A data-centric view of LQR algorithms in continuous time

by

Christopher Song

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Applied Mathematics

Waterloo, Ontario, Canada, 2025

© Christopher Song 2025

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

Chapters 3 and 4 of this thesis contain material from the following manuscript I coauthored with my supervisor Jun Liu, which we submitted to L-CSS and CDC 2025.

All Data-Driven LQR Algorithms Require at Least as Much Interval Data as System Identification

Christopher Song and Jun Liu

I contributed to the development of the theory, the numerical experiments, and the writing and editing of the manuscript. Jun Liu contributed to the development of the theory and the editing of the manuscript.

Abstract

Many control theorists are interested in how measurements of input and state trajectories can determine the properties of a control system, in lieu of the differential or difference equation models that usually play this role. In situations of practical interest, such models may be wholly or partially unknown, while input and state data are readily available. In the discrete- and continuous-time linear quadratic regulator (LQR) settings, this area of research has produced a proliferation of algorithms that use input and state data to solve the LQR problem, the system identification problem, or both. In different algorithms, these data and the requirements imposed on them take different forms that are not directly comparable, making it difficult to assess the relative efficiencies with which different algorithms makes use of data.

In [30], the authors show that the LQR and system identification problems are essentially equivalent in the discrete-time setting. In this thesis, we extend this result to the continuous-time setting, showing that, assuming input and state data are collected on intervals, every algorithm that solves the LQR problem requires at least as much data as system identification. From this, we show that the map from the data to the optimal gain defined by these algorithms is continuous, establishing a connection between interval data and sampled data algorithms. The possibility of using sampled data in place of interval data leads to a weaker convergence criterion on sampled data approximations, and a natural connection with numerical integration. We do some numerical experiments that show the critical importance of choosing when to make input and state measurements, and emphasize the possibility of doing so without knowledge of the system or its optimal gain.

Acknowledgements

I am very grateful to Jun Liu, who has consistently provided me with his indispensable guidance and warm support during my time in the math department. Many thanks are due also to Kirsten Morris and Henry Shum, for dedicating their time and energy to reading my thesis, providing their valuable feedback, and attending my defense.

Table of Contents

Author's Declaration ii

Statement of Contributions iii

Abstract iv

Acknowledgements v

List of Figures viii

1 Introduction 1

1.1 Existing work and motivations 2

1.2 Overview of the thesis 4

2 Linear Quadratic Regulator 5

2.1 Finite-horizon problem 6

2.2 Infinite-horizon problem 9

2.3 Kleinman's algorithm 10

2.4 Lyapunov equations 11

3 LQR with interval data from unknown systems 12

3.1 LQR directly from interval data 12

3.2 LQR via system identification 14

3.3 Comparing data requirements of ADP and system identification 15

3.3.1 Numerical experiments with intervals of equal size 15

3.3.2 An example with intervals of unequal size 21

3.4 Informativity of interval data for LQR 27

3.5 Direct LQR algorithms require as much data as system identification 28

3.6 Remarks 30

4 Sampled data approximations **32**
4.1 The map from data to system is continuous 33
4.2 The map from data to optimal gain is continuous 34
4.3 A weaker condition for approximation 36
4.4 Numerical experiments with sampled data 37
4.5 Remarks 41

5 Conclusions and future directions **43**
5.1 Conclusions 43
5.2 Future directions 45

References **47**

List of Figures

3.1 Error in K and P for a 2-dimensional system from 1 to 30 intervals of data. 17
3.2 Error in K and P for a 3-dimensional system from 1 to 30 intervals of data. 19
3.3 Error in K and P for a 6-dimensional system from 1 to 70 intervals of data. 21
3.4 Data for ADP and system identification over time for a 2-dimensional system. 23
3.5 "Pathological" sampling interval endpoints for ADP and system identification. 24

4.1 Error in K and P for a 2-dimensional system from 3 to 10,000 samples per interval. 39
4.2 Error in K and P for a 3-dimensional system from 3 to 10,000 samples per interval. 40
4.3 Error in K and P for a 6-dimensional system from 3 to 10,000 samples per interval. 41

Chapter 1

Introduction

Optimal control theory is a way of thinking about how to make decisions over time, wherein the objective is to find an action that optimizes a certain criterion. Commonly, this takes the form of seeking a control input for a dynamical system (usually represented by a differential or difference equation) that minimizes a cost functional representing the objectives one has for the system's behavior. The mathematical theory of optimal control is rooted in the calculus of variations, a classical branch of mathematical analysis originating in the 17^{th} century. Optimal control theorists developed their field rapidly during the second half of the 20^{th} century, spurred on by emerging applications to important problems in areas such as engineering, economics, and operations research. One of the most celebrated achievements of this development is the famous linear quadratic regulator (LQR) theory, which formulates and solves the optimal control problem in the setting of linear control-affine dynamical systems with quadratic cost functionals.

In the 21^{st} century, the techniques of optimal control theory in general and LQR theory in particular continue to enjoy broad and diverse applications. Armed with increasing amounts of computing power, modern control theorists are able to study dynamical systems of greater complexity, often without known differential or difference equation representations. Simultaneously, many of the fields where control theory is fruitfully applied have seen a massive increase in the availability of data, precipitated by the availability of sensing hardware.

These trends have conspired to generate large academic interest in the field of data-driven control, where observations of the input and state, rather than knowledge of the underlying dynamical system, play the role of constraining the system's possible behavior. LQR theory, seen from the new data-centric perspective, remains an active area of research. We presently give a brief review of some recent developments in this area that serves to motivate the remainder of this thesis.

1.1 Existing work and motivations

There are two dominant and contrasting approaches to data-driven optimal control problems in general. One approach calls for representations of the dynamical system to be estimated and constructed from measurements of the input and state trajectories it generates, which we call data. This step is known as system identification. Once the system is identified, the classical methods of optimal control may be directly applied to these representations in order to solve the optimal control problem. The other approach abandons representations of the system and attempts to find the optimal control directly from data. Accordingly, algorithms for the solution of LQR problems from input and state trajectories generated by unknown systems may be classified as direct or indirect, depending on whether they compute the system matrices A and B as an intermediate step.

It is natural to ask whether this step imposes additional requirements on the amount of data needed by indirect algorithms, regardless of their specific designs. Apart from the practical utility of developing algorithms which require the least possible amount of data to solve a given problem, this question is of theoretical interest because any answer to it would shed light on the relationship between the two problems of system identification and data-driven optimal control.

In the discrete-time setting, this question is essentially answered in the negative in [30], where the authors show that data rich enough for a direct LQR algorithm to find the optimal gain K are also sufficient to identify the system, provided that $K \neq 0$. They do this by considering the set containing only those systems that could have generated input and state trajectories consistent with the observed data. The systems belonging to such a subset are "compatible with the data", a definition we formalize for our setting in Chapter 3, Definition 3.41. The objective is not necessarily to narrow down such a subset to a singleton (as one does in system identification), or to solve the optimal control problem. Rather, what emerges are relationships between properties shared by the systems in such sets, and properties of the data which define those sets. Given a particular control-relevant property such as stabilization by a certain input, we ask what conditions the data must fulfill in order for the property to be ascribed to all systems in the set it defines. By taking such a perspective on data-driven control, we can glean important information about unknown systems from the data they generate, even without identifying them. This is known as the data informativity framework.

The data informativity framework introduced in [30] is further developed in the setting of continuous time in [21], where stabilization of the system is linked to the existence of a solution to a linear matrix inequality (LMI). In that paper, it is assumed that the values taken by the input and state are known on intervals of time, rather than being measured only at discrete points in time. We refer to these different forms of data as interval data and sampled data, respectively. In [9], informativity for stabilization with both interval and sampled data is investigated under the assumption of perfect knowledge of the state derivative. This is a simplifying assumption, since the state derivative is normally difficult to measure directly.

These results, among others, validate the usefulness of the data informativity framework for studying continuous-time linear systems. Nevertheless, to the best of our knowledge, no continuous-time analogue of the result from [30] establishing the equivalence between data informativity for direct LQR and system identification is known.

Given the question of whether direct and indirect LQR algorithms impose fundamentally different data requirements, it is useful to examine existing approaches in both discrete- and continuous-time settings. Direct LQR algorithms are numerous in the literature. See, for example, [22], [17], [29], [32], and [6] for discrete-time systems, and [14], [10], [31], [20], and [19] for continuous-time systems. All of the latter group of algorithms rely on interval data in the form of definite integrals of the input and state in order to perform a version of Kleinman iteration [15]. Other authors have proposed indirect algorithms that rely on estimating the state derivative. For example, in [21], interval data are used to obtain representations of the input u and state trajectory x in terms of a polynomial basis, and these are used to approximate the state derivative, which allows the system to be approximately identified. In [23], similar polynomial approximations to x and u are used in a direct LQR algorithm. In [11], the state derivative is approximated from sampled data using low-pass filtering in order to do system identification. Low-pass filtering on sampled data is also used in [5], but the state derivative is avoided in favour of LMIs. The authors of [7] present a unified treatment of discrete- and continuous-time systems and reformulate the sampled data LQR problem as a linear program.

Faced with such a large and diverse catalogue of data-driven LQR algorithms, one naturally hopes to get an idea of their relative advantages and drawbacks. These direct and indirect algorithms use interval or sampled data in different ways, making it difficult to directly compare the various requirements they impose on the data they use. Do the differences in the forms of data used by these algorithms lead to differences in their performance? The lack of comparability between different data requirements also precludes the drawing of conclusions about which of these algorithms are more data-efficient, in the sense of requiring less data to solve the same problem. Given these differences, it is even unclear how the "amount" of data should be defined. Furthermore, it remains to be seen how various requirements on data are sensitive not only to the quantity of measured data, but also to when exactly the measurements are made. Are some times better than others for collecting data? If so, can we know when those times are, in the absence of knowledge of the state or the optimal control?

Insight into these questions might be gained if only one key question were answered: can the fundamental equivalence between direct and indirect LQR algorithms, established for discrete-time systems in [30], be extended to continuous-time systems?

1.2 Overview of the thesis

The primary aim of this thesis is to address the question above by proving that from a data-centric perspective, the LQR and system identification problems are equivalent. It is well known that system identification may be used to solve the LQR problem, as we describe in Subsection 3.2. We show that additionally, all continuous-time data-driven LQR algorithms using interval data are doing system identification. We summarize the secondary questions this thesis aims to investigate as follows:

- How do interval data and sampled data LQR algorithms compare with one another?
- How can we compare the amounts of data required by different LQR algorithms when those data take differing forms?
- How important is the timing of the collection of data?

We address these aims in the next four chapters. In Chapter 2, we formulate the theory for the continuous-time infinite-horizon LQR problem and review its central results. We do this by first considering the finite-horizon problem, and then take the limit of its solution as the time horizon goes to infinity. In Chapter 3, we do some numerical experiments to investigate differences in performance between a direct and an indirect LQR algorithm. Then, we formulate the data informativity framework with interval data in order to show that every direct LQR algorithm using interval data requires as much data (expressed as the rank of a matrix) as system identification. In Chapter 4, we use this result to prove that the map from interval data generated by a system to the system's optimal gain is continuous. This enables the approximation of interval data in integral form by sampled data, furnishing a connection with the theory of numerical integration. We supplement our discussion of this connection with some more numerical experiments. In Chapter 5, we summarize our results, revisit the aims outlined above, and suggest some future research directions.

Chapter 2

Linear Quadratic Regulator

Control theorists are often interested in finding inputs that drive the state of a dynamical system toward the origin as time t increases. Informally speaking, it may be desirable to find an input that is both small in magnitude and able to drive the state to the origin quickly. In this respect, some choices of input are better than others, and we are motivated to search for a control which is in some sense optimal with respect to some criterion. The setup of the LQR problem formalizes this idea.

Let A, B, Q, and R be matrices (constant in time t) in $\mathbb{R}^{n\times n}$, $\mathbb{R}^{n\times m}$, $\mathbb{R}^{n\times n}$, and $\mathbb{R}^{m\times m}$, respectively. We require Q to be positive semidefinite and R to be positive definite. Let the input u be an $\mathbb{R}^m$-valued function of t, and let the state x be the $\mathbb{R}^n$-valued function of time that solves the initial value problem (IVP)

$$\dot{x}(t) = Ax(t) + Bu(t), \quad x(t_0) = x_0. \tag{2.1}$$

We assume that u is such that x exists uniquely on whatever interval of time we are interested in. Throughout the rest of this thesis, we refer to the matrices A and B together, the pair (A, B), and the block matrix $\begin{bmatrix} A & B \end{bmatrix}^\top$ as the system or the system matrices, sometimes without distinguishing between them. This is the setup for the finite- and infinite-horizon LQR theories we discuss shortly.

This chapter is organized as follows. In Section 2.1, we introduce the finite-horizon LQR theory and derive some of its results. We do this mainly for psychological reasons, just to see how the optimal control is actually found. The steps therein follow those in the first part of [2, Chapter 2]. The infinite-horizon theory develops out of the finite-horizon theory, but we gloss over the details of this development. In Section 2.2, we summarize the main results for infinite-horizon LQR, omitting their derivations. In Section 2.3, we state an algorithm for iteratively computing approximations to the solution of the infinite-horizon LQR problem that forms the basis for a data-driven algorithm we discuss in Section 3.1. Section 2.4 contains a small result about Lyapunov equations which comes in handy in Section 3.5.

2.1 Finite-horizon problem

Let t_0 and t_f be real numbers with $t_0 \leq t_f$, and let $\mathcal{U}$ be the set of $\mathbb{R}^m$-valued functions piecewise continuous on the interval $[t_0, t_f]$ that keep the state bounded on this interval. Fix a particular initial condition $x(t_0) = x_0$. We define

$$\begin{aligned} J_{x_0} : \mathcal{U} &\longrightarrow \mathbb{R} \\ u &\longmapsto \int_{t_0}^{t_f} x^\top(t)Qx(t) + u^\top(t)Ru(t)\,\mathrm{d}t + x^\top(t_f)Q_f x(t_f), \end{aligned} \tag{2.2}$$

where $x(t)$ solves the IVP (2.1) and $Q_f \in \mathbb{R}^{n\times n}$ is positive semidefinite. The terms $x^\top Qx$ and $u^\top Ru$ may be respectively thought of as weighted penalties on the state being far from the origin, and the control effort being large. The term $x^\top(t_f)Q_f x(t_f)$ penalizes the state at the time t_f. Our condition on u are to guarantee the existence of a unique solution of (2.1) on $[t_0, t_f]$.

At this stage, it may not be obvious whether there exists a $u \in \mathcal{U}$ which minimizes J_{x_0}, how such a minimizer may depend on x_0, or what form it may take. The following theorem addresses these questions.

Theorem 2.1.1. *The minimizer u^* of J_{x_0} exists, is unique, and is given by*

$$u^* = Kx^*, \tag{2.3}$$

where x^* is the unique solution to the IVP

$$\dot{x}^*(t) = (A + BK)x^*(t), \quad x^*(t_0) = x_0 \tag{2.4}$$

existing on the interval $[t_0, t_f]$, and

$$K = -R^{-1}B^\top P, \tag{2.5}$$

where P is the unique solution to the IVP

$$\dot{P} = PBR^{-1}B^\top P - A^\top P - PA - Q, \quad P(t_f) = Q_f \tag{2.6}$$

existing on the interval $[t_0, t_f]$.

We spend the remainder of this section building the proof of this result. We start by introducing the following ansatz. We assume such a minimizer exists, denote it by u^*, refer to it as the optimal input or optimal control, and assume it has the form

$$u^*(t) = K(t)x^*(t), \tag{2.7}$$

where K is a $n \times m$ matrix-valued function of time and x^* is the solution to the IVP

$$\dot{x}^*(t) = (A + BK)x^*(t), \quad x^*(t_0) = x_0. \tag{2.8}$$

It is possible to write every $u \in \mathcal{U}$ as

$$u(t) = Kx(t) + \mu(t), \tag{2.9}$$

where x solves (2.1) and μ is some $\mathbb{R}^m$ valued function on $[t_0, t_f]$. By doing this, we hope to show that with a judicious choice of K, we will obtain $\mu = 0$ as the minimizer of J_{x_0}, and in doing so justify our ansatz. We proceed by defining

$$\tilde{x} := x - x^*, \tag{2.10}$$

which lets us write (2.9) and (2.1) as

$$\begin{aligned} u &= K(x^* + \tilde{x}) + \mu \\ &= u^* + K\tilde{x} + \mu, \end{aligned} \tag{2.11}$$

$$\begin{aligned} \dot{x}^* + \dot{\tilde{x}} &= A(x^* + \tilde{x}) + B(K(x^* + \tilde{x}) + \mu), \quad x^*(t_0) + \tilde{x}(t_0) = x_0 \\ &= (A + BK)x^* + (A + BK)\tilde{x} + B\mu, \quad x^*(t_0) + \tilde{x}(t_0) = x_0, \end{aligned} \tag{2.12}$$

respectively. Subtracting (2.8) shows that $\tilde{x}$ solves the IVP

$$\dot{\tilde{x}} = (A + BK)\tilde{x} + B\mu, \quad \tilde{x}(t_0) = 0. \tag{2.13}$$

Now substituting (2.10) and (2.11) into (2.2) and doing some algebra, we get

$$\begin{aligned} J_{x_0}(u) = J_{x_0}(u^*) &\\ + \int_{t_0}^{t_f} \tilde{x}^\top(t)Q\tilde{x}(t) + (K(t)\tilde{x}(t) + \mu(t))^\top R(K(t)\tilde{x}(t) + \mu(t))\,\mathrm{d}t + \tilde{x}^\top(t_f)Q_f\tilde{x}(t_f) &\\ + 2\left(\int_{t_0}^{t_f} \tilde{x}^\top(t)Qx^*(t) + (K(t)\tilde{x}(t) + \mu(t))^\top RK(t)x^*(t)\,\mathrm{d}t + \tilde{x}^\top(t_f)Q_f x^*(t_f)\right), & \end{aligned} \tag{2.14}$$

where our conditions on Q, R, t_0, and t_f ensure that

$$\int_{t_0}^{t_f} \tilde{x}^\top(t)Q\tilde{x}(t) + (K(t)\tilde{x}(t) + \mu(t))^\top R(K(t)\tilde{x}(t) + \mu(t))\,\mathrm{d}t + \tilde{x}^\top(t_f)Q_f\tilde{x}(t_f) \geq 0. \tag{2.15}$$

If we can select K such that

$$\int_{t_0}^{t_f} \tilde{x}^\top(t)Qx^*(t) + (K(t)\tilde{x}(t) + \mu(t))^\top RK(t)x^*(t)\,\mathrm{d}t + \tilde{x}^\top(t_f)Q_f x^*(t_f) = 0, \tag{2.16}$$

then indeed $\mu = 0$ will give $\tilde{x}$ identically zero as the unique solution to (2.13), making (2.15) zero and $J_{x_0}(u) = J_{x_0}(u^*)$. This would show that J_{x_0} attains its minimum at $u^* = Kx^*$, justifying our ansatz.

In order to compute K from (2.16), we move the term $\tilde{x}^\top(t_f)Q_f x^*(t_f)$ under the integral. To do this, we introduce a $n \times n$ matrix-valued function of time P satisfying

$$P(t_f) = Q_f, \tag{2.17}$$

and we use the fundamental theorem of calculus to write

$$\begin{aligned}\tilde{x}^\top(t_f)Q_f x^*(t_f) &= \tilde{x}^\top(t_f)P(t_f)x^*(t_f)\\ &= \tilde{x}^\top(t_0)P(t_0)x^*(t_0) + \int_{t_0}^{t_f} \frac{\mathrm{d}}{\mathrm{d}t}\left(\tilde{x}^\top(t)P(t)x^*(t)\right)\mathrm{d}t\\ &= \tilde{x}^\top(t_0)P(t_0)x^*(t_0) + \int_{t_0}^{t_f} \dot{\tilde{x}}^\top(t)P(t)x^*(t)\\ &\quad + \tilde{x}^\top(t)\dot{P}(t)x^*(t) + \tilde{x}^\top(t)P(t)\dot{x}^*(t)\,\mathrm{d}t, \end{aligned} \tag{2.18}$$

into which we substitute (2.8) and (2.13) for $\dot{\tilde{x}}$ and $\dot{x}^*$ respectively, and $\tilde{x}(t_0) = 0$ before rearranging to get

$$\begin{aligned}&\tilde{x}^\top(t_f)Q_f x^*(t_f)\\ &= \int_{t_0}^{t_f} \tilde{x}^\top(t)\left(\dot{P}(t) + A^\top P(t) + P(t)A + P(t)BK(t) + K^\top(t)B^\top P(t)\right)x^*(t)\\ &\quad + \mu^\top(t)\left(B^\top P(t)\right)x^*(t)\,\mathrm{d}t. \end{aligned} \tag{2.19}$$

Substituting the above into (2.16) and collecting like terms yields

$$\begin{aligned}&\int_{t_0}^{t_f} \tilde{x}^\top(t)\left(\dot{P}(t) + (A+BK(t))^\top P(t) + P(t)(A+BK(t)) + Q + K^\top(t)RK(t)\right)x^*(t)\,\mathrm{d}t\\ &+ \int_{t_0}^{t_f} \mu^\top(t)\left(B^\top P(t) + RK(t)\right)x^*(t)\,\mathrm{d}t = 0. \end{aligned} \tag{2.20}$$

At this point it becomes clear that if we find a pair (P, K) that solves the system

$$0 = \dot{P} + (A+BK)^\top P + P(A+BK) + Q + K^\top RK, \tag{2.21a}$$

$$0 = B^\top P + RK, \tag{2.21b}$$

then this K will satisfy (2.16). Using the invertibility of R, we write

$$K = -R^{-1}B^\top P, \tag{2.22}$$

and substitute this back into (2.21a) to obtain the IVP

$$\dot{P} = PBR^{-1}B^\top P - A^\top P - PA - Q, \quad P(t_f) = Q_f, \tag{2.23}$$

which P must solve.

Remark 2.1.1. *While the optimal control depends on the choice of x_0, neither K nor P do.*

Equation (2.23) is known as the differential Riccati equation (DRE). A proof of the unique existence of a P on $[t_0, t_f]$ that solves the DRE is contained in [16, Section 6.1]. The results of the infinite-horizon LQR theory which we summarize in the following section are obtained by treating t_f as an increasing parameter. The idea is to consider the limit of a sequence of P's that solve a corresponding sequence of DREs which come from the sequence of finite-horizon LQR problems with increasing t_f's. This material is covered in detail in [16, Section 6.2], and we do not discuss it here.

2.2 Infinite-horizon problem

In this section, we formulate the infinite-horizon LQR problem in continuous time. We keep the definitions of x, u, A, B, Q, and R from the beginning of the section, and we assume u is piecewise continuous on $[t_0, \infty]$ so that the state stays bounded on this interval. We define

$$J^* := \min_u \int_0^\infty x^\top Q x + u^\top R u \, \mathrm{d}t. \tag{2.24}$$

It is necessary to introduce an assumption to guarantee that J^* is finite. Some definitions follow. We define stabilizability, detectability, and observability for pairs of matrices using the well-known Popov-Belevitch-Hautus (PBH) tests, which we state as they appear in [12].

Definition 2.2.1. *We say the pair (A, B) is stabilizable when*

$$\operatorname{rank}\left(\begin{bmatrix} A - \lambda I & B \end{bmatrix}\right) = n \tag{2.25}$$

for every eigenvalue λ of A with nonnegative real part.

Definition 2.2.2. *We say the pair (A, C) is detectable when*

$$\operatorname{rank}\left(\begin{bmatrix} A - \lambda I \\ C \end{bmatrix}\right) = n \tag{2.26}$$

for every eigenvalue λ of A with nonnegative real part.

Definition 2.2.3. *We say the pair (A, C) is observable when*

$$\operatorname{rank}\left(\begin{bmatrix} A - \lambda I \\ C \end{bmatrix}\right) = n \tag{2.27}$$

for all $\lambda \in \mathbb{C}$.

Assuming that (A, B) is stabilizable guarantees that J^* is finite and exists uniquely. It is well known (see for instance [28, Lemma 10.11]) that the minimizer is $u = Kx$, where

$$K := -R^{-1} B^\top P, \tag{2.28}$$

and P is the unique smallest positive semidefinite matrix that satisfies the algebraic Riccati equation (ARE)

$$A^\top P + P A - P B R^{-1} B^\top P + Q = 0. \tag{2.29}$$

This equation is the infinite-horizon counterpart of the DRE (2.23) we encountered in the previous section. It is of central importance in the infinite-horizon LQR theory, not least because the optimal gain K may be computed by solving it. However, before we discuss this, we introduce and discuss the concept of stability.

It is often desirable to ensure that the closed-loop system is stable, meaning the state, under the influence of the optimal gain, approaches the origin as time passes. We define stability as follows.

Definition 2.2.4. *We say a square matrix M is stable, or Hurwitz, when every eigenvalue of M has negative real part.*

The stabilizability assumption on (A, B) alone is insufficient to guarantee that the closed-loop system is stable, and the optimal gain may not be stabilizing even though the associated J^* is finite. Below, we give another useful definition of stabilizability, equivalent to the previous definition, which involves stability.

Definition 2.2.5. *We say the pair (A, B) is stabilizable when there exists an $n \times m$ matrix K such that $A + BK$ is Hurwitz.*

The stability of the closed-loop system $A + BK$ under the influence of the optimal gain is related to the matrix Q. Since Q is positive semidefinite, it is possible to write $Q = C^\top C$, where $C \in \mathbb{R}^{p \times n}$ and $p \leq n$. For the results of Chapters 3 and 4, we require $A + BK$ to be Hurwitz, so the closed-loop system with the optimal control is globally exponentially stable. The stability of the closed-loop system is a standard requirement for the LQR problem to be solvable (see for example [16, Section 6.2]). To guarantee this, it suffices to require every eigenvalue of A on the imaginary axis to be (A, C)-observable [28, Lemma 10.14]. Proposition 4.2.1 requires additionally that (A, C) be detectable. If we require that (A, C) be detectable, then not only is $A + BK$ stable, the solution of the ARE is unique. Stabilizability and stability are the weakest conditions commonly considered in the formulation of the infinite-horizon LQR theory.

Sometimes, the detectability assumption is strengthened to observability. Requiring (A, C) to be observable guarantees, in addition to the above, that P is positive definite. Many more details about all of the aforementioned are given in [16, Section 6.2]. Assuming observability of (A, C) has the advantage that it allows the J^* associated with the optimal input to be used as a Lyapunov function for x. The algorithms we discuss in Sections 2.3 and 3.1 require the observability of (A, C). For the remainder of this thesis, we assume that every eigenvalue of A on the imaginary axis is (A, C)-observable, and for Proposition 4.2.1, we assume that (A, C) is detectable.

While the optimal gain may be computed by directly solving the ARE, this is not the only way to solve the LQR problem. Next, we introduce an iterative method for computing the optimal gain that is useful when a stabilizing gain is already known.

2.3 Kleinman's algorithm

This is a policy iteration algorithm for computing solutions to the ARE first introduced in [15]. It assumes (A, B) is stabilizable, and (A, C) is observable, and its proof requires P to be positive definite and K to stabilize the system. We summarize the algorithm in the following proposition.

Proposition 2.3.1. *Let K_0 be such that $A + BK_0$ is stable. If we recursively compute sequences $\{K_{k+1}\}$ and $\{P_k\}$ from the system*

$$\begin{aligned} K_{k+1} &= -R^{-1}B^\top P_k, \\ 0 &= (A + BK_k)^\top P_k + P_k(A + BK_k) + Q + K_k^\top R K_k, \end{aligned} \tag{2.30}$$

then $\{P_k\}$ is monotonically decreasing, every K_{k+1} is such that $A + BK_{k+1}$ is stable, and

$$\lim_{k\to\infty} (P_k, K_{k+1}) = (P, K), \tag{2.31}$$

where P, K solve the ARE (2.29).

Remark 2.3.1. *The system (2.30) is affine in P. Kleinman's algorithm effectively allows us to compute the solution to the ARE (a quadratic equation in P) by solving a sequence of affine equations.*

Kleinman's algorithm forms the basis for many data-driven LQR algorithms, one of which we discuss in the first section of the next chapter.

2.4 Lyapunov equations

This section covers a small result used in the proof of Theorem 3.5.1.

Proposition 2.4.1. *A matrix P satisfying*

$$A^\top P + PA = Z, \tag{2.32}$$

where A, P, $Z \in \mathbb{R}^{n\times n}$, exists uniquely if A is Hurwitz.

Equations of the form (2.32) are commonly encountered in LQR theory and are known as Lyapunov equations. The ARE is a notable example. Before we give a proof, we introduce some notation.

Let $\otimes$ denote the Kronecker product and $\operatorname{vec}(A)$ denote the column vector formed by vertically stacking the columns of a matrix A. The Kronecker product satisfies the well-known identity

$$\operatorname{vec}(XYZ) = \left(Z^\top \otimes X\right) \operatorname{vec}(Y), \tag{2.33}$$

where X, Y, and Z are appropriately sized matrices. We make use of it below.

Proof. Applying the identity (2.33), we can write (2.32) as

$$\left(I_n \otimes A^\top + A^\top \otimes I_n\right) \operatorname{vec}(P) = \operatorname{vec}(Z), \tag{2.34}$$

where I_n is the $n \times n$ identity matrix. When $\left(I_n \otimes A^\top + A^\top \otimes I_n\right)$ is invertible, $\operatorname{vec}(P)$ exists uniquely. This matrix is known as the Kronecker sum, and its set of eigenvalues is the set of pairwise sums of the eigenvalues of $A^\top$ [13, Theorem 4.4.5]. If A is Hurwitz, then all of its eigenvalues have negative real part, so no pair of them are additive inverses, and P exists uniquely. □

We will again refer to the Kronecker product and the above identity in Section 3.1.

Chapter 3

LQR with interval data from unknown systems

In the previous chapter, we formulated the infinite horizon LQR problem and gave some methods by which the optimal gain K can be computed from the matrices A, B, Q, and R. In this chapter, we consider the situation where the system matrices A and B are unavailable, and instead we are only able to observe the values taken by $x(t)$ and $u(t)$ on q intervals

$$[t_0^1, t_f^1], \cdots, [t_0^q, t_f^q], \tag{3.1}$$

where we require $t_f^i > t_0^i$ for all $i \in \{1, \cdots, q\}$.

This chapter is organized as follows. In Section 3.1, we present a representative of the class of direct LQR algorithms. In Section 3.2, we cover a simple method for computing A and B from interval data, which together with the results covered in Chapter 2 comprise an indirect LQR algorithm. We do some numerical experiments to compare these algorithms in Section 3.3, and in Section 3.4, we set up a theoretical framework . Lastly, in Section 3.5 we state and prove the main theorem of the chapter before making some remarks in Section 3.6.

3.1 LQR directly from interval data

In this section, we introduce the adaptive dynamic programming (ADP) algorithm as it is described in [14, Chapter 2]. This algorithm is a modification of Kleinman's algorithm where the system matrices A and B in (2.30) are replaced by some matrices of measured data. We restate the main result and proof from [14] below.

Theorem 3.1.1. *Assume that (A, B) is detectable and (A, C) is observable. Let K_0 be a constant $n \times n$ matrix such that $A + BK_0$ is stable, and let u be an $\mathbb{R}^m$-valued function continuous on every interval $[t_0^i, t_f^i]$ for $1 \leq i \leq q$ which keeps the state bounded on these intervals. If we recursively compute sequences $\{K_{k+1}\}$ and $\{P_k\}$ from*

$$\Theta_k \begin{bmatrix} \operatorname{vec}(P_k) \\ \operatorname{vec}(K_{k+1}) \end{bmatrix} = \Xi_k \tag{3.2}$$

where

$$\begin{aligned}
\Theta_k &= \begin{bmatrix} X_+^{\otimes} & -2\left(X_-^{\otimes}\left(I_{n\times n}\otimes K_k^\top R\right) + U_-^{\otimes}\left(I_{n\times n}\otimes R\right)\right)\end{bmatrix}, \\
\Xi_k &= -X_-^{\otimes}\operatorname{vec}\left(Q + K_k^\top R K_k\right), \\
X_+^{\otimes} &= \begin{bmatrix}\left(x(t_f^1)\otimes x(t_f^1) - x(t_0^1)\otimes x(t_0^1)\right) & \cdots & \left(x(t_f^q)\otimes x(t_f^q) - x(t_0^q)\otimes x(t_0^q)\right)\end{bmatrix}^\top, \\
X_-^{\otimes} &= \begin{bmatrix}\int_{t_0^1}^{t_f^1} x(t)\otimes x(t)\,\mathrm{d}t & \cdots & \int_{t_0^q}^{t_f^q} x(t)\otimes x(t)\,\mathrm{d}t\end{bmatrix}, \\
U_-^{\otimes} &= \begin{bmatrix}\int_{t_0^1}^{t_f^1} x(t)\otimes u(t)\,\mathrm{d}t & \cdots & \int_{t_0^q}^{t_f^q} x(t)\otimes u(t)\,\mathrm{d}t\end{bmatrix},
\end{aligned} \tag{3.3}$$

x solves

$$\dot{x} = Ax + Bu, \tag{3.4}$$

$I_{n\times n}$ is the $n\times n$ identity matrix, and we assume q is such that

$$\operatorname{rank}\left(\begin{bmatrix} X_-^{\otimes} & U_-^{\otimes}\end{bmatrix}\right) = \frac{n(n+1)}{2} + mn, \tag{3.5}$$

then $\{P_k\}$ is monotonically decreasing, every K_{k+1} stabilizes (A,B), and

$$\lim_{k\to\infty}(P_k, K_{k+1}) = (P, K), \tag{3.6}$$

where P and K solve the ARE (2.29).

Below, we summarize the proof from [14, Sections 2.2 and 2.3].

Proof. First rearrange (2.30) into

$$0 = A^\top P_k + P_k A + K_k^\top B^\top P_k + P_k B K_k + Q + K_k^\top R K_k, \tag{3.7}$$

then substitute $B^\top P_k = -RK_{k+1}$ and note that $K_k^\top R K_{k+1}$ is symmetric to obtain

$$0 = A^\top P_k + P_k A - 2K_k^\top R K_{k+1} + Q + K_k^\top R K_k. \tag{3.8}$$

Now, the fundamental theorem of calculus gives

$$x^\top(t_f)P_k x(t_f) - x^\top(t_0)P_k x(t_0) = \int_{t_0}^{t_f}\frac{\mathrm{d}}{\mathrm{d}t}\left(x^\top P_k x\right)\mathrm{d}t, \tag{3.9}$$

where we compute

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\left(x^\top P_k x\right) &= \left(Ax+Bu\right)^\top P_k x + x^\top P_k\left(Ax+Bu\right) \\
&= x^\top\left(A^\top P_k + P_k A\right)x + 2u^\top B^\top P_k x \\
&= x^\top\left(2K_k^\top R K_{k+1} - Q - K_k^\top R K_k\right)x + 2u^\top B^\top P_k x \\
&= 2\left(x^\top K_{k+1}^\top + u^\top\right)R K_{k+1} x - x^\top\left(Q + K_k^\top R K_k\right)x
\end{aligned} \tag{3.10}$$

by substituting (3.8) and noting that P_k is constant in time and $u_0^\top B^\top P_k x$ is scalar. All that remains is to apply the identity (2.33) to (3.9) to obtain each row of the matrix equation (3.2). As the preceding computations make clear, the iterates P_k and K_{k+1} generated in ADP are identical to those generated in Kleinman's algorithm, so they share the same respective monotonicity and stabilizing properties, and converge to the same limits.

□

ADP is a method of doing Kleinman-type iteration which solves LQR problems without knowledge of A or B, provided an initial stabilizing gain is available. We now turn our attention to system identification, a technique for computing A and B from the data, enabling K to be found by directly solving the ARE.

3.2 LQR via system identification

Supposing as before that the full input and state over some time interval $[t_0^i, t_f^i]$ are exactly known, we can write the IVP (2.1) with initial time t_0^i in integral form as

$$x(t_f^i) - x(t_0^i) = \int_{t_0^i}^{t_f^i} Ax(s) + Bu(s)ds = \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} \int_{t_0^i}^{t_f^i} x(s)ds \\ \int_{t_0^i}^{t_f^i} u(s)ds \end{bmatrix}. \tag{3.11}$$

If we have input and state data on the q intervals (3.1), we can define the matrices

$$D_+^\top := \begin{bmatrix} x(t_f^1) - x(t_0^1) & \cdots & x(t_f^q) - x(t_0^q) \end{bmatrix}, \tag{3.12a}$$

$$D_-^\top := \begin{bmatrix} \int_{t_0^1}^{t_f^1} x(s)ds & \cdots & \int_{t_0^q}^{t_f^q} x(s)ds \\ \int_{t_0^1}^{t_f^1} u(s)ds & \cdots & \int_{t_0^q}^{t_f^q} u(s)ds \end{bmatrix}, \tag{3.12b}$$

$$S^\top := \begin{bmatrix} A & B \end{bmatrix}, \tag{3.12c}$$

and combine equation (3.11) for the q intervals into the system

$$D_+ = D_- S. \tag{3.13}$$

Remark 3.2.1. *Given measurements of x and u on the q intervals in (3.1), the integrals in the entries of D_- will be fully determined (and may be easily computed), even if the integrals themselves are not measured directly. It is on this basis that we consider D_+ and D_- to be known. We will consider approximations to these integrals constructed from sampled measurements of x and u using quadrature in Chapter 4.*

When

$$\operatorname{rank}(D_-) = m + n \tag{3.14}$$

holds, we can uniquely recover the true system from the data by solving system (3.13). Having identified the system matrices A and B, we can then solve the LQR problem by

solving the ARE. This is a very classical way of computing optimal controllers from data. However, as we saw in the previous section, explicitly identifying an unknown system from the data it generates is not necessary in order to compute the optimal gain. In the next section, we compare ADP and system identification in a series of numerical experiments. These experiments illustrate the limitations of directly comparing the data required by different algorithms and motivate the development of the data informativity framework in Section 3.4.

3.3 Comparing data requirements of ADP and system identification

We are interested in whether there exist sets of input and state measurements which are sufficient for one but not the other algorithm to solve the LQR problem. In general, given a suitable input and a large enough number of randomly chosen sampling intervals, both algorithms will be able to solve the LQR problem. On the other hand, there exist inputs and numbers of sampling intervals which are insufficient for both algorithms. Any situations in which ADP is more data-efficient than system identification could justify the greater complexity of the former algorithm. The data matrices $X_+^{\otimes}$, $X_-^{\otimes}$, and $U_-^{\otimes}$ from Section 3.1 are different sizes from the data matrices D_+ and D_- from Section 3.2 and have different entries, so rank conditions (3.5) and (3.14) cannot be directly compared.

What comparisons can we make? Each set of data matrices depends on definite integrals of x and u which, under our assumptions, are continuous and piecewise continuous functions of time, respectively. In order to exactly determine these functions (and therefore their integrals) on the intervals $[t_0^i, t_f^i]$, it is necessary and sufficient to measure the values x and u take on a dense subset of each interval. In practice, this is no easier than measuring values on entire intervals, so we abandon the idea of comparing cardinalities of subsets of intervals $[t_0^i, t_f^i]$. Furthermore, obtaining data for a single interval $[t_0^1, t_f^q]$ will be insufficient for both algorithms even when $m = n = 1$, while the same may not be true if this interval is broken into many smaller intervals $[t_0^1, t_f^1], \cdots, [t_0^q, t_f^q]$. For this reason, we also reject the idea of comparing the total length of the intervals on which data are collected. Instead, in the numerical experiments that follow, we fix the length of each sampling interval and compare the number of intervals required to satisfy the rank conditions of ADP and system identification. The choice of the input, and the number, length, and endpoints of the sampling intervals are made without knowledge of the true system.

3.3.1 Numerical experiments with intervals of equal size

In all of the following examples, we use MATLAB R2022a and `ode45` with `AbsTol` set to `1e-50` to simulate the system over a number of consecutive intervals where $t_f^i = t_0^{i+1}$ for all i. The data from the first i intervals are then assembled into the data matrices required by the algorithms, both of whose linear systems are solved using `mldivide` to compute P and K. Because the data matrices are generated from the same trajectories on the same intervals,

we can meaningfully compare the distances between the P and K thus obtained and those computed using `lqr` directly from A, B, Q, and R and observe how these errors change as i is varied. We treat the matrices returned by `lqr` as exact, so we use the distance

$$\text{error} := ||M - \hat{M}||_{\max} \tag{3.15}$$

between two matrices defined by the matrix max norm. We neglect the error introduced by `ode45` as it simultaneously integrates the ODE and generates the data. For a given algorithm and a given i, if the rank condition is not satisfied, `mldivide` may still be used to obtain a solution. However, the resulting K may not solve the optimal control problem at all. This is not useful for us, so in this case, nothing is computed or plotted.

Example 3.3.1. *We consider the system*

$$A = \begin{bmatrix} -1 & -1 \\ 1 & -2 \end{bmatrix}, \qquad B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \tag{3.16}$$

with

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad R = \begin{bmatrix} 1 \end{bmatrix}. \tag{3.17}$$

Using `lqr` we obtain

$$\begin{aligned} \texttt{P} &= \begin{bmatrix} \texttt{0.385988950543435} & \texttt{-0.06129939606183} \\ \texttt{-0.06129939606183} & \texttt{0.269783312208007} \end{bmatrix}, \\ \texttt{K} &= \begin{bmatrix} \texttt{0.324689554481599} & \texttt{0.208483916146171} \end{bmatrix}. \end{aligned} \tag{3.18}$$

For the simulation, we choose $t_f^i - t_0^i = 0.1$ for the interval length, $t_0^1 = 0$ for the initial time, $x(t_0^1) = \begin{bmatrix} 0.5 & -0.5 \end{bmatrix}^\top$ for the initial state, $u(t) = \begin{bmatrix} \sin(5t) \end{bmatrix}$ for the input, and integrate over $q = 30$ intervals along a single trajectory, so that $x(t_f^i) = x(t_0^{i+1})$. For ADP, we choose $K_0 = 0$ as the initial stabilizing gain and iterate 20 times, resulting in $||K_{21} - K_{20}||_{\max}$ always less than 10^{-13}. We allow i to vary from 1 to 30 and generate the following plot.

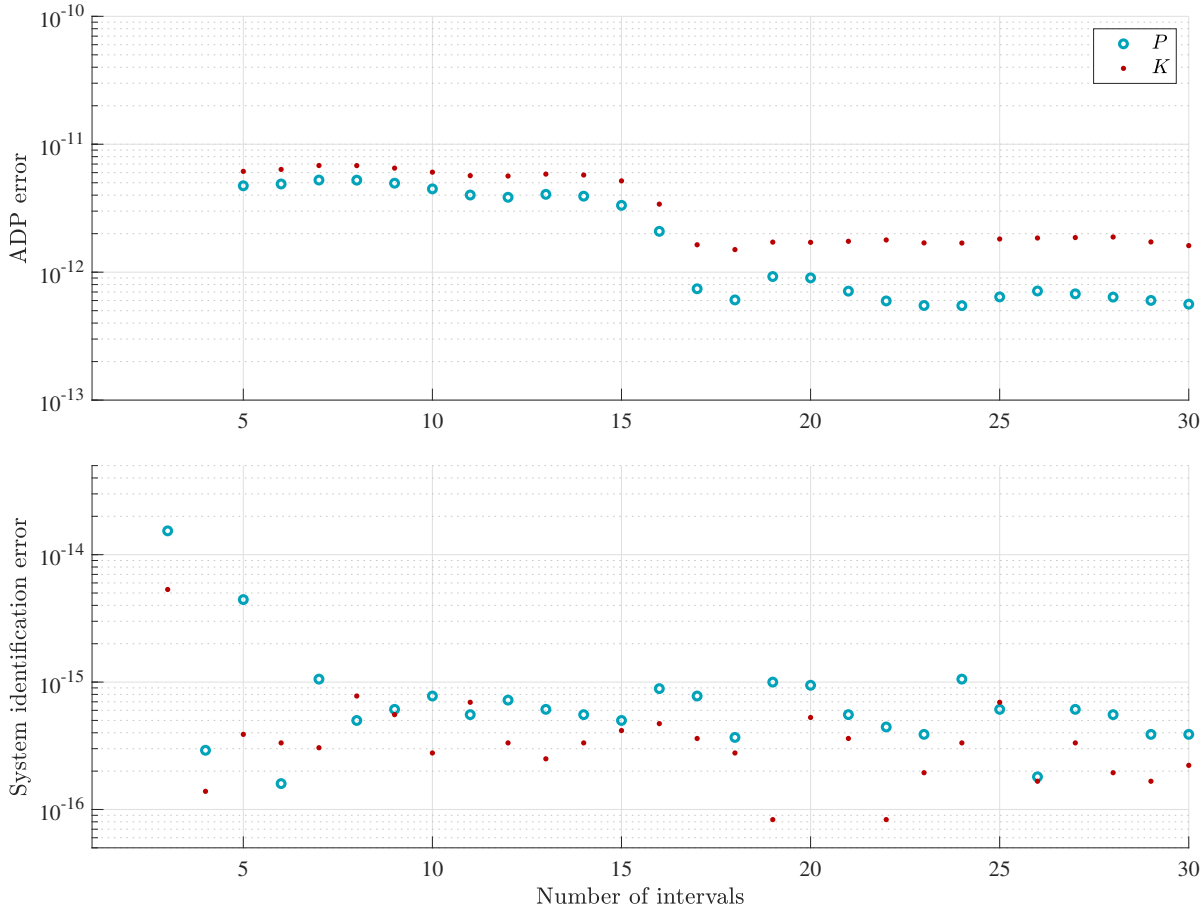


Figure 3.1: Error in K and P for a 2-dimensional system from 1 to 30 intervals of data.

Both algorithms perform well, with ADP resulting in notably larger errors than system identification. Presumably, this is due to the greater numerical error accumulated from the more numerous calculations necessitated by this algorithm's greater complexity. Both algorithms show a modest improvement in accuracy as the number of intervals increases. We attribute this to the averaging effect on numerical errors introduced by `mldivide` when it returns a least-squares solution upon being asked to solve an overdetermined system. The rank condition of ADP is first satisfied when the number of intervals increases to 5, whereas the corresponding number of intervals for system identification is 3. In this case, it appears that system identification sometimes succeeds with data that are too few for ADP, but never the reverse.

Example 3.3.2. *We consider the system*

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -0.1 & -0.5 & -0.7 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \tag{3.19}$$

with

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad R = \begin{bmatrix} 1 \end{bmatrix}. \tag{3.20}$$

Using `lqr` we obtain

$$\begin{aligned} \texttt{P} &= \begin{bmatrix} \texttt{2.355030933091258} & \texttt{2.238452370854355} & \texttt{0.904987562112091} \\ \texttt{2.238452370854355} & \texttt{4.241942496771637} & \texttt{1.893095222031230} \\ \texttt{0.904987562112091} & \texttt{1.893095222031230} & \texttt{1.596995960828505} \end{bmatrix}, \\ \texttt{K} &= \begin{bmatrix} \texttt{0.904987562112090} & \texttt{1.893095222031230} & \texttt{1.596995960828504} \end{bmatrix}. \end{aligned} \tag{3.21}$$

This system is the same as that given in 2.4.1 in [14]. For the simulation, we choose 0.1 for the interval length, 0 for the initial time, $u(t) = \begin{bmatrix} 2\sin(100t) \end{bmatrix}$ for the input, and integrate over $q = 30$ intervals. For each interval, we randomly choose an initial state $x(t_0^i)$ whose entries are drawn from a uniform distribution on $[-5, 5]$. For ADP, we choose $K_0 = 0$ and iterate 20 times, resulting in $||K_{21} - K_{20}||_{\max}$ always less than 10^{-13}. We allow i to vary from 1 to 30 and generate the following plot.

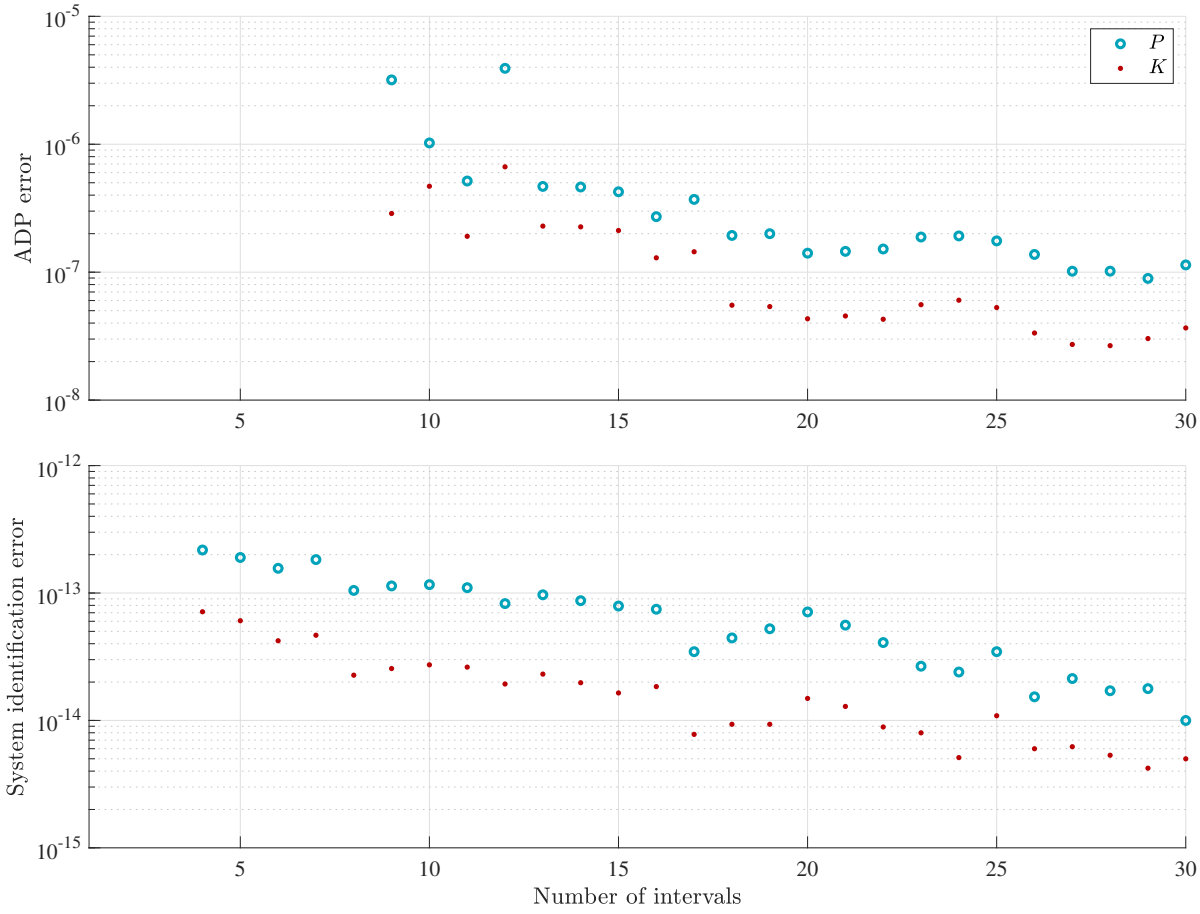


Figure 3.2: Error in K and P for a 3-dimensional system from 1 to 30 intervals of data.

The errors in the results from ADP are about four orders of magnitude greater than in the previous example, whereas system identification worsened only by about one order magnitude, so the gap in the accuracy of the two algorithms is greater than with the previous example. The errors from ADP could probably be reduced by trying different interval lengths, initial stabilizing gains, and inputs, but it seems that ADP is not competitive with system identification, which outperforms it considerably. Both algorithms show a similar improvement in error with an increasing number of samples. ADP requires 9 samples while system identification can cope with 4. It appears that system identification is more efficient with data in this example as well.

Example 3.3.3. *We consider the system*

$$A = \begin{bmatrix} 0.0741 & 0.0089 & 0 & 0 & & \\ 101.5873 & -7.2651 & 2.7608 & 2.8068 & 0 & 0 \\ 0.0704 & 0.0085 & -0.0741 & -0.0089 & 0 & 0.0200 \\ 0.0878 & 0.2672 & 0 & -0.3674 & 0.0044 & 0.3962 \\ -1.8414 & 0.0990 & 0 & 0 & -0.0343 & -0.0330 \\ 0 & 0 & 0 & -359.0000 & 187.5364 & -87.0316 \end{bmatrix},$$

$$B = \begin{bmatrix} -0.0042 & 0.0064 \\ -1.0360 & 1.5849 \\ 0.0042 & 0 \\ 0.1261 & 0 \\ 0 & -0.0168 \\ 0 & 0 \end{bmatrix}, \tag{3.22}$$

with

$$Q = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \end{bmatrix}, \qquad R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{3.23}$$

Using `lqr` we obtain

$$\texttt{P} = \begin{bmatrix} \texttt{766.758621} & \texttt{1.40195031} & \texttt{97.3265407} & \texttt{119.093696} & \texttt{-111.25490} & \texttt{0.62649452} \\ \texttt{1.40195031} & \texttt{0.10316092} & \texttt{-0.6581454} & \texttt{3.98928230} & \texttt{-1.8221334} & \texttt{0.01500913} \\ \texttt{97.3265407} & \texttt{-0.6581454} & \texttt{71.6914763} & \texttt{-1.3766798} & \texttt{-29.067836} & \texttt{0.02297498} \\ \texttt{119.093696} & \texttt{3.98928230} & \texttt{-1.3766798} & \texttt{233.810136} & \texttt{-126.09515} & \texttt{-1.1827543} \\ \texttt{-111.25490} & \texttt{-1.8221334} & \texttt{-29.067836} & \texttt{-126.09515} & \texttt{88.0775400} & \texttt{0.58591292} \\ \texttt{0.62649452} & \texttt{0.01500913} & \texttt{0.02297498} & \texttt{-1.1827543} & \texttt{0.58591292} & \texttt{0.56874023} \end{bmatrix},$$

$$\texttt{K} = \begin{bmatrix} \texttt{10.7536798} & \texttt{0.38752137} & \texttt{0.40057213} & \texttt{24.8445862} & \texttt{-13.667683} & \texttt{-0.1672295} \\ \texttt{8.99828865} & \texttt{0.20308407} & \texttt{0.06813470} & \texttt{9.20321182} & \texttt{-5.0796333} & \texttt{0.01795420} \end{bmatrix}. \tag{3.24}$$

This system is the same as that given in 2.4.2 in [14]. For the simulation, we choose $t_f^i - t_0^i = 0.005$ for the interval length, $t_0^1 = 0$ for the initial time,

$$u(t) = \begin{bmatrix} \sum_{j=1}^{100} \sin(\omega_j t) & \sum_{\ell=1}^{100} \sin(\omega_\ell t) \end{bmatrix}^\top$$

where each ω_j and ω_ℓ is drawn from a uniform distribution on $[-50, 50]$ for the input, and integrate over $q = 70$ intervals. For each interval, we randomly choose an initial state $x(t_0^i)$ whose entries are drawn from a uniform distribution on $[-0.5, 0.5]$. For ADP, we choose $K_0 = 0$ and iterate 30 times, resulting in $||K_{31} - K_{30}||_{\max}$ always less than 10^{-10}. We allow i to vary from 1 to 70 and generate the following plot.

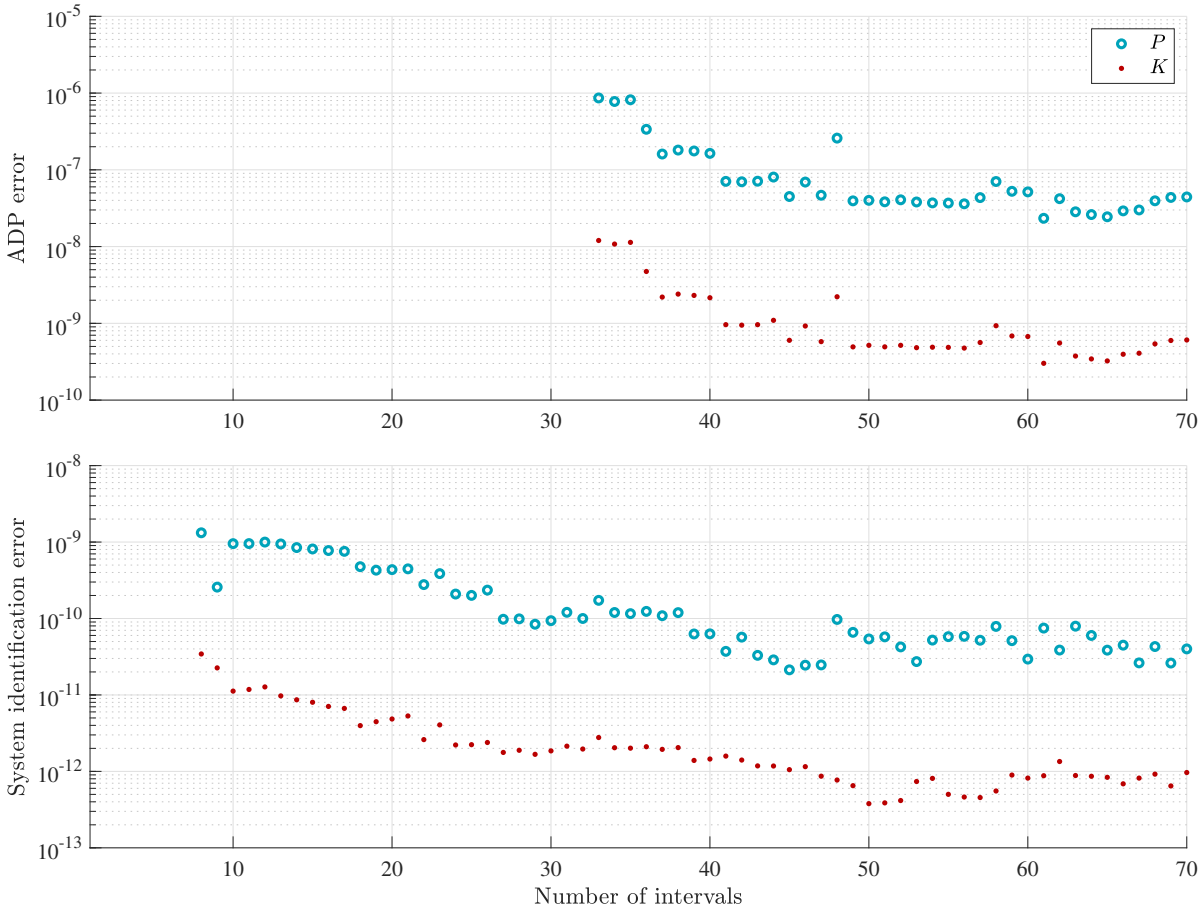


Figure 3.3: Error in K and P for a 6-dimensional system from 1 to 70 intervals of data.

The relationship between the errors and the number of intervals is broadly similar to that which we have observed in the previous examples. The gap in errors between ADP and system identification is smaller for this example than for the 3-dimensional system. The data requirement of ADP is first satisfied with 33 samples, and that of system identification with 8. It appears that the gap in data efficiency in favor of system identification increases with increasing system size, which is unsurprising since the rank condition for system identification is $O(m+n)$, compared with $O(mn)$ for ADP.

3.3.2 An example with intervals of unequal size

Our numerical experiments so far suggest strongly that ADP is plainly less data-efficient than system identification. In this subsection, we give an example with intervals of unequal size that casts doubt on this conclusion. We return to the 2-dimensional system of Example 3.3.1, but instead of integrating over intervals of fixed length and varying their number, we fix the number of intervals at 5, the initial time $t_0^1 = 0$, and vary the interval lengths. We keep $t_f^i = t_0^{i+1}$ and integrate along the a single trajectory. The input and K_0 remain

unchanged from Example 3.3.1, but we change the initial condition to $x(t_0^1) = \begin{bmatrix} 0.01 & 0.01 \end{bmatrix}^\top$. We then vary the t_f^i in a particular way that highlights the dependence of the ranks of the matrices $\begin{bmatrix} X_-^\otimes & U_-^\otimes \end{bmatrix}$ and D_- on the lengths of the intervals. To motivate our selection of the t_f^i, we plot the trajectory of the state, the input, and the entries of matrices

$$\begin{bmatrix} \int_{t_0^1}^t x_1(s)x_1(s)\,\mathrm{d}s & \int_{t_0^1}^t x_1(s)x_2(s)\,\mathrm{d}s & \int_{t_0^1}^t x_2(s)x_1(s)\,\mathrm{d}s & \int_{t_0^1}^t x_2(s)x_2(s)\,\mathrm{d}s \end{bmatrix},$$
$$\begin{bmatrix} \int_{t_0^1}^t x_1(s)u(s)\,\mathrm{d}s & \int_{t_0^1}^t x_2(s)u(s)\,\mathrm{d}s \end{bmatrix},$$
$$\begin{bmatrix} \int_{t_0^1}^t x(s)\,\mathrm{d}s \end{bmatrix},$$
$$\begin{bmatrix} \int_{t_0^1}^t u(s)\,\mathrm{d}s \end{bmatrix}, \tag{3.25}$$

as functions of t on $[0, 4]$ in Figure 3.4 on the next page. From the plot, we observe that the ratios between the entries of these matrices vary with time. For example, the quotient

$$\frac{\int_{t_0^1}^t x_2(s)\,\mathrm{d}s}{\int_{t_0^1}^t x_1(s)\,\mathrm{d}s} \tag{3.26}$$

is higher near $t = 1.5$ or $t = 2.75$ than it is near $t = 2.2$ or $t = 3.4$. If we choose the endpoints of our intervals so that this quotient has the same value at each endpoint, then the columns of D_- will be linearly dependent. We can select endpoints satisfying

$$\frac{\int_{t_0^1}^{t_f^i} x_2(s)\,\mathrm{d}s}{\int_{t_0^1}^{t_f^i} x_1(s)\,\mathrm{d}s} = 2, \quad i \in \{1, 2, 3, 4, 5\} \tag{3.27}$$

in order to force D_- to be rank deficient, and similarly select endpoints satisfying

$$\frac{\int_{t_0^1}^{t_f^i} x_1(s)x_1(s)\,\mathrm{d}s}{\int_{t_0^1}^{t_f^i} x_1(s)x_2(s)\,\mathrm{d}s} = 0.9, \quad i \in \{1, 2, 3, 4, 5\} \tag{3.28}$$

in order to force $\begin{bmatrix} X_-^\otimes & U_-^\otimes \end{bmatrix}$ to be rank deficient. We plot these points as well as the aforementioned quotients as they vary over time in Figure 3.5.

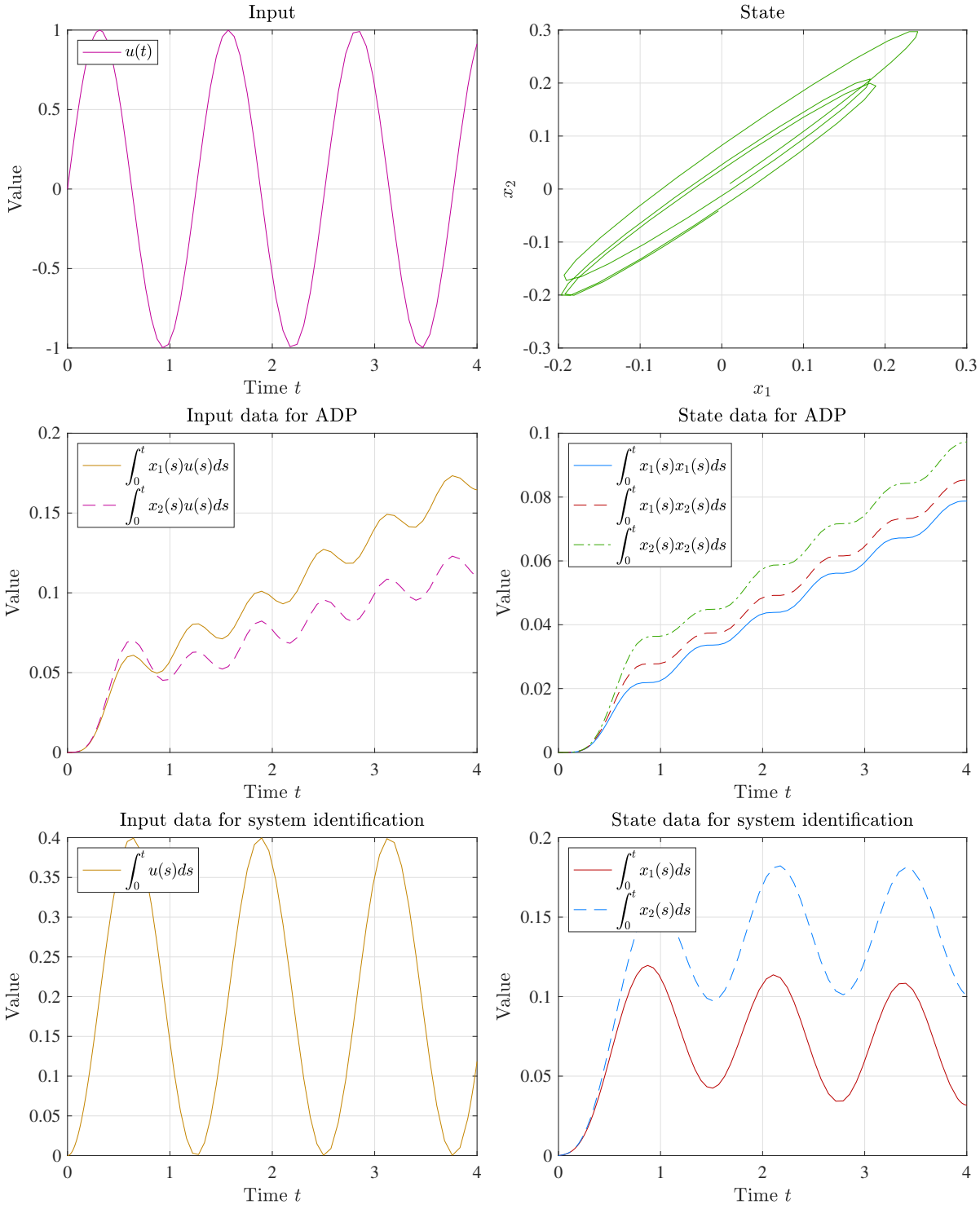


Figure 3.4: Data for ADP and system identification over time for a 2-dimensional system.

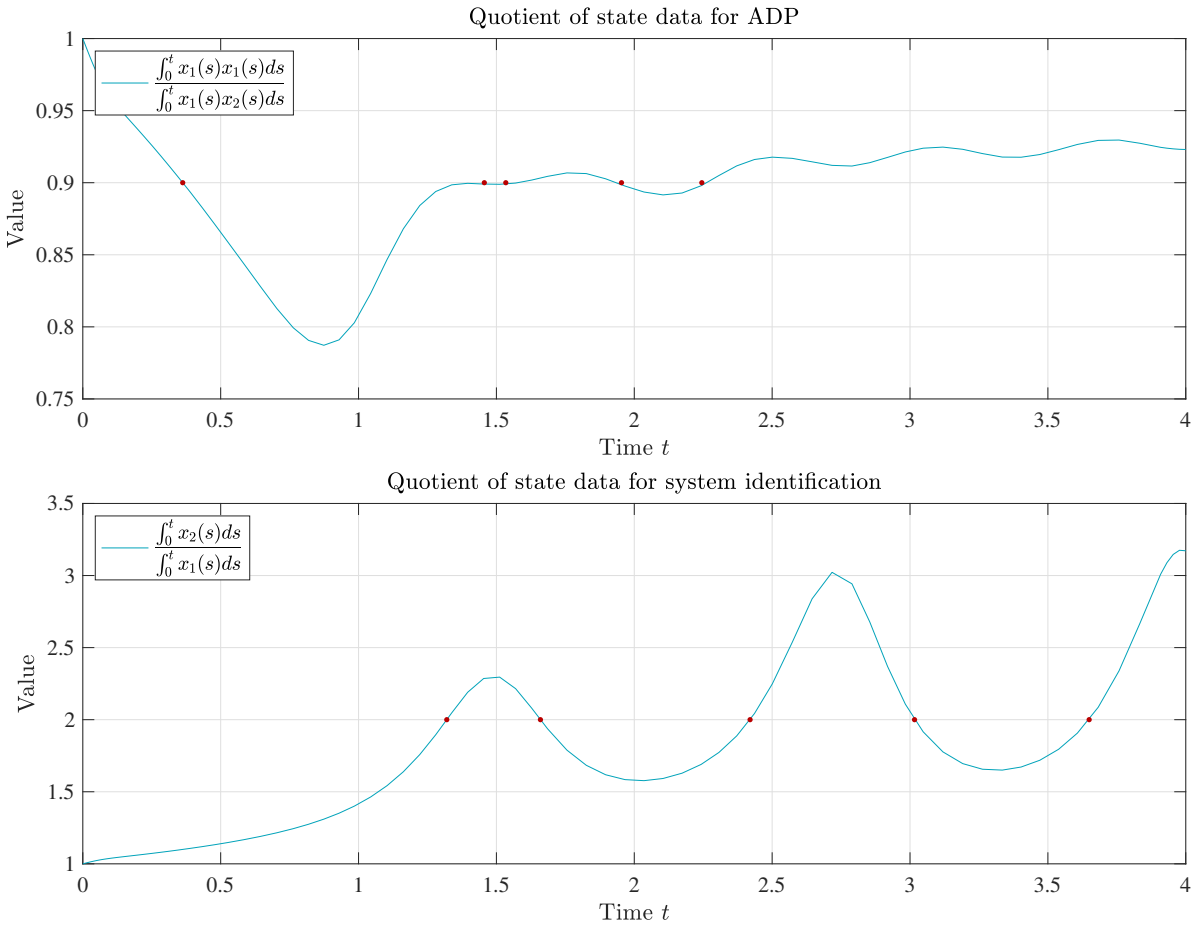


Figure 3.5: "Pathological" sampling interval endpoints for ADP and system identification.

The endpoints in the first subplot of the figure above are

$$(\mathtt{t_f^1}, \mathtt{t_f^2}, \mathtt{t_f^3}, \mathtt{t_f^4}, \mathtt{t_f^5}) = \tag{3.29}$$
$$(\mathtt{0.36235857765,\ 1.45617638255,\ 1.53404368934,\ 1.95361011479,\ 2.24496668603}),$$

and those in the second subplot are

$$(\mathtt{t_f^1}, \mathtt{t_f^2}, \mathtt{t_f^3}, \mathtt{t_f^4}, \mathtt{t_f^5}) = \tag{3.30}$$
$$(\mathtt{1.32001917243,\ 1.65964907930,\ 2.41995145884,\ 3.01633485068,\ 3.64965819368}).$$

Using the intervals defined by the endpoints in (3.29), we get

$$\begin{bmatrix} 0.00380596 & 0.00422884 & 0.00422884 & 0.00470051 & 0.02678606 & 0.02960169 \\ 0.02985536 & 0.03317262 & 0.03317262 & 0.04010324 & 0.04504974 & 0.02395866 \\ 0.00003632 & 0.00004035 & 0.00004035 & 0.00005671 & -0.0002035 & -0.0011072 \\ 0.00922823 & 0.01025359 & 0.01025359 & 0.01147989 & 0.02773984 & 0.02778419 \\ 0.00168993 & 0.00187770 & 0.00187770 & 0.00262059 & 0.00190112 & -0.0081537 \end{bmatrix}, \tag{3.31}$$

$$\begin{bmatrix} 0.0303911758 & 0.0128271450 & -0.000294574 & 0.0597236400 & 0.0028307283 \\ 0.0334292359 & 0.0654532923 & -0.001273909 & 0.0654112121 & 0.0154947045 \\ 0.2477341011 & -0.156181920 & 0.0719016850 & 0.2248777861 & -0.233782871 \end{bmatrix}, \tag{3.32}$$

for $\begin{bmatrix} X_-^\otimes & U_-^\otimes \end{bmatrix}$ and $D_-^\top$ respectively. As expected, the first column of $\begin{bmatrix} X_-^\otimes & U_-^\otimes \end{bmatrix}$ is 0.9 times the second column, and we use `cond` and `rank` to compute the condition number and rank of these matrices, yielding

$$\texttt{cond([X}_-^\otimes \ \texttt{U}_-^\otimes \texttt{]) = 1.492336577344e+17}, \quad \texttt{rank([X}_-^\otimes \ \texttt{U}_-^\otimes \texttt{]) = 4}, \tag{3.33}$$

$$\texttt{cond(D_) = 26.278488301511288}, \quad \texttt{rank(D_) = 3}. \tag{3.34}$$

This shows that using the data generated from the single trajectory broken into the five intervals defined by the endpoints in (3.29), the ADP rank condition $n(n+1)/2+mn = 5$ is not satisfied. On the other hand, using the same trajectory and intervals, the optimal gain produced by system identification has error $\approx 1.9 \times 10^{-16}$. If we instead use data generated from the same trajectory on intervals defined by the endpoints in (3.30), we get

$$\begin{bmatrix} 0.03218645 & 0.03583994 & 0.03583994 & 0.04314551 & 0.07925022 & 0.06167295 \\ 0.00256653 & 0.00259593 & 0.00259593 & 0.00265474 & 0.00293267 & 0.00048375 \\ 0.01492406 & 0.01573840 & 0.01573840 & 0.01736705 & 0.04094454 & 0.02927772 \\ 0.01060740 & 0.01097616 & 0.01097616 & 0.01171368 & 0.01922328 & 0.01009090 \\ 0.01156875 & 0.01202052 & 0.01202052 & 0.01292406 & 0.02438074 & 0.01461905 \end{bmatrix}, \tag{3.35}$$

$$\begin{bmatrix} 0.0567014832 & -0.002874064 & 0.0221630681 & -0.013374516 & 0.0152232123 \\ 0.1134029664 & -0.005748128 & 0.0443261362 & -0.026749033 & 0.0304464246 \\ 0.0099594997 & 0.2759991900 & -0.264578032 & 0.3406610391 & -0.326965789 \end{bmatrix}, \tag{3.36}$$

for $\begin{bmatrix} X_-^\otimes & U_-^\otimes \end{bmatrix}$ and $D_-^\top$ respectively. As expected, the second row of $D_-^\top$ is twice the first row, and we compute

$$\texttt{cond([X}_-^\otimes \ \texttt{U}_-^\otimes \texttt{]) = 7.884559620066274e+09}, \quad \texttt{rank([X}_-^\otimes \ \texttt{U}_-^\otimes \texttt{]) = 5}, \tag{3.37}$$

$$\texttt{cond(D_) = 1.153450412909037e+16}, \quad \texttt{rank(D_) = 2}, \tag{3.38}$$

which shows the system identification rank condition $m + n = 3$ is not satisfied. Using the same trajectory and intervals, ADP produces an optimal gain with error $\approx 6.8 \times 10^{-5}$.

In this example, we have demonstrated that the rank conditions for ADP and system identification may be satisfied or not satisfied when the number of data intervals is fixed and their lengths are varied. We have done this using the same system and input which, using intervals of equal length, earlier apparently showed system identification to require less data than ADP. With that being said, the data we obtained undoubtedly contained numerical errors from integrating the ODE and solving linear systems, the interval endpoints we chose were inexact, and the resulting matrices we computed were not perfectly singular. Although we believe the accuracy of our methods is sufficient to show the existence of endpoints that make the data sufficient for one algorithm but insufficient for the other, we are hesitant to make absolute statements about the relative data requirements of ADP and system identification. Nevertheless, it is clear that the rank conditions of both algorithms depend strongly on the interval endpoints.

In this section, we have presented some numerical experiments aimed at understanding the differences in the amount of data required to solve LQR problems via a direct algorithm (ADP) and an indirect algorithm (system identification). In the case where intervals are of equal length, for our 2-dimensional, 3-dimensional, and 6-dimensional examples, system identification appears to be more accurate and to require less data than ADP. This gives the impression that ADP tends to be less efficient with data than system identification, making the hope that ADP may be used to solve LQR problems using data insufficient for system identification seem remote.

However, by using the same number of intervals of varying lengths, we showed that each algorithm can be made to satisfy its rank condition when the other fails simply by careful (or pathological) choice of the interval endpoints. It seems somewhat counterintuitive that the success or failure of either algorithm should be so sensitive to integration endpoints. This is especially interesting since the process we used to choose our intervals made use only of input and state measurements, and did not depend on any *a priori* knowledge of the unknown system. It is therefore conceivable that in practical settings, both algorithms' data efficiencies could benefit from more deliberate choice of data collection intervals than defaulting to intervals of equal length. This observation makes it difficult to make any confident statements about which algorithm may be more data efficient than the other in practice or in theory, in spite of any apparent advantage enjoyed by system identification.

Besides, we have only considered two particular algorithms as representatives of direct and indirect algorithms for solving the LQR problem from data. As such, the results of our numerical experiments provide limited insight into the comparative data efficiencies of the two approaches in general. It remains to be seen whether there exist any direct algorithms that can consistently solve LQR problems from data insufficient for system identification. To investigate this question, it is necessary to take a more theoretical approach which is independent of the details of any particular algorithm. In the next section, we do this by extending the data informativity framework introduced in [30] to our present continuous time setting.

3.4 Informativity of interval data for LQR

In Section 3.2, we considered the case where D_- defined in (3.12) has full rank. We now consider the case where it is is rank-deficient. In this case the linear equation (3.13) has infinitely many solutions. We can then define sets

$$\mathcal{S}_D := \{S : D_+ = D_- S\}, \tag{3.39}$$

$$\mathcal{A}_D := \left\{A : \begin{bmatrix} A & B \end{bmatrix}^\top \in \mathcal{S}_D \text{ for some } B\right\} \tag{3.40}$$

of systems that are compatible with the data. It is convenient to additionally define the set

$$\mathcal{S}_D^0 := \{S_0 : 0 = D_- S_0\} \tag{3.41}$$

of matrices that solve the corresponding homogeneous system. From now on, we denote the system matrices of S_0 by A_0 and B_0, the true system that generated the data by $\hat{S}$, and its system matrices by $\hat{A}$ and $\hat{B}$. When system identification is possible from the data, $\mathcal{S}_D$ is a singleton containing only the true system. When system identification is impossible, $\mathcal{S}_D$ contains in addition to the true system infinitely many systems between which the data cannot distinguish. The main idea of the data informativity framework is to investigate which properties relevant to control may be attributed to all members of a given $\mathcal{S}_D$. By doing so, we can extract some useful information about the true system even in situations where it cannot be identified.

For a particular choice of m and n, the set of linear systems $\begin{bmatrix} A & B \end{bmatrix}^\top$ is the set $\mathbb{R}^{n(m+n)}$. We define the subset

$$\mathcal{S}_{\text{stab}} := \left\{ \begin{bmatrix} A & B \end{bmatrix}^\top \text{stabilizable} \right\} \tag{3.42}$$

of $\mathbb{R}^{n(m+n)}$ containing all systems which are stabilizable according to Definition 2.2.1. For every fixed system and Q and R satisfying our conditions from Chapter 2, the uniqueness of the corresponding P and K established in Section 2.2 makes it possible to define functions

$$p_{Q,R} : \mathcal{S}_{\text{stab}} \longrightarrow \mathbb{R}^{n\times n} \tag{3.43}$$

$$\begin{bmatrix} A & B \end{bmatrix}^\top \longmapsto P \quad \text{where } P \text{ solves (2.29)}, \tag{3.44}$$

$$k_{Q,R} : \mathcal{S}_{\text{stab}} \longrightarrow \mathbb{R}^{m\times n} \tag{3.45}$$

$$\begin{bmatrix} A & B \end{bmatrix}^\top \longmapsto K \quad \text{where } K \text{ solves (2.28)}. \tag{3.46}$$

Even when Q and R are fixed, different systems may have the same K. For example, given any stabilizable pair (A, B), every member of the set

$$\{(\alpha A, \alpha B) : \alpha \in \mathbb{R}, \alpha > 0\} \tag{3.47}$$

is also a stabilizable pair, and shares the same optimal K. For this reason, in general we cannot expect $k_{Q,R}$ to be injective. Instead, it is natural to consider level sets.

Definition 3.4.1. *For particular Q, R, P, and K, we refer to the sets*

$$\mathcal{S}_{Q,R,P} := \left\{ \begin{bmatrix} A & B \end{bmatrix}^\top \in \mathcal{S}_{stab} : p_{Q,R}\left(\begin{bmatrix} A & B \end{bmatrix}^\top \right) = P \right\} \tag{3.48}$$

$$\mathcal{S}_{Q,R,K} := \left\{ \begin{bmatrix} A & B \end{bmatrix}^\top \in \mathcal{S}_{stab} : k_{Q,R}\left(\begin{bmatrix} A & B \end{bmatrix}^\top \right) = K \right\} \tag{3.49}$$

as level sets of $p_{Q,R}$ and $k_{Q,R}$ respectively.

We are finally ready to state the definition of informativity for LQR for continuous-time systems. This definition extends the notion of informativity for LQR given in [30, Definition 24] to the continuous-time setting.

Definition 3.4.2. *We say the data are informative for LQR when*

$$\mathcal{S}_D \subset \left\{ \begin{bmatrix} A & B \end{bmatrix}^\top \in \mathcal{S}_{Q,R,K} : A + BK \text{ is Hurwitz} \right\} \tag{3.50}$$

holds for some $K \in \mathbb{R}^{n\times m}$. In this case, $u = Kx$ is a stabilizing optimal control for every $\begin{bmatrix} A & B \end{bmatrix}^\top \in \mathcal{S}_D$.

Remark 3.4.1. *Since $\mathcal{S}_{Q,R,K} \subset \mathcal{S}_{stab}$, Definition 3.4.2 implies that every system compatible with the data is stabilizable.*

Remark 3.4.2. *No matter what the data, the K in Definition 3.4.2 will be unique if it exists because the minimizer of (2.24) is unique.*

3.5 Direct LQR algorithms require as much data as system identification

We are now ready to state and prove the main theorem of this chapter.

Theorem 3.5.1. *The data are informative for LQR if and only if $\mathcal{S}_D = \left\{ \begin{bmatrix} \hat{A} & \hat{B} \end{bmatrix}^\top \right\}$ or $K = 0$.*

That $\mathcal{S}_D = \left\{ \begin{bmatrix} \hat{A} & \hat{B} \end{bmatrix}^\top \right\}$ or $K = 0$ imply the data are informative for LQR is immediate, so it only remains to prove the converse implication. To this end, we give the following proposition.

Proposition 3.5.1. *If the data are informative for LQR, then $A_0 + B_0 K = 0$ for every $\begin{bmatrix} A_0 & B_0 \end{bmatrix}^\top \in \mathcal{S}_D^0$, where $K = k_{Q,R}\left(\begin{bmatrix} A & B \end{bmatrix}^\top \right)$ for every $\begin{bmatrix} A & B \end{bmatrix}^\top \in \mathcal{S}_D$.*

Proof. For every positive number α, $\begin{bmatrix} A & B \end{bmatrix}^\top \in \mathcal{S}_D$, and $\begin{bmatrix} A_0 & B_0 \end{bmatrix}^\top \in \mathcal{S}_D^0$, we have

$$\begin{bmatrix} A & B \end{bmatrix}^\top + \frac{1}{\alpha} \begin{bmatrix} A_0 & B_0 \end{bmatrix}^\top \in \mathcal{S}_D. \tag{3.51}$$

Defining

$$F := A + BK, \tag{3.52}$$

$$F_0 := A_0 + B_0 K, \tag{3.53}$$

we see that (3.51) implies that the matrices $F + (1/\alpha)F_0$ and $F - (1/\alpha)F_0$ are Hurwitz, since every system in $\mathcal{S}_D$ is stabilized by K. Scaling these matrices by α also scales their eigenvalues by α, and since α is positive, $\alpha F + F_0$ and $\alpha F - F_0$ are also Hurwitz. The eigenvalues of all of the aforementioned depend continuously on α. Consider a sequence $\{\alpha_j\}_{j=1}^{\infty}$ of positive α_j converging to zero. Every element of the sequences $\{\alpha_j F + F_0\}_{j=0}^{\infty}$ and $\{\alpha_j F - F_0\}_{j=0}^{\infty}$ is Hurwitz, so the limits

$$\lim_{j\to\infty} \alpha_j F + F_0 = F_0, \tag{3.54}$$

$$\lim_{j\to\infty} \alpha_j F - F_0 = -F_0 \tag{3.55}$$

must have only eigenvalues with nonpositive real part. We conclude that F_0 has eigenvalues only on the imaginary axis. Next, we observe that for $\begin{bmatrix} A_0 & B_0 \end{bmatrix}^\top \in \mathcal{S}_D^0$, we have $\begin{bmatrix} NA_0 & NB_0 \end{bmatrix}^\top \in \mathcal{S}_D^0$ for every $N \in \mathbb{R}^{n\times n}$, so replacing A_0 and B_0 in (3.53) with NA_0 and NB_0, respectively, our argument above shows that NF_0 has eigenvalues only on the imaginary axis for every N. Setting $N = F_0^\top$ in particular, we conclude that $F_0^\top F_0$ has eigenvalues only on the imaginary axis. Being a symmetric real matrix, its eigenvalues must also be real. Thus, $F_0^\top F_0$ must be nilpotent. The only symmetric real nilpotent matrix is the zero matrix, so we are done. □

Remark 3.5.1. *The proofs of the above lemma and of Lemma 15 in [30] differ in that continuous-time stability of a matrix is related to the real parts of is eigenvalues, whereas discrete-time stability is related to the spectral radius.*

An easy corollary of this proposition follows.

Corollary 3.5.1. *If the data are informative for LQR, then $A + BK = \hat{A} + \hat{B}K$ for every $\begin{bmatrix} A & B \end{bmatrix}^\top \in \mathcal{S}_D$, where $\hat{A} + \hat{B}K$ is the closed-loop true system under input $u = Kx$.*

The following proposition makes use of this corollary to show that data informative for LQR is sufficient for P to be determined uniquely.

Proposition 3.5.2. *If the data are informative for LQR, then $\mathcal{S}_D \subset \mathcal{S}_{Q,R,P}$ for a unique P independent of A and B.*

Proof. Substituting $RK = -B^\top P$ from (2.28) and $\hat{A} + \hat{B}K = A + BK$ from Corollary 3.5.1, the ARE (2.29) becomes

$$0 = \left(\hat{A} + \hat{B}K\right)^\top P + P\left(\hat{A} + \hat{B}K\right) + Q + K^\top R K, \tag{3.56}$$

where K, $\hat{A} + \hat{B}K$, Q, and R are common to all $\begin{bmatrix} A & B \end{bmatrix}^\top \in \mathcal{S}_D$. Since $\hat{A} + \hat{B}K$ is Hurwitz, Proposition 2.4.1 guarantees that the P which solves the Lyapunov equation (3.56) exists uniquely. Since (3.56) holds for all $\begin{bmatrix} A & B \end{bmatrix}^\top \in \mathcal{S}_D$, P is independent of A and B, and the proof is complete. □

Now we are ready to give the proof of Theorem 3.5.1.

Proof. Assuming the data are informative for LQR, we rewrite (2.29) and (2.28) as

$$A^\top P = -P\left(\hat{A} + \hat{B}K\right) - Q, \tag{3.57}$$

$$B^\top P = -RK, \tag{3.58}$$

by a substitution similar to the one performed in (3.56). Proposition 3.5.2 shows that equations (3.57) and (3.58) hold for every $\begin{bmatrix} A & B \end{bmatrix}^\top \in \mathcal{S}_D$ and the right-hand sides of both equations are independent of A and B. Since $\begin{bmatrix} A + A_0 & B + B_0 \end{bmatrix}^\top \in \mathcal{S}_D$ for every $\begin{bmatrix} A_0 & B_0 \end{bmatrix}^\top \in \mathcal{S}_D^0$ and every $\begin{bmatrix} A & B \end{bmatrix}^\top \in \mathcal{S}_D$, (3.57) and (3.58) imply that $A_0^\top P = 0$ and $B_0^\top P = 0$ for every $\begin{bmatrix} A_0 & B_0 \end{bmatrix}^\top \in \mathcal{S}_D^0$. Furthermore, for $\begin{bmatrix} A_0 & B_0 \end{bmatrix}^\top \in \mathcal{S}_D^0$, we also have $\begin{bmatrix} NA_0 & NB_0 \end{bmatrix}^\top \in \mathcal{S}_D^0$ for every $N \in \mathbb{R}^{n\times n}$, giving

$$A_0^\top N^\top P = 0, \tag{3.59}$$

$$B_0^\top N^\top P = 0 \tag{3.60}$$

for every $N \in \mathbb{R}^{n\times n}$. This implies that either $P = 0$ or $\begin{bmatrix} A_0 & B_0 \end{bmatrix} = 0$. To see this, suppose $P \neq 0$, $A_0^\top \neq 0$, and let p_{ij} and $a_{k\ell}$ be nonzero entries of P and $A_0^\top$, respectively. If we choose $N^\top$ such that $n_{\ell i}$ is the only nonzero entry, then the entry in row k and column j of $A_0^\top N^\top P$ will be nonzero. The same argument works with $B_0^\top$ in place of $A_0^\top$. In the case $\begin{bmatrix} A_0 & B_0 \end{bmatrix} = 0$, we get $\mathcal{S}_D = \left\{\hat{S}\right\}$, and $P = 0$ gives $K = 0$. □

Corollary 3.5.2. *The data are informative for LQR if and only if $K = 0$ or D_- has a trivial nullspace (meaning that the only S that solves $0 = D_- S$ is $S = 0$).*

Remark 3.5.2. *When $K = 0$, Proposition 3.5.1 gives $A_0 = 0$, meaning $\mathcal{A}_D = \left\{\hat{A}\right\}$ and the data uniquely identify $\hat{A}$.*

3.6 Remarks

In this section, we have shown that under the typical assumptions for LQR, in order for $\mathcal{S}_D$, an affine subspace of $\mathbb{R}^{n(m+n)}$, to contain only systems optimally stabilized by a single K, it must be a singleton or the optimal gain must be zero. The latter is a pathological case where A is stable, the open-loop system is already optimal, and the cost cannot be improved by any feedback control. If we require $(\hat{A}, C)$ to be observable, P will be positive definite and this case will be eliminated. Aside from the pathological case, every algorithm, direct or indirect, that solves LQR problems using interval data is implicitly doing system identification. This gives a positive answer to the key question posed at the end of Section 1.1. It also strengthens a link between the various conditions of data-driven LQR algorithms and some recent work on persistent excitation [4] and Willems' lemma [18] in continuous time. One

imagines the possibility of using Willems' lemma to investigate whether a given condition on data is strictly necessary (in addition to being sufficient), or whether the algorithm can be made to work with less data by relaxing it.

We now know that the map from input and state data on the q intervals in (3.1) to the optimal gain K defined by any data-driven LQR algorithm has the same domain as (and is identical to) that defined by system identification together with the ARE. In the next section, we use this fact to prove that this map is continuous, which leads to the possibility of attaining arbitrarily close approximations to K by constructing approximations to D_- from sampled data.

Chapter 4

Sampled data approximations

We begin with a simple example that shows the impossibility of using sampled data to exactly and uniquely recover input and state trajectories, or the systems that generated them, even when the samples are spaced arbitrarily closely.

Example 4.0.1. *Let z be an integer and let φ be a positive real number. Let x_z and u_z be the state and input, where x_z solves the IVP*

$$\dot{x}_z(t) = A_z x(t) + B_z u_z(t), \quad x_z(0) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \tag{4.1}$$

and

$$\begin{aligned} A_z &= 2\pi\varphi z \begin{bmatrix} 1 & 0 \\ -2 & 1 \end{bmatrix}, \\ B_z &= 2\pi\varphi z \begin{bmatrix} 1 \\ -1 \end{bmatrix} \end{aligned} \tag{4.2}$$

are the system matrices parameterized by z. Incidentally, (A_z, B_z) is stabilizable for every $z \neq 0$. Applying input $u_z = \begin{bmatrix} 1 & -1 \end{bmatrix} x_z$, the closed-loop system state evolves according to

$$x_z(t) = \begin{bmatrix} \cos(2\pi\varphi z t) \\ -\sin(2\pi\varphi z t) \end{bmatrix}. \tag{4.3}$$

Suppose we choose to sample periodically with frequency φ, so that the state and input are measured at every time $t \in \{\tau/\varphi, \tau \in \mathbb{Z}\}$. Then we will always observe $x_z(t) = x_z(0)$ and $u_z(t) = \begin{bmatrix} 1 & -1 \end{bmatrix} x_z(0)$, for every choice of z. We can make the sample times arbitrarily closely spaced by making φ arbitrarily large, yet we still cannot distinguish between any of the pairs (A_z, B_z) or the trajectories of their states x_z. For a given pair of systems, there always exists a sampling frequency high enough to distinguish them, but there exists no sampling frequency high enough to distinguish between all pairs of systems in $\mathbb{R}^{n(m+n)}$.

While many physical systems of practical interest exist in continuous time, sensors and controllers are often implemented using digital computers running in discrete time. We are

therefore motivated to consider the LQR problem for continuous-time systems with sampled data, in spite of the difficulties illustrated by the example above. In this chapter, we extend our results for the interval data formulation using approximations to interval data constructed from sampled data. Rather than measuring values taken by the input and state on an entire interval $[t_0^i, t_f^i]$, we sample them at a finite subset of points in the interval.

The chapter is organized as follows. In Section 4.1, we show that the map from data matrices D_+ and D_- to the true system $\hat{S}$ is continuous. This enables us to show in Section 4.2 that approximations to D_+ and D_- can be used to obtain arbitrarily close approximations to the optimal gain K of the true system. In Section 4.3, we discuss the relation between this result and some preexisting methods for approximating x and u from sampled data. In Section 4.4, we do some numerical experiments demonstrating the usefulness of the theory from Section 4.2 and highlighting the importance of choosing sampling points. We conclude with some remarks in Section 4.5.

4.1 The map from data to system is continuous

Let $\left\{\tilde{D}_+^d\right\}_{d=1}^{\infty}$ and $\left\{\tilde{D}_-^d\right\}_{d=1}^{\infty}$ be matrix-valued sequences satisfying

$$\lim_{d\to\infty} \tilde{D}_+^d = D_+, \tag{4.4}$$

$$\lim_{d\to\infty} \tilde{D}_-^d = D_-. \tag{4.5}$$

Since D_- is not square, we require the following easy proposition.

Proposition 4.1.1. *D_- has a trivial nullspace if and only if $D_-^\top D_-$ is invertible.*

Proof. Suppose D_- has a trivial nullspace and consider an S that solves $D_-^\top D_- S = 0$. Then, $D_- S$ is simultaneously in the nullspace of $D_-^\top$ and in the column space of D_-. These spaces are orthogonal complements so we must have $S = 0$, which shows that $D_-^\top D_-$ is invertible. Now suppose $D_-^\top D_-$ is invertible and consider an S that solves $D_- S = 0$. The same S solves $D_-^\top D_- S = 0$, and by the supposition must therefore be $S = 0$, which shows D_- has a trivial nullspace. □

This is useful later on when we show the continuous dependence of $\hat{S}$ on D_+ and D_-. Corollary 3.5.2 shows that if $K \neq 0$, D_- must have a trivial nullspace for the LQR problem to be solvable.

Proposition 4.1.2. *Assume that D_- has a trivial nullspace. Then, there exists some δ such that for every $d \geq \delta$ there exists a unique matrix $\tilde{S}^d$ that solves*

$$\tilde{D}_+^d = \tilde{D}_-^d \tilde{S}^d, \tag{4.6}$$

and these $\tilde{S}^d$ satisfy

$$\lim_{d\to\infty} \tilde{S}^d = \hat{S}. \tag{4.7}$$

Proof. Since D_- has a trivial nullspace, $D_-^\top D_-$ has only nonzero eigenvalues, and these are all bounded away from zero because there are finitely many of them. Since the eigenvalues of a matrix are continuous in its entries, the eigenvalues of $\tilde{D}_-^{d\top}\tilde{D}_-^d$ approach those of $D_-^\top D_-$ as $d \to \infty$, meaning there exists some δ such that for every $d \geq \delta$ the matrix $\tilde{D}_-^{d\top}\tilde{D}_-^d$ is invertible. From the continuity of matrix inversion, the sequence $\left\{\left(\tilde{D}_-^{d\top}\tilde{D}_-^d\right)^{-1}\right\}_{d=\delta}^{\infty}$ satisfies

$$\lim_{d\to\infty}\left(\tilde{D}_-^{d\top}\tilde{D}_-^d\right)^{-1} = \left(D_-^\top D_-\right)^{-1}. \tag{4.8}$$

To show convergence of the sequence $\{\tilde{S}^d\}_{d=\delta}^{\infty}$, we first write

$$\begin{aligned}
0 &= \lim_{d\to\infty} D_+ - \tilde{D}_+^d \\
&= \lim_{d\to\infty} D_- S - \tilde{D}_-^d \tilde{S}^d \\
&= \lim_{d\to\infty} D_- S - \tilde{D}_-^d S + \tilde{D}_-^d S - \tilde{D}_-^d \tilde{S}^d \\
&= \lim_{d\to\infty} \left(D_- - \tilde{D}_-^d\right) S + \tilde{D}_-^d \left(S - \tilde{S}^d\right) \\
&= \lim_{d\to\infty} \left(D_- - \tilde{D}_-^d\right) S + \lim_{d\to\infty} \tilde{D}_-^d \left(S - \tilde{S}^d\right) \\
&= 0 + \lim_{d\to\infty} \tilde{D}_-^d \left(S - \tilde{S}^d\right).
\end{aligned} \tag{4.9}$$

We then substitute this into

$$\begin{aligned}
&\lim_{d\to\infty}\left(\tilde{D}_-^{d\top}\tilde{D}_-^d\right)^{-1}\tilde{D}_-^{d\top}\tilde{D}_-^d\left(\hat{S} - \tilde{S}^d\right) \\
&= \lim_{d\to\infty}\left(\tilde{D}_-^{d\top}\tilde{D}_-^d\right)^{-1}\tilde{D}_-^{d\top}\lim_{d\to\infty}\tilde{D}_-^d\left(\hat{S} - \tilde{S}^d\right).
\end{aligned} \tag{4.10}$$

to complete the proof.

□

4.2 The map from data to optimal gain is continuous

Having just proved in the previous section that systems may be approximately identified from approximations to the input and state data they generate, we now show that close enough approximations to the true system are stabilizable, and will lead to close approximations to the true optimal gain. We achieve this using the PBH test for stabilizability we introduced in Definition 2.2.1 in Chapter 2. We summarize our main result in this proposition.

Proposition 4.2.1. *Let* $\tilde{S}^d = \begin{bmatrix}\tilde{A}^d & \tilde{B}^d\end{bmatrix}^\top$ *be the unique solution to* (4.6)*. There exists some* δ *such that for every* $d \geq \delta$ *the pair* $(\tilde{A}^d, \tilde{B}^d)$ *is stabilizable. Furthermore,*

$$\lim_{d\to\infty} k_{Q,R}\left(\tilde{S}^{d\top}\right) = k_{Q,R}\left(\hat{S}^\top\right), \tag{4.11}$$

provided that $(\hat{A}, C)$ *is detectable.*

Proof. The eigenvalue $\tilde{\lambda}^d$ of $\tilde{A}^d$ corresponding to eigenvalue λ of A depends continuously on the entries of $\tilde{A}^d$, so defining

$$M_c := \begin{bmatrix} A - \lambda I & B \end{bmatrix}, \tag{4.12}$$

$$\tilde{M}_c^d := \begin{bmatrix} \tilde{A}^d - \tilde{\lambda}^d I & \tilde{B}^d \end{bmatrix}, \tag{4.13}$$

we get

$$\lim_{d\to\infty} \tilde{M}_c^d = M_c \tag{4.14}$$

from Proposition 4.1.2.

Since $M_c^\top$ has more rows than columns, equation (2.25) implies it has a trivial nullspace. This means that for each eigenvalue λ of A with nonnegative real part, $M_c M_c^\top$ is invertible, so all of its (finitely many) eigenvalues are bounded away from zero. We now show that $\tilde{M}_c^d \tilde{M}_c^{d\top}$ is also invertible when d is large enough. First, consider a sequence $\{\tilde{\lambda}^d\}_{d=1}^\infty$ converging to a λ with nonnegative real part. There exists a δ_λ such that for every $d \geq \delta_\lambda$ the eigenvalues of $\tilde{M}_c^d \tilde{M}_c^{d\top}$ are all nonzero. Now consider a sequence $\{\tilde{\lambda}^d\}_{d=1}^\infty$ converging to a λ with negative real part. There exists a δ_λ such that for every $d \geq \delta_\lambda$, $\tilde{\lambda}^d$ also has negative real part.

It follows that, for every $d \geq \delta := \max\{\delta_\lambda\}$, we have $\tilde{M}_c^d \tilde{M}_c^{d\top}$ invertible, and therefore $\tilde{M}_c^d$ has a trivial nullspace for every $\tilde{\lambda}^d$ with nonnegative real part. Applying Definition 2.2.1, we conclude that $(\tilde{A}^d, \tilde{B}^d)$ is stabilizable for $d \geq \delta$. This means that, for d large enough, the unique smallest positive semidefinite solution $\tilde{P}^d$ of the ARE

$$\tilde{A}^{d\top} \tilde{P}^d + \tilde{P}^d \tilde{A}^d - \tilde{P}^d \tilde{B}^d R^{-1} \tilde{B}^{d\top} \tilde{P}^d + Q = 0 \tag{4.15}$$

defines the optimal gain $\tilde{K}^d$

$$\tilde{K}^d := -R^{-1} \tilde{B}^{d\top} \tilde{P}^d \tag{4.16}$$

which minimizes the LQR cost for $\tilde{A}^d, \tilde{B}^d$, Q, and R. We know from [8] that when $(\hat{A}, \hat{B})$ and $(\hat{A}, C)$ are respectively stabilizable and detectable, $\tilde{P}^d$ depends continuously on $\tilde{A}^d$ and $\tilde{B}^d$, so it must converge to P. If we replace $\tilde{D}_+^d$, $\tilde{D}_-^d$, and $\tilde{S}^d$ in Proposition 4.1.2 with $\tilde{B}^{d\top} \tilde{P}^d$, $-R$, and $\tilde{K}^d$, respectively, we can conclude that

$$\lim_{d\to\infty} \tilde{K}^d = K. \tag{4.17}$$

□

In summary, we have shown that the map from data matrices D_+ and D_- to optimal gain K defined by system identification together with the ARE is continuous. This naturally leads to the idea of finding approximations to D_+ and D_- in order to compute approximations to K. There may be a connection between this result and some recent related work [33] on noise sensitivity of data-driven LQR algorithms in discrete time. Assuming the endpoints of the sampling intervals may be freely chosen, there is no reason why D_+ should not be directly measured by selecting endpoints among the points at which x and u are sampled. Therefore, it only remains to approximate D_-, whose entries are definite integrals of the entries of x and u over time. This contrasts with the convergence in norm in $L^2\left([t_0^i, t_f^i], \mathbb{R}\right)$ to x and u required in [21], which we discuss presently.

4.3 A weaker condition for approximation

Since x and u are continuous and piecewise continuous in time, respectively, their entries belong to the space $L^2\left([t_0^i, t_f^i]\right)$ of square-integrable real-valued functions on $[t_0^i, t_f^i]$ equipped with the standard inner product. From [21, Section II], a real-valued entry f of x or u can be represented as

$$f = \sum_{k=0}^{\infty} \tilde{f}_k b_k, \tag{4.18}$$

where $\{b_k\}_{k\in\mathbb{N}}$ is a complete basis and

$$\tilde{f}_k = \langle f, b_k \rangle_w, \tag{4.19}$$

where $\langle \cdot, \cdot \rangle_w$ is a weighted inner product defined by

$$\langle f, g \rangle_w := \int_{t_0^i}^{t_f^i} f(t) g(t) w(t)\, \mathrm{d}t, \tag{4.20}$$

and $w : [t_0^i, t_f^i] \mapsto \mathbb{R}$ is a weight function chosen to ensure $\{b_k\}$ is orthogonal with respect to $\langle \cdot, \cdot \rangle_w$. In [21], the basis functions considered are orthogonal polynomials, and the examples given are Chebyshev and Legendre polynomials, which are orthogonal on $[-1, 1]$ with respect to weigh functions $w(t) = \frac{1}{\sqrt{1-t^2}}$ and $w(t) = 1$, respectively. Affine transformations that map $[-1, 1]$ to an arbitrary interval $[t_0^i, t_f^i]$ may be applied to the basis functions to achieve approximation on any interval (we say more about this in the next section). In both cases, the weight functions are bounded below on $[-1, 1]$ by a positive constant. In this case, convergence in the norm given by the limit (4.18) implies convergence to the entries of D_-, as stated in the following proposition.

Proposition 4.3.1. *Let α be a positive number and let $\|\cdot\|_p$ and $\|\cdot\|_w$ be the norm induced by the standard L^p inner product and $\langle \cdot, \cdot \rangle_w$ respectively. Assume w satisfies $w(t) \geq \alpha$ for all $t \in [t_0^i, t_f^i]$. Then,*

$$\lim_{N\to\infty} \left\| \sum_{k=0}^{N} \tilde{f}_k b_k - f \right\|_w = 0 \tag{4.21}$$

implies

$$\lim_{N\to\infty} \int_{t_0^i}^{t_f^i} \sum_{k=0}^{N} \tilde{f}_k b_k(t)\, \mathrm{d}t = \int_{t_0^i}^{t_f^i} f(t)\, \mathrm{d}t. \tag{4.22}$$

Proof. From the condition on w we immediately have $\|f\|_w \geq \alpha \|f\|_2$ and

$$\lim_{N\to\infty} \left\| \sum_{k=0}^{N} \tilde{f}_k b_k - f \right\|_2 = 0. \tag{4.23}$$

Then, applying Hölder's inequality, we write

$$\begin{aligned}\left\|\sum_{k=0}^{N}\tilde{f}_k b_k - f\right\|_1 &= \left\|\left(\sum_{k=0}^{N}\tilde{f}_k b_k - f\right)\cdot 1\right\|_1 \\ &\leq \left\|\sum_{k=0}^{N}\tilde{f}_k b_k - f\right\|_2 \|1\|_2 \\ &= \left(t_f^i - t_0^i\right)^{\frac{1}{2}}\left\|\sum_{k=0}^{N}\tilde{f}_k b_k - f\right\|_2, \end{aligned} \tag{4.24}$$

which, since our intervals satisfy $t_f^i > t_0^i$, shows

$$\lim_{N\to\infty}\left\|\sum_{k=0}^{N}\tilde{f}_k b_k - f\right\|_1 = 0. \tag{4.25}$$

The last step is to observe that

$$\left\|\sum_{k=0}^{N}\tilde{f}_k b_k - f\right\|_1 = \int_{t_0^i}^{t_f^i}\left|\sum_{k=0}^{N}\tilde{f}_k b_k(t) - f(t)\right|\mathrm{d}t \geq \int_{t_0^i}^{t_f^i}\sum_{k=0}^{N}\tilde{f}_k b_k(t) - f(t)\,\mathrm{d}t, \tag{4.26}$$

and taking the limit as $N \to \infty$ completes the proof. □

Clearly, the converse of Proposition 4.3.1 does not hold. In fact, by considering sequences converging to D_-, we no longer require any approximations to the entries of x and u to be constructed. By only requiring the limits (4.4) and (4.5) to hold, we are seeking approximations to a matrix of definite integrals (which exists in the finite-dimensional space $\mathbb{R}^{q(m+n)}$) instead of approximations to the integrands (which exist in infinite-dimensional spaces) in the fashion of [21] or [23]. In doing so, we weaken the conditions on approximations to interval data necessary to guarantee convergence of data-driven LQR algorithms to the optimal gain.

4.4 Numerical experiments with sampled data

There are numerous numerical integration methods in which function values on a finite subset of an interval are used to approximate definite integrals on the interval. If we sample at $d+1$ equally spaced points and use polynomial interpolants, we are doing Newton-Cotes quadrature. If we move these samples to the Chebyshev nodes, and instead integrate a Chebyshev polynomial interpolant, we get Clenshaw-Curtis quadrature. If the interpolant is a Legendre polynomial, we are doing Gaussian quadrature. Interested readers can find more methods and details in standard numerical integration references, such as [24, Chapter 9].

We choose to focus on Clenshaw-Curtis quadrature, where the main idea is to approximate the integrand with a Chebyshev series, so that we get

$$\lim_{d\to\infty} \int_{-1}^{1} \sum_{k=0}^{d} c_k T_k(t)\,\mathrm{d}t = \int_{-1}^{1} f(t)\,\mathrm{d}t, \tag{4.27}$$

where c_k are some coefficients and T_k are the Chebyshev polynomials. In practice, rather than constructing the approximation explicitly, we sample at $d+1$ Chebyshev points of the second kind, given by

$$t_k = \cos(k\pi/d), \quad 0 \le k \le d. \tag{4.28}$$

Then, since the map from the $d+1$ sampled values $f(t_k)$ to the integral $\int_{-1}^{1} \sum_{k=0}^{d} c_k T_k(t)\,\mathrm{d}t$ of the degree d Chebyshev interpolant is a linear functional, we can directly obtain the integral by computing

$$\sum_{k=0}^{d} w_k f(t_k), \tag{4.29}$$

where w_k are particular weights. A more detailed discussion may be found in [26, Chapter 19]. The points t_k lie in the interval $[-1, 1]$, so given an arbitrary interval $[t_0^i, t_f^i]$, we use the affine bijection

$$t \longleftrightarrow \frac{t_f^i - t_0^i}{2} t + \frac{t_f^i + t_0^i}{2} \tag{4.30}$$

to map the former to the latter, and also apply a coefficient of $\frac{t_f^i - t_0^i}{2}$ to the weights w_k . This algorithm has the advantage that the interval endpoints are included among these points, so we can directly measure D_+ without introducing additional sampling points.

Curtis-Clenshaw and Gaussian quadrature are known experimentally [25] to be competitive with one another. Newton-Cotes formulas suffer from the major drawback that they diverge [27, Section 2] as the degree of the interpolating polynomial increases to infinity, but retain some attractiveness because they use periodically sampled data which is naturally generated by many digital sensors and computer hardware. This is the main reason we are interested in them, although higher-order formulas are also in use, and techniques such as Richardson extrapolation can help improve their relatively poor convergence rate. In the following experiments, we assess how the ADP and system identification algorithms perform with approximated data constructed using Newton-Cotes quadrature of degree one (commonly known as the trapezoid rule), and with data constructed using Clenshaw-Curtis quadrature of degree d.

We repeat the examples of Chapter 3, keeping the same parameters as before and fixing the number of intervals. Now, instead of using `ode45` to compute the integrals in D_- as we do in Chapter 3, we use it to measure x and u at $d+1$ times during each interval.

We do this once with equally spaced samples and do quadrature with the trapezoid rule, and again with $d+1$ samples at Chebyshev nodes and do Clenshaw-Curtis quadrature. In both cases, the sample times include all t_0^i and t_f^i , so we measure D_+ exactly. Using both LQR algorithms, we separately compute $\tilde{K}^d$ from the $\tilde{D}_-^d$ obtained from both quadrature methods. We allow $d+1$ to vary from 3 to $10,000$ and compare the $\tilde{P}^d$ and $\tilde{K}^d$ thus obtained with the P and K returned by `lqr`. In the our error plots, we include for comparison solid and dashed horizontal lines representing the error in the K and P respectively computed from `ode45`-generated interval data. The results for the 2-dimensional Example 3.3.1 with 30 intervals are shown in the following plot.

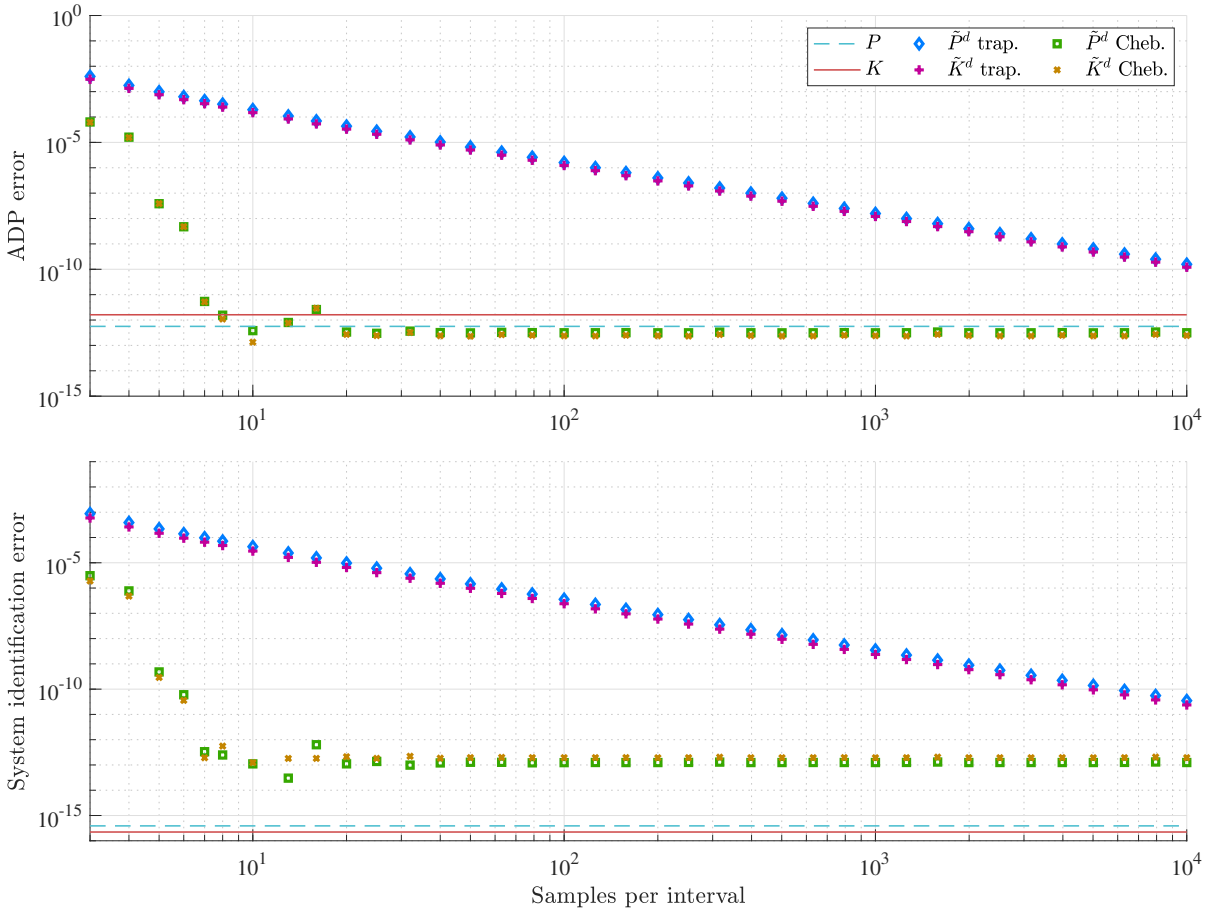


Figure 4.1: Error in K and P for a 2-dimensional system from 3 to 10,000 samples per interval.

ADP and system identification respond to the errors in approximated data in similar ways. In ADP, Curtis-Clenshaw quadrature produces approximations to K and P as good as those produced from interval data generated by `ode45` for $d+1 \geq 10$, and the errors remain around 10^{-12} for larger d. In system identification, similar behavior is observed where the errors stop decaying after $d+1 = 10$ and remain around 10^{-13}. However, there is a significant and persistent gap between these errors and the errors obtained from `ode45` interval data. This indicates that the more dramatic difference in the error produced by both

algorithms using interval data we observed in Figure 3.1 is at least partly an artifact of our numerical simulation, and not necessarily predictive of the relative performances in practice. Nevertheless, it seems likely that a gap of at least one order of magnitude does exist, which we continue to attribute to the greater complexity of ADP. Overall, approximated data from Curtis-Clenshaw quadrature performs very well in both algorithms. In contrast, the approximation errors of trapezoid rule with equally spaced samples decay much more slowly as d increases, and the errors are still markedly larger even with $d + 1 = 10,000$ sampling points per interval. Next, we show the results for the 3-dimensional Example 3.3.2, also with 30 intervals.

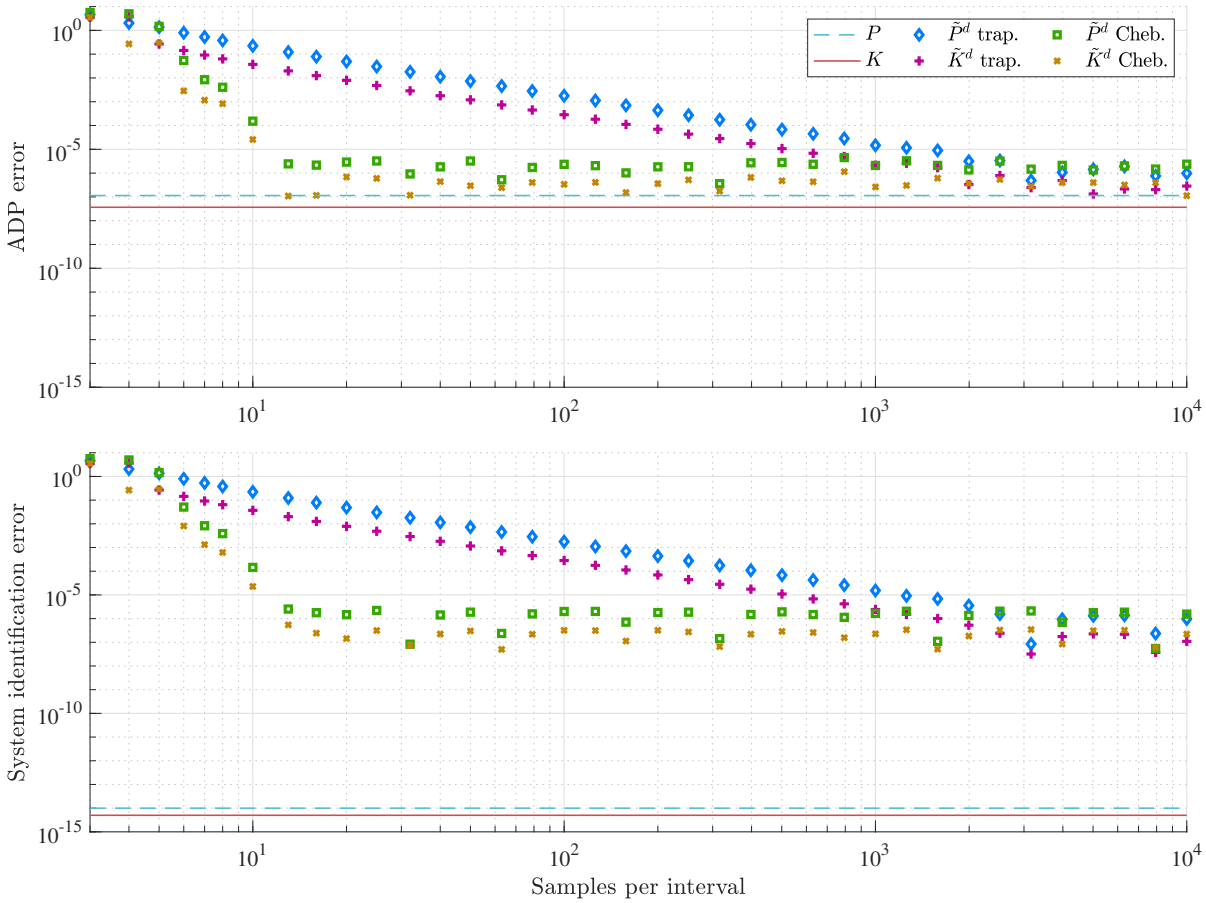


Figure 4.2: Error in K and P for a 3-dimensional system from 3 to 10,000 samples per interval.

The results are similar to those for the 2-dimensional system. In ADP, both quadrature methods eventually succeed in almost matching the errors given by interval data. Both ADP and system identification with both quadrature methods achieve errors around 10^{-6}, so the gap in performance between LQR algorithms appears to be smaller than the roughly one degree of magnitude observed for the previous system. In comparison, the gap in performance for interval data is striking. The errors produced by Clenshaw-Curtis quadrature stop decreasing past $d + 1 = 20$ in both LQR algorithms. The errors produced by trapezoid

rule again decay far more slowly, but eventually match those produced by Clenshaw-Curtis quadrature for $d + 1 \geq 2,000$. Finally, we show the results for the 6-dimensional Example 3.3.3 with 70 intervals.

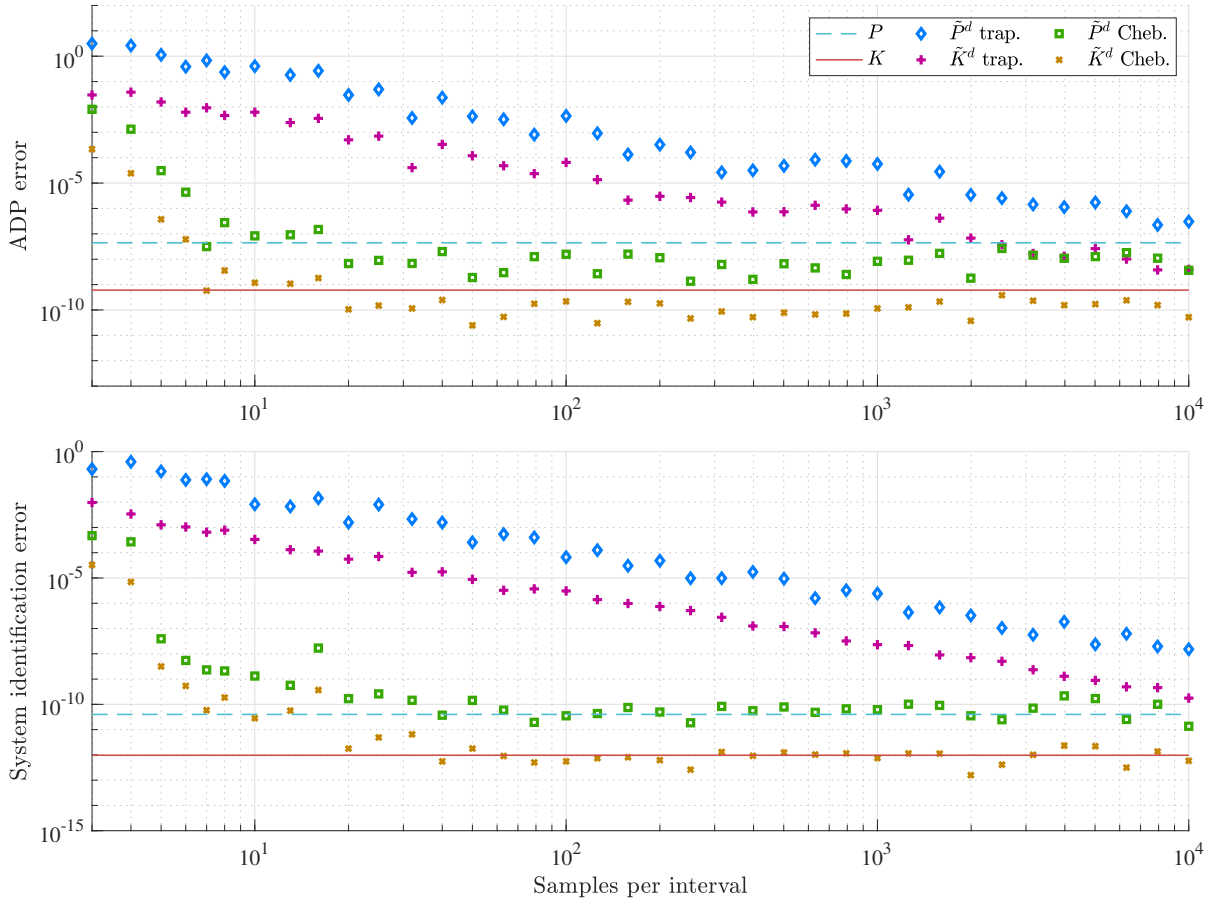


Figure 4.3: Error in K and P for a 6-dimensional system from 3 to 10,000 samples per interval.

In this example, Curtis-Clenshaw quadrature matches the errors of interval data in both ADP and system identification for $d + 1 \geq 20$. Using this quadrature method, the errors produced by the LQR algorithms persistently differ by about two orders of magnitude. The errors produced by trapezoid rule remain orders of magnitude larger, even with $d+1 = 10,000$ samples per interval.

4.5 Remarks

In this chapter, we have used our results from Chapter 3 to show that the map from data matrices D_+ and D_- to optimal gain K is continuous. We take the approach of seeking approximations to definite integrals of x and u on an interval, rather than approximations to x and u themselves. This results in a weaker condition on approximations to interval data.

When combined with our result from Chapter 3, this means we can use any approximate data on the q intervals in (3.1) for which (4.4) and (4.5) hold in place of exact interval data in a data-driven LQR algorithm such as [14], [10], [31], [20], or [19] that is formulated using exact interval data, and we will obtain arbitrarily close approximations to K.

This opens the possibility of applying the varied and powerful techniques of numerical integration to the task of approximating interval data from sampled data. By seeking approximations to integrals, we avoid certain issues associated with computing approximations to the state derivative. For example, in [21], the derivative approximation error caused by partition in [21, Equation 19] does not always go to zero as the order of the polynomial interpolant grows, so instead the authors provide an upper bound.

We have demonstrated the application of techniques from numerical integration with our numerical experiments. All three examples showcase the large advantage enjoyed by Clenshaw-Curtis quadrature over trapezoid rule, which highlights the significance of the choice of sampling points. This suggests that methods which make use of periodic data sampling may be dramatically improved by more judicious choice of sampling points. We stress that we have made the choice of sampling points without knowledge of the system or its optimal gain. This points to the possibility of exploring different methods by which sampling points may be chosen in different situations. In some circumstances, the greater practical difficulty of implementing aperiodic sampling with digital sensors and computers could be more than justified by the improvements in performance it affords.

Chapter 5

Conclusions and future directions

In this thesis, we have developed a data-centric view of direct and indirect LQR algorithms in continuous time. In Chapters 1 and 2, we have given an overview of the diverse approaches taken by various authors to the LQR problem in discrete and continuous time, and a review of the classical theory of the linear quadratic regulator which forms the foundation for our subsequent analysis.

In Chapter 3, we have used a data informativity framework to show that all direct LQR algorithms require at least as much interval data as system identification. This means that all interval data LQR algorithms, direct and indirect, evaluate the same map from data D_+ and D_- to optimal gain K. We have complimented this result with some insights about the system identification and ADP algorithms gleaned from our numerical experiments. These experiments suggest that ADP requires more intervals of data than system identification when the intervals are equally sized, but this is not true in general if the interval size varies.

In Chapter 4, we have shown that the map from D_+ and D_- to K is continuous, which enables the use of sampled data with every LQR algorithm formulated using interval data to approximate K. This leads to a connection with numerical integration, and furnishes a different and weaker condition on the convergence of approximations used for this purpose. Our numerical experiments have shown a large advantage in the convergence rate of approximations to P and K for a certain aperiodic sampling scheme, compared with periodic sampling.

5.1 Conclusions

Having addressed the central question of this thesis, we now return to the three questions we posed in Section 1.2.

We have compared interval and sampled data LQR algorithms with one another primarily through the lens of using sampled data to approximate interval data in integral form. There are certainly sampled data algorithms that do not take this approach, and we have

a little more to say about this in the next section. Nonetheless, our numerical experiments in Chapter 4 show that sampled data can be used to solve LQR problems as effectively as interval data, provided the sampling times are well chosen. For our examples, any more than approximately 10 to 20 samples per interval was excessive, and resulted in no further reduction in error. We acknowledge that significant gaps between Curtis-Clenshaw quadrature and interval data persist even with rapid sampling in system identification for the 2-dimensional and 3-dimensional systems, as shown in Figures 4.1 and 4.2. These gaps are likely due not to poor performance of sampled data, but to unrealistically good performance of the interval data we obtained using `ode45`. We see no reason why sampled data cannot match interval data in a realistic application, as it also does in Figure 4.3. Admittedly, we draw these conclusions based on numerical experiments with only two algorithms and three relatively small linear systems, but they are also justified by the theory from Chapter 4.

We state and prove the main result (Theorem 3.5.1) from Chapter 3 in the hope that it will allow a degree of comparison between the data requirements of different interval data LQR algorithms. Each LQR algorithm shapes the interval data into its own form and imposes its own conditions on that data (for example the rank conditions on Ω and Γ in [20], condition (18) in [19], condition (2.38) in [14], uniqueness of solution to (22) in [31]), and neither the data nor the conditions can be directly compared. While Theorem 3.5.1 indeed shows the equivalence of the system identification and data-driven LQR problems for data of the form defined in equations (3.12), the example of Subsection 3.3.2 shows that it is possible for an interval data LQR algorithm to have its data requirement satisfied by data that is insufficient for system identification. However, this example is highly contrived, since the interval endpoints are specifically chosen to cause D_- to be rank-deficient, a choice which would never be made in practice. We expect that in the absence of such special choices of interval endpoints, each of the rank conditions mentioned above holding will generally imply that the D_- matrix constructed from the values taken by x and u on the q intervals in (3.1) will have a trivial nullspace, and the rank condition (3.14) will be satisfied. However, this remains to be formally investigated. On the other hand, because (3.14) is is necessary and sufficient for system identification, those interval data LQR algorithms that impose requirements on their data which fail to hold when (3.14) holds can be said to be inefficient. They require more data than is strictly necessary in order to solve the LQR problem. From our numerical experiments in Section 3.3, it seems that ADP is sometimes but not always inefficient in this sense, and the degree of inefficiency for a given problem depends on the particular choice of sampling interval endpoints.

Example 3.3.2 shows the strong and somewhat unexpected sensitivity of the rank conditions of ADP and system identifications to the endpoints of sampling intervals, when their number is held constant. This points to the possibility of improving the data efficiency of both algorithms by careful selection of interval endpoints, based not on knowledge of the underlying system or its optimal gain, but solely on input and state data already collected. This, along with the massive difference in performance between trapezoid rule and Clenshaw-Curtis quadrature we observed in Section 4.4, suggests that a primary focus of data-driven LQR algorithm design should be using existing input and state data to decide when and

how more data should be collected. Our experiments show that large advantages may be gained without knowledge of the system matrices or the optimal gain. The timing of data collection matters to such a degree that it seems more appropriate to categorize data-driven LQR algorithms based on how they decide how and when to collect data, rather than based on whether they explicitly identify the system.

5.2 Future directions

Our investigations into the continuous-time LQR problem remain substantially incomplete. As mentioned in the previous subsection, there exist sampled data LQR algorithms that do not attempt to approximate interval data. We presently sketch the setup for an exact system identification algorithm.

Example 5.2.1. *Consider the LQR problem for the linear system (2.1). Suppose the state and input are measured at times t^i and $t^i + h$, and that the input is constant on the time interval $[t^i, t^i+h]$. Then, using e^{-hA} as an integrating factor, we obtain by the time invariance of the system*

$$x(t^i + h) = e^{hA}\left(x(t^i) + \int_0^h e^{-\tau A} d\tau B u\right) = \begin{bmatrix} e^{hA} & e^{hA}\int_0^h e^{-\tau A} d\tau B \end{bmatrix} \begin{bmatrix} x(t^i) \\ u(t^i) \end{bmatrix} \tag{5.1}$$

for this input. If we have state and input data on q pairs of times

$$(t^1, t^1 + h), \cdots, (t^q, t^q + h), \tag{5.2}$$

we can define

$$D_+^\top := \begin{bmatrix} x(t^1 + h) & \cdots & x(t^q + h) \end{bmatrix}, \tag{5.3a}$$

$$D_-^\top := \begin{bmatrix} x(t^1) & \cdots & x(t^q) \\ u(t^1) & \cdots & u(t^q) \end{bmatrix}, \tag{5.3b}$$

much as we did in (3.12), and introduce the notation

$$A_\Delta := e^{hA}, \tag{5.4a}$$

$$\Phi := \int_0^h e^{-\tau A} d\tau, \tag{5.4b}$$

$$B_\Delta := A_\Delta \Phi B, \tag{5.4c}$$

$$S_\Delta^\top := \begin{bmatrix} A_\Delta & B_\Delta \end{bmatrix}, \tag{5.4d}$$

which allows us to combine equation (5.1) for the q pairs into the system

$$D_+ = D_- S_\Delta. \tag{5.5}$$

Equations (5.4) define an operator $\begin{bmatrix} A & B \end{bmatrix}^\top \longmapsto \begin{bmatrix} A_\Delta & B_\Delta \end{bmatrix}^\top$ on $\mathbb{R}^{n(m+n)}$ induced by the periodic sampling of the input and state of the continuous-time system, and these data constrain A and B via (5.5). What conditions on D_- and h must we impose in order to invert the map and recover A and B? What can we say about the set of $\begin{bmatrix} A & B \end{bmatrix}^\top$ compatible with the data when D_- is not invertible?

Generalizing the data informativity framework to exact sampled data algorithms like the one sketched above would do much to advance our understanding of the LQR problem in the continuous time setting.

Having made some remarks about the LQR problem, we naturally turn toward generalizations to other settings in optimal control. The ADP algorithm has been adapted to stochastic linear systems [3], and to nonlinear systems whose right-hand side is spanned by a finite basis set [14, Chapter 3]. In [1], the optimal control problem is solved by an iterative procedure for approximating solutions to the Hamilton-Jacobi-Bellman (HJB) equation reminiscent of Kleinman's algorithm for solving the ARE.

The similarities between direct and indirect optimal control algorithms developed for linear, stochastic, and nonlinear continuous-time systems inspire hope that a result analogous to those of [30] and Chapter 3 establishing the equivalence of data informativity for the optimal control and system identification problems may hold for settings beyond LQR. Such a result may be attained by a generalization of the methods used to establish the existing results, or there may be easy counterexamples showing that such a result does not exist for large classes of systems beyond the deterministic linear setting. In the latter event, it may yet be possible to leverage the data informativity framework in different ways to provide more insight on the relationship between the problems of system identification and data-driven optimal control. This relationship has been the central theme of our investigations in this thesis, and many questions along similar lines of inquiry remain.

References

[1] Randal W. Beard, George N. Saridis, and John T. Wen. Galerkin approximations of the generalized Hamilton-Jacobi-Bellman equation. *Automatica*, 33(23):2159–2177, 1997.

[2] Alain Bensoussan. *Stochastic Control of Partially Observable Systems*. Cambridge University Press, 1992.

[3] Tao Bian, Yu Jiang, and Zhong-Ping Jiang. Adaptive dynamic programming for stochastic systems with state and control dependent noise. *IEEE Trans. Automat. Contr.*, 61(12):4170–4175, 2016.

[4] Marco Borghesi, Simone Baroncini, Guido Carnevale, Alessandro Bosso, and Giuseppe Notarstefano. On sufficient richness for linear time-invariant systems. *2025*, arXiv:2502.04062.

[5] Alessandro Bosso, Marco Borghesi, Andrea Iannelli, Giuseppe Notarstefano, and Andrew R. Teel. Derivative-free data-driven control of continuous-time linear time-invariant systems. *2024*, arXiv:2410.24167.

[6] Gustavo G. da Silva, Alexandre S. Banzanella, Charles Lorenzini, and Lucíola Campestrini. Data-driven LQR control design. *IEEE Control Syst. Lett.*, 3(1):180–185, 2019.

[7] Tianyu Dai and Mario Sznaier. Data-driven quadratic stabilization and LQR control of LTI systems. *Automatica*, 153:111041, 2023.

[8] David F. Delchamps. Analytic stabilization and the algebraic riccati equation. In *The 22nd IEEE Conference on Decision and Control*, pages 1396–1401. IEEE, 1983.

[9] Jaap Eising and Jorge Cortés. When sampling works in data-driven control: informativity for stabilization in continuous time. *IEEE Trans. Automat. Contr.*, 70(1):565–572, 2025.

[10] Justin Fong, Ying Tan, Vincent Crocher, Denny Oetomo, and Iven Mareels. Dual-loop iterative optimal control for the finite horizon LQR problem with unknown dynamics. *Systems & Control Letters*, 111:49–57, 2018.

[11] Hugues Garnier, Michel Mensler, and Alain Richard. Continuous-time model identification from sampled data: Implementation issues and performance evaluation. *International Journal of Control*, 76(13):1337–1357, 2003.

[12] João P. Hespanha. *Linear systems theory*. Princeton University Press, Princeton, NJ, 2009.

[13] Roger A. Horn and Charles R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991.

[14] Yu Jiang and Zhong-Ping Jiang. *Robust adaptive dynamic programming*. Wiley, Hoboken, NJ, 2017.

[15] David Kleinman. On an iterative technique for riccati equation computations. *IEEE Trans. Automat. Contr.*, 13(1):114–115, 1968.

[16] Daniel Liberzon. *Calculus of variations and optimal control theory*. Princeton University Press, Princeton, NJ, 2012.

[17] Victor G. Lopez, Mohammad Alsalti, and Matthias A. Müller. Efficient off-policy q-learning for data-based discrete-time LQR problems. *IEEE Trans. Automat. Contr.*, 68(5):2922–2933, 2023.

[18] Victor G. Lopez and Matthias A. Müller. On a continuous-time version of willems' lemma. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 2759–2764, 2022.

[19] Bo Pang, Tao Bian, and Zhong-Ping Jiang. Robust policy iteration for continuous-time linear quadratic regulation. *IEEE Trans. Automat. Contr.*, 67(1):504–511, 2022.

[20] Corrado Possieri and Mario Sassano. Q-learning for continuous-time linear systems: a data-driven implementation of the Kleinman algorithm. *IEEE Trans. Syst., Man, Cybern.*, 52(10):6487–6497, 2022.

[21] Paolo Rapisarda, Henk J. van Waarde, and Mehmet K. Çamlibel. Orthogonal polynomial bases for data-driven analysis and control of continuous-time systems. *IEEE Trans. Automat. Contr.*, 69(7):4307–4319, 2024.

[22] Monica Rotulo, Claudio De Persis, and Pietro Tesi. Data-driven linear quadratic regulation via semidefinite programming. *IFAC-PapersOnLine*, 53(2):3995–4000, 2020.

[23] Philipp Schmitz, Timm Faulwasser, Paolo Rapisarda, and Karl Worthmann. A continuous-time fundamental lemma and its application in data-driven optimal control. *Systems & Control Letters*, 194:105950, 2024.

[24] Tim J. Sullivan. *Introduction to Uncertainty Quantification*. Springer, Cham, Switzerland, 2015.

[25] Lloyd N. Trefethen. Is Gauss quadrature better than Clenshaw-Curtis? *SIAM Review*, 50(1):67–87, 2008.

[26] Lloyd N. Trefethen. *Approximation theory and approximation practice*. SIAM, Philadelphia, PA, 2020.

[27] Lloyd N. Trefethen. Exactness of quadrature formulas. *SIAM Review*, 64(1):132–150, 2022.

[28] Harry L Trentelman, Anton A Stoorvogel, and Malo Hautus. *Control Theory for Linear Systems*. Springer, London, 2001.

[29] Abdul A. Umar, Kunhee Ryu, Juhoon Back, and Jung-Su Kim. Stability margin of data-driven LQR and its application to consensus problem. *Mathematics*, 12(199), 2024.

[30] Henk J. van Waarde, Jaap Eising, Harry L. Trentelman, and Mehmet K. Çamlibel. Data informativity: A new perspective on data-driven analysis and control. *IEEE Trans. Automat. Contr.*, 65(11):4753–4768, 2020.

[31] Draguna Vrabie, Octavian Pastravanu, Murad Abu-Khalaf, and Frank L. Lewis. Adaptive optimal control for continuous-time linear systems based on policy iteration. *Automatica*, 45:477–484, 2009.

[32] Zhemin Wang, Guillaume O. Berger, and Raphaël M. Jungers. Data-driven control of unknown switched linear systems using scenario optimization. *IEEE Trans. Automat. Contr.*, 69(11):7310–7325, 2024.

[33] Xiong Zeng, Laurent Bako, and Necmiye Ozay. Noise sensitivity of the semidefinite programs for direct data-driven LQR. *2024*, arXiv:2412.19705.