i = \{1, \ldots, n\}, \tag{5.15}$$

with α_i, β_i defined in (5.13), and where $(I_T \otimes K)Z\Lambda_i = K$ from $T = 1$. We can write condition (5.15) on K as

$$\begin{bmatrix} KX_0K_U & UK_0 \end{bmatrix} \begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_n \\ \\ \beta_1 & \beta_2 & \cdots & \beta_n \end{bmatrix} = 0, \tag{5.16}$$

from which we find that

$$KX_0K_U \begin{bmatrix} \alpha_1 & \dots & \alpha_n \end{bmatrix} = -UK_0 \begin{bmatrix} \beta_1 & \dots & \beta_n \end{bmatrix}.$$

We notice that $X_0K_U \begin{bmatrix} \alpha_1 & \dots & \alpha_n \end{bmatrix}$ is invertible since both X_0K_U and $\begin{bmatrix} \alpha_1 & \dots & \alpha_n \end{bmatrix}$ are square matrices with full rank.[2] This ensures the existence and uniqueness of K and concludes the proof. ■

Thanks to Theorem 31 a closed-form solution for the eigenstructure assignment problem is found. When needed, Theorem 31 can be also used to find a solution for the pole placement problem by simply selecting any arbitrary $\mathcal{V}$ such that $v_i \in \mathcal{P}_i$, by leveraging Theorem 29. As we have discussed, fixing $\mathcal{L}$ while leaving more freedom on the choice of eigenvectors $\mathcal{V}$ renders K not unique. This allows for more interesting problems to be solved, for example, by imposing some sparsity constraints on matrix K. In the next section we explore strategies to leverage Theorem 29 and Theorem 30 to find a sparse $K \in \mathcal{K}$. These solutions are not closed-form but rather based on the solution of bilinear optimization programs.

5.5 Data-driven pole placement with sparse feedback matrices

Consider the pole placement problem with desired closed-loop eigenvalues $\mathcal{L}$. As previously discussed, each eigenvector v_i corresponding to eigenvalue λ_i must belong to a

[2]The fact that X_0K_U is full rank is a direct consequence of the assumption in (2.4), see also [Celi and Pasqualetti, 2022b]. The fact that $\Gamma = \begin{bmatrix} \alpha_1 & \cdots & \alpha_n \end{bmatrix}$ is full rank can be proved by contradiction. Assume, without loss of generality, that $\{\delta_1, \cdots, \delta_{n-1}\}$, $\delta_i \in \mathbb{R}$, exist such that $\alpha_n = \sum_{k=1}^{n-1} \delta_k \alpha_k$, i.e., α_n is a linear combination of the remaining columns of Γ, and therefore Γ is singular. Then, from (5.8), $v_n = X_0K_U\alpha_n = X_0K_U \sum_{k=1}^{n-1} \delta_k\alpha_k = \sum_{k=1}^{n-1} \delta_k v_k$. This would imply that the elements in $\mathcal{V}$ are not linearly independent, contradicting Assumption 26.

subspace $\mathcal{P}_i$. Let $\mathcal{K}$ in (5.9) be the set containing all matrices K such that $\rho(A-BK)=\mathcal{L}$. Then, we can look for a pair of K and $\mathcal{V}=\{v_1,\ldots,v_n\}$ that satisfy

$$\begin{aligned} \underset{K,\,\mathcal{V}}{\arg\min} \quad & f_1(K) \\ \text{subject to} \quad & K \in \mathcal{K}, \\ & v_i \in \mathcal{P}_i, \quad \forall i \in \{1,\ldots,n\}, \\ & f_2(K) = 0, \end{aligned} \tag{5.17}$$

where $f_1(K):\mathbb{R}^{m\times n}\to\mathbb{R}$ and $f_2(K):\mathbb{R}^{m\times n}\to\mathbb{R}^q$ are some functions of K. We notice that (5.17) is a bilinear optimization problem in the variables K and $\mathcal{V}$, which follows from the definition of $\mathcal{K}$ in (5.9). Different choices of $f_1(K)$ and $f_2(K)$ lead to the solution of different problems, as will be detailed next.

5.5.1 Data-driven minimum-gain pole placement with sparse static feedback

Problem (5.17) can be cast as a data-driven optimization problem thanks to the results of Section 5.4. In particular, the condition on $v_i \in \mathcal{P}_i$ can be imposed through Theorem 29, while $\mathcal{K}$ is characterized in Theorem 30. We now propose an optimization-based strategy to compute a static feedback K with sparsity constraints, directly from data.

Let $S \in \{0,1\}^{m\times n}$ be the binary matrix that specifies the sparsity structure of

feedback K. That is, we wish to find K such that

$$K_{ij} = \begin{cases} 0 & \text{if } S_{ij} = 1, \\ \star & \text{if } S_{ij} = 0. \end{cases} \tag{5.18}$$

Now, consider the following optimization problem to find a $K \in \mathcal{K}$ which has sparsity constraints as specified by S

$$\begin{aligned} \underset{\substack{K \\ \{\gamma_1,\ldots,\gamma_n\}}}{\arg\min} \quad & \frac{1}{2}\|K\|_F^2 \\ \text{subject to} \quad & \left\| \begin{bmatrix} (I_T \otimes K) Z \Lambda_i X_0 K_U & U K_0 \end{bmatrix} w_i \right\| = 0, \\ & w_i = \mathbf{Ker}\left(\begin{bmatrix} (X - W\Lambda_i X_0) K_U & X K_0 \end{bmatrix} \right) \gamma_i, \\ & \forall i \in \{1, \ldots, n\}, \\ & S \circ K = 0_{m \times n}. \end{aligned} \tag{5.19}$$

The above is a data-driven implementation of (5.17), with $f_1(K) = \|K\|_F^2$ and $f_2(K) = S \circ K$. Minimizing the norm of the gain in $f_1(K)$ reduces the overall control effort, while $f_2(K)$ imposes the desired sparsity pattern on K. We notice that υ_i depends on the choice of γ_i, since $\upsilon_i = X_0 K_U \begin{bmatrix} I & 0 \end{bmatrix} w_i$, as a direct consequence of (5.4).

Remark 32 (Feasibility of (5.19)) *There is no known procedure to determine the feasibility of* (5.19)*, i.e., if $K \in \mathcal{K}$ exists such that $S \circ K = 0$. In general, assessing the existence of a sparse static feedback K is an NP-hard problem even when (A, B) are known [Katewa and Pasqualetti, 2021]. Therefore, in the following, we shall assume the feasibility of* (5.19)*. We refer the interested reader to [Katewa and Pasqualetti, 2021] for a detailed analytical characterization of the locally optimal solution of* (5.17) *in terms of the eigenvector matrices of the closed-loop system.* □

Remark 33 (Fixed modes of $(A - BK)$ [Sezer and Šiljak, 1981]) *We recall that the fixed modes of (A, B) with respect to the sparsity constraints S are the eigenvalues of A that cannot be changed using a sparse state feedback. When the fixed modes of (A, B) do not belong to $\mathcal{L}$, problem* (5.19) *becomes unfeasible and a different sparsity constraint S must be selected.* □

We now discuss a numerical implementation of (5.19).

Example 34 (Data-driven sparse feedback) *We consider the discretized version of a batch reactor system [Walsh and Ye, 2001] (with sampling time of* $0.1s$*), with*

$$A = \begin{bmatrix} 1.178 & 0.001 & 0.511 & -0.403 \\ -0.051 & 0.661 & -0.011 & 0.061 \\ 0.076 & 0.335 & 0.560 & 0.382 \\ 0 & 0.335 & 0.089 & 0.849 \end{bmatrix}, B = \begin{bmatrix} 0.004 & -0.087 \\ 0.467 & 0.001 \\ 0.213 & -0.235 \\ 0.213 & -0.016 \end{bmatrix},$$

which is open-loop unstable with $\sigma(A) = \{1.2200, 1.0049, 0.4206, 0.6025\}$. We set $T = 10$ and collect a series of $N = n + mT = 24$ experiments (2.2) *satisfying the assumption in* (2.4)*. We select the desired closed-loop eigenvalues $\mathcal{L} = \{-0.3, 0.2, 0.5, 0.7\}$ and the sparsity pattern*

$$S = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Problem (5.19) *is implemented in* `MATLAB` *and solved using the* `fmincon` *routine. A stabilizing K, satisfying the conditions required by $\mathcal{L}$ and S is found, with*

$$K = \begin{bmatrix} 0.0000 & 2.7633 & 2.7324 & 0.4122 \\ -2.3621 & 1.2654 & -0.0000 & 1.1906 \end{bmatrix}, \tag{5.20}$$

and

$$V = \begin{bmatrix} 0.0475 & -0.1938 & -0.2007 & -0.5204 \\ 0.9606 & -0.8211 & -0.6873 & 0.3220 \\ -0.2581 & 0.5266 & 0.5699 & -0.2354 \\ 0.0914 & 0.1046 & 0.4032 & -0.7550 \end{bmatrix}, \tag{5.21}$$

where $V = \begin{bmatrix} v_1 & \dots & v_n \end{bmatrix}$. *We remark that* K *and* V *are not unique, in general.* □

5.5.2 Data-driven maximally sparse feedback

As a second example of the application of the results of Section 5.4 to the data-driven design of sparse controllers, we consider the problem of finding the maximally sparse feedback K. In this scenario we are not seeking for a feedback with a defined sparsity pattern, as specified in S, but rather the one which has the most number of entries at zero. This is done by removing the specification on the sparsity pattern and by minimizing $\sum_{ij} |K_{ij}|$ which is often used as a proxy for the L_0-norm of a matrix. In this case, problem (5.17) can be written as

$$\begin{aligned} \underset{\substack{K \\ \{\gamma_1,\dots,\gamma_n\}}}{\arg\min} \quad & \sum_{ij} |K_{ij}| \\ \text{subject to} \quad & \begin{bmatrix} (I_T \otimes K) Z \Lambda_i X_0 K_U & U K_0 \end{bmatrix} w_i = 0, \\ & w_i = \mathbf{Ker} \left(\begin{bmatrix} (X - W \Lambda_i X_0) K_U & X K_0 \end{bmatrix} \right) \gamma_i, \\ & \forall i \in \{1, \dots, n\}. \end{aligned} \tag{5.22}$$

This is, again, a bilinear optimization problem. We now show a numerical implementation of this approach.

Example 35 (Data-driven maximally sparse feedback) *Consider the same problem setting as in Example 34. We let* $\mathcal{L} = \{-0.3, 0.2, 0.5, 0.7\}$ *be sets of the desired closed-loop eigenvalues. By running* (5.22) *we find*

$$K = \begin{bmatrix} 0.0000 & 1.6901 & 0.0000 & 4.4741 \\ -1.9515 & 0.0000 & -1.0042 & 0.0000 \end{bmatrix}$$

and

$$V = \begin{bmatrix} 0.7460 & 0.5153 & 0.0752 & -0.0053 \\ 0.1892 & 0.4018 & 0.9439 & 0.9901 \\ -0.6318 & -0.7451 & -0.2829 & 0.1127 \\ -0.0922 & -0.1332 & -0.1532 & 0.0834 \end{bmatrix}.$$

Despite the non-convexity of (5.22) *we obtain a sparse controller with a total of* 4 *entries at zero.* □

5.6 Conclusions

In this chapter we consider a data-driven strategy for the design of, possibly sparse, feedback gains for pole placement and eigenstructure assignment. Given a set of desired closed-loop eigenvalues $\mathcal{L}$, we characterize the allowable eigenvector subspaces of the dynamical system described by unknown (A, B), as well as the set $\mathcal{K}$ of feedback gains such that $\rho(A - BK) = \mathcal{L}$ for all $K \in \mathcal{K}$. For the eigenstructure assignment problem, we give a closed-form data-driven expression for the gain K which assigns the desired closed-loop eigenvalues $\mathcal{L}$ together with the associated desired eigenvectors $\mathcal{V}$. Further, we discuss optimization-based strategies to find a sparse gain $K \in \mathcal{K}$ when a desired sparsity structure needs to be imposed on K, or when the overall sparsity of K (i.e., the number of

zero elements of K) needs to be maximized. Numerical simulations complement our analysis. Future work includes a characterization of the performance of these tools when data is collected with noise, together with a comparison with model-based methods based on system identification from data. In the next chapter we take a step further: instead of studying a specific problem to solve through the lenses of data, we propose a framework for a data-driven approach to geometric control. As we will see, this will unlock a collection of problems (and their respective solutions) especially in security, e.g., designing and mitigating attacks on systems.

Chapter 6

A data-driven approach to geometric control

Studying structural properties of linear dynamical systems through invariant subspaces is one of the key contributions of the geometric approach to system theory. In general, a model of the dynamics is required in order to compute the invariant subspaces of interest. In this chapter we overcome this limitation by finding direct data-driven formulas for some of the foundational tools of the geometric approach. We use our results to (i) find a feedback gain that confines the system state within a subspace, (ii) compute the invariant zeros of the unknown system, and (iii) design attacks that remain undetectable.

6.1 Introduction

The geometric approach is a collection of notions and algorithms for the analysis and control of dynamical systems. Differently from the classic methods in the frequency

and state space domains [Åström and Murray, 2010], the geometric approach offers an intuitive and coordinate-free analysis of the properties of dynamical systems in terms of appropriately defined subspaces, and synthesis algorithms based on subspace operations, such as sum, intersection, and orthogonal complementation. The geometric approach has been successfully used to solve a variety of complex control and estimation problems; we refer the interested reader to [Basile and Marro, 1991] for a detailed treatment of the main geometric control notions and their applications.

Similarly to the frequency and state-space approaches to control, the geometric approach assumes an accurate, in fact exact, representation of the system dynamics. To overcome this limitation and in response to an ever-increasing availability of sensors, historical data, and machine learning algorithms, the behavioral approach, and more generally a data-driven approach, has seen a rapid increase in popularity. Here, system analysis and control synthesis do not require a model of the dynamics and are instead obtained directly from experimental data reflecting the system dynamics [Markovsky and Rapisarda, 2008].

While analysis, control and estimation problems can often be solved equivalently using different methods, the frequency, state-space, geometric, and data-driven approaches all offer complementary insights into the structure and properties of the system dynamics, and together contribute to forming a comprehensive theory of systems. In this chapter we create the first connections between the geometric and data-driven approaches to system analysis and control. In particular, we derive data-driven expressions of the fundamental sets used in the geometric approach to solve a variety of control and estimation problems, and show how these sets have an even more insightful and straightforward interpretation

when analyzed in the higher-dimensional data space as compared to their geometric view in the lower-dimensional state space.

6.2 Related work

From the first definitions of controlled and conditioned invariant subspaces, the geometric approach to control has evolved over the last decades into a full theory and a set of algorithms for linear [Basile and Marro, 1991] and nonlinear [Isidori, 1995] systems. Applications of the geometric approach include the disturbance decoupling [Wonham and Morse, 1970] and fault detection [Massoumnia et al., 1989] problems, the characterization of stealthy attacks in cyber-physical systems [Pasqualetti et al., 2013], and the secure state estimation problem [Fawzi et al., 2014].

The data-driven approach to system analysis and control is receiving renewed and increased interest. While traditional indirect data-driven methods use data to identify a model of the system [Gevers, 2005] and proceed to synthesize a controller in a second step, direct data-driven methods bypass (at least apparently [Krishnan and Pasqualetti, 2021, Dörfler et al., 2021, van Waarde et al., 2020]) the identification step and design control actions directly from data. In this framework, recent results tackle various problems for linear systems, including optimal [Baggio et al., 2021, Monshizadeh, 2020], robust [De Persis and Tesi, 2020] and distributed [Allibhoy and Cortés, 2020, Celi et al., 2021, Garrabé and Russo, 2021, Jiao et al., 2021] control, as well as unknown-input estimation [Turan and Ferrari-Trecate, 2021]. We refer the reader to [Markovsky and Dörfler, 2021] for a recent survey on data-driven control. The intersection of geometric and data-driven control has

remained relatively unexplored, with the remarkable exception of [Eising, 2021]. Next we recall some of the key tools of the geometric approach to control, namely controlled invariants and conditioned invariants.

6.3 Controlled and conditioned invariant subspaces

The notions of controlled and conditioned invariant subspaces are the basis of the geometric approach for the analysis and control of linear systems. We now recall their definition and basic properties. We refer the interested reader to [Basile and Marro, 1991, Wonham, 1985, Trentelman et al., 2001] for a detailed treatment of this subject.

Definition 36 ($(A, \mathcal{B})$-controlled invariant) *Given a matrix $A \in \mathbb{R}^{n\times n}$ and a subspace $\mathcal{B} \subseteq \mathbb{R}^n$, a subspace $\mathcal{V} \subseteq \mathbb{R}^n$ is an $(A, \mathcal{B})$-controlled invariant subspace if*

$$A\mathcal{V} \subseteq \mathcal{V} + \mathcal{B}. \tag{6.1}$$

When $\mathcal{B} = \mathbf{Im}(B)$, the notion of controlled invariance refers to the possibility of confining the state trajectory of the system (2.1) within a subspace. Specifically, the subspace $\mathcal{V}$ is an $(A, \mathbf{Im}(B))$-controlled invariant subspace if, for every initial state in $\mathcal{V}$, there exists a control input such that the state belongs to $\mathcal{V}$ at all times. Of particular interest is $\boldsymbol{\mathcal{V}^*}$, the largest $(A, \mathbf{Im}(B))$-controlled invariant subspace contained in $\mathbf{Ker}(C)$. The subspace $\boldsymbol{\mathcal{V}^*}$ contains all trajectories of (2.1) that generate an identically zero output. Hence, it holds that $\boldsymbol{\mathcal{V}^*} = \{0\}$ if and only if the system (2.1) features no invariant zeros, a notion that is at the basis of the analysis of stealthy attacks and unknown-input observers [Pasqualetti et al., 2013], among others.

Definition 37 ($(A,\mathcal{C})$-conditioned invariant) *Given a matrix $A \in \mathbb{R}^{n\times n}$ and a subspace $\mathcal{C} \subseteq \mathbb{R}^n$, a subspace $\mathcal{S} \subseteq \mathbb{R}^n$ is an $(A,\mathcal{C})$-conditioned invariant subspace if*

$$A(\mathcal{S} \cap \mathcal{C}) \subseteq \mathcal{S}. \tag{6.2}$$

When $\mathcal{C} = \mathbf{Ker}(C)$, the notion of conditioned invariance arises in the context of state estimation. Specifically, the subspace $\mathcal{S}$ is an $(A, \mathbf{Ker}(C))$-conditioned invariant subspace if it is possible to design an (asymptotic) observer that reconstructs the state up to a canonical projection onto $\mathbb{R}^n \setminus \mathcal{S}$ by processing the initial condition, the input, and the measurements of the system (2.1). Of particular interest is $\boldsymbol{\mathcal{S}}^*$, the smallest $(A, \mathbf{Ker}(C))$-conditioned invariant subspace containing $\mathbf{Im}(B)$. In fact, the orthogonal complement of $\boldsymbol{\mathcal{S}}^*$ is the largest subspace that can be estimated with a dynamic observer in the presence of an unknown input.

The subspaces $\boldsymbol{\mathcal{V}}^*$ and $\boldsymbol{\mathcal{S}}^*$ can be conveniently computed using simple recursive algorithms [Basile and Marro, 1991]. Further, these subspaces can be used to characterize important properties of the system (2.1). For instance, the system (2.1) is right invertible if and only if $\boldsymbol{\mathcal{V}}^* \cup \boldsymbol{\mathcal{S}}^* = \mathbb{R}^n$, and left invertible if and only if the subspace $\boldsymbol{\mathcal{R}}^* = \boldsymbol{\mathcal{V}}^* \cap \boldsymbol{\mathcal{S}}^*$ is the trivial subspace [Basile and Marro, 1991]. It should be noticed that $\boldsymbol{\mathcal{R}}^*$ coincides with the largest subspace that can be reached from the origin with trajectories that belong to $\boldsymbol{\mathcal{V}}^*$ at all times (hence, generating an identically zero output).

The definition of the subspaces $\boldsymbol{\mathcal{V}}^*$, $\boldsymbol{\mathcal{S}}^*$ and $\boldsymbol{\mathcal{R}}^*$, as well as the algorithms to compute them, assume the exact knowledge of the system matrices. Instead, in the remainder of the chapter we derive purely data-driven expressions of these subspaces, which also offer an alternative interpretation of them. Similarly to how $\boldsymbol{\mathcal{V}}^*$, $\boldsymbol{\mathcal{S}}^*$ and $\boldsymbol{\mathcal{R}}^*$ are used in the geo-

metric approach, our data-driven formulas can also be used to solve a variety of estimation and control problems.

6.4 Data-driven geometric control

We begin with finding a data-driven expression of the subspace $\boldsymbol{\mathcal{V}}^*$ for the system (2.1), the largest $(A, \mathbf{Im}(B))$-controlled invariant subspace contained in $\mathbf{Ker}(C)$.

Theorem 38 (Data driven formula for $\boldsymbol{\mathcal{V}}^*$) *Let X_0 and Y be as in (2.2), with $T \geq n$. Then,*

$$\boldsymbol{\mathcal{V}}^* = X_0 \mathbf{Ker}(Y). \tag{6.3}$$

Notice that (6.3) requires only the knowledge of the initial state X_0 and the output trajectory Y of N experiments . To prove Theorem 38, recall that $\boldsymbol{\mathcal{V}}^*$ is the set of initial states for which there exists a control input such that the resulting state trajectory generates an identically zero output. Since the system is linear, under our assumption of persistently exciting experimental inputs, any system trajectory can be expressed as an appropriate linear combination of the experimental trajectories. This has been formalized for our framework in Lemma 2.

The following instrumental Lemma shows that it is sufficient to consider trajectories of any finite length $T \geq n$ to compute $\boldsymbol{\mathcal{V}}^*$, and is instrumental to the proof of Theorem 38.

Lemma 39 (Computing $\boldsymbol{\mathcal{V}}^*$ from trajectories of finite length) *For the system (2.1), any initial state x_0, and any finite horizon $T \geq n$, the following statements are equivalent:*

(i) $x_0 \in \boldsymbol{\mathcal{V}}^*$*;*

(ii) there exists an input sequence $u(0), \ldots, u(T-1)$ *such that* $y(t) = 0$ *for all* $t \in \{0, \ldots, T-1\}$.

Proof.

(i) $\Rightarrow$ (ii) Follows from the definition of $\boldsymbol{\mathcal{V}}^*$.

(ii) $\Rightarrow$ (i) Notice that $y(T-1) = Cx(T-1) = 0$. Thus, $x(T-1) \in \mathbf{Ker}(C) = \mathcal{V}_0$.

Similarly, $x(T-2)$ satisfies

$$x(T-1) = Ax(T-2) + Bu(T-2), \text{ and}$$

$$y(T-2) = Cx(T-2) = 0.$$

This implies that

$$x(T-2) \in A^{-1}(x(T-1) - Bu(T-2))$$

$$\subseteq A^{-1}(\mathcal{V}_0 + \mathbf{Im}(B)) \cap \mathbf{Ker}(C) = \mathcal{V}_1$$

Iterating this procedure yields

$$x(T-1) \in \mathcal{V}_0 = \mathbf{Ker}(C), \text{ and} \tag{6.4a}$$

$$x(T-i) \in \mathcal{V}_i = A^{-1}(V_{i-1} + \mathbf{Im}(B)) \cap \mathbf{Ker}(C). \tag{6.4b}$$

Since $\mathcal{V}_i$ converges to $\boldsymbol{\mathcal{V}}^*$ is at most n steps [Basile and Marro, 1991], we have that $x(T-\tau) \in \boldsymbol{\mathcal{V}}^*$ for all $\tau \geq n$, which concludes the proof. ■

We now prove Theorem 38 using Lemma 2 and 39.

Proof of Theorem 38: From Lemma 39 we seek all initial conditions x_0 for which the output can be maintained at zero for $T \geq n$ steps. From (2.5), the vectors α and β that identify state trajectories with identically zero output must satisfy

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} \in \mathbf{Ker} \begin{bmatrix} YK_U & YK_0 \end{bmatrix}. \tag{6.5}$$

The initial condition corresponding to such trajectories is $x_0 = X_0 K_U \alpha$ (cf. Lemma 2). Thus, using (6.5), $\boldsymbol{\mathcal{V}}^*$ becomes $\boldsymbol{\mathcal{V}}^* = \begin{bmatrix} X_0 K_U & 0 \end{bmatrix} \mathbf{Ker} \begin{bmatrix} Y K_U & Y K_0 \end{bmatrix} = X_0 \mathbf{Ker}(Y)$. The last equality follows from some algebraic manipulations, that are here omitted. ■

We next find a data-driven expression for $\boldsymbol{\mathcal{S}}^*$, the smallest $(A, \mathbf{Ker}(C))$ conditioned invariant containing $\mathbf{Im}(B)$.

Theorem 40 (Data driven formula for $\boldsymbol{\mathcal{S}}^*$) *Let X_0, X_F and Y be as in (2.2), with $T \geq n$. Then,*

$$\boldsymbol{\mathcal{S}}^* = X_F K_0 \mathbf{Ker}\,(Y K_0). \tag{6.6}$$

To prove Theorem 40, we first show that, similarly to the case of $\boldsymbol{\mathcal{V}}^*$, the subspace $\boldsymbol{\mathcal{S}}^*$ can be computed from a collection of trajectories of finite length $T \geq n$.

Lemma 41 (Computing $\boldsymbol{\mathcal{S}}^*$ from trajectories of finite length) *For the system (2.1) and any finite horizon $T \geq n$, the following statements are equivalent:*

(i) $x(T) \in \boldsymbol{\mathcal{S}}^*$;

(ii) there exists an input sequence $u(0), \ldots, u(T-1)$ such that $y(t) = 0$ for all $t \in \{0, \ldots, T-1\}$ and $x(0) = 0$.

Proof.

(i) $\Rightarrow$ (ii) Follows from the definition of $\boldsymbol{\mathcal{S}}^*$. In fact, let $u(t) = 0$, for $t \in \{0, \ldots, T-2\}$ and $u(T-1) \neq 0$. Then $x(T) = Ax(T-1) + Bu(T-1) = Bu(t-1) \in \mathbf{Im}(B) \subseteq \boldsymbol{\mathcal{S}}^*$.

(ii) $\Rightarrow$ (i) Because $x(1) = Bu(0)$ and $y(1) = Cx(1) = 0$, we have $x(1) \in \mathbf{Im}(B) \cap \mathbf{Ker}(C) = \mathcal{S}_1 \cap \mathbf{Ker}(C)$. Similarly,

$$x(2) \in A(\mathcal{S}_1 \cap \mathbf{Ker}(C)) + \mathbf{Im}(B) = \mathcal{S}_2,$$

and $x(2) \in \mathbf{Ker}(C)$ since $y(2) = Cx(2) = 0$. Recursively:

$$x(1) \in \mathcal{S}_1 = \mathbf{Im}(B), \text{ and} \tag{6.7a}$$

$$x(i) \in \mathcal{S}_i = A(\mathcal{S}_{i-1} \cap \mathbf{Ker}(C)) + \mathbf{Im}(B). \tag{6.7b}$$

Since $\mathcal{S}_i$ converges to $\boldsymbol{\mathcal{S}}^*$ in at most n steps [Basile and Marro, 1991], we have that $x(\tau) \in \boldsymbol{\mathcal{S}}^*$ for all $\tau \geq n$, which concludes the proof. ■

We are now ready to prove Theorem 40.

Proof of Theorem 40: From (2.5), when $x(0) = 0$, any state trajectory of length T that generates an identically zero output of length T can be parametrized with $\alpha = 0$ and $\beta \in \mathbf{Ker}(YK_0)$. Using Lemma 41, the set $\boldsymbol{\mathcal{S}}^*$ can be equivalently written as the final states reached by such trajectories, that is, $\boldsymbol{\mathcal{S}}^* = X_F K_0 \mathbf{Ker}(YK_0)$, which concludes the proof. ■

Remark 42 (Obtaining $\boldsymbol{\mathcal{R}}^*$ from $\boldsymbol{\mathcal{V}}^*$ and $\boldsymbol{\mathcal{S}}^*$) *The combined knowledge of $\boldsymbol{\mathcal{V}}^*$ and $\boldsymbol{\mathcal{S}}^*$ allows us to find $\boldsymbol{\mathcal{R}}^*$ as [Basile and Marro, 1991]*

$$\boldsymbol{\mathcal{R}}^* = \boldsymbol{\mathcal{V}}^* \cap \boldsymbol{\mathcal{S}}^*. \tag{6.8}$$

□

Remark 43 (Direct vs indirect invariant subspaces computation) *A direct comparison between our formulas and classic indirect data-driven approaches is not straightforward. First, our formulas for $\boldsymbol{\mathcal{V}}^*$ and $\boldsymbol{\mathcal{S}}^*$ use data that is not sufficient to estimate the system matrices (e.g., the inputs U and the state trajectories X are not used in Theorems 38 and 40). Second, when inputs and outputs are available, system identification only provides the*

system matrices up to a similarity transformation. Hence, the subspaces computed with the identified matrices would not match the original subspaces since the similarity transformation remains unknown without state information. Third, system identification methods make assumptions (such as controllability and observability) that are not required for our approach. For example, let $A = \begin{bmatrix} 0.5 & 0 \\ 0 & 1 \end{bmatrix}$, $B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $C_1 = \begin{bmatrix} 3 & -5 \end{bmatrix}$, *and* $C_2 = \begin{bmatrix} 3 & 4 \end{bmatrix}$ *where* (A, B) *is not controllable. We notice that* $\Sigma_1 = (A, B, C_1)$ *and* $\Sigma_2 = (A, B, C_2)$ *share the same input-output relationship, i.e., there exist input-output trajectories which are compatible with both systems. However,* Σ_1 *and* Σ_2 *do not share the same* $\boldsymbol{\mathcal{V}}^*$*, in fact*

$$\boldsymbol{\mathcal{V}}^*{}_{\Sigma_1} = \begin{bmatrix} 1.00 \\ 0.60 \end{bmatrix}, \quad \boldsymbol{\mathcal{V}}^*{}_{\Sigma_2} = \begin{bmatrix} 1.00 \\ -0.75 \end{bmatrix}.$$

System identification using input-output data cannot distinguish between Σ_1 *and* Σ_2 *and, as a consequence, fails at estimating the correct invariant subspaces. Instead, our data-driven formulas work also in this situation.*

Although a formal discussion of how noise affects the identification of subspaces is beyond the scope of this work, in Fig. 6.1 we offer a qualitative comparison of how our approach performs with respect to a traditional system identification followed by a model based geometric approach. We observe that the tools presented in this chapter are robust to noise. We refer to the image caption for the implementation details and to [Celi and Pasqualetti, 2022a] for the code to reproduce this result. □

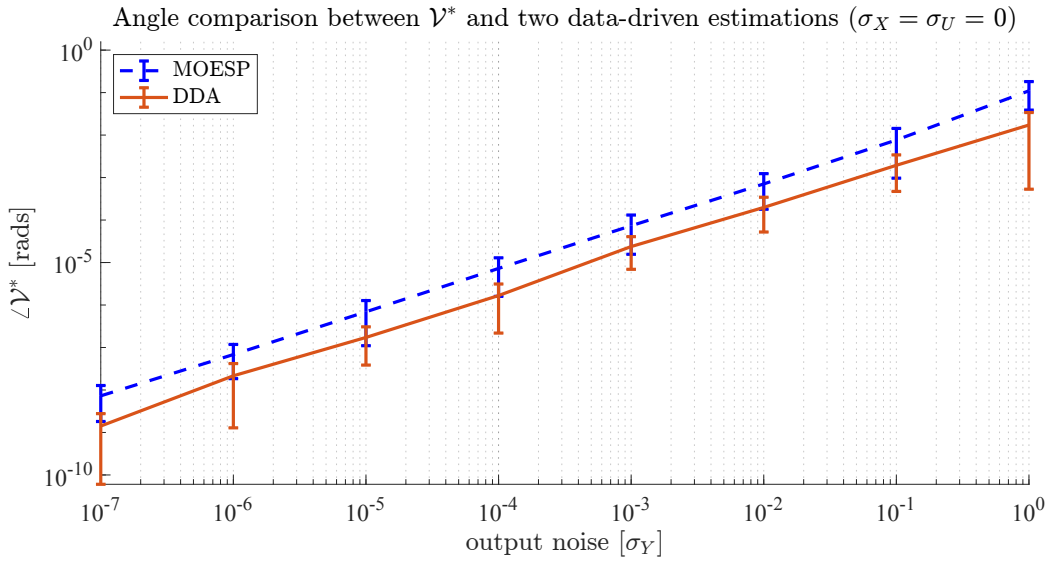

Figure 6.1: This figure shows a comparison for computing $\boldsymbol{\mathcal{V}^*}$ with two different data driven approaches for minimal system [Katayama, 2005, Example 6.6]. Outputs Y are collected with noise, i.e., $Y = \hat{Y} + \Delta_Y$, where Y is the measured data and Δ_Y is the noise matrix with i.i.d. entries, zero mean, and variance σ_Y. In blue (dashed) we show the result obtained trough the MOESP algorithm (see [Katayama, 2005]). In red (solid) we show the result obtained trough the approach proposed in this chapter (DDA). State trajectories X are assumed to be known only by MOESP, while DDA requires only X_0. For every approach and for every value of σ_Y we perform a total of 100 Montecarlo simulations and plot the mean value of the angle between the estimated and the model based $\boldsymbol{\mathcal{V}^*}$ (true subspace). $T = 50$ for MOESP, and $T = 3$, $N = 20$ for DDA.

6.5 Data-driven feedbacks for invariant subspaces and invariant zeros

The state of a system can be confined within a subspace $\mathcal{V}$ through a state-feedback controller if and only if $\mathcal{V}$ is a controlled invariant subspace. We continue this chapter with the data-driven design of such state-feedback controller, that is, the data-driven design of a matrix F such that

$$(A + BF)\mathcal{V} \subseteq \mathcal{V}. \tag{6.9}$$

For a trajectory X_T and input U_T, let

$$X_{0,T} = \begin{bmatrix} x(1) & x(2) & \cdots & x(T-1) \end{bmatrix}, \tag{6.10a}$$

$$X_{1,T} = \begin{bmatrix} x(2) & x(3) & \cdots & x(T) \end{bmatrix}, \text{ and} \tag{6.10b}$$

$$U_{0,T} = \begin{bmatrix} u(0) & u(1) & \cdots & u(T-1) \end{bmatrix}. \tag{6.10c}$$

Theorem 44 (Data-driven feedback for invariant subspace) *Let X_T be the trajectory of (2.1) with input U_T and some initial condition. Let $\mathcal{V} = \mathbf{Im}(V)$ be an $(A, \mathbf{Im}(B))$-controlled invariant subspace, and let*

$$F = U_{0,T}(X_{0,T}^{\dagger} + K\gamma), \tag{6.11}$$

with $K = \mathbf{Ker}(X_{0,T})$ and

$$\gamma = -((I - VV^{\dagger})X_{1,T}K)^{\dagger}(I - VV^{\dagger})X_{1,T}X_{0,T}^{\dagger}VV^{\dagger}. \tag{6.12}$$

If $[U_{0,T}^{\top}\ X_{0,T}^{\top}]^{\top}$ is full-row rank,[1] then $(A+BF)\mathcal{V} \subseteq \mathcal{V}$.

Proof. From [De Persis and Tesi, 2020, Theorem 2], for any state-feedback gain F, the closed loop matrix can be written as

$$A + BF = X_{1,T}G,$$

where the matrix G satisfies $X_{0,T}G = I$ and $U_{0,T}G = F$. Further, F renders the subspace $\mathcal{V}$ invariant if and only if $(A+BF)\mathcal{V} = X_{1,T}G\mathcal{V} \subseteq \mathcal{V}$, or, equivalently,

$$(I - VV^{\dagger})X_{1,T}GV = 0,$$

[1] This condition requires the trajectory to be sufficiently informative and is related to the notion of persistency of excitation [Willems et al., 2005, De Persis and Tesi, 2020, Markovsky and Dörfler, 2021].

where $V = \mathbf{Basis}(\mathcal{V})$ and $(I - VV^\dagger)$ is a projector onto $\mathcal{V}^\perp$. From $X_{0,T}G = I$ we obtain $G = X_{0,T}^\dagger + K\gamma$, where γ is any matrix verifying the equality

$$(I - VV^\dagger)X_{1,T}(X_{0,T}^\dagger + K\gamma)V = 0.$$

Solving for γ (a solution γ exists because $\mathcal{V}$ is an (A, B)-controlled invariant subspace and $[U_{0,T}^\top \; X_{0,T}^\top]^\top$ is full-row rank) and using $U_{0,T}G = F$ concludes the proof. ■

Theorem 44 details the computation of a feedback matrix that renders a subspace invariant, from sufficiently informative state and input trajectories. It should be noticed that Theorem 44 does not guarantee the internal, nor external, stability of the subspace, which imposes additional constraints on γ. This is left as a topic of future investigation.

To conclude this section we present a strategy to identify the invariant zeros of (2.1) from data. We make the assumption that (2.1) is such that $\boldsymbol{\mathcal{R}}^*$ is the trivial subspace. Systems with $\boldsymbol{\mathcal{R}}^* \neq \{0\}$ are intrinsically vulnerable to, e.g., undetectable malicious attacks with unstable state trajectories. On the other hand, when $\boldsymbol{\mathcal{R}}^* = \{0\}$, the existence of unstable invisible trajectories depends on the modulo of its invariant zeros. In fact, the knowledge of the number and magnitude of the invariant zeros when $\boldsymbol{\mathcal{R}}^* = \{0\}$ is essential when studying problems such as noninteracting control [Basile and Marro, 1991] and attack detection [Pasqualetti et al., 2013], motivating our interest in their identification.

Theorem 45 (Data-driven invariant zeros I) *Let X and $\boldsymbol{\mathcal{V}}^*$ be as in* (2.2) *and* (6.3)*, respectively, with $T \geq n$. Let $V = \mathbf{Basis}(\boldsymbol{\mathcal{V}}^*)$ and assume that $\boldsymbol{\mathcal{R}}^* = \{0\}$. Then, $z \in \mathbb{C}$ is an invariant zero of* (2.1) *if and only if the matrix*

$$\begin{bmatrix} XX^\dagger(I \otimes V) & -\left([z \; z^2 \; \cdots \; z^T] \otimes I\right)^\top \end{bmatrix} \tag{6.13}$$

has a nontrivial kernel.

Proof. When $\boldsymbol{\mathcal{V}}^* \neq \{0\}$ and $\boldsymbol{\mathcal{R}}^* = \{0\}$, there exists a trajectory $x(t) = z^t x(0)$, with $x(t) \in \boldsymbol{\mathcal{V}}^*$ for all $t \geq 0$ and z an invariant zero of (2.1) [Basile and Marro, 1991]. We write such trajectory as

$$X_T^V = [(zI)^\top \ \cdots \ (z^T I)^\top]v = \left([z \ \cdots \ z^T] \otimes I\right)^\top v. \tag{6.14}$$

With assumption in (2.4), any trajectory belongs to the image of the data matrix X. Then, when the trajectory X_T^V above exists, there also exists a vector $\bar{w} \in X^\dagger(I \otimes V)$ such that $X\bar{w} = X_T^V$. The condition on $\bar{w}$ imposes that the trajectory is compatible with (2.1) while evolving inside $\boldsymbol{\mathcal{V}}^*$. Both vectors $v \neq 0$ and $\bar{w} = X^\dagger(I \otimes V)w \neq 0$ exist if and only if

$$XX^\dagger(I \otimes V)w = \left([z \ z^2 \ \cdots \ z^T] \otimes I\right)^\top v \tag{6.15}$$

i.e., the kernel of $[XX^\dagger(I \otimes V) \quad -\left([z \ z^2 \ \cdots \ z^T] \otimes I\right)^\top]$ is non-empty, concluding the proof.

■

The invariant zeros of the system (2.1) can be equivalently characterized using data collected as in (6.10).

Lemma 46 (Data-driven invariant zeros II) *Let $\boldsymbol{\mathcal{V}}^*$ be as in* (6.3) *and assume that $\boldsymbol{\mathcal{R}}^* = \{0\}$. Let $T = \begin{bmatrix} T_1 & T_2 \end{bmatrix}$, with $T_1 = \textbf{Basis}(\boldsymbol{\mathcal{V}}^*)$, and T_2 chosen such that T is nonsingular. Finally, let $G = X_{0,T}^\dagger + K\gamma$, with γ defined as in* (6.12)*. Then, the invariant zeros of* (2.1) *are the eigenvalues of A_{11}, where*

$$T^{-1}(X_{1,T}G)T = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}. \tag{6.16}$$

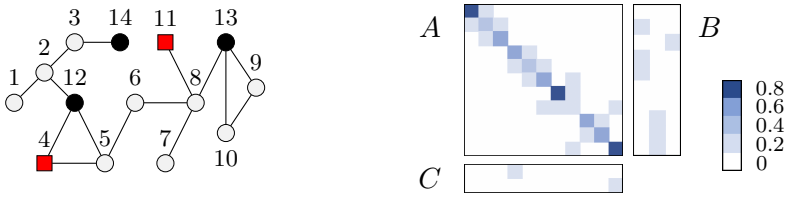

Figure 6.2: An example of consensus network. On the left, agents are numbered from 1 through 14, where nodes $\{12, 13, 14\}$ (in black) are the leaders and nodes $\{4, 11\}$ (square) are the network monitors. On the right, the weighted adjacency matrix for the follower nodes $\{1, \ldots, 11\}$ is shown, together with the input and output matrices (the numerical values of the entries of the matrices are color coded and belong to the set $\{0, 0.2, 0.4, 0.6, 0.8\}$).

Proof. This result derives from the facts that (i) the closed loop system with the state feedback $u = Fx$ satisfies

$$A + BF = X_{1,T}G, \tag{6.17}$$

(ii) the subspace $\boldsymbol{\mathcal{V}}^*$ is invariant for the closed-loop matrix $A + BF$, and (iii) the invariant zeros of (2.1) are the eigenvalues of the closed-loop matrix $A + BF$ contained in $\boldsymbol{\mathcal{V}}^*$. ■

6.6 Malicious attacks: an illustrative example

To illustrate a possible use of the theory we developed, consider the leader-follower consensus network in Fig. 6.2.

The network is equipped with two monitoring nodes, specifically, nodes 4 and 11. The state of the monitoring nodes is used to detect any anomalous behavior of the network from its nominal dynamics (see also [Pasqualetti et al., 2011]). We let an attacker take control of the leader nodes, and seek for an attack strategy that remains undetectable from the monitoring nodes, and leverages only historical data of the network dynamics. The attacker strategy is designed as follows: (i) compute $\boldsymbol{\mathcal{V}}^*$ and $\boldsymbol{\mathcal{S}}^*$ using Theorems 38 and 40,

respectively, and find $\boldsymbol{\mathcal{R}}^* = \boldsymbol{\mathcal{V}}^* \cap \boldsymbol{\mathcal{S}}^*$; (ii) for $R = \mathbf{Basis}(\boldsymbol{\mathcal{R}}^*)$, and X, U and K_0 defined as in (2.2), compute $P_1 \neq 0$ as

$$\begin{bmatrix} XK_0 & I \otimes R \end{bmatrix} \begin{bmatrix} P_1 \\ \\ P_2 \end{bmatrix} = 0; \tag{6.18}$$

and (iii) choose the attack input A_T as $A_T \in \mathbf{Im}(UK_0P_1)$. Then, for any initial state $x(0)$ and nominal control input U_T, the output of (2.1) with input U_T is indistinguishable from the output with input $U_T + A_T$. As can be seen in Fig. 6.3 from time $t = 10s$, the attacker strategy perturbs the state of the network but does not affect the monitoring nodes, thus remaining undetectable. In fact, it can be shown that any input $A_T \in \mathbf{Im}(UK_0P)$ moves the state trajectory within the controlled invariant $\boldsymbol{\mathcal{R}}^* \subseteq \mathbf{Ker}(C)$, thus affecting the state of the system but not its output.

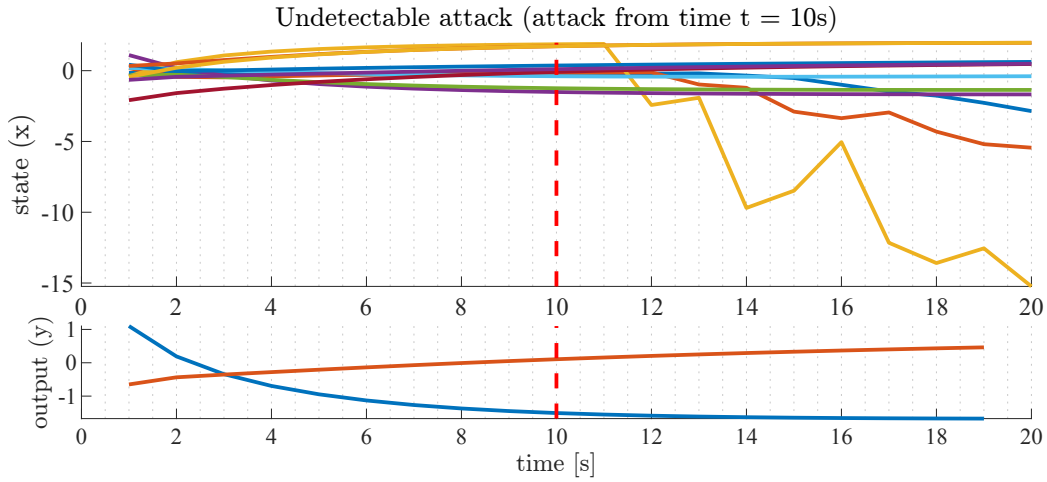

Figure 6.3: In this figure we show an attack on the network of Fig. 6.2. The systems initial condition is chosen randomly and the leaders impose $u = [-2\ 2\ 4]^\top$. The attacker waits for the system to reach its equilibrium and then, at time $t = 10s$, injects an attack A_T as proposed in Sec 6.6. We notice how the system state its perturbed from the equilibrium, while the output of the system remains unaffected by the attack, rendering the attack action effectively invisible at the output. We use the following parameters: time horizon $T = 49 > n$, and number of experimental trajectories $N = n + mT = 158$, with X_0 and U satisfying assumption in (2.4).

6.7 Conclusions

In this chapter we have shown how experimental data can be used to learn key invariant subspaces of a linear system. In particular, we derived data-driven expressions for $\boldsymbol{\mathcal{V}}^*$, the largest $(A, \mathbf{Im}(B))$-controlled invariant contained in $\mathbf{Ker}(C)$, and $\boldsymbol{\mathcal{S}}^*$, the smallest $(A, \mathbf{Ker}(C))$-conditioned invariant containing $\mathbf{Im}(B)$. Being able to identify these subspaces from data suggests that much of the results and intuitions of the geometric approach to control can be conveniently reworked in a data-driven framework. To support this point, we leveraged the identified invariant subspaces to design a data-driven feedback controller to force the state inside a desired controlled invariant subspace, and to compute the invariant zeros of the system. Finally, as an example of the theoretical results, we designd a data-driven undetectable attack. Applications and extensions of the proposed results are numerous, and are left as the subject of future investigation.

Chapter 7

Distributed data-driven control for network systems

In this chapter we discuss a strategy to directly learn control actions when data from past system trajectories is distributed among multiple agents in a network. The approach we develop provably converges to a suboptimal solution in a finite number of steps, bounded by the diameter of the network, and with a sub-optimality gap that can be characterized as a function of data, and that can be made arbitrarily small.

7.1 Introduction

Networks and multi-agent systems have been studied extensively in the control community in the context of formation control [Pallottino et al., 2007, Oh et al., 2015], consensus algorithms [Ren and Beard, 2005], and coordination of multi-agent systems [Mesbahi and Egerstedt, 2010], to cite a few. These results have seen application in various real-world

scenarios including robotics [Moore et al., 2004, Bullo et al., 2009, Celi et al., 2019], power grids [Molzahn et al., 2017, Swartz et al., 2022] and brain networks [Nozari et al., 2023, Celi et al., 2020, Allibhoy et al., 2022]. Together with its opportunities, modeling the interaction between possibly heterogeneous systems through networks presents practical challenges. In fact, building accurate models of large networks is a burdensome task and modeling errors (e.g., missing or extra links, incorrect link weights) are often unavoidable [Butts, 2003, Amaral, 2008]. Network models can be built either through first principles or via system identification. In the former, a dynamical model is deduced from the physical properties of the system, which are often not fully characterized. Alternatively, system identification can be used when data are available [Ljung, 1987, Katayama, 2005]. Unfortunately, this approach, too, has limitations and has returned a set of mixed results in application to modern network systems [Angulo et al., 2017, Achlioptas et al., 2009, Jung et al., 2010]. This is especially problematic when the end goal of system identification is to provide the basis for controller design since errors in the modeling phase can compound in the controller design phase. In this chapter, we discuss a model-agnostic control approach for a network system. That is, we bypass the system modeling phase altogether by using data to directly learn suitable control actions.

7.2 Related work

The direct data-driven control literature leverages data to design controllers in a one-step solution, as opposed to the indirect approach, where data are used to identify a model that is then used to design a controller. Although direct data-driven controls can

be dated as far back as [Ziegler and Nichols, 1942], they have recently been the subject of renewed interest from the control community. As detailed throughout this thesis, a modern approach to designing direct data-driven controls includes data-driven open-loop optimal control [Baggio et al., 2019, Baggio et al., 2021, Monshizadeh, 2020], closed-loop and robust control [Markovsky and Rapisarda, 2008, De Persis and Tesi, 2020, van Waarde et al., 2020, Berberich et al., 2020a], and predictive and nonlinear control [Coulson et al., 2019, Berberich et al., 2020b, Tabuada et al., 2017, Coulson et al., 2021, van Waarde et al., 2022]. The problem of learning optimal controls in network systems has remained, however, relatively unexplored. To the best of our knowledge, the first result in this direction is [Allibhoy and Cortés, 2020], soon followed by [Celi et al., 2021] and [Alonso et al., 2022]. In [Allibhoy and Cortés, 2020], the authors propose a modified version of the DeePC algorithm [Coulson et al., 2019] to stabilize a network system through a primal-dual flow (while minimizing a quadratic function of the inputs and the states). This strategy is based on a Model Predictive Control approach (MPC) where a behavioral systems representation built upon a recorded system trajectory is used in place of the system's model. A limit of this approach is that it assumes that input-state data from past trajectories, as well as the current state, can be freely shared among neighboring agents. In addition, the distribution of the primal-dual flow among the agents in the network requires a substantial number of messages to be shared [Cherukuri et al., 2017]. This assumption is limiting because it presupposes that the time scales of communication and computation processes are significantly shorter than that of the controller action. An alternative approach appeared in [Alonso et al., 2022], where the same problem is tackled through the system level synthesis

framework (SLS) [Wang et al., 2019]. SLS parameterizes closed-loop system responses directly from collected open-loop trajectories. The result of [Alonso et al., 2022] reduces to a distributed computation of a dynamic state-feedback controller via the alternating direction method of multipliers (ADMM) [Boyd et al., 2011]. This approach shares much of the limitations of [Allibhoy and Cortés, 2020], namely the necessity of high frequency communication between nodes in the network, and the need to share input-state trajectories among the networks' nodes. Further recent results on the control of interconnected systems include [Eising and Cortés, 2022,Jiao et al., 2021,Wang et al., 2023,Li et al., 2022]. However, these works focus on finding stabilizing controllers, and do not include optimization over an objective function, robustness bounds, nor explicit data-driven formulas.

7.3 Setup for multiagent learning

We here study the state evolution of the controllable linear system in (2.1)

$$x(t+1) = Ax(t) + Bu(t), \tag{7.1}$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ are the state and the input vectors at time $t \in \mathbb{N}^+$, respectively, and with $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$. Similarly to Capter 3, we seek to compute the input sequence $\mathbf{u}_T = \mathbf{col}(u(0), \dots, u(T-1))$ that minimizes the following finite-horizon quadratic objective function:

$$\begin{aligned} \arg\min_{\mathbf{u}_T} \quad & \mathbf{x}_T^\top Q \mathbf{x}_T + \mathbf{u}_T^\top R \mathbf{u}_T \\ \text{subject to} \quad & x(t+1) = Ax(t) + Bu(t), \\ & x(0) = x_0, \end{aligned} \tag{7.2}$$

where $Q \in \mathbb{R}^{n(T+1)\times n(T+1)} \succeq 0$, $R \in \mathbb{R}^{mT\times mT} \succ 0$ are the state and input weighting matrices, respectively, and $\mathbf{x}_T = \mathbf{col}(x(0), \dots, x(T))$. However, differently from previous chapters of this thesis, here we assume that data are not centrally available. Instead, past data will be distributed among different actors in a network. We next detail how data is collected and distributed among agents, and propose a strategy to leverage this sparse structure to achieve a (sub)-optimal controller for the above problem.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be the graph associated with the matrix A in (7.1), where $\mathcal{V} = \{1, \dots, n\}$ and $(i, j) \in \mathcal{E}$ if and only if $A_{ji} \neq 0$. Let $\mathcal{V}$ be partitioned as $\mathcal{V} = \mathcal{V}_1 \cup \dots \cup \mathcal{V}_M$, with $|\mathcal{V}_i| = n_i$. Then, after reordering the nodes, the matrix A reads as

$$A = \begin{bmatrix} A_{11} & \dots & A_{1M} \\ \vdots & \vdots & \vdots \\ A_{M1} & \dots & A_{MM} \end{bmatrix}.$$

We assume[1] that the input matrix in (7.1) can be written as

$$B = \mathbf{blkdiag}(B_1, \dots, B_M). \tag{7.3}$$

Then, system (7.1) can equivalently be written as the interconnection of M subsystems of the form

$$x_i(t+1) = A_{ii}x_i(t) + \sum_{j=1,\, j\neq i}^{M} A_{ij}x_j(t) + B_i u_i(t), \tag{7.4}$$

where $x_i \in \mathbb{R}^{n_i}$ and $u_i \in \mathbb{R}^{m_i}$ are the states and inputs, respectively, of the nodes in $\mathcal{V}_i$.

We assume the presence of M agents, each one responsible for one subsystem (see also Fig. 7.1). In particular, agents are interconnected according to a directed communica-

[1] Although some of our results hold also for general input matrices, this assumption simplifies the presentation and is also common in distributed data-driven studies [Allibhoy and Cortés, 2020, Alonso et al., 2022].

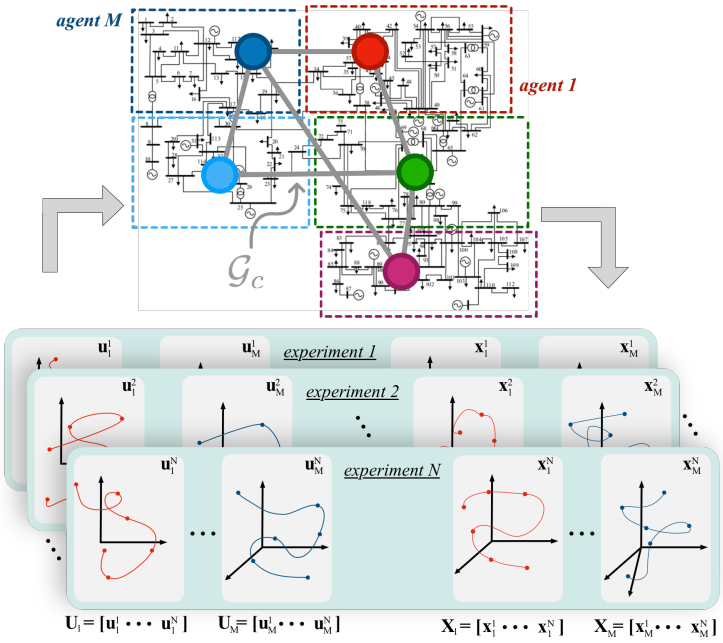

Figure 7.1: In this figure we show a visual depiction of the data collection phase. N experiments of length T are carried out over a network system. Each agent $i \in \{1, \dots, M\}$ collects input-state trajectories of a subset of the network nodes, i.e., the initial state $x_{0,i}^j$, the state trajectory $\mathbf{x}_{T,i}^j$, and the input trajectory $\mathbf{u}_{T,i}^j$, for each experiment j.

tion graph $\mathcal{G}_c = (\mathcal{V}_c, \mathcal{E}_c)$, and agent i selects the input u_i to the ith subsystem using local data and information exchanged with neighboring agents $\mathcal{N}_i = \{j : (i,j) \in \mathcal{E}_c\}$. The local data available to agent i are

$$X_i = \begin{bmatrix} \mathbf{x}_{T,i}^1 & \cdots & \mathbf{x}_{T,i}^N \end{bmatrix}, \tag{7.5a}$$

$$U_i = \begin{bmatrix} \mathbf{u}_{T,i}^1 & \cdots & \mathbf{u}_{T,i}^N \end{bmatrix}, \tag{7.5b}$$

$$X_{0,i} = \begin{bmatrix} x_{0,i}^1 & \cdots & x_{0,i}^N \end{bmatrix}, \tag{7.5c}$$

that is, the components indexed by $\mathcal{V}_i$ of the state trajectories, control inputs, and initial

conditions of the N experiments. Thus, after a possible reordering of the system states,

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_M \end{bmatrix}, \; U = \begin{bmatrix} U_1 \\ \vdots \\ U_M \end{bmatrix}, \text{ and } X_0 = \begin{bmatrix} X_{0,1} \\ \vdots \\ X_{0,M} \end{bmatrix}.$$

Agents cooperate to collectively compute the local inputs that solve the minimization problem (7.2). A naive solution to this problem requires the agents to share all their local experimental data with all other agents, and then employ centralized data-driven formulas for the solution of the minimization problem (7.2) (see [De Persis and Tesi, 2020, Coulson et al., 2019, Celi et al., 2022] and Theorem 47 below). Instead, we will develop algorithms that require the agents to exchange reduced information to compute, in finite time, a solution to the minimization problem (7.2) in a distributed manner.

7.4 Multiagent learning of optiamal controls

We start by providing a data-driven solution to the minimization problem (7.2) when there is only one agent. This solution is general, and appeared previously in [Celi et al., 2022]. In particular, the optimal control $\mathbf{u}_T^*$ minimizing (7.2) can be computed in closed form as

$$\mathbf{u}_T^* = -UK_0S^\dagger(XK_0)^\top QXK_U(X_0K_U)^\dagger x_0, \tag{7.6}$$

with

$$S = (XK_0)^\top Q(XK_0) + (UK_0)^\top R(UK_0),$$

and $K_0 = \mathbf{Basis}(\mathbf{Ker}(X_0))$ and $K_U = \mathbf{Basis}(\mathbf{Ker}(U))$, highlighting the dependency of the optimal controls on the data. We now give an alternative formula to compute $\mathbf{u}_T^*$ in closed

form. This formula requires a stronger assumption on Q, but is more compact and holds in most scenarios.

Theorem 47 (Single-agent data-driven solution (7.2)**)** *Let X_0, X and U be as in* (2.2) *and satisfying assumption in* (2.4)*. Define $P \in \mathbb{R}^{(n+mT)\times(n+mT)}$ as*

$$P = \left(X \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\dagger} \right)^{\mathsf{T}} Q \left(X \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\dagger} \right) + \begin{bmatrix} 0_{n\times n} & 0 \\ 0 & R \end{bmatrix}. \tag{7.7}$$

If $Q^{1/2}O_T$ has full column rank the matrix P is invertible, then the solution $\mathbf{u}_T^*$ to (7.2) *can be extracted from*

$$\begin{bmatrix} x_0 \\ \mathbf{u}_T^* \end{bmatrix} = P^{-\frac{1}{2}} \left(\begin{bmatrix} I & 0 \end{bmatrix} P^{-\frac{1}{2}} \right)^{\dagger} x_0, \tag{7.8}$$

with $x(0) = x_0$, □

Proof. Consider the following problem

$$\begin{aligned} \arg\min_{\gamma} \quad & \left\| P^{\frac{1}{2}} \gamma \right\|^2 \\ \text{subject to} \quad & \begin{bmatrix} I & 0 \end{bmatrix} \gamma = x_0. \end{aligned} \tag{7.9}$$

where P is as in (7.7). Note that, the solution to the above problem is $\gamma^* = \mathbf{col}(x_0, \mathbf{u}_T^*)$ with $\mathbf{u}_T^*$ being the solution to (7.2). This follows from (2.6) and the fact that

$$\begin{aligned} P &= \left(X \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\dagger} \right)^{\mathsf{T}} Q \left(X \begin{bmatrix} X_0 \\ U \end{bmatrix}^{\dagger} \right) + \begin{bmatrix} 0 & 0 \\ 0 & R \end{bmatrix} \\ &= \begin{bmatrix} O_T & F_T \end{bmatrix}^{\mathsf{T}} Q \begin{bmatrix} O_T & F_T \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & R \end{bmatrix} \end{aligned} \tag{7.10}$$

when condition (2.4) holds. If P is invertible, letting $v = P^{1/2}\gamma$ problem (7.9) becomes

$$\begin{aligned} \arg\min_{v} \quad & \|v\|^2 \\ \text{subject to} \quad & \begin{bmatrix} I & 0 \end{bmatrix} P^{-1/2} v = x_0. \end{aligned} \tag{7.11}$$

whose solution is $v_T^* = (\begin{bmatrix} I & 0 \end{bmatrix} P^{-1/2})^\dagger x_0$. Thus, the solution to (7.9) is

$$\gamma^* = \begin{bmatrix} x_0 \\ \mathbf{u}_T^* \end{bmatrix} = P^{-1/2} v_T^* = P^{-1/2} (\begin{bmatrix} I & 0 \end{bmatrix} P^{-1/2})^\dagger x_0. \tag{7.12}$$

We show that if $Q^{1/2}O_T$ has full column rank, then $P \succ 0$, which in turn implies that P is invertible. To this end, notice that, from (7.10),

$$P = \underbrace{\begin{bmatrix} O_T^\mathsf{T} Q O_T & O_T^\mathsf{T} Q F_T \\ F_T^\mathsf{T} Q O_T & F_T^\mathsf{T} Q F_T \end{bmatrix}}_{P_1} + \begin{bmatrix} 0 & 0 \\ 0 & R \end{bmatrix}.$$

Since $O_T^\mathsf{T} Q O_T \succ 0$ (which directly follows from the assumption that $Q^{1/2}O_T$ has full column rank) and $P_1 \succeq 0$, the Schur complement of the block $O_T^\mathsf{T} Q O_T$ of P_1 must be positive semidefinite [Zhang, 2005, Theorem 1.12], namely $S = F_T^\mathsf{T} Q F_T - F_T^\mathsf{T} Q O_T^\mathsf{T} (O_T^\mathsf{T} Q O_T)^{-1} O_T^\mathsf{T} Q F_T \succeq 0$. Next, since $S \succeq 0$ and $R \succ 0$, the Schur complement of the block $O_T^\mathsf{T} Q O_T$ of P, namely $S + R$, is positive definite. Since the block $O_T^\mathsf{T} Q O_T$ of P and its Schur complement are both positive definite, follows that $P \succ 0$ [Zhang, 2005, Theorem 1.12].

We complement the proof by noticing that, for P not invertible, and for any $Q \succ 0$, problem (7.9) becomes

$$\begin{aligned} \arg\min_{v_T,\, w} \quad & \|v_T\|^2 \\ \text{subject to} \quad & \begin{bmatrix} I & 0 \end{bmatrix} ((P^{1/2})^\dagger v_T + K_P w) = x_0, \end{aligned} \tag{7.13}$$

where w is a vector of appropriate size, from which a solution alternative to (7.12) can be found. ■

Theorem 47 provides a data-driven expression of the optimal control input for the minimization problem (7.2) alternative to (7.6). The assumption that $Q^{1/2}O_T$ has full column rank is required to ensure the invertibility of P and is typically satisfied in practice. For instance, it is satisfied when $Q \succ 0$, or, in the case of block diagonal Q with diagonal blocks $Q_d \in \mathbb{R}^{n\times n}$ (standard LQR setup), when the pair $(A, Q_d^{1/2})$ is observable. This assumption on $Q^{1/2}O_T$ is made in order to keep expression (7.6) more compact, which helps simplifying notation in the rest of this chapter. Both expressions (7.8) and (7.6) require the knowledge of all the experimental data, as well as the knowledge of the cost function, which is undesirable for interconnected and possibly large systems. While a distributed data-driven solution to the minimization problem (7.2) could be obtained using standard techniques for distributed optimization, e.g., see [Allibhoy and Cortés, 2020, Alonso et al., 2022], we follow a different approach that will lead to an algorithm with finite-time convergence guarantees. To this aim, we define the following auxiliary problems:

$$\begin{aligned} \underset{\alpha,\,\beta}{\arg\min} \quad & \left\| \begin{bmatrix} Q^{\frac{1}{2}}X & Q^{\frac{1}{2}}X \\ 0 & R^{\frac{1}{2}}U \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \right\|^2 \\ \text{subject to} \quad & \begin{bmatrix} X_0 & 0 \\ U & 0 \\ 0 & X_0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} x_0 \\ 0 \\ 0 \end{bmatrix} \end{aligned} \tag{7.14}$$

and

$$\begin{aligned} \underset{\alpha,\beta,v,w}{\arg\min} \quad & \left\| \begin{bmatrix} \alpha \\ \beta \\ v \\ w \end{bmatrix} \right\|^2 \\ \text{subject to} \quad & \begin{bmatrix} X_0 & 0 & 0 & 0 \\ U & 0 & 0 & 0 \\ 0 & X_0 & 0 & 0 \\ Q^{\frac{1}{2}}X & Q^{\frac{1}{2}}X & \varepsilon I_X & 0 \\ 0 & R^{\frac{1}{2}}U & 0 & \varepsilon I_U \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ v \\ w \end{bmatrix} = \begin{bmatrix} x_0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \end{aligned} \tag{7.15}$$

where $\varepsilon \in \mathbb{R}^+$ is a tunable parameter, with $I_X = I_{n(T+1)}$, $I_U = I_{mT}$ and therefore $\alpha \in \mathbb{R}^N$, $\beta \in \mathbb{R}^N$, $v \in \mathbb{R}^{n(T+1)}$, $w \in \mathbb{R}^{mT}$. We now characterize the feasibility and optimality properties of (7.14) and (7.15).

Lemma 48 (Relationship between the solution of (7.2) **and** (7.14)**)** *If α^* and β^* are minimizers of problem* (7.14)*, then $\mathbf{u}_T^* = U\beta^*$ is the minimizer of problem* (7.2)*.* □

Proof. From Lemma 2 we can write

$$\mathbf{x}_T = \begin{bmatrix} XK_U & XK_0 \end{bmatrix} \begin{bmatrix} \bar{\alpha} \\ \bar{\beta} \end{bmatrix},$$

with $x_0 = X_0 K_U \bar{\alpha}$ and $\mathbf{u}_T = UK_0\bar{\beta}$. Equivalently, we can write

$$\mathbf{x}_T = \begin{bmatrix} X & X \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

where $\alpha = K_U\bar{\alpha}$ (i.e., $U\alpha = 0$), and $\beta = K_0\bar{\beta}$ (i.e., $X_0\beta = 0$), and consequently $x_0 = X_0\alpha$, $\mathbf{u}_T = U\beta$ and the cost function of (7.14) equals that of (7.2). When $\mathbf{u}_T^*$ is the solution to (7.2) and α^*, β^* is the solution to (7.14), it is then straightforward to see that $\mathbf{u}_T^* = U\beta^*$.

We can further derive an explicit form of the optimal vectors α^* and β^* instrumental to other results in this work. By substituting the constraints of (7.14) into the cost function, and for H, Y, and $\bar{x}_0$ as in (7.23), problem (7.14) can be written as

$$\arg\min_{w} \qquad \left\| H\left(Y^{\dagger}\bar{x}_0 + K_Y w\right)\right\|^2, \tag{7.16}$$

where $K_Y = \mathbf{Basis}(\mathbf{Ker}(Y))$. The minimizers to (7.16) define the set of optimal vectors α^*, β^* via

$$\begin{aligned}\begin{bmatrix}\alpha^* \\ \beta^*\end{bmatrix} &= Y^{\dagger}\bar{x}_0 + K_Y w^* \\ &= \left(I - K_Y(HK_Y)^{\dagger}H\right)Y^{\dagger}\bar{x}_0 + r \end{aligned} \tag{7.17}$$

where $r \in \mathbf{Ker}(H) \subseteq \mathbf{Ker}(U)$. The resulting, unique, optimal control $\mathbf{u}_T = U\beta^*$ is the solution to (7.2). ■

Lemma 49 (Relationship between the solution of (7.2) and (7.15)) *The minimization problem (7.15) is feasible and admits a unique solution when $\varepsilon > 0$. Furthermore, if $\alpha^*(\varepsilon)$, $\beta^*(\varepsilon)$, $v^*(\varepsilon)$ and $w^*(\varepsilon)$ are minimizers of problem (7.15), then*

$$\lim_{\varepsilon \to 0^+} U\beta^*(\varepsilon) = \mathbf{u}_T^*, \tag{7.18}$$

where $\varepsilon > 0$, and $\mathbf{u}_T^*$ is the minimizer of problem (7.2). □

Proof. From the constraints in (7.15) it holds, for all $\varepsilon > 0$,

$$v = -\frac{1}{\varepsilon}Q^{\frac{1}{2}}X(\alpha + \beta), \quad w = -\frac{1}{\varepsilon}R^{\frac{1}{2}}U\beta.$$

Substituting these equations in the cost function, and for H, Y and $\bar{x}_0$ as in (7.23), problem (7.15) can be rewritten as

$$\gamma^*(\varepsilon) = \begin{bmatrix} \alpha^*(\varepsilon) \\ \beta^*(\varepsilon) \end{bmatrix} = \arg\min_{\alpha,\beta} \quad \varepsilon^2 \left\| \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \right\|^2 + \left\| H \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \right\|^2 \\ \text{s.t.} \quad Y \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \bar{x}_0. \tag{7.19}$$

If $\gamma^*(\varepsilon)$ is bounded as $\varepsilon \to 0^+$, then $\gamma^*(\varepsilon)$ converges to the (minimum norm) solution to problem (7.14) as $\varepsilon \to 0^+$, and (7.18) holds. Thus, it remains to prove the boundedness of $\gamma^*(\varepsilon)$ as $\varepsilon \to 0^+$. To this end, we note that by writing $\gamma^*(\varepsilon) = \gamma_1(\varepsilon) + \gamma_2(\varepsilon)$ with

$$\gamma_1(\varepsilon) \perp \mathbf{Ker}H \cap \mathbf{Ker}Y \text{ and } \gamma_2(\varepsilon) \in \mathbf{Ker}H \cap \mathbf{Ker}Y,$$

the cost of (7.19) evaluated at $\gamma^*(\varepsilon)$ reads as

$$C(\gamma^*(\varepsilon)) = \varepsilon^2 \left\|\gamma_1(\varepsilon)\right\|^2 + \varepsilon^2 \left\|\gamma_2(\varepsilon)\right\|^2 + \left\|H\gamma_1(\varepsilon)\right\|^2, \tag{7.20}$$

since $\gamma_2(\varepsilon) \in \mathbf{Ker}H$. Further, the vector $\gamma_1(\varepsilon)$ satisfies $Y\gamma_1(\varepsilon) = Y\gamma^*(\varepsilon) = \bar{x}_0$ since $\gamma_2(\varepsilon) \in \mathbf{Ker}Y$. From the latter fact and (7.20), it follows that it must be $\gamma_2(\varepsilon) = 0$, $\forall \varepsilon > 0$, for $\gamma^*(\varepsilon)$ to be optimal. Further, from (7.20) and the fact that there always exists a γ (e.g., $\gamma = Y^\dagger \bar{x}_0$) which is independent of ε and satisfies the constraint in (7.19) (thus yielding a cost which is bounded $\forall \varepsilon > 0$), it follows that $\gamma_1(\varepsilon) = \gamma^*(\varepsilon)$ must be bounded as $\varepsilon \to 0^+$.

■

Since the constraints in the minimization problem (7.15) can be partitioned row-wise in a way that each row depends only on the data available to a single agent (see below),

Algorithm 1: Distributed data-driven optimal control

Input: $x_{0,i}$, W_i, d

$\gamma_i = W_i^{\dagger}\mathbf{col}(x_{0,i}, 0)$

$K_i = \mathbf{Basis}(\mathbf{Ker}(W_i))$

Until convergence

  for $j \in \mathcal{N}_i$ **do**

    Receive: γ_j, K_j

    $\gamma_i^+ = \gamma_i + \begin{bmatrix} K_i & 0 \end{bmatrix} \begin{bmatrix} -K_i & K_j \end{bmatrix}^{\dagger} (\gamma_i - \gamma_j)$

    $K_i = \mathbf{Basis}(\mathbf{Im}(K_i) \cap \mathbf{Im}(K_j))$

    $\gamma_i = \gamma_i^+$

  end

  Transmit: γ_i, K_i

end

Return: $\beta_i = [\gamma_i]_{N+1:2N}$

a distributed algorithm can be readily obtained. Further, given the equivalence between the minimization problems (7.2), (7.14), and (7.15) as stated in Lemma 48 and Lemma 49, a distributed solution to (7.2) can be obtained by solving (7.15) in a distributed manner. Our distributed algorithm for the agents to solve the minimization problem (7.2) via distributed computation is in Algorithm 1, where W_i is defined in (7.21) and d denotes the diameter of $\mathcal{G}_c$.[2] We now provide an informal description of the algorithm:

(S1) Initially, each agent i computes the minimum norm solution to

$$W_i\gamma_i = \mathbf{col}(x_{0,i}, 0_{mT}, 0_n, 0_{n(T+1)}, 0_{mT}),$$

[2]The diameter of a graph $\mathcal{G}$ is the maximum distance between any two nodes of $\mathcal{G}$.

where $W_i \in \mathbb{R}^{(n_i+m_iT+n_i+n_i(T+1)+m_iT)\times(N+N+n(T+1)+mT)}$

$$W_i = \begin{bmatrix} X_{0,i} & 0 & 0 & 0 \\ U_i & 0 & 0 & 0 \\ 0 & X_{0,i} & 0 & 0 \\ Q_i^{\frac{1}{2}} X_i & Q_i^{\frac{1}{2}} X_i & \varepsilon I_X^i & 0 \\ 0 & R_i^{\frac{1}{2}} U_i & 0 & \varepsilon I_U^i \end{bmatrix}, \tag{7.21}$$

and $K_i = \mathbf{Basis}(\mathbf{Ker}(W_i))$. Here, I_X^i (resp. I_U^i) is a matrix whose rows are the rows of I_{nT} (resp. I_{mT}) corresponding to the indices that extract $Q_i^{\frac{1}{2}} X_i$ (resp. $R_i^{\frac{1}{2}} U_i$) from $Q^{\frac{1}{2}} X$ (resp. $R^{\frac{1}{2}} U$). From the notation in (7.15), let $\gamma_i = \mathbf{col}(\alpha_i, \beta_i, v_i, w_i) \in \mathbb{R}^{N+N+n(T+1)+mT}$ be such solution.

(S2) At each iteration, each agent i transmits γ_i and K_i to its neighboring agents j, and receives γ_j and K_j from each neighbor j.

(S3) At each iteration, each agent i updates γ_i and K_i as

$$\gamma_i^+ = \gamma_i + \begin{bmatrix} K_i & 0 \end{bmatrix} \begin{bmatrix} -K_i & K_j \end{bmatrix}^\dagger (\gamma_i - \gamma_j),$$

$$K_i = \mathbf{Basis}(\mathbf{Im}(K_i) \cap \mathbf{Im}(K_j)).$$

(S4) Convergence of this iterative procedure is guaranteed after a number of steps equal to the diameter of communication graph (see below). Upon convergence, each agent returns the vector $\beta_i = [\gamma_i]_{N+1:2N}$, extracted from γ_i at the algorithm's final iteration.

A high level walkthrough of the algorithm is in order. Step (S1) is simply the initialization step, in which we compute a preliminary solution $\gamma_i = W_i^\dagger \mathbf{col}(x_{0,i}, 0)$ which uses only local data and is feasible for agent i. In step (S2) neighboring agents share the information

that is needed to update the provisory solution γ_i. In step (S3) γ_i is updated to γ_i^+ with information from its neighbors in such a way that γ_i^+ is a feasible solution for agent i and its neighbors. More specifically we find the new γ_i^+ to satisfy

$$\gamma_i^+ = W_i^\dagger \mathbf{col}(x_{0,i}, 0) + K_i \kappa_i \tag{7.22a}$$

$$= W_j^\dagger \mathbf{col}(x_{0,j}, 0) + K_j \kappa_j, \tag{7.22b}$$

for all $j \in \mathcal{N}$, where κ_i and κ_j are vectors of appropriate dimension. In the proof we show how κ_i and κ_j such that γ_i^+ satisfies (7.22) always exist, and show how this can be computed through the procedure described in (S3). Finally, (S4) gives one condition for ending the algorithm once it converges, which is also detailed in the proof.

Theorem 50 (Distributed learning of data-driven optimal controls) *Let $\mathcal{G}_c$ be a strongly connected communication graph. Let $\beta_i(\varepsilon)$ be the value returned by Algorithm 1 when W_i is as in* (7.21)*, and let $\alpha^*(\varepsilon)$, $\beta^*(\varepsilon)$, $v^*(\varepsilon)$, $w^*(\varepsilon)$ be minimizers of problem* (7.15)*, for some $\varepsilon > 0$. Then, for all $i \in \{1, \ldots, M\}$, $\beta_i(\varepsilon) = \beta^*(\varepsilon)$.* □

Proof. The proof follows an argument similar to the one of [Pasqualetti et al., 2012, Theorem 3.3]. Let γ_i be the estimate of agent i, W_i be defined as in (7.21), and $K_i = \mathbf{Basis}(\mathbf{Ker}(W_i))$. Observe that $\gamma_i = W_i^\dagger \mathbf{col}(x_{i,0}, 0, 0, 0, 0) \perp \mathbf{Ker}(W_i)$. Let i and j be two neighboring agents, i.e, $(i, j) \in \mathcal{E}_c$, then there exist two vectors κ_i and κ_j such that $\gamma_i + K_i \kappa_i = \gamma_j + K_j \kappa_j$. In particular, such vectors can be chosen as

$$\begin{bmatrix} \kappa_i \\ \\ \kappa_j \end{bmatrix} = \begin{bmatrix} -K_i & K_j \end{bmatrix}^\dagger (\gamma_i - \gamma_j).$$

Substituting κ_i back in γ_i we have that the vector

$$\gamma_i^+ = \gamma_i + \begin{bmatrix} K_i & 0 \end{bmatrix} \begin{bmatrix} -K_i & K_j \end{bmatrix}^\dagger (\gamma_i - \gamma_j)$$

is such that $\mathbf{col}(x_{0,i}, 0, 0, 0, 0) = W_i \gamma_i^+$ and $\mathbf{col}(x_{0,j}, 0, 0, 0, 0) = W_j \gamma_i^+$. Moreover, we have that $\gamma_i^+ \perp (\mathbf{Im}(K_i) \cap \mathbf{Im}(K_j))$, since

$$\begin{bmatrix} \kappa_i \\ \\ \kappa_j \end{bmatrix} \perp \mathbf{Ker}(\begin{bmatrix} -K_i & K_j \end{bmatrix}).$$

We notice that $K_i \kappa_i \perp \mathbf{Im}(K_j)$; by contradiction, if $K_i \kappa_i \not\perp \mathbf{Im}(K_j)$, then one can find $\kappa_i = \tilde{\kappa}_i + \bar{\kappa}_i$, where $K_i \tilde{\kappa}_i \perp \mathbf{Im}(K_j)$ and $K_i \bar{\kappa}_i \in \mathbf{Im}(K_j)$. Let $\bar{\kappa}_j = K_j^\dagger K_i \bar{\kappa}_i$ and $\tilde{\kappa}_j = \kappa_j - \bar{\kappa}_j$. Then $\mathbf{col}(\bar{\kappa}_i, \bar{\kappa}_j) \in \mathbf{Ker}(\begin{bmatrix} -K_i & K_j \end{bmatrix})$ and hence $\mathbf{col}(\bar{\kappa}_i, \bar{\kappa}_j) \not\perp \mathbf{Ker}(\begin{bmatrix} -K_i & K_j \end{bmatrix})$, which contradicts the hypothesis. We conclude that $\begin{bmatrix} K_i & 0 \end{bmatrix} \begin{bmatrix} -K_i & K_j \end{bmatrix}^\dagger (\gamma_i - \gamma_j) \perp \mathbf{Im}(K_j)$, and, since $\gamma_i \perp \mathbf{Im}(K_i)$, we can conclude that $\gamma_i^+ \perp (\mathbf{Im}(K_i) \cap \mathbf{Im}(K_j))$. The theorem follows by noticing that after a number of steps equal to the diameter of $\mathcal{G}_c$, each vector γ_i verifies all the measurements, since we will have that $\gamma_i \perp \bigcap_j^M \mathbf{Im}(K_j)$. Finally, we remark that the solution we are interested in involves only the second N elements of γ_i, corresponding to β_i. ■

From Theorem 50, Algorithm 1 returns in a finite number of steps the solution of the minimization problem (7.15). Due to Lemma 49, such solution yields the minimizer of (7.2) as the parameter ε decreases to zero. In fact, for any finite value of ε, the sub-optimality gap between the minimizer of (7.2) and the input $U\beta^*(\varepsilon)$ reconstructed from the

minimizer of (7.15) can also be quantified. Let

$$Y = \begin{bmatrix} X_0 & 0 \\ U & 0 \\ 0 & X_0 \end{bmatrix}, \quad H = \begin{bmatrix} Q^{\frac{1}{2}}X & Q^{\frac{1}{2}}X \\ 0 & R^{\frac{1}{2}}U \end{bmatrix}, \quad \bar{x}_0 = \begin{bmatrix} x_0 \\ 0 \\ 0 \end{bmatrix}. \tag{7.23}$$

Lemma 51 (Optimality gap of (7.18) for finite ε) *Let $\beta^*(\varepsilon)$ be the minimizer of Problem (7.15), and let $\delta(\varepsilon) = \|\mathbf{u}_T^* - U\beta^*(\varepsilon)\|$. Then,*

$$\delta(\varepsilon) \leq \left\| U \begin{bmatrix} 0_{N,N} & I_N \end{bmatrix} K_Y Z(\varepsilon) K_Y^\top H^\top H Y^\dagger x_0 \right\|, \tag{7.24}$$

where $Z(\varepsilon) = (K_Y^\top(\varepsilon^2 I + H^\top H)K_Y)^\dagger - (K_Y^\top H^\top H K_Y)^\dagger$, and $K_Y = \mathbf{Basis}(\mathbf{Ker} Y)$. □

Proof. For H, Y and $\bar{x}_0$ as in (7.23), problem (7.15) can be rewritten as

$$\begin{aligned} \begin{bmatrix} \alpha^*(\varepsilon) \\ \beta^*(\varepsilon) \end{bmatrix} = \arg\min_{\alpha,\beta} \quad & \left\| \begin{bmatrix} \varepsilon I \\ H \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \right\|^2 \\ \text{s.t.} \quad & Y \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \bar{x}_0. \end{aligned} \tag{7.25}$$

The solution to (7.25) is $Y^\dagger \bar{x}_0 - K_Y w^*(\varepsilon)$ with

$$\begin{aligned} w^*(\varepsilon) &= \arg\min_w \quad \left\| \begin{bmatrix} \varepsilon I \\ H \end{bmatrix} \left(Y^\dagger \bar{x}_0 - K_Y w \right) \right\|^2 \\ &= \left(\begin{bmatrix} \varepsilon I \\ H \end{bmatrix} K_Y \right)^\dagger \begin{bmatrix} \varepsilon I \\ H \end{bmatrix} Y^\dagger \bar{x}_0 \\ &= \left(K_Y^\top(\varepsilon^2 I + H^\top H) K_Y \right)^\dagger K_Y^\top (\varepsilon^2 I + H^\top H) Y^\dagger \bar{x}_0 \\ &= \left(K_Y^\top(\varepsilon^2 I + H^\top H) K_Y \right)^\dagger K_Y^\top H^\top H Y^\dagger \bar{x}_0, \end{aligned} \tag{7.26}$$

where in the last-but-one step we used that $A^\dagger = (A^\top A)^\dagger A^\top$ for any matrix A, and in the last step $K_Y^\top Y^\dagger = 0$. Similarly, we can rewrite any minimizer to Problem (7.14) as (cf. (7.17))

$$\begin{bmatrix} \alpha^* \\ \beta^* \end{bmatrix} = \left(I - K_Y \left(H K_Y \right)^\dagger H \right) Y^\dagger \bar{x}_0 + r$$

$$= \left(I - K_Y \left(K_Y^\top (H^\top H) K_Y \right)^\dagger K_Y^\top H^\top H \right) Y^\dagger \bar{x}_0 + r, \tag{7.27}$$

where $r \in \mathbf{Ker}(H) \subseteq \mathbf{Ker}(U)$ and $r \in \mathbf{Ker}(H) \subseteq \mathbf{Ker}(X_0)$. From (7.26) and (7.27) and from the fact that $\|\mathbf{col}(\alpha, \beta)\| \geq \|\beta\|$ for any α and β, it follows that $\|U(\beta^* - \beta^*(\varepsilon))\| = \|\mathbf{u}_T^* - U\beta^*(\varepsilon)\|$ is bounded as in (7.24). ■

Using Lemma 51, Algorithm 1 can be used to compute a sub-optimal solution to (7.2) in a finite number of distributed calculations and within any desired sub-optimality guarantee. In fact, once β_i is computed by each agent i, the sub-optimal and local control at each agent is simply $\mathbf{u}_{T,i} = U_i \beta_i$.

Remark 52 (Convergence of Algorithm 1) *Algorithm 1 converges after d iterations, where d is the diameter of $\mathcal{G}_c$. When d is not available the algorithm can be stopped after n iterations, since n is always available to each agent through the size of the vector v_i, and $d \leq n$ always holds.*

7.5 Numerical results and illustrative examples

We now provide numerical validations of the results presented in this chapter. First, we show how Algorithm 1 and Theorem 50 can be used to solve the problem in (7.2) when the system (7.1) is unknown but data (7.5) are available. We further discuss how

our formulas can be used directly, for a fixed-window control, in a receding horizon fashion. Finally, we compare and discuss our approach to alternative approaches in the literature.

7.5.1 An application to scaling ring networks

We assess the results of this chapter by analyzing the convergence of Algorithm 1 for networks of varying diameter. We consider randomly generated ring networks, with communication graph $\mathcal{G}_c$ as in Fig. 7.2a, here shown for a ring network with $M = 6$ and $d = 3$. In Fig. 7.2b we plot the error between the solution of (7.2) and the result of Algorithm 1. For the kth iteration of Algorithm 1, subnetwork i computes its own control inputs as $u_i[k] = U_i\beta_i[k]$, where $\beta_i[k]$ is the interim value of β_i at iteration k. The dynamics of (7.1) when $u_i[k]$ is injected to all $i \in \{1, \ldots, M\}$ is compared to the one induced by the model-based optimal control (7.8). As discussed in Theorem 50, Algorithm 1 converges to the solution in a number of steps equal to the diameter of the communication graph $\mathcal{G}_c$.

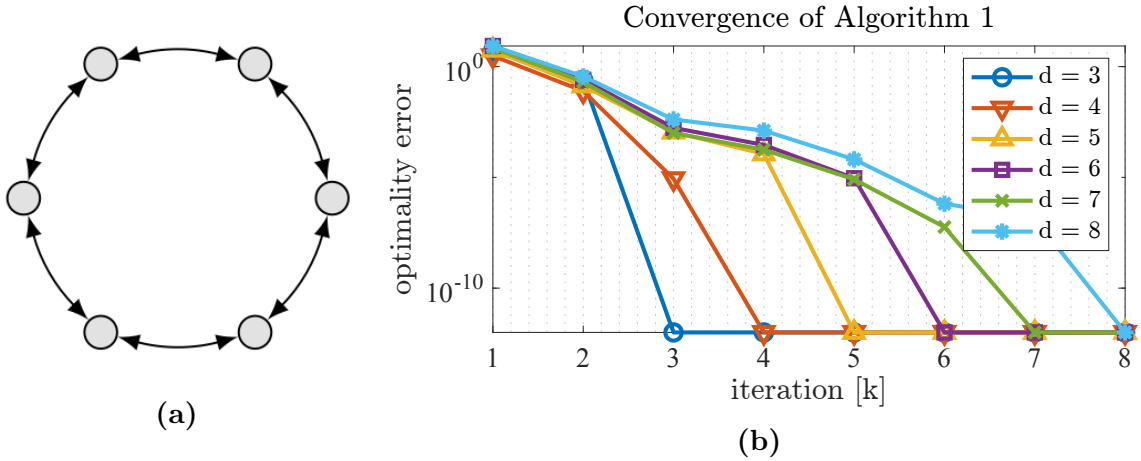

Figure 7.2: This figure shows the results associated with the experiment of Section 7.5.1. In panel (a) we show the communication graph of an example network, with $M = 6$ and $d = 3$. All other networks in this example are similar in structure (ring network), with varying diameters $d \in \{3, \ldots, 8\}$. All experiments shown in panel (b) are performed with $T = 5$, $\varepsilon = 10^{-3}$ and the tolerance for the pseudoinverse operation is set to $\texttt{tol} = 10^{-8}$.

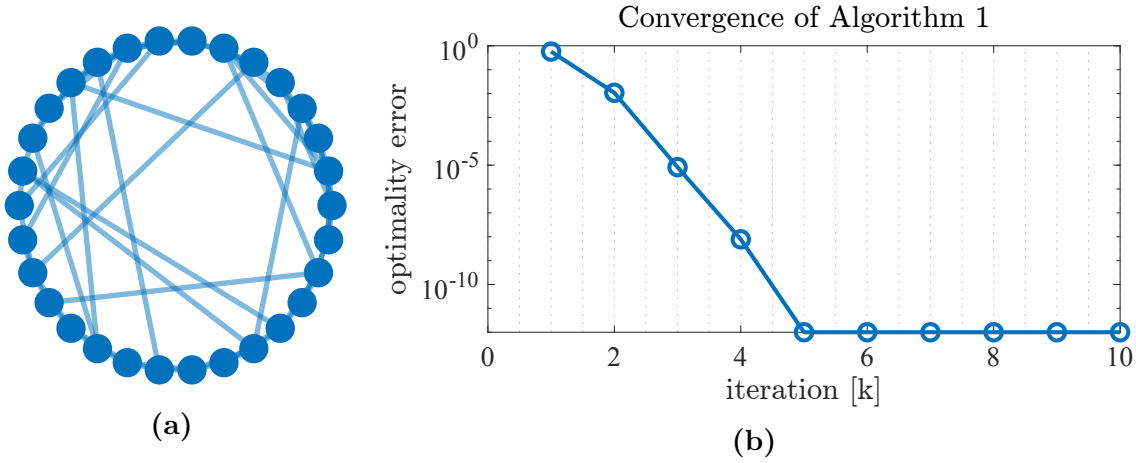

Figure 7.3: This figure shows the results associated with the experiment of Section 7.5.2. In panel (a) we show a randomly generated Watts-Strogats network with $M = 30$ agents. In panel (b) we show the convergence of Algorithm 1, comparing it to the model-based centralized solution of (7.2). Additional simulation parameters are $T = 5$, $\varepsilon = 10^{-3}$ and the tolerance for the pseudoinverse operation is set to $\texttt{tol} = 10^{-8}$.

7.5.2 An application to Watts-Strogatz networks

To prove the effectiveness of this approach in more complex network structures, we test Algorithm 1 on a Watts-Strogats network of larger size [Watts and Strogatz, 1998]. In Fig. 7.3 we run Algorithm 1 over a randomly generated Watts-Strogats network with $M = 30$, mean node degree 4 and rewiring probability of 0.15, ref. the `WattsStrogatz` function in MATLAB [MathWorks, 2023]. The particular realization of Fig. 7.3a is a network with diameter 5, as testified by the convergence of Algorithm 1, see Fig. 7.3b.

7.5.3 Receding horizon implementation

Theorem 47 assumes that T is the control horizon of problem (7.2) as well as of the dataset (7.5). It is possible to lift this requirement by implementing Algorithm 1 in a receding horizon fashion. That is, once Algorithm 1 is executed, each agent applies only

Algorithm 2: Receding horizon Algorithm 1

Input: T_{sim}, $\{X_1, \ldots, X_M\}$, $\{U_1, \ldots, U_M\}$, $\{X_{F,1}, \ldots, X_{F,M}\}$,

$\{X_{0,1}, \ldots, X_{0,M}\}$, ε, Q, R, $\{x_{0,1}, \ldots, x_{0,M}\}$, h

set $t = 0$

for T_{sim} *times* **do**

run Algorithm 1, where each agent computes $\mathbf{u}_{T,i} = U_i\beta_i$

let each agent apply h steps of the computed control, i.e.,

$u_i(t) = \mathbf{u}_{T,i}(1 : m_i)$

set t to $t + 1$ and $x_{0,i}$ to the updated state $x_i(t + 1)$, for all $i = \{1, \ldots, M\}$

end

a finite horizon $h \leq T$ of the computed controller $\mathbf{u}_{T,i} = U_i\beta_i$. After h time steps, then, Algorithm 1 is executed again, and a new controller is found for the subsequent horizon h. This approach is formally described in Algorithm 2. Clearly, there is no limit to the number of times that this algorithm can be run, i.e., we can use this approach to design an arbitrarily long controller. A receding horizon implementation is common throughout the literature, often termed Model Predictive Control (MPC), and is also implemented in related works [Allibhoy and Cortés, 2020, Alonso et al., 2022]. Comparison of these approaches with our method are discussed next.

7.5.4 Comparison with splitting methods

Several strategies have been proposed to solve MPC problems in a distributed fashion. Among these, splitting methods [Cherukuri et al., 2017, Clason and Valkonen,

2020] have been explored in a model-based [Farokhi et al., 2014] and data-driven [Allibhoy and Cortés, 2020] setting. These consist in *splitting* one optimization problem in a family of smaller optimization problems. In the context of the problem setup (7.2), each agent needs to solve a local optimization problem, while iteratively exchanging information with other agents in order to properly converge to the global optimal solution. The accuracy of the approach is closely related to the number of iterations of the algorithm, a higher number of them leading to a more accurate solution. A known downside of these approaches is their slow convergence; typically, a large number of iterations is needed to converge to an acceptable solution (measured as its distance from the exact optimal solution). Recently [Allibhoy and Cortés, 2020] proposed splitting (7.2) trough a primal-dual flow. In Fig. 7.4 we compare the convergence properties of Algorithm 2 with a primal-dual algorithm based on [Allibhoy and Cortés, 2020]. In particular, we keep the problem formulation and data-collecting phase of [Allibhoy and Cortés, 2020], while modifying the cost function through the Augmented Lagrangian method, which is known to improve the convergence speed of the flow. As a standard practice, we distribute the Augmented Lagrangian through the Alternating Directions Method of Multipliers (ADMM), see [Clason and Valkonen, 2020, Chapter 8]. As highlighted in Fig. 7.4, Algorithm 2 returns a stabilizing controller even for higher noise values.

7.5.5 Comparison with [Alonso et al., 2022] with noisy data

We now compare the performance of our approach to [Alonso et al., 2022], when data are collected with noise. We consider the same system used in [Alonso et al., 2022,

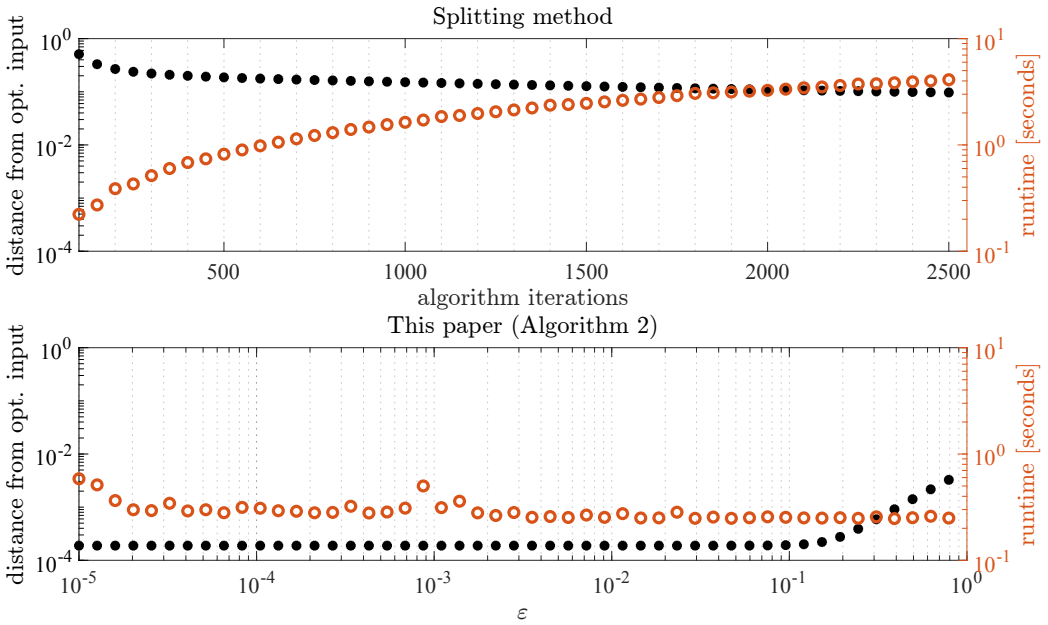

Figure 7.4: This figure shows a comparison between the approach in Algorithm 2 and a distributed data-driven solver based on [Allibhoy and Cortés, 2020] (splitting method). The experiments are performed with noiseless data from the same system, a randomly generated network with $M = 3$ agents, and diverse state size. This comparison highlights the *computational time to accuracy tradeoff* that the approaches have with respect to a particular design parameter, namely the number of iterations of the splitting method, and the parameter ε for Algorithm 2. The runtime is reported in seconds (lower is better), and the distance from the optimal input is computed as the norm of the difference of the solution and the exact optimal control computed through a model-based LQR solver (lower is better). Additional simulation parameters are $T = 5$, $\texttt{tol} = 10^{-8}$.

Sec. VI], a ring network with $M = 4$ subsystems, i.e., $\mathcal{E}_c = \{(i, i+1), (i+1, i), i = 1, \dots, M\} \cup \{(1, M), (M, 1)\}$. Each subsystem (7.4) is a two-dimensional linearized and discretized ($\Delta t = 0.2$) swing dynamics, with

$$A_{ii} = \begin{bmatrix} 1 & \Delta t \\ -\frac{k_i}{m_i}\Delta t & 1 - \frac{d_i}{m_i}\Delta t \end{bmatrix}, \quad A_{ij} = \begin{bmatrix} 0 & 0 \\ \frac{k_{ij}}{m_i}\Delta t & 0 \end{bmatrix},$$

and $B_{ii} = \begin{bmatrix} 0 & 1 \end{bmatrix}^\mathsf{T}$. The design parameters are drawn randomly from continuous uniform distributions $m_i \sim \mathcal{U}[0, 2]$, $d_i \sim \mathcal{U}[0.5, 1]$, and $k_{ij} \sim \mathcal{U}[1, 1.5]$, with $k_i = \sum_{j \in \mathcal{N}_i} k_{ij}$. We run the same experiment three times with $Q = I$ and $R = I$, the first assuming that $\Delta_X = 0$ (i.e., noiseless case), the second that $\Delta_X \sim \mathcal{N}(0, \sigma^2 I)$, with $\sigma^2 = 0.1$, and finally with the same noise distribution but with $\sigma^2 = 5.0$. We run our algorithm with the receding horizon implementation described in Algorithm 2, with $h = 1$. We notice from Fig. 7.5 that our method and that of [Alonso et al., 2022] perform well when $\Delta_X = 0$. Crucially, however, the convergence of [Alonso et al., 2022] to a stabilizing controller is significantly slower than Algorithm 2, when data are corrupted by noise.

7.6 Conclusions

In this final chapter we build on our results on data-driven optimal control design discussed throughout this work and propose an algorithm to distributedly learn optimal controllers for a network system with data. Our approach provably converges to a suboptimal solution in a finite number of steps, with a subopotimality gap that can be characterized as a function of the available data. Moreover, although data are distributed among multiple

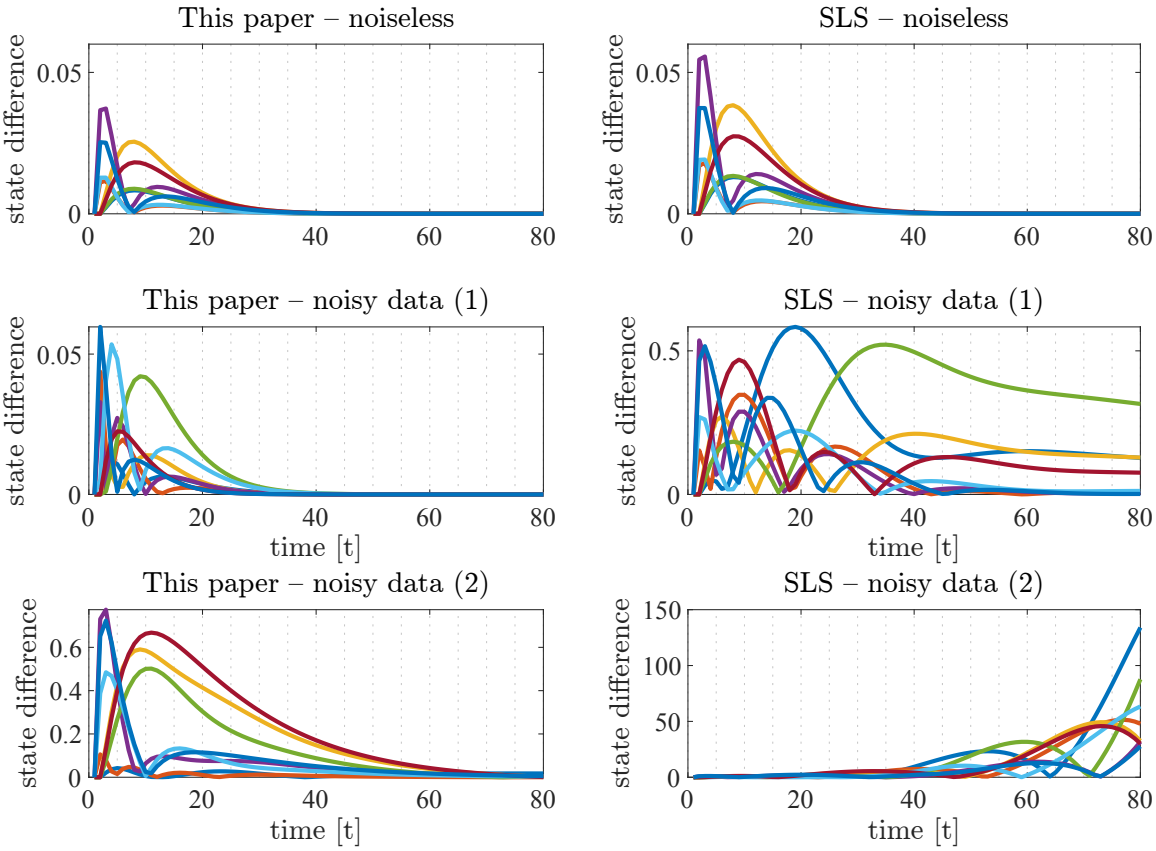

Figure 7.5: Comparison of the robustness of two different distributed data driven controllers with noisy data. The underlying model is a chain of four interconnected bi-dimensional oscillators, as discussed in [Alonso et al., 2022, Sec. VI]. Problem (7.2) is considered with $Q = 1$, $R = 1$, and random initial condition. The top panel shows the difference between the reference state evolution and the solution of Algorithm 2 (left) and of [Alonso et al., 2022] (right) when data are collected without noise. As expected, both methods converge to the equilibrium ($T = 10$). In the middle panel the same comparison is shown when data are collected with noise (each input and state trajectory is perturbed with an additive i.i.d. disturbance with zero mean and variance $\sigma^2 = 0.1$). In the bottom panel we run the same simulation for data collected with variance $\sigma^2 = 5$. $\varepsilon = 0.01$ and $\texttt{tol} = 10^{-8}$.

agents in the network, and communication between agents is therefore necessary to find a globally optimal control, this approach does not require to directly share trajectory data between agents. We discuss how these features are attractive, especially when compared to alternative approaches in the literature. Finally, we characterize the robustness properties of our approach and show that we can bound, in probability, the error on the cost function of the control computed with corrupted data.

Chapter 8

Conclusions

In this thesis we proposed a framework to systematically design closed-form expressions for the data-driven control of unknown systems. We restricted our analysis to the class of linear, time-invariant, discrete-time systems. We further made some assumptions on the nature of the noise affecting the experimental data we collect, which allowed us to study some notions of robustness for our approach. Within this framework, we covered a diverse set of problems, including open-loop input-output optimal control and minimum energy control (cf. Chapter 3), closed-loop state-feedback control design (cf. Chapter 4), eigenstructure assignment and sparse pole placement (cf. Chapter 5). In Chapter 6 we studied how data can be used to rethink the geometric approach to control, discovering new ways to link system's properties and data. Further, we proposed a strategy to apply our approach in network systems and showcased how this can be efficiently applied to a variety of complex network structures (cf. Chapter 7). Crucially, we complemented the discussion by studying the robustness properties of our approach, through formally provable bounds

that characterize how noise in the datasets might affect the approach's convergence to the desired controller (cf. Chapter 3). Here, we discovered that our closed-form expressions can be appropriately modified to account for the system noise, at least asymptotically as the number of experiments increases. When no information about the noise distribution is known, we can still guarantee some degree of convergence of our approach, with the general conclusion that more data equal better results, as one would expect.

Naturally, there are numerous avenues to expand over the present work. Some authors have already proposed solutions to new problems based on this framework or building directly over the seminal work in [Baggio et al., 2019], including the study of Imitation and Transfer Learning [Guo et al., 2023], attack detection [Krishnan and Pasqualetti, 2020, Gadginmath et al., 2022b], as well as the design of the Kalman-filter and the LQR controller [Al Makdah and Pasqualetti, 2023]. These works deal with various problems and different working assumptions, but all rely on the assumption that the underlying dynamics are linear.

Lifting the assumptions on the linearity of the underlying unknown system would be a logical next step. The study of linear systems is common practice in control systems, and it is often the starting point when developing new theories. At the same, this is a rather limiting assumption in practical applications and expanding these ideas to nonlinear systems is a natural next step. Some preliminary work in the direction of direct data-driven control for nonlinear systems has been done, for example, in [Gadginmath et al., 2022a, De Persis et al., 2023]. As with the model-based case, it seems that there is not a one-size-fits-all strategy to deal with nonlinear systems and therefore one can anticipate a fertile area

of research in this direction.

Another interesting research avenue is to rekindle the link between the originally motivating machine learning framework and our control theoretic approach. In fact, in this thesis we started from a control theoretical setting to study data-driven techniques. This comes with a formal framework which introduces all the benefits that we throughly discussed throughout the thesis, but which falls distant from being truly applicable to modern machine learning applications. At the same time, very recent and promising architectures such as Transformers [Vaswani et al., 2017] and Structured State Space Model [Gu et al., 2021, Gu and Dao, 2023], seem to suggest an ever stronger link between machine learning, state space-based control theory and system analysis [Fu et al., 2022, Du et al., 2023, Goel and Bartlett, 2023]. The study of these modern architectures with a control-theoretic focus should be a major focus for future research directions. Truly leveraging the power of machine learning in control science in a reliable, provably safe and robust fashion is one of the great challenges of our time. The prospect of merging the strengths and benefits of both fields is an exciting opportunity with potentially disruptive consequences for the future of science.

Bibliography

[Aangenent et al., 2005] Aangenent, W., Kostic, D., de Jager, B., van de Molengraft, R., and Steinbuch, M. (2005). Data-based optimal control. In *American Control Conference*, pages 1460–1465. IEEE.

[Abbasi-Yadkori et al., 2019] Abbasi-Yadkori, Y., Lazic, N., and Szepesvári, C. (2019). Model-free linear quadratic control via reduction to expert prediction. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3108–3117. PMLR.

[Achlioptas et al., 2009] Achlioptas, D., Clauset, A., Kempe, D., and Moore, C. (2009). On the bias of traceroute sampling: or, power-law degree distributions in regular graphs. *Journal of the ACM (JACM)*, 56(4):1–28.

[Al Makdah et al., 2022] Al Makdah, A. A., Krishnan, V., Katewa, V., and Pasqualetti, F. (2022). Behavioral feedback for optimal LQG control. In *IEEE Conf. on Decision and Control*, pages 4660–4666, Cancún, Mexico.

[Al Makdah and Pasqualetti, 2023] Al Makdah, A. A. and Pasqualetti, F. (2023). On the sample complexity of the linear quadratic gaussian regulator. In *IEEE Conf. on Decision and Control*, Marina Bay Sands, Singapore. To appear.

[Allibhoy et al., 2022] Allibhoy, A., Celi, F., Pasqualetti, F., and Cortés, J. (2022). Optimal network interventions to control the spreading of oscillations. *IEEE Open Journal of Control Systems*, 1:141–151.

[Allibhoy and Cortés, 2020] Allibhoy, A. and Cortés, J. (2020). Data-based receding horizon control of linear network systems. *IEEE Control Systems Letters*, 5(4):1207–1212.

[Alonso et al., 2022] Alonso, C. A., Yang, F., and Matni, N. (2022). Data-driven distributed and localized model predictive control. *IEEE Open Journal of Control Systems*, 1:29–40.

[Amaral, 2008] Amaral, L. A. N. (2008). A truer measure of our ignorance. *Proceedings of the National Academy of Sciences*, 105(19):6795–6796.

[Andry et al., 1983] Andry, A. N., Shapiro, E. Y., and Chung, J. C. (1983). Eigenstructure assignment for linear systems. *IEEE Transactions on Aerospace and Electronic Systems*, (5):711–729.

[Angulo et al., 2017] Angulo, M. T., Moreno, J. A., Lippner, G., Barabási, A.-L., and Liu, Y.-Y. (2017). Fundamental limitations of network reconstruction from temporal data. *Journal of the Royal Society Interface*, 14(127):20160966.

[Anguluri et al., 2020] Anguluri, R., Al Makdah, A. A., Katewa, V., and Pasqualetti, F. (2020). On the robustness of data-driven controllers for linear systems. In *Learning for Dynamics & Control*, volume 120 of *Proceedings of Machine Learning Research*, pages 404–412, San Francisco, CA, USA.

[Åström and Murray, 2010] Åström, K. J. and Murray, R. M. (2010). *Feedback systems: an introduction for scientists and engineers*. Princeton university press.

[Baggio et al., 2021] Baggio, G., Bassett, D. S., and Pasqualetti, F. (2021). Data-driven control of complex networks. *Nature Communications*, 12(1429).

[Baggio et al., 2019] Baggio, G., Katewa, V., and Pasqualetti, F. (2019). Data-driven minimum-energy controls for linear systems. *IEEE Control Systems Letters*, 3(3):589–594.

[Baggio and Pasqualetti, 2020] Baggio, G. and Pasqualetti, F. (2020). Learning minimum-energy controls from heterogeneous data. In *American Control Conference*, Denver, CO, USA.

[Basile and Marro, 1991] Basile, G. and Marro, G. (1991). *Controlled and Conditioned Invariants in Linear System Theory*. Prentice Hall.

[Berberich et al., 2020a] Berberich, J., Koch, A., Scherer, C. W., and Allgöwer, F. (2020a). Robust data-driven state-feedback design. In *American Control Conference*, pages 1532–1538, Denver, CO, USA.

[Berberich et al., 2020b] Berberich, J., Köhler, J., Müller, M. A., and Allgöwer, F. (2020b). Data-driven model predictive control with stability and robustness guarantees. *IEEE Transactions on Automatic Control*, 66(4):1702–1717.

[Berberich et al., 2022] Berberich, J., Scherer, C. W., and Allgöwer, F. (2022). Combining prior knowledge and data for robust controller design. *IEEE Transactions on Automatic Control*.

[Bernstein, 2009] Bernstein, D. S. (2009). *Matrix Mathematics*. Princeton University Press, 2 edition.

[Bianchin, 2023] Bianchin, G. (2023). Data-driven exact pole placement for linear systems. *arXiv preprint arXiv:2303.11469*.

[Bisoffi et al., 2021] Bisoffi, A., De Persis, C., and Tesi, P. (2021). Trade-offs in learning controllers from noisy data. *Systems & Control Letters*, 154:104985.

[Bisoffi et al., 2022] Bisoffi, A., De Persis, C., and Tesi, P. (2022). Data-driven control via Petersen's lemma. *Automatica*, 145:110537.

[Boyd et al., 2011] Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122.

[Boyd and Vandenberghe, 2004] Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.

[Bradtke et al., 1994] Bradtke, S. J., Ydstie, B. E., and Barto, A. G. (1994). Adaptive linear quadratic control using policy iteration. In *American Control Conference*, volume 3, pages 3475–3479. IEEE.

[Breschi et al., 2023] Breschi, V., Chiuso, A., and Formentin, S. (2023). Data-driven predictive control in a stochastic setting: A unified framework. *Automatica*, 152:110961.

[Brunton and Kutz, 2019] Brunton, S. L. and Kutz, J. N. (2019). *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press.

[Bullo et al., 2009] Bullo, F., Cortes, J., and Martinez, S. (2009). *Distributed control of robotic networks: a mathematical approach to motion coordination algorithms*. Princeton University Press.

[Butts, 2003] Butts, C. T. (2003). Network inference, error, and informant (in)accuracy: a bayesian approach. *Social Networks*, 25(2):103–140.

[Celi et al., 2020] Celi, F., Allibhoy, A., Pasqualetti, F., and Cortés, J. (2020). Linear-threshold dynamics for the study of epileptic events. *IEEE Control Systems Letters*, 5:1405–1410.

[Celi et al., 2021] Celi, F., Baggio, G., and Pasqualetti, F. (2021). Distributed learning of optimal controls for linear systems. In *IEEE Conf. on Decision and Control*, pages 5764–5769, Austin, TX.

[Celi et al., 2022] Celi, F., Baggio, G., and Pasqualetti, F. (2022). Closed-form estimates of the LQR gain from finite data. In *IEEE Conf. on Decision and Control*, pages 4016–4021, Cancún, Mexico.

[Celi et al., 2023a] Celi, F., Baggio, G., and Pasqualetti, F. (2023a). Closed-form and robust formulas for data-driven LQ control. *Annual Reviews in Control*, 56.

[Celi et al., 2023b] Celi, F., Baggio, G., and Pasqualetti, F. (2023b). Data-driven eigenstructure assignment for sparse feedback design. In *IEEE Conf. on Decision and Control*, Marina Bay Sands, Singapore. To appear.

[Celi et al., 2023c] Celi, F., Baggio, G., and Pasqualetti, F. (2023c). Distributed data-driven control of network systems. *IEEE Open Journal of Control Systems*, 2:93–107.

[Celi and Pasqualetti, 2022a] Celi, F. and Pasqualetti, F. (2022a). Data-driven geometric approach to control. [Online]. Available: https://github.com/federicoceli/ddga (Accessed: Feb. 7, 2024).

[Celi and Pasqualetti, 2022b] Celi, F. and Pasqualetti, F. (2022b). Data-driven meets geometric control: Zero dynamics, subspace stabilization, and malicious attacks. *IEEE Control Systems Letters*, 6:2569–2574.

[Celi et al., 2019] Celi, F., Wang, L., Pallottino, L., and Egerstedt, M. (2019). Deconfliction of motion paths with traffic inspired rules. *IEEE Int. Conf. on Robotics and Automation*, 4(2):2227–2234.

[Cherukuri et al., 2017] Cherukuri, A., Gharesifard, B., and Cortés, J. (2017). Saddle-point dynamics: conditions for asymptotic stability of saddle points. *SIAM Journal on Control and Optimization*, 55(1):486–511.

[Clason and Valkonen, 2020] Clason, C. and Valkonen, T. (2020). *Introduction to Nonsmooth Analysis and Optimization.* arXiv preprint arXiv:2001.00216.

[Coulson et al., 2019] Coulson, J., Lygeros, J., and Dörfler, F. (2019). Data-enabled predictive control: In the shallows of the DeePC. In *European Control Conference*, pages 307–312, Naples, Italy.

[Coulson et al., 2021] Coulson, J., Lygeros, J., and Dörfler, F. (2021). Distributionally robust chance constrained data-enabled predictive control. *IEEE Transactions on Automatic Control*, 67(7):3289–3304.

[da Silva et al., 2018] da Silva, G. R. G., Bazanella, A. S., Lorenzini, C., and Campestrini, L. (2018). Data-driven LQR control design. *IEEE Control Systems Letters*, 3(1):180–185.

[De Persis et al., 2023] De Persis, C., Gadginmath, D., Pasqualetti, F., and Tesi, P. (2023). Data-driven feedback linearization with complete dictionaries. In *IEEE Conf. on Decision and Control*, Marina Bay Sands, Singapore. To appear.

[De Persis and Tesi, 2020] De Persis, C. and Tesi, P. (2020). Formulas for data-driven control: Stabilization, optimality and robustness. *IEEE Transactions on Automatic Control*, 65(3):909–924.

[De Persis and Tesi, 2021] De Persis, C. and Tesi, P. (2021). Low-complexity learning of linear quadratic regulators from noisy data. *Automatica*, 128:109548.

[Dean et al., 2020] Dean, S., Mania, H., Matni, N., Recht, B., and Tu, S. (2020). On the sample complexity of the linear quadratic regulator. *Foundations of Computational Mathematics*, 20(4):633–679.

[Diem and Anderson, 1976] Diem, N. D. and Anderson, B. D. O. (1976). New results for linear-quadratic discrete-time games. *International Journal of Control*, 23(1):17–38.

[Dörfler et al., 2022] Dörfler, F., Tesi, P., and De Persis, C. (2022). On the role of regularization in direct data-driven LQR control. In *IEEE Conf. on Decision and Control*, pages 1091–1098, Cancún, Mexico. IEEE.

[Dörfler et al., 2023] Dörfler, F., Tesi, P., and De Persis, C. (2023). On the certainty-equivalence approach to direct data-driven LQR design. *IEEE Transactions on Automatic Control*.

[Du et al., 2023] Du, Z., Balim, H., Oymak, S., and Ozay, N. (2023). Can transformers learn optimal filtering for unknown systems? *IEEE Control Systems Letters*, 7:3525–3530.

[Dörfler et al., 2021] Dörfler, F., Coulson, J., and Markovsky, I. (2021). Bridging direct & indirect data-driven control formulations via regularizations and relaxations. Submitted. Available at https://arxiv.org/abs/2101.01273.

[Eising, 2021] Eising, J. (2021). *A geometric framework for constraints and data: from linear systems to convex processes*. PhD thesis, University of Groningen. [Thesis fully internal (DIV), University of Groningen].

[Eising and Cortés, 2022] Eising, J. and Cortés, J. (2022). Informativity for centralized design of distributed controllers for networked systems. In *European Control Conference*, pages 681–686.

[Eykholt et al., 2018] Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., and Song, D. (2018). Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1625–1634.

[Farokhi et al., 2014] Farokhi, F., Shames, I., and Johansson, K. H. (2014). Distributed MPC via dual decomposition and alternative direction method of multipliers. In *Distributed model predictive control made easy*, pages 115–131. Springer.

[Fawzi et al., 2014] Fawzi, H., Tabuada, P., and Diggavi, S. (2014). Secure estimation and control for cyber-physical systems under adversarial attacks. *IEEE Transactions on Automatic Control*, 59(6):1454–1467.

[Fazel et al., 2018] Fazel, M., Ge, R., Kakade, S., and Mesbahi, M. (2018). Global convergence of policy gradient methods for the linear quadratic regulator. In *International Conference on Machine Learning*, pages 1467–1476, Stockholm, Sweden.

[Fu et al., 2022] Fu, D. Y., Dao, T., Saab, K. K., Thomas, A. W., Rudra, A., and Ré, C. (2022). Hungry Hungry Hippos: Towards language modeling with state space models. In *The Eleventh International Conference on Learning Representations*.

[Furuta and Wongsaisuwan, 1993] Furuta, K. and Wongsaisuwan, M. (1993). Closed-form solutions to discrete-time LQ optimal control and disturbance attenuation. *Systems & Control Letters*, 20(6):427–437.

[Gadginmath et al., 2022a] Gadginmath, D., Krishnan, V., and Pasqualetti, F. (2022a). Data-driven feedback linearization using the koopman generator. *IEEE Transactions on Automatic Control*. Submitted.

[Gadginmath et al., 2022b] Gadginmath, D., Krishnan, V., and Pasqualetti, F. (2022b). Direct vs indirect methods for behavior-based attack detection. In *IEEE Conf. on Decision and Control*, Cancún, Mexico.

[Garrabé and Russo, 2021] Garrabé, E. and Russo, G. (2021). On the design of autonomous agents from multiple data sources. *IEEE Control Systems Letters*, 6:698 – 703.

[Gevers, 2005] Gevers, M. (2005). Identification for control: From the early achievements to the revival of experiment design. *European Journal of Control*, 11:1–18.

[Goel and Bartlett, 2023] Goel, G. and Bartlett, P. (2023). Can a transformer represent a kalman filter? *arXiv preprint arXiv:2312.06937*.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

[Gravell et al., 2020] Gravell, B., Esfahani, P. M., and Summers, T. (2020). Learning optimal controllers for linear systems with multiplicative noise via policy gradient. *IEEE Transactions on Automatic Control*, 66(11):5283–5298.

[Gu and Dao, 2023] Gu, A. and Dao, T. (2023). Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

[Gu et al., 2021] Gu, A., Goel, K., and Ré, C. (2021). Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*.

[Guo et al., 2023] Guo, T., Al Makdah, A. A., Krishnan, V., and Pasqualetti, F. (2023). Imitation and transfer learning for LQG control. *IEEE Control Systems Letters*, 7:2149–2154.

[Ho and Kalman, 1966] Ho, B. L. and Kalman, R. E. (1966). Editorial: Effective construction of linear state-variable models from input/output functions: Die konstruktion von linearen modeilen in der darstellung durch zustandsvariable aus den beziehungen für ein- und ausgangsgrößen. *at - Automatisierungstechnik*, 14(1-12):545–548.

[Innocenti and Stanziola, 1990] Innocenti, M. and Stanziola, C. (1990). Performance—robustness trade off of eigenstructure assignment applied to rotorcraft. *The Aeronautical Journal*, 94(934):124–131.

[Isidori, 1995] Isidori, A. (1995). *Nonlinear Control Systems*. Communications and Control Engineering Series. Springer, 3 edition.

[Jiao et al., 2021] Jiao, J., van Waarde, H. J., Trentelman, H. L., Camlibel, M. K., and Hirche, S. (2021). Data-driven output synchronization of heterogeneous leader-follower multi-agent systems. In *IEEE Conf. on Decision and Control*, pages 466–471, Austin, TX.

[Jung et al., 2010] Jung, K., Shah, D., and Shin, J. (2010). Distributed averaging via lifted Markov chains. *IEEE Transactions on Information Theory*, 56(1):634–647.

[Kailath, 1980] Kailath, T. (1980). *Linear Systems*. Prentice-Hall.

[Kailath et al., 2000] Kailath, T., Sayed, A. H., and Hassibi, B. (2000). *Linear estimation*. Prentice Hall.

[Kalman, 1960] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45.

[Katayama, 2005] Katayama, T. (2005). *Subspace methods for system identification*. Communications and Control Engineering. Springer-Verlag London.

[Katewa and Pasqualetti, 2021] Katewa, V. and Pasqualetti, F. (2021). Minimum-gain pole placement with sparse static feedback. *IEEE Transactions on Automatic Control*, 66(8):1558–2523.

[Kautsky et al., 1985] Kautsky, J., Nichols, N. K., and Van Dooren, P. (1985). Robust pole assignment in linear state feedback. *International Journal of control*, 41(5):1129–1155.

[Khalil, 2002] Khalil, H. K. (2002). *Nonlinear Systems*. Prentice Hall, 3 edition.

[Klein and Moore, 1977] Klein, G. and Moore, B. C. (1977). Eigenvalue-generalized eigenvector assignment with state feedback. *IEEE Transactions on Automatic Control*, 22(1):140–141.

[Kleinman, 1968] Kleinman, D. L. (1968). On an iterative technique for Riccati equation computations. *IEEE Transactions on Automatic Control*, 13(1):114–115.

[Krishnan and Pasqualetti, 2020] Krishnan, V. and Pasqualetti, F. (2020). Data-driven attack detection for linear systems. *IEEE Control Systems Letters*, 5(2):671–676.

[Krishnan and Pasqualetti, 2021] Krishnan, V. and Pasqualetti, F. (2021). On direct vs indirect data-driven predictive control. In *IEEE Conf. on Decision and Control*, pages 736–741, Austin, TX.

[Lancaster and Rodman, 1995] Lancaster, P. and Rodman, L. (1995). *Algebraic Riccati equations*. Clarendon press.

[Lewis, 1981] Lewis, F. (1981). A generalized inverse solution to the discrete-time singular Riccati equation. *IEEE Transactions on Automatic Control*, 26(2):395–398.

[Li et al., 2022] Li, Y., Wang, X., Sun, J., Wang, G., and Chen, J. (2022). Data-driven consensus control of fully distributed event-triggered multi-agent systems. *Sci China Inf Sci*, 66(5):152202.

[Lin et al., 2011] Lin, F., Fardad, M., and Jovanović, M. R. (2011). Augmented Lagrangian approach to design of structured optimal state feedback gains. *IEEE Transactions on Automatic Control*, 56(12):2923–2929.

[Liu and Patton, 1998] Liu, G. P. and Patton, R. J. (1998). *Eigenstructure assignment for control system design*. John Wiley & Sons, Inc.

[Ljung, 1987] Ljung, L. (1987). *System Identification: Theory for the User*. Prentice Hall.

[Lopez and Müller, 2022] Lopez, V. G. and Müller, M. A. (2022). On a continuous-time version of willems' lemma. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 2759–2764. IEEE.

[Markovsky and Dörfler, 2021] Markovsky, I. and Dörfler, F. (2021). Behavioral systems theory in data-driven analysis, signal processing, and control. *Annual Reviews in Control*, 52:42–64.

[Markovsky and Rapisarda, 2008] Markovsky, I. and Rapisarda, P. (2008). Data-driven simulation and control. *International Journal of Control*, 81(12):1946–1959.

[Massoumnia et al., 1989] Massoumnia, M. A., Verghese, G. C., and Willsky, A. S. (1989). Failure detection and identification. *IEEE Transactions on Automatic Control*, 34(3):316–321.

[MathWorks, 2023] MathWorks (2023). Build Watts-Strogatz small world graph model. [Online]. Available: https://www.mathworks.com/help/matlab/math/build-watts-strogatz-small-world-graph-model.html (Accessed: Jan. 16, 2023).

[Mesbahi and Egerstedt, 2010] Mesbahi, M. and Egerstedt, M. (2010). *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press.

[Miller et al., 2022] Miller, L. J., Grauer, J. A., Pei, J., and Nelson, S. L. (2022). Reconstruction of the Apollo 11 Moon landing final descent trajectory. Technical Report TM-20220007267, NASA.

[Mohammadi et al., 2019] Mohammadi, H., Zare, A., Soltanolkotabi, M., and Jovanović, M. R. (2019). Global exponential convergence of gradient methods over the nonconvex landscape of the linear quadratic regulator. In *IEEE Conf. on Decision and Control*, pages 7474–7479, Nice, France.

[Molzahn et al., 2017] Molzahn, D. K., Dörfler, F., Sandberg, H., Low, S. H., Chakrabarti, S., Baldick, R., and Lavaei, J. (2017). A survey of distributed optimization and control algorithms for electric power systems. *IEEE Transactions on Smart Grid*, 8(6):2941–2962.

[Monshizadeh, 2020] Monshizadeh, N. (2020). Amidst data-driven model reduction and control. *IEEE Control Systems Letters*, 4(4):833–838.

[Moore, 1976] Moore, B. C. (1976). On the flexibility offered by state feedback in multivariable systems beyond closed loop eigenvalue assignment. *IEEE Transactions on Automatic Control*, 21(5):689–692.

[Moore et al., 2004] Moore, D., Leonard, J., Rus, D., and Teller, S. (2004). Robust distributed network localization with noisy range measurements. In *ACM Conference on Embedded Networked Sensor Systems*, pages 50–61, Baltimore, MD, USA.

[Mukherjee and Hossain, 2022] Mukherjee, S. and Hossain, R. R. (2022). Data-driven pole placement in LMI regions with robustness guarantees. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 4010–4015. IEEE.

[Ng and Jordan, 2001] Ng, A. and Jordan, M. (2001). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems*, 14.

[Nozari et al., 2023] Nozari, E., Bertolero, M. A., Stiso, J., Caciagli, L., Cornblath, E. J., He, X., Mahadevan, A. S., Pappas, G. J., and Bassett, D. S. (2023). Macroscopic resting-state brain dynamics are best described by linear models. *Nature biomedical engineering*, pages 1–17.

[Oh et al., 2015] Oh, K. K., Park, M. C., and Ahn, H. S. (2015). A survey of multi-agent formation control. *Automatica*, 53:424–440.

[Padula et al., 2021] Padula, F., Ferrante, A., and Ntogramatzidis, L. (2021). Eigenstructure assignment in linear geometric control. *Automatica*, 124:109363.

[Pallottino et al., 2007] Pallottino, L., Scordio, V. G., Bicchi, A., and Frazzoli, E. (2007). Decentralized cooperative policy for conflict resolution in multivehicle systems. *IEEE Transactions on Robotics*, 23(6):1170–1183.

[Pasqualetti et al., 2011] Pasqualetti, F., Bicchi, A., and Bullo, F. (2011). Consensus computation in unreliable networks: A system theoretic approach. *IEEE Transactions on Automatic Control*, 56(12):90–104.

[Pasqualetti et al., 2012] Pasqualetti, F., Carli, R., and Bullo, F. (2012). Distributed estimation via iterative projections with application to power network monitoring. *Automatica*, 48(5):747–758.

[Pasqualetti et al., 2013] Pasqualetti, F., Dörfler, F., and Bullo, F. (2013). Attack detection and identification in cyber-physical systems. *IEEE Transactions on Automatic Control*, 58(11):2715–2729.

[Pasqualetti et al., 2014] Pasqualetti, F., Zampieri, S., and Bullo, F. (2014). Controllability metrics, limitations and algorithms for complex networks. *IEEE Transactions on Control of Network Systems*, 1(1):40–52.

[Pellegrino et al., 2023a] Pellegrino, F. A., Blanchini, F., Fenu, G., and Salvato, E. (2023a). Closed-loop control from data driven open-loop optimal control trajectories. In *European Control Conference*, pages 1379–1384.

[Pellegrino et al., 2023b] Pellegrino, F. A., Blanchini, F., Fenu, G., and Salvato, E. (2023b). Data-driven dynamic relatively optimal control. *European Journal of Control*, page 100839.

[Polderman and Willems, 1997] Polderman, J. W. and Willems, J. C. (1997). *Introduction to mathematical systems theory: a behavioral approach*, volume 26. Springer Science & Business Media.

[Recht, 2019] Recht, B. (2019). A tour of reinforcement learning: The view from continuous control. *Annual Review of Control, Robotics, and Autonomous Systems*, 2:253–279.

[Ren and Beard, 2005] Ren, W. and Beard, R. W. (2005). Consensus seeking in multiagent systems under dynamically changing interaction topologies. *IEEE Transactions on Automatic Control*, 50(5):655–661.

[Rotulo et al., 2020] Rotulo, M., De Persis, C., and Tesi, P. (2020). Data-driven linear quadratic regulation via semidefinite programming. *IFAC-PapersOnLine*, 53(2):3995–4000.

[Russell and Norvig, 2021] Russell, S. J. and Norvig, P. (2021). *Artificial intelligence a modern approach*. Pearson.

[Schmitz et al., 2022] Schmitz, P., Faulwasser, T., and Worthmann, K. (2022). Willems' fundamental lemma for linear descriptor systems and its use for data-driven output-feedback MPC. *IEEE Control Systems Letters*, 6:2443–2448.

[Sezer and Šiljak, 1981] Sezer, M. E. and Šiljak, D. D. (1981). Structurally fixed modes. *Systems & Control Letters*, 1(1):60–64.

[Swartz et al., 2022] Swartz, J., Celi, F., Pasqualetti, F., and von Meier, A. (2022). Parameter conditions to prevent voltage oscillations caused by ltc-inverter hunting on power distribution grids. In *European Control Conference*, London, UK.

[Szegedy et al., 2014] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2014). Intriguing properties of neural networks. In *International Conference on Learning Representations*, Banff, Canada.

[Tabuada et al., 2017] Tabuada, P., Ma, W., Grizzle, J., and Ames, A. D. (2017). Data-driven control for feedback linearizable single-input systems. In *IEEE Conf. on Decision and Control*, pages 6265–6270, Melbourne, Australia.

[Taylor, 2019] Taylor, A. (2019). Math invented for Moon landing helps your flight arrive on time. [Online]. Available: https://www.nasa.gov/aeronautics/math-invented-for-moon-landing-helps-your-flight-arrive-on-time (Accessed: Feb. 8, 2024).

[Teoh et al., 2022] Teoh, J. Q., Tehrani, M. G., Ferguson, N. S., and Elliott, S. J. (2022). Eigenvalue sensitivity minimisation for robust pole placement by the receptance method. *Mechanical Systems and Signal Processing*, 173:108974.

[Trentelman et al., 2001] Trentelman, H. L., Stoorvogel, A., and Hautus, M. (2001). *Control Theory for Linear Systems*. Springer.

[Tu et al., 2022] Tu, S., Frostig, R., and Soltanolkotabi, M. (2022). Learning from many trajectories. *arXiv preprint arXiv:2203.17193*.

[Turan and Ferrari-Trecate, 2021] Turan, S. M. and Ferrari-Trecate, G. (2021). Data-driven unknown-input observers and state estimation. *IEEE Control Systems Letters*, 6:1424–1429.

[Van der Vaart, 2000] Van der Vaart, A. W. (2000). *Asymptotic statistics*, volume 3 of *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press.

[van Waarde et al., 2020a] van Waarde, H. J., Camlibel, M. K., and Mesbahi, M. (2020a). From noisy data to feedback controllers: Nonconservative design via a matrix S-lemma. *IEEE Transactions on Automatic Control*, 67(1):162–175.

[van Waarde et al., 2022] van Waarde, H. J., Camlibel, M. K., and Mesbahi, M. (2022). From noisy data to feedback controllers: non-conservative design via a matrix S-lemma. *IEEE Transactions on Automatic Control*, 67:162–175.

[van Waarde et al., 2020b] van Waarde, H. J., De Persis, C., Camlibel, M. K., and Tesi, P. (2020b). Willems' fundamental lemma for state-space systems and its extension to multiple datasets. *IEEE Control Systems Letters*, 4(3):602–607.

[van Waarde et al., 2020] van Waarde, H. J., Eising, J., Trentelman, H. L., and Camlibel, M. K. (2020). Data informativity: a new perspective on data-driven analysis and control. *IEEE Transactions on Automatic Control*, 65(11):4753–4768.

[van Waarde and Mesbahi, 2020] van Waarde, H. J. and Mesbahi, M. (2020). Data-driven parameterizations of suboptimal LQR and H_2 controllers. *IFAC World Congress*, 53(2):4234–4239.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

[Verhoek et al., 2021] Verhoek, C., Tóth, R., Haesaert, S., and Koch, A. (2021). Fundamental lemma for data-driven analysis of linear parameter-varying systems. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 5040–5046. IEEE.

[Vit, 1972] Vit, K. (1972). Iterative solution of the Riccati equation. *IEEE Transactions on Automatic Control*, 17(2):258–259.

[Walsh and Ye, 2001] Walsh, G. C. and Ye, H. (2001). Scheduling of networked control systems. *IEEE control systems magazine*, 21(1):57–65.

[Wang et al., 2016] Wang, P., Man, Z., Cao, Z., Zheng, J., and Zhao, Y. (2016). Dynamics modelling and linear control of quadcopter. In *2016 International Conference on Advanced Mechatronic Systems (ICAMechS)*, pages 498–503.

[Wang et al., 2023] Wang, X., Sun, J., Wang, G., Allgöwer, F., and Chen, J. (2023). Data-driven control of distributed event-triggered network systems. *IEEE/CAA Journal of Automatica Sinica*, 10:1–14.

[Wang et al., 2019] Wang, Y.-S., Matni, N., and Doyle, J. C. (2019). A system-level approach to controller synthesis. *IEEE Transactions on Automatic Control*, 64(10):4079–4093.

[Watts and Strogatz, 1998] Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442.

[Willems and Polderman, 1997] Willems, J. C. and Polderman, J. W. (1997). *Introduction to mathematical systems theory: a behavioral approach*, volume 26. Springer Science & Business Media.

[Willems et al., 2005] Willems, J. C., Rapisarda, P., Markovsky, I., and Moor, B. L. M. D. (2005). A note on persistency of excitation. *Systems & Control Letters*, 54(4):325–329.

[Wonham, 1967] Wonham, W. M. (1967). On pole assignment in multi-input controllable linear systems. *IEEE Transactions on Automatic Control*, 12(6):660–665.

[Wonham, 1985] Wonham, W. M. (1985). *Linear Multivariable Control: A Geometric Approach*. Springer, 3 edition.

[Wonham and Morse, 1970] Wonham, W. M. and Morse, A. S. (1970). Decoupling and pole assignment in linear multivariable systems: a geometric approach. *SIAM Journal on Control*, 8(1):1–18.

[Yu et al., 2021] Yu, Y., Talebi, S., van Waarde, H. J., Topcu, U., Mesbahi, M., and Açıkmeşe, B. (2021). On controllability and persistency of excitation in data-driven control: Extensions of willems' fundamental lemma. In *IEEE Conf. on Decision and Control*, pages 6485–6490. IEEE.

[Zhang, 2005] Zhang, F. (2005). *The Schur Complement and Its Applications*. Springer.

[Zhou et al., 1996] Zhou, K., Doyle, J. C., and Glover, K. (1996). *Robust and Optimal Control*. Prentice Hall.

[Ziegler and Nichols, 1942] Ziegler, J. G. and Nichols, N. B. (1942). Optimum settings for automatic controllers. *Transactions of the ASME*, 64(11).