Robust Least-Squares Optimization for Data-Driven Predictive Control

Submitted in the partial fulfillment of the requirements of the degree of

Master of Technology

by

Shreyas N. B.

(Roll No: 210010061)

Supervisor:

Prof. Ravi Banavar

Co-Supervisors:

Prof. Cyrus Mostajeran

Dr. Bamdev Mishra

Center for Systems and Controls

INDIAN INSTITUTE OF TECHNOLOGY BOMBAY 2025

(Draft - October 24, 2025)

Approval Sheet

This thesis entitled *Robust Least-Squares Optimization for Data-Driven Predictive Control* by **Shreyas N. B.** is approved for the degree of *Master of Technology.*

Examiner

____

____

____

Supervisor (s)

____

____

____

Chairperson

____

Date: ________________

Place: ________________

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources.

I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission.

I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

I declare that I have not used any of the modern AI tools or any other similar tools for writing the thesis.

____

(Signature)

____

(Name of the student)

____

(Roll No.)

Date: ____________________

Abstract

This thesis introduces a new framework for addressing a geometrically robust least-squares optimization problem, developed in the context of finite-time, data-driven predictive control. Traditional least-squares methods, while foundational in system identification and estimation, often struggle to maintain performance in the presence of model uncertainty or noisy data. To address this, the proposed formulation embeds robustness directly into the optimization process, rather than treating it as an external correction or regularization term. The central idea is to reinterpret the least-squares problem through a geometric lens and formulate it as a minimax problem on a product manifold, allowing for a principled treatment of uncertainty and nonlinearity.

The core formulation considers two sets of variables: one representing the decision variable of interest, and the other representing uncertainty, bounded within a geometric constraint described as a ball. This ball constraint captures possible perturbations or variations in the data, which can arise from measurement noise, modeling errors, or unmodeled system dynamics. By doing so, the method directly encodes robustness against such variations into the optimization problem itself. The resulting minimax structure can be interpreted as the controller or estimator seeking a solution that minimizes the worst-case residual error induced by the uncertainty. This formulation thus bridges ideas from robust optimization, geometric control, and estimation on manifolds.

A key theoretical contribution of this work lies in the explicit solvability of the inner maximization problem. Despite the high-level geometric structure, the maximization over the uncertainty variable admits a closed-form expression, simplifying the overall computation and enabling efficient implementation. This property distinguishes the approach from conventional robust least-squares methods that often rely on iterative or conservative approximations to handle uncertainty. By leveraging the geometry of the manifold and the structure of the ball constraint, the inner problem collapses into a tractable form that preserves interpretability while ensuring robustness.

When applied to data-driven predictive control, the proposed method demonstrates strong performance, particularly for linear time-invariant (LTI) systems whose dynamics are not explicitly known but can be inferred from data. Under mild assumptions of controllability and observability, the algorithm is able to generate predictive control inputs that stabilize the system and track desired trajectories effectively. The finite-time formulation ensures that the optimization remains computationally feasible for online implementation, an essential property for real-time control applications. The method's ability to integrate data-driven modeling with geometric robustness makes it particularly suitable for scenarios where accurate models are unavailable or costly to obtain, such as in aerial robotics, autonomous systems, human-robot interactions and other complex systems.

Beyond its direct application to predictive control, this thesis contributes conceptually to the intersection of geometry and optimization in control theory. By formulating the problem on a product manifold, it emphasizes the role of intrinsic structure in ensuring stability and convergence properties, while the minimax perspective naturally connects to ideas from game theory and robust estimation. Overall, the work provides both theoretical insight and practical algorithms for robust, geometry-aware control, advancing the broader goal of reliable control strategies from uncertain data.

The thesis is organized as follows: Chapter 1 introduces the problem statement and motivation. Chapter 2 reviews relevant literature on robust least-squares methods, behavioral system theory, and data-driven control techniques. Chapter 3 introduces the mathematical preliminaries necessary for understanding the geometric framework, including Riemannian geometry and optimization on manifolds, min-max optimization. Chapter 4 introduces the behavioral systems theory, including system identification and connection to the Grassmannian. Chapter 5 details the application of the proposed method to data-driven predictive control, including algorithmic implementation, performance analysis and application to a simple LTI system. Finally, we conclude with a summary of contributions and suggestions for future research directions.

Acknowledgments

I would like to express my sincere gratitude to my supervisor Prof. Ravi Banavar, for his invaluable guidance, support, and encouragement throughout the course of this research. His insights and expertise have been instrumental in shaping the direction of this thesis.

I would also like to thank my co-supervisors Prof. Cyrus Mostajeran (*NTU Singapore*) and Dr. Bamdev Mishra (*Microsoft*, *India*), for their valuable feedback and suggestions, which have greatly improved the quality of this work.

I am grateful to Prof. Alberto Padoan (*UBC Vancouver*) and Prof. Jeremy Coulson (*UW—Madison*) for generously sharing their expertise in data-driven control and behavioral theory; their insightful discussions and feedback have substantially contributed to the ideas presented in this thesis.

Finally, I would like to acknowledge the support from my Department at the Center for Systems and Controls, Indian Institute of Technology Bombay, which made this research possible.

Contents

1 Introduction

The objective of this thesis is to study the application of robust least-squares optimization in the context of data-driven predictive control. The fundamental idea lies in the behavioral approach to systems theory, which allows us to represent a dynamical system purely based on its observed input-output data, without requiring an explicit parametric model. This approach is particularly useful in scenarios where the underlying system dynamics are complex or unknown.

1.1 Robust Least-Squares

The robust least-squares problem has been extensively studied in the context of optimization and control theory. The foundational work by El Ghaoui and Lebret [1] introduced a robust optimization framework for least-squares problems, addressing uncertainties in the data.

The classical least-squares problem aims to minimize the sum of squared residuals between observed and predicted values. This problem can be formulated as follows:

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2^2,$$

where $A \in \mathbb{R}^{m \times n}$ is the data matrix, $b \in \mathbb{R}^m$ is the observation vector, and $x \in \mathbb{R}^n$ is the vector of unknown parameters to be estimated. This can be visualized in Figure 1.1.

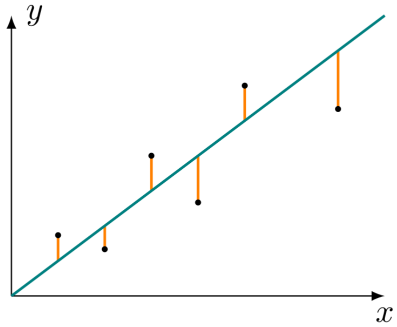


Figure 1.1 / Classical Least-Squares Problem

However, in many practical scenarios, the data matrix A and observation vector b may be subject to uncertainties or perturbations. The robust (or total) least-squares problem extends the classical formulation by considering these uncertainties, leading to a more resilient estimation process:

$$\min_{\Delta A, \Delta b} \|[\Delta A \quad \Delta b]\|_F^2 \quad \text{subject to} \quad (A + \Delta A)x = b + \Delta b,$$

where ΔA and Δb represent the uncertainties in the data matrix and observation vector, respectively. The problem can be visualized in Figure 1.2.

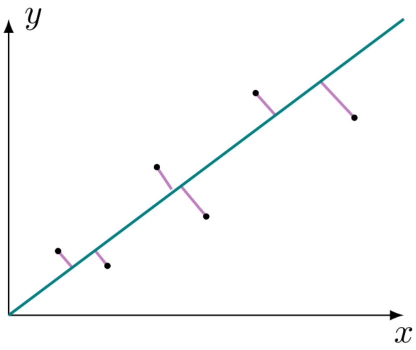


Figure 1.2 / Total Least-Squares Problem

Significant advancements have been made in solving robust least-squares problems, including the development of efficient algorithms and theoretical guarantees for their performance [1]. This framework led to an equivalent reformulation of the robust least-squares problem as a minimax optimization problem, posed as:

$$\min_{x \in \mathbb{R}^n} \max_{A \in \mathbb{B}_\rho(\hat{A})} \|Ax - b\|_2^2,$$

where $\mathbb{B}_\rho(\hat{A})$ denotes a ball of radius ρ centered at the nominal data matrix $\hat{A}$. This reformulation allows for the application of bilevel optimization techniques to solve the robust least-squares problem efficiently, especially when A has a special structure (e.g. elements on a manifold). The geometrically robust least-squares problem can be visualized in Figure 1.3.

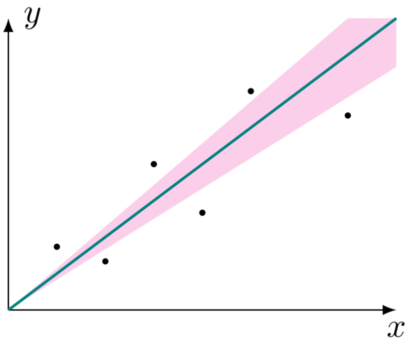
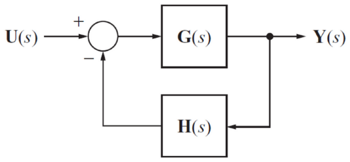


Figure 1.3 / (Geometrically) Robust Least-Squares

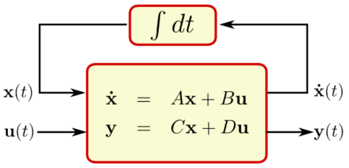
All these variants of least-squares problems indeed have explicit solutions that can be computed efficiently using singular value decomposition (SVD) techniques [2]. However, in practice, we usually encounter constraints on the feasible set of solutions. This leads to constrained robust least-squares problems, which are generally more challenging to solve. Various approaches have been proposed to tackle these problems, including regularization, convex relaxation techniques, iterative algorithms, and heuristic methods.

1.2 Behavioral Approach to System Theory

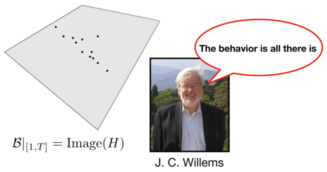
The behavioral approach to system theory, pioneered by Jan C. Willems in the 1980s [3], provides a framework for modeling and analyzing dynamical systems based on their observed behaviors rather than traditional state-space representations. This approach emphasizes the importance of trajectories and the relationships between inputs and outputs, allowing for a more flexible and general representation of systems. The behavioral approach has been applied to various areas, including system identification, control design, and fault detection.



Transfer function


$\int dt$
$\mathbf{x}(t)$
$\mathbf{u}(t)$
$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$
$\mathbf{y} = C\mathbf{x} + D\mathbf{u}$
$\dot{\mathbf{x}}(t)$
$\mathbf{y}(t)$

State-space


$\mathcal{B}|_{[1,T]} = \mathrm{Image}(H)$
The behavior is all there is
J. C. Willems

Behavior

Willems' seminal work laid the foundation for the behavioral approach, introducing key concepts such as the notion of a system's behavior as a set of trajectories and the idea of controllability and observability in this context. Subsequent research expanded on these ideas, exploring various aspects of behavioral systems, including their interconnections, feedback control, and robustness properties [4] [5].

1.3 Data-Driven Control Methods

Data-driven control methods have gained significant attention in recent years due to their ability to leverage data directly for control design, bypassing the need for explicit system identification. These methods are particularly useful in scenarios where obtaining accurate models is challenging or impractical. One prominent class of data-driven control methods is predictive control, which utilizes historical data to predict future system behavior and optimize control actions accordingly. Predictive control methods, such as Model Predictive Control (MPC), have been widely studied and applied in various domains, including process control, robotics, and autonomous systems [6].

Recent advancements in data-driven predictive control via behavioral system theory have led to the development of novel control strategies that directly utilize input-output data for control synthesis. These approaches often involve constructing Hankel matrices from collected data and formulating optimization problems that ensure desired closed-loop properties, such as stability and performance. A notable example is the work by Coulson et al. [7], which introduced a data-driven predictive control (DeePC) framework based on Willems' fundamental lemma. This framework allows for the design of controllers that guarantee closed-loop stability and performance without requiring an explicit model of the system, including complex nonlinear systems like quadcopters.

2 Literature Survey

The purpose of this chapter is to provide a survey of the existing literature related to robust least-squares, behavioral system theory, and data-driven control methods, particularly focusing on predictive control strategies. There has been significant research in this area over the past few decades, leading to various methodologies and frameworks for controlling dynamical systems using data.

Least-squares optimization is a classical problem which has been utilized across many domains in science and engineering over the past few centuries. The earliest documented usage of the problem can be traced back to Legendre (1805) [8], where it was described as an algebraic procedure for fitting linear equations to data. Legendre demonstrated the new method by analyzing the same data as Laplace for the shape of the Earth. On the other hand, Gauss (1809) [9] went beyond Legendre and succeeded in connecting the method of least-squares with the principles of probability. The least-squares technique soon became an indispensable tool in astronomy, geodesy and laid the foundation of many core concepts in modern engineering problems.

2.1 Problem Overview

Robust Least-Squares

The least-squares problem is most commonly seen in the form shown below:

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2^2, \tag{2.1}$$

where $A \in \mathbb{R}^{m \times n}$ is a matrix of coefficients, $b \in \mathbb{R}^m$ is a vector of observations, and $x \in \mathbb{R}^n$ is the vector of unknowns to be determined. Thus, by minimizing the residual $\Delta b = Ax - b$, it determines a solution that closely matches b in the Euclidean 2-norm. As the domain of applied statistics progressed, the total least squares problem [2] was studied, which considered perturbations in both dependent and independent variables, i.e., $(\Delta A, \Delta b)$. The optimal solutions may be sensitive to perturbations in the data (A, b). One way to mitigate this is to consider the robust least squares problem:

$$\min_{x \in \mathbb{R}^n} \max_{A \in \mathbb{B}_\rho^{\mathrm{F}}(\hat{A})} \|Ax - b\|_2^2, \tag{2.2}$$

where $\mathbb{B}_\rho^{\mathrm{F}}(\hat{A}) = \{A \in \mathbb{R}^{m \times n} : \|A - \hat{A}\|_2 \leq \rho\}$ is a ball centered around $\hat{A}$ with radius ρ, endowed with the Frobenius norm.

Building on these ideas, alternative perturbation models have been explored, yielding different robust versions of the least-squares problem. For instance, El

Ghaoui and Lebret (1997) [1] considered the robust least squares problem

$$\min_{x\in\mathbb{R}^n} \max_{[A\ \ b]\in\mathbb{B}^{\mathrm{F}}_{\rho}([\hat{A}\ \ \hat{b}])} \|Ax-b\|_2^2. \tag{2.3}$$

This approach is motivated by scenarios where the exact data (A, b) are unknown, but belong to a family of matrices $(\hat{A}+\Delta A, \hat{b}+\Delta b)$ and the residual $[\Delta A\ \Delta b]$ lies in a norm-bounded matrix ball.

In this thesis, we introduce a robust optimization framework that accounts for the geometric nature of perturbations found in diverse instances of the problem. Specifically, we consider the optimization problem,

$$\min_{x\in\mathbb{R}^n} \max_{\mathcal{S}\in\mathbb{B}^{d}_{\rho}(\hat{\mathcal{S}})} \|P_{\mathcal{S}}x-b\|_2^2, \tag{2.4}$$

where $P_{\mathcal{S}}$ is the orthogonal projection onto the k-dimensional subspace $\mathcal{S}$ of $\mathbb{R}^n$, and $\mathbb{B}^{d}_{\rho}(\hat{\mathcal{S}})$ is a ball centered at $\hat{\mathcal{S}}$ with radius ρ defined by the metric d on the *Grassmannian* $\mathrm{Gr}(k, n)$, which is the set of all k-dimensional subspaces in $\mathbb{R}^n$ endowed with the structure of a smooth Riemannian manifold. This approach is motivated by a diverse range of applications where the linear model A is a matrix representation of a subspace subject to bounded perturbations (due to uncertainty or approximations errors), quantified naturally in terms of distances between subspaces.

Data-Driven Predictive Control

The availability of large datasets coupled with unprecedented storage and computational power has recently reignited interest in direct data-driven control methods, which aim to infer optimal decisions directly from measured data (bypassing system identification). At the heart of this emerging trend lies the behavioral approach to system theory (Willems, 2007 [10]) and a seminal result by Willems and his collaborators [11], commonly referred to as the *fundamental lemma.* The lemma establishes that finite-horizon behaviors of Linear Time-Invariant (LTI) systems can be represented as images of raw data matrices.

The proposed idea behind data-driven predictive control via the geometric approach is posed as:

(1) Collect a set of input-output trajectories of the system, denoted by $\{u_k, y_k\}_{k=0}^{N-1}$, where $u_k \in \mathbb{R}^m$ and $y_k \in \mathbb{R}^p$ are the input and output at time step k, respectively. Store trajectories as $w_k = [u_k^\top\ y_k^\top]^\top \in \mathbb{R}^q$ where $q = m + p$.

(2) Construct the Hankel matrix of depth L from the collected data:

$$\mathcal{H}_L(w) = \begin{bmatrix} w_0 & w_1 & w_2 & \cdots & w_{N-L} \\ w_1 & w_2 & w_3 & \cdots & w_{N-L+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{L-1} & w_L & w_{L+1} & \cdots & w_{N-1} \end{bmatrix} \in \mathbb{R}^{qL\times(N-L+1)}.$$

(3) According to the fundamental lemma, if the input sequence $\{u_k\}$ is persistently exciting of order $L+n$ (where n is the order of the system), then any trajectory of length L can be expressed as a linear combination of the columns of $\mathcal{H}_L(w)$. In other words, the behavior $\mathfrak{B}_L$ of the system over a horizon L is given by:

$$\mathfrak{B}_L = \mathrm{im}(\mathcal{H}_L(w))$$

We identify $\mathfrak{B}_L$ as a k-dimensional subspace of $\mathbb{R}^{qL}$, where $k \leq qL$ is the rank of $\mathcal{H}_L(w)$. Thus, it is an element of the Grassmannian $\mathrm{Gr}(k, qL)$, denoted as $\mathcal{S} \equiv \mathfrak{B}_L$.

(4) At each time step t, solve a constrained least-squares problem to find the optimal trajectory that minimizes the cost function while satisfying the system's behavior. The optimization problem can be formulated as:

$$\min_{x \in \mathbb{R}^{qL}} \max_{\mathcal{S} \in \mathbb{B}^d_\rho(\hat{\mathcal{S}})} \|P_{\mathcal{S}} x - b\|_2^2, \quad \text{s.t.} \quad P_{\mathcal{S}} x \in \mathfrak{B}_L$$

As a result, various data-driven modeling, estimation, filtering, and control problems can be formulated as weighted or constrained least squares problems (Markovsky and Dörfler, 2021 [12]). Moreover, being finite-dimensional subspaces, finite-horizon LTI behaviors can be identified with points on the Grassmannian $\mathrm{Gr}(k, n)$ and uncertainty can be naturally quantified using Grassmannian (subspace) metrics. This approach has demonstrated its effectiveness in data-driven mode recognition and control applications (DeePC [7]) and shows promise to open new avenues in adaptive control (Padoan et al., 2022 [5]).

In summary, the main focus of this thesis is to explore the robust least-squares problem with subspace uncertainty and its application in data-driven predictive control via behavioral systems theory. We investigate the theoretical foundations of this approach, develop efficient algorithms for solving the robust optimization problem, and demonstrate its effectiveness through numerical simulations and real-world applications. The ultimate goal is to provide a geometric approach to robust and reliable framework for data-driven control that can handle uncertainties in the system dynamics and improve the performance of control systems in practice.

3 Preliminaries

This chapter introduces mathematical concepts and notations related to optimization theory, behavioral approach to systems theory, and data-driven predictive control that will be used throughout this thesis.

3.1 Geometry

In this section, we briefly review some fundamental concepts from Riemannian geometry and optimization on manifolds. For a more comprehensive treatment, the reader is referred to [13] [14].

Euclidean Spaces

Definition 3.1.1 (Inner Product). An inner product on a real vector space $\mathcal{E}$ is a function $\langle\cdot,\cdot\rangle : \mathcal{E} \times \mathcal{E} \to \mathbb{R}$ that satisfies the following properties for all $u, v, w \in \mathcal{E}$ and $a, b \in \mathbb{R}$:

- Symmetry: $\langle u, v\rangle = \langle v, u\rangle$,
- Linearity: $\langle au + bv, w\rangle = a\langle u, w\rangle + b\langle v, w\rangle$,
- Positive-definiteness: $\langle u, u\rangle \geq 0$ and $\langle u, u\rangle = 0 \iff u = 0$.

Definition 3.1.2 (Euclidean Space). A linear space $\mathcal{E}$ equipped with an inner product $\langle\cdot,\cdot\rangle$ is called a Euclidean space. An inner product induces a norm on $\mathcal{E}$ called the Euclidean norm:

$$\|u\| = \sqrt{\langle u, u\rangle}, \quad \forall u \in \mathcal{E}.$$

The standard inner product on $\mathbb{R}^n$ and the associated norm are given by:

$$\langle u, v\rangle = u^\top v, \quad \|u\|_2 = \sqrt{u^\top u}, \quad \forall u, v \in \mathbb{R}^n. \tag{3.1}$$

Similarly, the standard inner product on the space of real matrices $\mathbb{R}^{n\times k}$ is the Frobenius inner product, with the associated Frobenius norm:

$$\langle A, B\rangle = \operatorname{Tr}(A^\top B), \quad \|A\|_{\mathrm{F}} = \sqrt{\operatorname{Tr}(A^\top A)}, \quad \forall A, B \in \mathbb{R}^{n\times k}, \tag{3.2}$$

where $\operatorname{Tr}(M) = \sum_i M_{ii}$ denotes the trace of a matrix. We often use the following properties of the above inner product, with matrices U, V, W, A, B of compatible sizes:

$$\begin{aligned} \langle U, V\rangle &= \langle U^\top, V^\top\rangle \\ \langle AB, W\rangle &= \langle A, WB^\top\rangle = \langle B, A^\top W\rangle. \end{aligned} \tag{3.3}$$

Definition 3.1.3 (Differential). Let U, V be two open sets in two Euclidean spaces $\mathcal{E}_1$ and $\mathcal{E}_2$, respectively. A map $F : U \to V$ is said to be *smooth* if it is infinitely differentiable (class $\mathcal{C}^\infty$) on its domain.

Also, F is said to be *differentiable* at a point $x \in U$ if there exists a neighborhood $U'$of x such that $F \mid_{U'}$ is smooth. Thus, if $F : U \to V$ is smooth at x, then its *differential* at x, is the linear map $\mathrm{D}F(x) : \mathcal{E}_1 \to \mathcal{E}_2$ defined as:

$$\mathrm{D}F(x)[v] = \frac{d}{dt} F(x+tv)\bigg|_{t=0} = \lim_{t\to 0} \frac{F(x+tv) - F(x)}{t}$$

Definition 3.1.4 (Euclidean Gradient). Consider a smooth function $f : \mathcal{E} \to \mathbb{R}$, where $\mathcal{E}$ is a linear space. The *Euclidean gradient* with respect to an inner product $\langle \cdot, \cdot \rangle : \mathcal{E} \times \mathcal{E} \to \mathbb{R}$, denoted by $\boldsymbol{\nabla} f(x)$ is a unique element of $\mathcal{E}$ such that, for all $v \in \mathcal{E}$,

$$\langle v, \boldsymbol{\nabla} f(x) \rangle = \mathrm{D}f(x)[v], \quad x, v \in \mathcal{E},$$

where $\mathrm{D}f(x) : \mathcal{E} \to \mathbb{R}$ is the differential of f at x.

Definition 3.1.5 (Euclidean Hessian). The *Euclidean Hessian* of a smooth function $f : \mathcal{E} \to \mathbb{R}$ is the linear map $\mathrm{Hess}f(x) : \mathcal{E} \to \mathcal{E}$ defined as:

$$\mathrm{Hess}f(x)[v] = \mathrm{D}(\boldsymbol{\nabla} f)(x)[v] = \frac{d}{dt} \boldsymbol{\nabla} f(x+tv)\bigg|_{t=0}, \quad \forall x, v \in \mathcal{E}.$$

Definition 3.1.6 (Embedded Submanifold). Let $\mathcal{E}$ be a linear space of dimension k. A non-empty subset $\mathcal{M}$ of $\mathcal{E}$ is a (smooth) *embedded submanifold* of $\mathcal{E}$ of dimension n if either of the following conditions hold:

(1) $n = k$ and $\mathcal{M}$ is open in $\mathcal{E}$, also called an open submanifold; or

(2) $n = k - q$ for some $q \geq 1$ and, for each $x \in \mathcal{M}$, there exists an open neighborhood U of x in $\mathcal{E}$ and a smooth map $h : U \to \mathbb{R}^q$ such that:

(a) If y is in U, then $h(y) = 0$ if and only if $y \in \mathcal{M}$; and

(b) rank $\mathrm{D}h(x) = q$.

Such a function h is called a *local defining function* for $\mathcal{M}$ at x.

Definition 3.1.7 (Tangent Space). Let $\mathcal{M}$ be a subset of $\mathcal{E}$. For all $x \in \mathcal{M}$, the *tangent space* to $\mathcal{M}$ at x, denoted by $T_x\mathcal{M}$, is defined as the set of all vectors that are tangents at x to smooth curves on $\mathcal{M}$, i.e.,

$$T_x\mathcal{M} = \{\gamma'(0) \mid \gamma : (-\varepsilon, \varepsilon) \to \mathcal{M} \text{ is a smooth curve with } \gamma(0) = x\}.$$

Definition 3.1.8 (Tangent Bundle). The *tangent bundle* of a manifold $\mathcal{M}$ is the set of all tangent spaces over all points in $\mathcal{M}$:

$$T\mathcal{M} = \{(x, v) \mid x \in \mathcal{M}, v \in T_x\mathcal{M}\}.$$

Riemannian Geometry

Definition 3.1.9 (Differential). The differential of $F : \mathcal{M} \to \mathcal{M}'$ at the point $x \in \mathcal{M}$ is the linear map $DF(x) : T_x\mathcal{M} \to T_{F(x)}\mathcal{M}'$ defined as:

$$\mathrm{D}F(x)[v] = \frac{d}{dt} F(\gamma(t))\Big|_{t=0} = (F \circ \gamma)'(0),$$

Let $\mathcal{M}$ and $\mathcal{M}'$ be two embedded submanifolds of two Euclidean spaces $\mathcal{E}$ and $\mathcal{E}'$, respectively. Then, the map $F : \mathcal{M} \to \mathcal{M}'$ admits a smooth extension $\bar{F} : U \to \mathcal{E}'$ defined on an open neighborhood U of $\mathcal{M}$ in $\mathcal{E}$. Thus, for all $x \in \mathcal{M}$ and $v \in T_x\mathcal{M}$, the differential of F at x can be computed as:

$$\mathrm{D}F(x) = \mathrm{D}\bar{F}(x) \mid_{T_x\mathcal{M}} .$$

Definition 3.1.10 (Riemannian Metric). A metric $\langle \cdot, \cdot \rangle$ on $T_x\mathcal{M}$ is a *Riemannian metric* if it varies smoothly with x, in the sense that for all smooth vector fields V, W on $\mathcal{M}$, the function $x \mapsto \langle V(x), W(x) \rangle_x$ is smooth.

Definition 3.1.11 (Riemannian Manifold). A *Riemannian manifold* is a smooth manifold $\mathcal{M}$ endowed with a Riemannian Metric $\langle \cdot, \cdot \rangle$.

Definition 3.1.12 (Riemannian Submanifold). An embedded submanifold $\mathcal{M}$ of a Euclidean space $\mathcal{E}$ is a *Riemannian submanifold* if it is endowed with the Riemannian metric induced by the inner product of $\mathcal{E}$, i.e., for all $x \in \mathcal{M}$ and $u, v \in T_x\mathcal{M}$,

$$\langle u, v \rangle_x = \langle u, v \rangle,$$

where $\langle \cdot, \cdot \rangle$ is the Inner Product on $\mathcal{E}$.

Definition 3.1.13 (Riemannian Gradient). Let $f : \mathcal{M} \to \mathbb{R}$ be smooth on a Riemannian manifold $\mathcal{M}$. The *Riemannian gradient* of f is the vector field $\mathrm{grad} f$ on $\mathcal{M}$ uniquely defined by the following identities:

$$\forall (x, v) \in \mathrm{T}\mathcal{M}, \quad \mathrm{D}f(x)[v] = \langle v, \mathrm{grad} f(x) \rangle_x,$$

where $\mathrm{D}f(x) : T_x\mathcal{M} \to \mathbb{R}$ is the differential of f at x, and $\langle \cdot, \cdot \rangle_x$ is the Riemannian metric on $T_x\mathcal{M}$.

Definition 3.1.14 (Projection). Let $\mathcal{M}$ be an embedded submanifold of a Euclidean space $\mathcal{E}$ equipped with a Euclidean metric $\langle \cdot, \cdot \rangle$. The *orthogonal projection* onto the tangent space $T_x\mathcal{M}$ at a point $x \in \mathcal{M}$ is the linear map $P_x^{\perp} : \mathcal{E} \to T_x\mathcal{M}$ which satisfies the following properties:

(1) *Range*: $\mathrm{im}(P_x^{\perp}) = T_x\mathcal{M}$;

(2) *Projector*: $P_x^{\perp} \circ P_x^{\perp} = P_x^{\perp}$;

(3) *Orthogonal*: $\langle u - P_x^{\perp}(u), v \rangle = 0$, for all $u \in \mathcal{E}$ and $v \in T_x\mathcal{M}$.

Lemma 3.1.1. Let $\mathcal{M}$ be an embedded submanifold of a Euclidean space $\mathcal{E}$ equipped with a Euclidean metric $\langle\cdot,\cdot\rangle$. For a smooth function $f : \mathcal{M} \to \mathbb{R}$, the Riemannian gradient at a point $x \in \mathcal{M}$ is given by:

$$\operatorname{grad} f(x) = P_x^{\perp}(\boldsymbol{\nabla} \bar{f}(x)),$$

where $\bar{f}$ is a smooth extension of f to an open neighborhood of $\mathcal{M}$ in $\mathcal{E}$, and $\boldsymbol{\nabla}\bar{f}(x)$ is the Euclidean gradient of $\bar{f}$ at x.

Grassmannians

The Grassmannian $\operatorname{Gr}(k,n)$ is the set of all k-dimensional subspaces of $\mathbb{R}^n$. It can be endowed with the structure of a smooth Riemannian Manifold of dimension $k(n-k)$. Each point $\mathcal{S} \in \operatorname{Gr}(k,n)$ can be represented by an orthogonal matrix $Y \in \mathbb{R}^{n\times k}$ such that $Y^\top Y = \mathbb{I}_k$.

The Grassmannian is a quotient manifold of the Stiefel manifold $\operatorname{St}(k,n)$, which is the set of all orthogonal $n \times k$ matrices (embedded submanifold of $\mathbb{R}^{n\times k}$). The equivalence class of $Y \in \operatorname{St}(k,n)$ is given by $[Y] = \{YQ \mid Q \in O(k)\}$, where $O(k)$ is the orthogonal group of $k\times k$ orthogonal matrices. Thus, $\operatorname{Gr}(k,n) \cong \operatorname{St}(k,n)/O(k)$ (refer [14] for more details).

The projection operator onto the subspace $\mathcal{S}$ is given by $P_{\mathcal{S}} = YY^\top$ and the orthogonal complement projection is $P_{\mathcal{S}}^{\perp} = \mathbb{I}_n - YY^\top$.

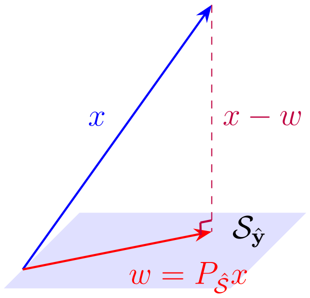

x
$x - w = P_{\hat{\mathcal{S}}}^{\perp}x$
$\mathcal{S}_{\hat{\mathbf{y}}}$
$w = P_{\hat{\mathcal{S}}}x$

Figure 3.4 / Depiction of the orthogonal projection of a vector onto a subspace $\hat{\mathcal{S}}$.

Definition 3.1.15 (Chordal Distance). The *chordal distance* between two subspaces $\mathcal{S}_1, \mathcal{S}_2 \in \operatorname{Gr}(k,n)$ is defined as:

$$d_2(\mathcal{S}_1,\mathcal{S}_2) = \sqrt{\operatorname{Tr}\left(P_{\mathcal{S}_1}^{\perp}P_{\mathcal{S}_2}\right)} = \frac{1}{\sqrt{2}}\|P_{\mathcal{S}_1} - P_{\mathcal{S}_2}\|_{\mathrm{F}} = \sqrt{\sum_{i=1}^{k}\sin^2(\theta_i)},$$

where $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm, and $0 \le \theta_1 \le \cdots \le \theta_k \le \pi/2$ are the principal angles between $\mathcal{S}_1$ and $\mathcal{S}_2$.

Definition 3.1.16 (Gap Distance). The *gap distance* between two subspaces $\mathcal{S}_1, \mathcal{S}_2 \in \operatorname{Gr}(k,n)$ is defined as:

$$d_\infty(\mathcal{S}_1,\mathcal{S}_2) = \|P_{\mathcal{S}_1} - P_{\mathcal{S}_2}\|_2 = \sin(\theta_k),$$

where $\|\cdot\|_2$ is the spectral norm, and θ_k is the largest principal angle between $\mathcal{S}_1$ and $\mathcal{S}_2$.

3.2 Optimization

Definition 3.2.1 (Convexity). Let $\mathcal{X}$ be a real vector space. Let $f : \mathcal{X} \to \mathbb{R}$ be a function, such that $x \mapsto f(x)$. Then, f is convex over $\mathcal{X}$, for all $x \in \mathcal{X}$, if and only if

$$f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2),$$

for all $0 \leq t \leq 1$, and all $x_1, x_2 \in \mathcal{X}$.

Definition 3.2.2 (Constrained Optimization). For the constrained minimization problem,

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{subject to} \quad \begin{cases} c_i(x) = 0, & i \in \mathcal{E}, \\ c_i(x) \leq 0, & i \in \mathcal{I}, \end{cases} \tag{3.5}$$

where f and the functions c_i are all smooth, real-valued functions on a subset of $\mathbb{R}^n$, and $\mathcal{I}$ and $\mathcal{E}$ are two finite sets of indices. We call f the *objective function*, while c_i, $i \in \mathcal{E}$ are the equality constraints and c_i, $i \in \mathcal{I}$ are the inequality constraints. We define the *feasible set* Ω to be the set of points x that satisfy the constraints; that is,

$$\Omega = \{x \mid c_i(x) = 0, i \in \mathcal{E}; c_i(x) \leq 0, i \in \mathcal{I}\},$$

so that we rewrite the problem more compactly as,

$$\min_{x \in \Omega} f(x).$$

Definition 3.2.3 (Lagrangian function). The *Lagrangian* function for (3.5) is defined as:

$$\mathcal{L}(x, \lambda) = f(x) + \sum_{i \in \mathcal{I} \cap \mathcal{E}} \lambda_i c_i(x)$$

Definition 3.2.4 (First-Order Optimality Conditions). Let x^* be a local solution of (3.5), that the functions f and c_i in (3.5) are continuously differentiable. Then there exists a Lagrange multiplier λ^*, with components $\lambda_i^*, i \in \mathcal{E} \cap \mathcal{I}$, such that the following conditions are satisfied at (x^*, λ^*),

$$\nabla_x \mathcal{L}(x^*, \lambda^*) = 0, \tag{3.6a}$$

$$c_i(x^*) = 0, \quad i \in \mathcal{E}, \tag{3.6b}$$

$$c_i(x^*) \leq 0, \quad i \in \mathcal{I}, \tag{3.6c}$$

$$\lambda^* \geq 0, \quad i \in \mathcal{I}, \tag{3.6d}$$

$$\lambda_i^* c_i(x^*) = 0, \quad i \in \mathcal{E} \cap \mathcal{I}. \tag{3.6e}$$

These conditions (3.6) are often known as the *KKT (Karush-Kuhn-Tucker) conditions*. The conditions (3.6e) are *complementarity slackness conditions*; they imply that either constraint i is active or $\lambda_i^* = 0$, or possibly both.

Definition 3.2.5 (Min-Max Optimization). In general, the min-max problems (see [15], [16]) on Riemannian manifolds that we focus on, are of the kind

$$\min_{x \in \mathcal{M}_x} \max_{y \in \mathcal{M}_y} f(x, y),$$

where f is atleast $\mathcal{C}^2$ and $\mathcal{M}_x, \mathcal{M}_y$ are the Riemannian manifolds containing x, y respectively. Without loss of generality, we assume that minimization takes place first, followed by maximization (cannot be interchanged). These problems are termed *min-max optimization*, whereas the candidate solution points are termed *minimax points*.

Definition 3.2.6 (Global Minimax point). Let $\mathcal{M}_x, \mathcal{M}_y$ be two smooth Riemannian manifolds. Consider two subsets $\mathcal{X}, \mathcal{Y}$ such that $\mathcal{X} \times \mathcal{Y} \subseteq \mathcal{M}_x \times \mathcal{M}_y$. Let $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, be a function such that $(x, y) \mapsto f(x, y)$. The point (x^*, y^*) is called a global minimax point if for any $(x, y) \in \mathcal{X} \times \mathcal{Y}$, it satisfies:

$$f(x^*, y) \leq f(x^*, y^*) \leq \max_{y' \in \mathcal{Y}} f(x, y').$$

Definition 3.2.7 (Local Minimax point). A point (x^*, y^*) is a local minimax point of $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ if there exists some $\delta_0 > 0$, and a function h such that $\lim_{\delta \to 0} h(\delta) = 0 \ \forall \ \delta \in (0, \delta_0]$ and for every $(x, y) \in \mathcal{X} \times \mathcal{Y}$ satisfying $d(x, x^*) \leq \delta$ and $d(y, y^*) \leq \delta$ such that

$$f(x^*, y) \leq f(x^*, y^*) \leq \max_{y' : d(y, y') \leq h(\delta)} f(x, y').$$

Robust Least-Squares

Consider the robust least-squares problem:

$$\min_{x \in \mathbb{R}^n} \max_{\mathcal{S} \in \mathbb{B}^d_\rho(\hat{\mathcal{S}})} \|YY^\top x - b\|^2, \tag{3.7}$$

where $\mathbb{B}^d_\rho(\hat{S}) = \{\mathcal{S} \in \mathrm{Gr}(k, n) \mid d(\mathcal{S}, \hat{\mathcal{S}}) \leq \rho\}$ is a ball on the Grassmannian centered at the subspace $\hat{\mathcal{S}}$ with radius $\rho > 0$, and $d(\cdot, \cdot)$ is a distance metric on the Grassmannian. For purposes of this discussion, we consider the chordal distance $d_2(\cdot, \cdot)$ (see 3.1.15 for definition).

Convexity

The robust least-squares problem in (3.7) is a non-concave in the inner maximization variable $\mathcal{S}$. However, it can be shown that the inner problem has an explicit solution, and hence as long as the outer minimization problem is convex, the overall problem remains tractable.

Lemma 3.2.1. The robust least-squares problem in (3.7) is convex in the outer minimization variable x.

Proof. Let $f(x, Y) = \|YY^\top x - b\|^2$ denote the cost function in (3.7). To show that f is convex in x, we need to show that for any $x_1, x_2 \in \mathbb{R}^n$ and $0 \leq t \leq 1$, the following inequality holds (see 3.2.1):

$$f(tx_1 + (1-t)x_2, Y) \leq tf(x_1, Y) + (1-t)f(x_2, Y).$$

Expanding the left-hand side, we have:

$$\begin{aligned} f(tx_1 + (1-t)x_2, Y) &= \|YY^\top(tx_1 + (1-t)x_2) - b\|^2 \\ &= \|t(YY^\top x_1 - b) + (1-t)(YY^\top x_2 - b)\|^2. \end{aligned}$$

Using the convexity of the squared norm (using *Jensen's inequality*), we have:

$$\begin{aligned} f(tx_1 + (1-t)x_2, Y) &\leq t\|YY^\top x_1 - b\|^2 + (1-t)\|YY^\top x_2 - b\|^2 \\ &= tf(x_1, Y) + (1-t)f(x_2, Y). \end{aligned}$$

Thus, $f(x, Y)$ is convex in x for any fixed Y. Since the maximum of convex functions is also convex (*preservation of convexity over maximization*), the overall problem in (3.7) is convex in x. □

Proposition 3.2.1. If $(x^*, \mathcal{S}^*)$ is a local minimax point of the robust least-squares problem, then it is also a local minimax point of the following problem:

$$\min_{x \in \mathbb{R}^n} \max_{\mathcal{S} \in \mathbb{B}_\rho^{d_2}(\hat{\mathcal{S}})} \operatorname{tr}\Big(Y^\top B(x, \lambda) Y\Big),$$

where $B(x, \lambda) = A(x) + \lambda \hat{Y}\hat{Y}^\top$, $A(x) = (xx^\top - bx^\top - xb^\top)$, and $\lambda \geq 0$ is a *KKT* multiplier.

Proof. Let $f(x, Y) = \|YY^\top x - b\|^2$ denote the cost function in (3.7). The ball constraint can be expressed as $g(Y) = d_2^2(Y, \hat{Y}) - \rho^2 \leq 0$. The Lagrangian function for the inner maximization problem is given by:

$$\mathcal{L}(x, Y, \lambda) = f(x, Y) - \lambda g(Y),$$

where $\lambda \geq 0$ is the *KKT* multiplier (see (3.6) for details). Expanding (3.7), and ignoring constants, we have:

$$\begin{aligned} \mathcal{L}(x, Y, \lambda) &= \langle YY^\top x - b, YY^\top x - b\rangle - \lambda\Big(\operatorname{tr}(Y^\top(\mathbb{I} - \hat{Y}\hat{Y}^\top)Y) - \rho^2\Big) \\ &= \operatorname{tr}\Big(Y^\top A(x) Y\Big) - \lambda\Big(k - \operatorname{tr}\Big(Y^\top \hat{Y}\hat{Y}^\top Y\Big) - \rho^2\Big) \\ &= \operatorname{tr}\Big(Y^\top\Big(A(x) + \lambda\hat{Y}\hat{Y}^\top\Big)Y\Big) \\ &= \operatorname{tr}\Big(Y^\top B(x, \lambda) Y\Big), \end{aligned}$$

where $A(x)$ and $B(x, \lambda)$ are defined as above. □

Lemma 3.2.2. Let $B \in \mathbb{R}^{n \times n}$ be a symmetric matrix. Define the function $h : \operatorname{Gr}(k, n) \to \mathbb{R}$ such that $h(\mathcal{S}) = \operatorname{tr}(Y^\top BY)$, where $Y \in \mathbb{R}^{n \times k}$ is any orthogonal basis of the subspace $\mathcal{S}$. Then, $\max_{\mathcal{S} \in \operatorname{Gr}(k,n)} h(\mathcal{S})$ has an explicit solution such that

$$\mathcal{S}^* = \{\text{top-k eigenspace of } B\}$$

Proof. Let the eigen-decomposition of B be given by $B = U\Lambda U^\top$, where $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$ with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ being the eigenvalues of B, and $U = [u_1, u_2, \ldots, u_n]$ is the orthogonal matrix of corresponding eigenvectors. For any subspace $\mathcal{S} \in \operatorname{Gr}(k, n)$ with orthogonal basis Y, we can express Y in terms of the eigenvectors of B as $Y = UQ$, where $Q \in \mathbb{R}^{n\times k}$ satisfies $Q^\top Q = I_k$. Then,

$$\begin{aligned} h(\mathcal{S}) &= \operatorname{tr}(Y^\top BY) = \operatorname{tr}(Q^\top U^\top BUQ) = \operatorname{tr}(Q^\top \Lambda Q) \\ &= \sum_{i=1}^{k} q_i^\top \Lambda q_i = \sum_{i=1}^{k}\sum_{j=1}^{n} \lambda_j (q_{ij})^2, \end{aligned}$$

where q_i is the i-th column of Q. To maximize $h(\mathcal{S})$, we need to choose Q such that it maximizes the sum of the weighted eigenvalues. The maximum is achieved when the columns of Q are chosen to be the top-k eigenvectors of B. Thus, the optimal subspace $\mathcal{S}^*$ is spanned by the top-k eigenvectors of B, i.e.,

$$Y^* = \begin{bmatrix} | & | & & | \\ | & | & & | \\ u_1 & u_2 & \cdots & u_k \\ | & | & & | \\ | & | & & | \end{bmatrix}$$

where u_i is the i-th eigenvector of B. The optimal subspace is thus given by

$$\mathcal{S}^* = \operatorname{span}\{u_1, u_2, \ldots, u_k\} = \operatorname{span}\{Y^*\}.$$

□

Lemma 3.2.3. If $(x^*, \mathcal{S}^*)$ is a local minimax point of the robust least-squares problem (5.5), then the *KKT* conditions for both the inner and outer problems are satisfied at $(x^*, \mathcal{S}^*)$ with some *KKT* multiplier $\lambda^* \geq 0$.

Proof. Since $(x^*, \mathcal{S}^*)$ is a local minimax point of the robust least-squares problem (3.7), it satisfies the first-order optimality conditions for both the inner maximization and outer minimization problems. Constructing the Lagrangian function for the inner maximization problem as in the previous proposition (3.2.1), we have:

$$\mathcal{L}(x, \mathcal{S}, \lambda) = \operatorname{tr}\Big(Y^\top B(x, \lambda) Y\Big),$$

where $B(x, \lambda) = A(x) + \lambda \hat{Y}\hat{Y}^\top$. The first-order optimality conditions for the inner maximization problem require that the gradient of the Lagrangian with respect to $\mathcal{S}$ vanishes at $\mathcal{S}^*$, i.e.,

$$\begin{aligned} \operatorname{grad}_{\mathcal{S}} \mathcal{L}(x^*, \mathcal{S}^*, \lambda^*) &= 0, \\ P_{\mathcal{S}^*}^{\perp} \boldsymbol{\nabla}_Y \bar{\mathcal{L}}(x^*, Y^*, \lambda^*) &= 0, \\ P_{\mathcal{S}^*}^{\perp}(2B(x^*, \lambda^*)Y^*) &= 0, \\ \implies B(x^*, \lambda^*)Y^* &= Y^* {Y^*}^\top B(x^*, \lambda^*)Y^*. \end{aligned}$$

Hence, the columns of Y^* are eigenvectors of $B(x^*, \lambda^*)$, corresponding to the top-k eigenvalues (from 3.2.2).

Similarly, the first-order optimality conditions for the outer minimization problem require that the gradient of the Lagrangian with respect to x vanishes at x^*, i.e.,

$$\begin{aligned} \boldsymbol{\nabla}_x \mathcal{L}(x^*, \mathcal{S}^*, \lambda^*) &= 0 \\ \boldsymbol{\nabla}_x \mathrm{tr}\Big(Y^{*\top} B(x^*, \lambda^*) Y^* \Big) &= 0 \\ P_{\mathcal{S}^*} x^* = P_{\mathcal{S}^*} b \end{aligned}$$

□

Lemma 3.2.4. If $\lambda > 0$, then the optimal subspace $\mathcal{S}^*$ for the inner problem lies on the boundary of the ball $\mathbb{B}_\rho(\hat{\mathcal{S}})$.

Proof. Assume, for the sake of contradiction, that the optimal subspace $\mathcal{S}^*$ lies in the interior of the ball $\mathbb{B}_\rho^{d_2}(\hat{\mathcal{S}})$. Then, we have $d_2(\mathcal{S}^*, \hat{\mathcal{S}}) < \rho$, which implies that $g(\mathcal{S}^*) < 0$. From the *KKT* conditions (see (3.6)), we have the complementary slackness condition $\lambda g(\mathcal{S}^*) = 0$. Since $\lambda > 0$, it follows that $g(\mathcal{S}^*)$ must be equal to zero, which contradicts our assumption that $\mathcal{S}^*$ lies in the interior of the ball. Therefore, the optimal subspace $\mathcal{S}^*$ must lie on the boundary of the ball $\mathbb{B}_\rho^{d_2}(\hat{\mathcal{S}})$. □

Remark 1. Practically, the *KKT* multiplier λ can be interpreted as a regularization parameter that balances the trade-off between fitting the data and adhering to the constraint defined by the ball $\mathbb{B}_\rho^{d}(\hat{\mathcal{S}})$. A larger value of λ places more emphasis on satisfying the constraint, while a smaller value allows for more flexibility in fitting the data. However, the choice of λ value is not trivial and must be tuned based on satisfying the boundary condition in 3.2.4. This is typically implemented using techniques like line-search or bisection methods.

Constrained Robust Least-Squares

From the perspective of application in predictive control, it is often desirable to impose a specific type of constraint, called the *past constraint*. Thus, the constrained robust least-squares problem is formulated as:

$$\min_{x \in \mathbb{R}^n} \max_{\mathcal{S} \in \mathbb{B}_\rho^d(\hat{\mathcal{S}})} \|YY^\top x - b\|^2 \quad \text{subject to} \quad MYY^\top x = z, \tag{3.8}$$

where $M \in \mathbb{R}^{d \times n}$ and $z \in \mathbb{R}^d$ define the equality constraint. The *KKT* conditions for this problem can be derived similarly to the unconstrained case, with the addition of Lagrange multipliers for the equality constraints.

Note. Typically, $d < k$ and M is a full row-rank matrix such that $M = [\mathbb{I}_d, 0]$ is a selection matrix. This constraint ensures that the first d components of the projected vector $YY^\top x$ match the given vector z. This is particularly useful in scenarios such as in model predictive control applications where past states or inputs are fixed.

Remark 2. Although M is always a full row-rank matrix, the matrix $C = MYY^\top$ may not be full row-rank for all orthogonal matrices Y. This is usually undesirable as it may lead to infeasibility of the equality constraint $Cx = z$. We will see how to address this issue using *regularization* techniques.

Proposition 3.2.2. The explicit solution $\mathcal{S}^*$ to the inner maximization problem in (3.8) is given by

$$\mathcal{S}^* = \{\text{top-k eigenspace of } B_c(x, \lambda^*, \gamma^*)\}$$

for some *KKT* multipliers $\lambda, \{\gamma_i\}_{i=1}^d$, where

$$B_c(x, \lambda, \gamma) = A(x) + \lambda \hat{Y}\hat{Y}^\top + M^\top \gamma x^\top$$

Proof. The problem (3.8) can be posed as

$$\begin{aligned} &\min_{x \in \mathbb{R}^n} \max_{\mathcal{S} \in \mathrm{Gr}(k,n)} f(x, Y) \\ &\text{subject to} \quad h(x, Y) = 0 \quad \text{and} \quad g(Y) \leq 0, \end{aligned} \tag{3.9}$$

where the following functions are defined as:

$$\begin{aligned} f(x, Y) &= \|YY^\top x - b\|^2 \\ h(x, Y) &= MYY^\top x - z \\ g(Y) &= d_2^2(Y, \hat{Y}) - \rho^2 \end{aligned}$$

Constructing the Lagrangian function,

$$\bar{\mathcal{L}}(x, Y, \lambda, \gamma) = f(x, Y) - \lambda g(Y) + \gamma^\top h(x, Y)$$

which satisfies the *KKT* stationarity condition

$$\mathrm{grad}_{\mathcal{S}} \mathcal{L}(x^*, \mathcal{S}^*, \lambda^*, \gamma^*) = P_{\mathcal{S}^*}^\perp \boldsymbol{\nabla}_Y \bar{\mathcal{L}}(x^*, Y^*, \lambda^*, \gamma^*) = 0$$

Using Proposition 3.2.1, we also have that

$$\bar{\mathcal{L}}(x, Y, \lambda, \gamma) = \mathrm{tr}\Big(Y^\top B_c(x, \lambda, \gamma) Y\Big),$$

where $B_c(x, \lambda, \gamma)$ is as proposed.

Thus, the worst-case subspace is given by

$$\mathcal{S}^* = \{\text{top-k eigenspace of } B_c(x, \lambda^*, \gamma^*)\}$$

□

Regularization

In practical implementations of the optimization problems discussed above, it is often beneficial to incorporate regularization terms to enhance numerical stability and ensure well-posedness.

For example, in the above problems, we can easily observe that if $x^* \longrightarrow x^* + \eta$ for some $\eta \in \text{Null}(YY^\top)$, then there are infinitely many optimal solutions that yield the same cost. To mitigate such issues, we can introduce a regularization term to the objective function. A common choice is to add ℓ_2-norm regularization terms. This is a widely used technique under the name of *Tikhonov regularization* (see [17]) which has shown to improve the conditioning of least-squares problems.

Norm Regularization

Consider the regularized robust least-squares problem:

$$\min_{x\in\mathbb{R}^n} \max_{\mathcal{S}\in\mathbb{B}^d_\rho(\hat{\mathcal{S}})} \Big(\|YY^\top x - b\|^2 + \mu\|x\|^2 \Big), \tag{3.10}$$

where $\mu > 0$ is the regularization parameter. The addition of the term $\mu\|x\|^2$ helps to penalize large values of x, promoting solutions with smaller norms. This regularization can improve the conditioning of the optimization problem and lead to more stable solutions.

We also observe that the problem becomes strongly convex in x due to the presence of the ℓ_2-norm regularization term. In fact, the Hessian of the objective function with respect to x is given by:

$$\text{Hess}_x f(x, Y) = 2\Big(YY^\top + \mu\mathbb{I}\Big),$$

which is positive definite for any $\mu > 0$. This strong convexity ensures the uniqueness of the optimal solution x^*.

Constraint Regularization

Consider the regularized robust least-squares problem:

$$\min_{x\in\mathbb{R}^n} \max_{\mathcal{S}\in\mathbb{B}^d_\rho(\hat{\mathcal{S}})} \Big(\|YY^\top x - b\|^2 + \gamma\|MYY^\top x - z\|^2 \Big), \tag{3.11}$$

where $\gamma > 0$ is a regularization parameter. The relaxation of the equality constraint to an inequality constraint allows for some flexibility in satisfying the constraint, which can be particularly useful in scenarios where exact satisfaction of the constraint may not be feasible due to noise or modeling errors. This regularization can help to ensure the feasibility of the optimization problem while still promoting solutions that are close to satisfying the original equality constraint.

This also helps in ensuring that the matrix $C = MYY^\top$ is full row-rank, thereby avoiding infeasibility issues. The equality constraint $h(x, Y) = 0$ is a coupled constraint involving both x and Y, which affects convexity of the Lagrangian.

Algorithm 1: Constrained Robust Least-Squares Algorithm

```
Data: Subspace Estimate $\hat{Y} \in \mathbb{R}^{n \times k}$, Vectors $b \in \mathbb{R}^n, z \in \mathbb{R}^d$
Input   : Initial guess $x_0 \in \mathbb{R}^n$, time-step $\alpha > 0$, $\lambda_0, \rho, \gamma > 0$, tolx
Output : Optimal subspace $Y^*$, Optimal point $x^*$
while true do
    $A_i = x_i x_i^\top - x_i b^\top - b x_i^\top$;
    $Y = \texttt{eigs}(A_i, k)$;
    if $d_2(Y, \hat{Y}) \leq \rho$ then
        $\lambda^* = 0$;
        $Y^* = Y$;
    else
        $\lambda^* = \arg\min_\lambda \Big( d_2(Y(\lambda), \hat{Y}) - \rho \Big)$;
        $B_i = A_i + \lambda^* \hat{Y} \hat{Y}^\top$;
        $Y^* = \{\texttt{top-k eigenvectors of } B_i\}$;
    end
    $v_i = \boldsymbol{\nabla}_{x_i} f(x_i, Y^*) = 2Y^* Y^{*\top}(x_i - b) + 2\gamma Y^* Y^{*\top} M^\top (M Y^* Y^{*\top} x_i - z)$;
    if $\|v_i\| \leq$ tolx then
        break;
    else
        $x_{i+1} = x_i - \alpha v_i$;
    end
end
```

3.3 Linear Systems Theory

In this section, we briefly review some fundamental concepts in linear systems theory. Refer to [18] for a comprehensive treatment.

Definition 3.3.1 (LTI System). A discrete-time linear time-invariant (LTI) system is described by the state-space equations:

$$x(t+1) = Ax(t) + Bu(t), \quad y(t) = Cx(t) + Du(t),$$

where $x(t) \in \mathbb{R}^n$ is the state vector, $u(t) \in \mathbb{R}^m$ is the input vector, $y(t) \in \mathbb{R}^p$ is the output vector, and A, B, C, D are constant matrices of appropriate dimensions.

Definition 3.3.2 (Controllability). An LTI System is said to be controllable if, for any initial state x_0 and any final state x_f, there exists a finite time T and an input sequence $\{u(0), u(1), \ldots, u(T-1)\}$ that drives the state from x_0 to x_f in T steps.

Definition 3.3.3 (Observability). An LTI System is said to be observable if, for any initial state x_0, the output sequence $\{y(0), y(1), \ldots, y(T-1)\}$ over a finite time T uniquely determines the initial state x_0.

Definition 3.3.4 (Minimal Realization). A realization of an LTI System is said to be minimal if it has the smallest possible state dimension among all realizations that produce the same input-output behavior.

Definition 3.3.5 (Hankel Matrix). The Hankel matrix of an LTI System is a structured matrix that captures the input-output behavior of the system. For a system with impulse response coefficients $\{h_1, h_2, h_3, \ldots\}$, the Hankel matrix $\mathcal{H}$ is defined as:

$$\mathcal{H} = \begin{bmatrix} h_1 & h_2 & h_3 & \cdots \\ h_2 & h_3 & h_4 & \cdots \\ h_3 & h_4 & h_5 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}.$$

Definition 3.3.6 (Kalman Rank Condition). An LTI System is controllable if and only if the controllability matrix

$$\mathcal{C} = [B, AB, A^2B, \ldots, A^{n-1}B]$$

satisfies the rank condition:

$$\text{rank}(\mathcal{C}) = n.$$

Similarly, the system is observable if and only if the observability matrix

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

satisfies the rank condition:

$$\text{rank}(\mathcal{O}) = n.$$

4 Behavioral Systems Theory

The behavioral approach to systems theory, pioneered by Willems [10], offers a framework for modeling and analyzing dynamical systems based on their observed behaviors, rather than relying on explicit parametric models. In this chapter, we provide an overview of the key concepts in behavioral systems theory, including the definition of behaviors, the construction of Hankel matrices, and the fundamental lemma that underpins data-driven control methods.

4.1 Behaviors

In the behavioral framework, a dynamical system is characterized by its *behavior*, which is defined as the set of all possible trajectories (input-output sequences) that the system can exhibit.

Notation and terminology

Sets and functions

The set of positive and non-negative integers are denoted by $\mathbb{N}$ and $\mathbb{Z}_+$, respectively. The set of real and non-negative real numbers are denoted by $\mathbb{R}$ and $\mathbb{R}_+$, respectively. For $T \in \mathbb{N}$, the set of integers $\{1, 2, \ldots, T\}$ is denoted by $\mathbf{T}$. A map f from X to Y is denoted by $f : X \to Y$; $(Y)^X$ denotes the set of all such maps. The *restriction* of $f : X \to Y$ to a set X', with $X' \cap X \neq \emptyset$, is denoted by $f|_{X'}$ and is defined by $f|_{X'}(x)$ for $x \in X'$; if $\mathcal{F} \subseteq (Y)^X$, then $\mathcal{F}|_{X'}$ is defined as $\{f|_{X'} : f \in \mathcal{F}\}$.

Linear algebra

The transpose, image, and kernel of a matrix $M \in \mathbb{R}^{m\times n}$ are denoted by $M^\top$, **im** M, and $\operatorname{Ker} M$, respectively. The $m \times p$ zero matrix is denoted by $0_{m\times p}$ and the $m \times p$ matrix with ones on the main diagonal and zeros elsewhere is denoted by $\mathbb{I}_{m\times p}$. For $m = p$, we use 0_p and $\mathbb{I}_p$, and subscripts are omitted when dimensions are clear from context. The Kronecker product of the matrices $M \in \mathbb{R}^{m\times n}$ and $N \in \mathbb{R}^{p\times q}$ is denoted by $M \otimes N$.

Sequences, shift operator, Hankel matrices

A *sequence* is a function $w : S \to \mathbb{R}^q$, where $S \subseteq \mathbb{N}$; the sequence is *finite* if S has finite cardinality, and *infinite* otherwise. An infinite sequence w is also denoted by $\{w_k\}_{k\in\mathbb{N}}$. We use the terms *sequence*, *time series*, and *trajectory* interchangeably. By a convenient abuse of notation, a finite sequence $w \in (\mathbb{R}^q)^{\mathbf{T}}$ is often identified with the corresponding column vector $w = \operatorname{col}(w(1), \ldots, w(T)) \in \mathbb{R}^{qT}$.

The *shift operator* $\sigma : (\mathbb{R}^q)^{\mathbb{N}} \to (\mathbb{R}^q)^{\mathbb{N}}$ is defined as

$$\sigma w(t) = w(t+1). \tag{4.1}$$

By convention, the shift operator acts on all elements of a set $\mathcal{W} \subseteq (\mathbb{R}^q)^{\mathbb{N}}$, that is, $\sigma\mathcal{W} = \{\sigma w \,:\, w \in \mathcal{W}\}$.

The *Hankel matrix* of depth $L \in \mathbf{T}$ associated with the sequence $w \in (\mathbb{R}^q)^{\mathbf{T}}$ is defined as

$$\mathcal{H}_L(w) = \begin{bmatrix} w(1) & w(2) & w(3) & \cdots & w(T-L+1) \\ w(2) & w(3) & w(4) & \cdots & w(T-L+2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w(L) & w(L+1) & w(L+2) & \cdots & w(T) \end{bmatrix} \tag{4.2}$$

Systems

In behavioral systems theory [19], a *system* is a triple $\Sigma = (\mathbb{T}, \mathbb{W}, \mathfrak{B})$, where $\mathbb{T}$ is the *time set*, $\mathbb{W}$ is the *signal set*, and $\mathfrak{B} \subseteq (\mathbb{W})^{\mathbb{T}}$ is the *behavior* of the system. Throughout this work, we exclusively focus on *discrete-time* systems, with $\mathbb{T} = \mathbb{N}$ and $\mathbb{W} = \mathbb{R}^q$. We adapt definitions accordingly, emphasizing this assumption when necessary. By a convenient abuse of notation, we also identify a system Σ with the corresponding behavior $\mathfrak{B}$.

LTI systems

A system $\mathfrak{B}$ is *linear* if $\mathfrak{B}$ is a linear subspace, *time-invariant* if $\mathfrak{B}$ is shift-invariant, *i.e.*, $\sigma(\mathfrak{B}) \subseteq \mathfrak{B}$, and *complete* if $w \in \mathfrak{B}$ if and only $w|_{[t_0,t_1]} \in \mathfrak{B}|_{[t_0,t_1]}$ for every $-\infty < t_0 \leq t_1 < \infty$. Completeness of a discrete-time linear system $\mathfrak{B}$ is equivalent to $\mathfrak{B}$ being closed in the topology of pointwise convergence Proposition 4 [3]. The set of all (discrete-time) complete LTI systems is denoted by $\mathcal{L}^q$. By a convenient abuse of notation, we often write $\mathfrak{B} \in \mathcal{L}^q$.

Kernel representations

Every LTI system $\mathfrak{B} \in \mathcal{L}^q$ admits a *kernel representation* of the form

$$\mathfrak{B} = \operatorname{Ker} R(\sigma), \tag{4.3}$$

where $R(\sigma)$ is the operator defined by the polynomial matrix

$$R(z) = R_1 + R_2 z + \cdots + R_d z^{d-1},$$

with $R_i \in \mathbb{R}^{g \times q}$ for $i \in \boldsymbol{d}$, and

$$\operatorname{Ker} R(\sigma) = \{w \in (\mathbb{R}^q)^{\mathbb{N}} \,:\, R(\sigma)w = 0\}.$$

Without loss of generality, one may assume that $\operatorname{Ker} R(\sigma)$ is a *minimal* kernel representation of $\mathbb{B}$, *i.e.*, $R(\sigma)$ has full row rank [4].

Partitions

Given a permutation matrix $\Pi \in \mathbb{R}^{q\times q}$ and an integer $0 < m < q$, the map defined by the equation

$$(u, y) = \Pi w \tag{4.4}$$

induces a *partition* of each $w \in \mathbb{R}^q$ into the variables $u \in \mathbb{R}^m$ and $y \in \mathbb{R}^{q-m}$. For LTI systems, we can also simply identify trajectories as

$$w = \begin{bmatrix} u \\ y \end{bmatrix}. \tag{4.5}$$

We write $w \sim (u, y)$ if (4.4) holds for some permutation matrix $\Pi \in \mathbb{R}^{q\times q}$ and integer $0 < m < q$. Any partition (4.4) induces natural projections $\pi_u : w \mapsto u$ and $\pi_y : w \mapsto y$. We call (4.4) a *partition of* $\mathfrak{B} \in \mathcal{L}^q$ if (4.4) holds for all $w \in \mathfrak{B}$; the partition (4.4) is an *input-output partition* if u is an *input* and y is an *output* (see p.568 [3] for definitions).

Integer invariants

The structure of an LTI system $\mathfrak{B} \in \mathcal{L}^q$ is characterized by a set of *integer invariants* [3], defined as

- the *number of inputs* $m(\mathfrak{B}) = q - \text{row dim} R(\sigma)$,
- the *number of outputs* $p(\mathfrak{B}) = \text{row dim} R(\sigma)$,
- the *lag* $\ell(\mathfrak{B}) = \max_{i\in\mathbf{p}}\{\deg \text{row}_i R(\sigma)\}$, and
- the *order* $n(\mathfrak{B}) = \sum_{i\in\mathbf{p}} \deg \text{row}_i R(\sigma)$,

where $\text{Ker}\, R(\sigma)$ is a minimal kernel representation of $\mathfrak{B}$, while row $\dim R$ and $\deg \text{row}_i R$ are the number of rows and the degree of the i-th row of $R(z)$, respectively. The integer invariants are intrinsic properties of a system, as they do not depend on its representation [3]. Thus, we omit the dependence on the particular behavior if clear from context.

Latent variables

A *system with latent variables* is a tuple $\Sigma = (\mathbb{N}, \mathbb{R}^q, \mathbb{R}^\kappa, \mathfrak{B}_{\text{full}})$, where $\mathbb{N}$ is the *time set*, $\mathbb{R}^q$ is the set of *manifest variables*, $\mathbb{R}^\kappa$ is the set of *latent variables*, and $\mathfrak{B}_{\text{full}} \subseteq (\mathbb{R}^q \times \mathbb{R}^\kappa)^{\mathbb{N}}$ is the *full behavior*, respectively. The *manifest behavior* of the system is defined as

$$\mathfrak{B}_{\text{man}} = \{w \in (\mathbb{R}^q)^{\mathbb{N}} \mid \exists\, \eta \in (\mathbb{R}^\kappa)^{\mathbb{N}} \text{ s. t. } (w, \eta) \in \mathfrak{B}_{\text{full}}\}, \tag{4.6}$$

where w is the *manifest variable* and η is the *latent variable*. Manifest variables are *external* and directly observable (e.g., inputs and outputs); latent variables are *internal* (e.g., state and interconnection variables) and usually play an auxiliary role.

State-space representations

Every finite-dimensional LTI dynamic system $\mathbb{B} \in \mathcal{L}^q$ can be described by the equations

$$\sigma x = Ax + Bu, \quad y = Cx + Du, \tag{4.7}$$

and admits a (*minimal*) *input/state/output representation*

$$\mathfrak{B} = \Big\{ (u, y) \sim w \in (\mathbb{R}^q)^{\mathbb{N}} \;:\; \exists x \in (\mathbb{R}^n)^{\mathbb{N}} \text{ s.t. } (4.7) \text{ holds} \Big\}, \tag{4.8}$$

where

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \in \mathbb{R}^{(n+p)\times(n+m)}$$

and m, n, and p are the number of inputs, the order, and the number of outputs of $\mathfrak{B}$, respectively. State-space representations define systems where the latent variable is the state x.

Data-driven representations

Data-driven representations connect data directly to system behavior using data matrices. The restricted behavior of a finite-dimensional, discrete-time, LTI system is a subspace, whose dimension is determined only by the integer invariants and time horizon.

Lemma 4.1.1. Lemma 2.1 [20] Let $\mathfrak{B} \in \mathcal{L}^q$ and $L \in \mathbb{N}$. The restricted behavior $\mathfrak{B}|_{[1,L]}$ is a subspace of $(\mathbb{R}^q)^{\mathbf{L}}$ and $\dim \mathfrak{B}|_{[1,L]} = m(\mathfrak{B})L + n(\mathfrak{B})$ for all $L > \ell(\mathfrak{B})$.

As a direct consequence of Lemma 4.1.1, the restricted behavior of a finite-dimensional, discrete-time, LTI system can be represented as the image of a raw data matrix. Next, we summarize a version of this principle known as the *fundamental lemma* [11].

Lemma 4.1.2. Corollary 19 [21] Let $\mathfrak{B} \in \mathcal{L}^q$ and $w \in \mathfrak{B}|_{[1,T]}$. Fix $L \in \mathbf{T}$, with $L > \ell(\mathfrak{B})$. Then $\mathfrak{B}|_{[1,L]} = \mathbf{im}\, \mathcal{H}_L(w)$ if and only if

$$\mathbf{rank}\, \mathcal{H}_L(w) = m(\mathfrak{B})L + n(\mathfrak{B}). \tag{4.9}$$

The rank condition (4.9) is referred to as the *generalized persistency of excitation* condition [21]. It can be proven that this condition holds if and only if Kalman's rank condition 3.3.6 holds for the given system, *i.e.*, the system must be controllable and observable. Thus, we call a trajectory $w \in \mathfrak{B}|_{[1,T]}$ of a system $\mathfrak{B} \in \mathcal{L}^q$ *generalized persistently exciting (GPE) of order* L if (4.9) holds. Different variations of this principle can be formulated under a range of assumptions, see, *e.g.*, the recent survey [12] for an overview and [5] for some extensions to broader classes of systems.

4.2 System Identification

System identification in the behavioral framework involves constructing a behavior $\mathfrak{B}$ that accurately captures the dynamics of a system based on observed data.

Using the fundamental lemma (Lemma 4.1.2), one can identify the behavior of a finite-dimensional, discrete-time, LTI system from a set of GPE trajectories. Specifically, given a set of trajectories $\{w_i\}_{i=1}^{T_d}$ that are GPE of order L, the restricted-behavior can be reconstructed as

$$\mathbf{im}\,\mathcal{H}_L(w) = \mathfrak{B}_{[1,L]} \tag{4.10}$$

Effect of noise

In practical scenarios, data is often corrupted by noise, which can affect the accuracy of (behavioral) system identification. The rank of the Hankel matrix may be inflated due to noise, masking the system's actual order. To mitigate this, techniques such as singular value decomposition (SVD) can be employed to approximate the Hankel matrix and extract the dominant dynamics.

The singular values of the Hankel matrix can provide insights into the underlying system order, allowing for a more robust identification process in the presence of noise. An analysis of the magnitude of singular values for noisy data and ideal data is shown in Figure 4.11, which was performed for a simple SISO system (double integrator) with 10% additive Gaussian noise. It can be observed that the presence of noise inflates the singular values, masking the true system order. The noise floor, indicated by the dashed line, is where the singular values suddenly drop off, suggesting a cutoff for identifying the system order in the ideal case.

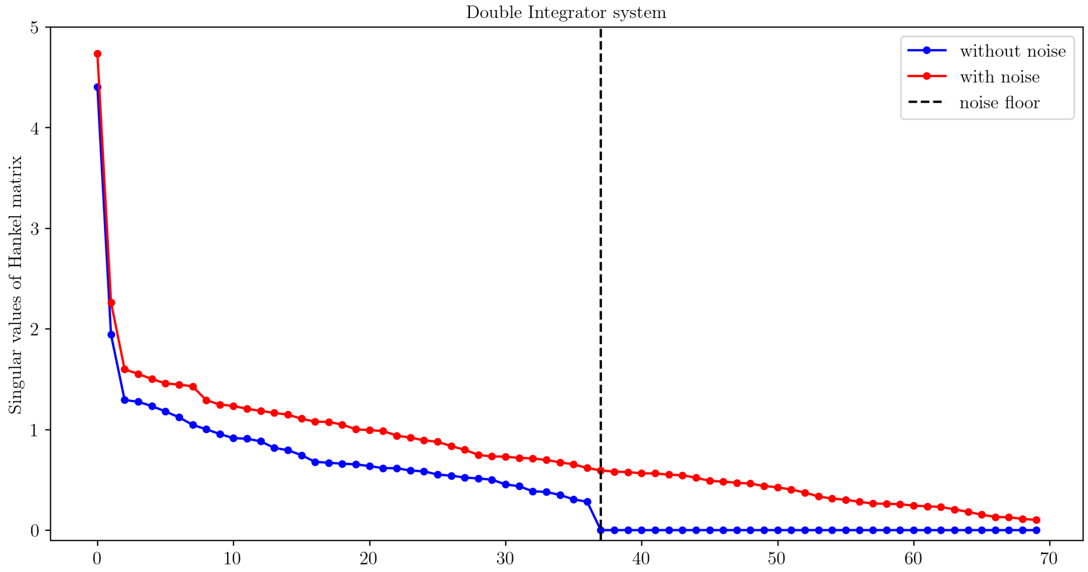


Figure 4.11 / Singular values of Hankel matrices constructed from ideal data (blue) and noisy data (red). The noise floor is shown by the dashed line.

Grassmannians and Behaviors

Behaviors can be viewed as points on a Grassmannian manifold, which is the space of all k-dimensional subspaces of an n-dimensional vector space. This perspective allows for the application of geometric methods in system identification and analysis.

Given restricted behavior $\mathfrak{B}|_{[1,L]}$ of dimension $k = m(\mathfrak{B})L + n(\mathfrak{B})$, we can associate it with a point on the Grassmannian manifold $\mathrm{Gr}(k, qL)$.

Denote $r_H = \mathbf{rank}\,\mathcal{H}_L(w)$ for trajectories $\{w_i\}_{i=1}^{T_d}$. Depending on the noise level, r_H satisfies the following:

$$k \leq r_H \leq n, \tag{4.12}$$

where $n = qL, k = m(\mathfrak{B})L + n(\mathfrak{B})$. Still, only the first k singular vectors of $\mathcal{H}_L(w)$ correspond to the behavior $\mathfrak{B}$. Thus, the behavior can be identified with a point on the Grassmannian manifold $\mathrm{Gr}(k, n)$. To obtain an orthonormal basis for the behavior, we can perform an SVD on $\mathcal{H}_L(w)$ and extract the first k left singular vectors. This basis spans the same subspace as $\mathfrak{B}|_{[1,L]}$, providing a representation of the behavior in the presence of noise. Thus,

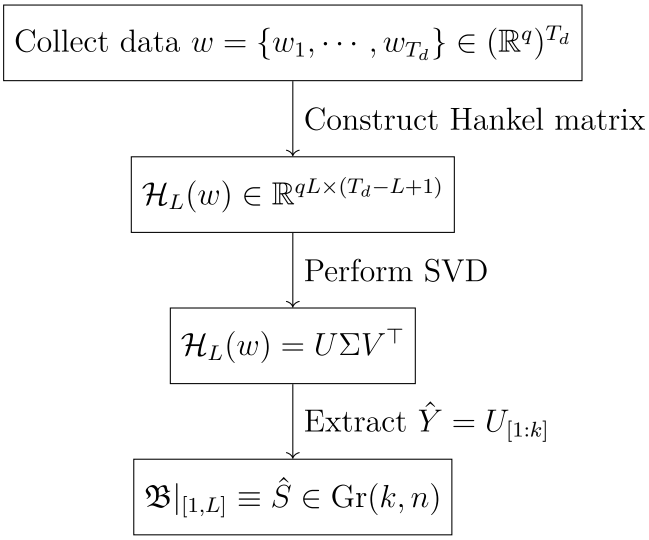
$$\mathcal{H}_L(w) = U\Sigma V^\top, \tag{4.13}$$

where $U \in \mathbb{R}^{n\times r_H}$, $\Sigma \in \mathbb{R}^{r_H\times r_H}$, and $V \in \mathbb{R}^{(T_d-L+1)\times r_H}$. The first k columns of U, is equivalently denoted by $\hat{Y}$ such that

$$\hat{Y} \equiv U_{[1:k]}, \tag{4.14}$$

where $\hat{Y} \in \mathbb{R}^{n\times k}$ and $\hat{Y}^\top\hat{Y} = \mathbb{I}_k$. The subspace spanned by the columns of $\hat{Y}$ corresponds to the behavior $\mathfrak{B}|_{[1,L]}$, is also identified by $\hat{S} \in \mathrm{Gr}(k, n)$.

Summary


Collect data $w = \{w_1, \cdots, w_{T_d}\} \in (\mathbb{R}^q)^{T_d}$
Construct Hankel matrix
$\mathcal{H}_L(w) \in \mathbb{R}^{qL\times(T_d-L+1)}$
Perform SVD
$\mathcal{H}_L(w) = U\Sigma V^\top$
Extract $\hat{Y} = U_{[1:k]}$
$\mathfrak{B}|_{[1,L]} \equiv \hat{S} \in \mathrm{Gr}(k, n)$

5 Data-Driven Control

Data-driven control is an emerging paradigm in control theory that focuses on designing controllers directly from data, without relying on explicit parametric models of the system dynamics. This chapter explores the principles and methodologies of predictive control, both in the context of known system models and in scenarios where only data is available. We focus specifically on finite-time linear quadratic tracking problems, and highlight the advantages of the behavioral framework in enabling data-driven control strategies.

5.1 Linear Quadratic Tracking

Finite-time linear quadratic tracking (LQT) is a fundamental optimal control problem [22] where the objective is to minimize a quadratic cost function over a finite time horizon while tracking a desired reference trajectory. Consider a discrete-time LTI System given by

$$\begin{cases} x(t+1) = Ax(t) + Bu(t), \\ y(t) = Cx(t) + Du(t), \end{cases} \tag{5.1}$$

where $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}, C \in \mathbb{R}^{p\times n}, D \in \mathbb{R}^{p\times m}$ are system matrices, $x(t) \in \mathbb{R}^n$ is the state vector, $u(t) \in \mathbb{R}^m$ is the control input, and $y(t) \in \mathbb{R}^p$ is the output vector. Given a desired trajectory $r = (r_0, r_1, \cdots) \in (\mathbb{R}^p)^{\mathbb{N}}$, input constraint set $\mathcal{U} \subseteq \mathbb{R}^m$, and output constraint set $\mathcal{Y} \subseteq \mathbb{R}^p$, we wish to apply control inputs such that the output tracks the reference trajectory while satisfying the constraints and optimizing a cost function.

In the case where the system is *known*, i.e., the matrices A, B, C, D are given, the finite-horizon LQT problem is approached using *Model Predictive Control*.

Model Predictive Control

Model Predictive Control (MPC) is an advanced control strategy that utilizes an explicit model of the system to predict future behavior and optimize control actions over a finite horizon [[6], [23]]. Consider the optimization problem:

$$\begin{aligned} \underset{u,x,y}{\text{minimize}} \quad & \sum_{k=0}^{N-1} \left(\|y_k - r_{t+k}\|_Q^2 + \|u_k\|_R^2 \right) \\ \text{subject to} \quad & x_{k+1} = Ax_k + Bu_k, \\ & y_k = Cx_k + Du_k, \\ & x_0 = \hat{x}(t), \\ & u_k \in \mathcal{U}, \forall k \in \{0, \dots, N-1\} \\ & y_k \in \mathcal{Y}, \forall k \in \{0, \dots, N-1\}, \end{aligned} \tag{5.2}$$

where $N \in \mathbb{Z}_{>0}$ is the time horizon, $u = (u_0, u_1, \dots, u_{N-1}), x = (x_0, x_1, \dots, x_N)$, $y = (y_0, y_1, \dots, y_{N-1})$ are the decision variables, and r_{t+k} is the desired reference at time $t+k$, where $t \in \mathbb{Z}_{>0}$ is the time at which the optimization is solved. The matrices $Q \succeq 0$ and $R \succ 0$ are weighting matrices for the output tracking error and control effort, respectively. The initial state $\hat{x}(t)$ is typically obtained from state estimation or measurement at time t.

The classical MPC algorithm operates in a *receding horizon* fashion. The algorithm is summarized in Algorithm 2 and illustrated in Figure 5.3.

Algorithm 2: Model Predictive Control

```
Input : $(A, B, C, D)$, reference trajectory $r$, past input/output data $(u, y)$,
        constraint sets $\mathcal{U}, \mathcal{Y}$, weighting matrices $Q, R$, time horizon $N$

Generate state estimate $\hat{x}(t)$ from past data $(u, y)$.;
Solve (5.2) for $u^* = (u_0^*, u_1^*, \dots, u_{N-1}^*)$.;
Apply inputs $(u(t), \dots, u(t+s)) = (u_0^*, \dots, u_s^*)$ for some $s \leq N-1$.;
Set $t$ to $t+s$ and update past input/output data.;
Return to start.
```

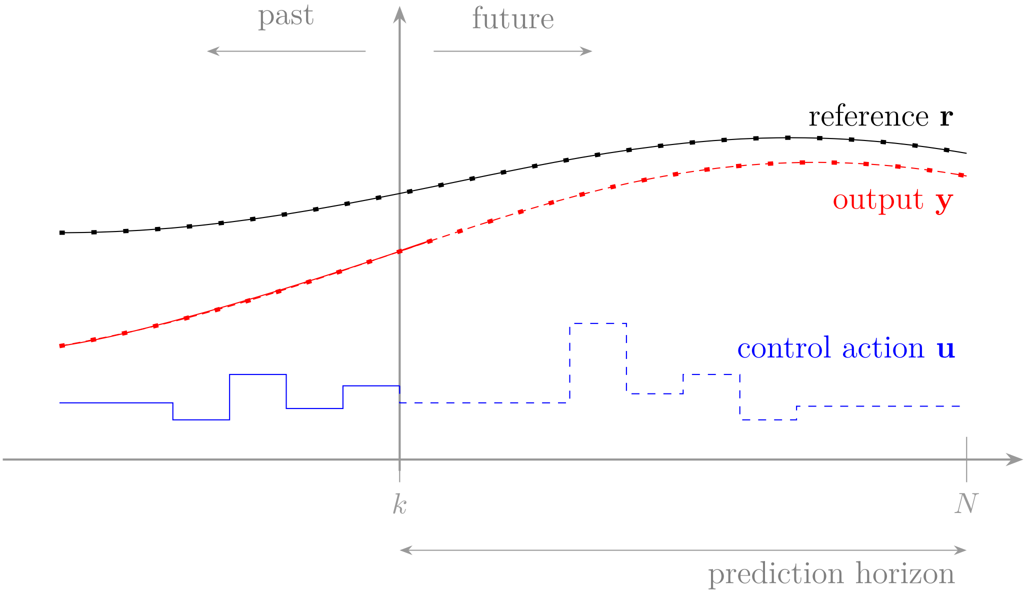


Figure 5.3 / Classical MPC in a receding horizon fashion. At each time step, an optimization problem is solved over a future horizon using the current state estimate and past data. The first s control inputs are applied, and the horizon recedes forward in time.

Data-Driven Predictive Control

When the system model is *unknown*, i.e., the matrices A, B, C, D are not available, traditional model-based control strategies like MPC cannot be directly applied. Instead, we turn to data-driven predictive control methods [e.g. DeePC [7]] that leverage historical input-output data to design controllers without explicit knowledge of the system dynamics. The behavioral framework introduced in Chapter 4 provides a powerful toolset for such data-driven approaches, since it bypasses the need for parametric models and directly utilizes data to characterize system behavior.

On a finite-horizon, the problem can be formulated as follows. Given,

- an LTI system $\Sigma = (\mathbb{T}, \mathbb{W}, \mathfrak{B})$, with $\mathbb{W} = \mathbb{R}^q$, and $\mathbb{T} = [1, N]$,
- an initial trajectory $w_{\text{ini}} \in \mathfrak{B}|_{\mathbb{T}_{\text{past}}}$ with $\mathbb{T}_{\text{past}} = [1, T_{\text{ini}}]$,
- a reference trajectory $w_{\text{ref}} \in (\mathbb{R}^q)^{\mathbb{T}_{\text{fut}}}$, with $\mathbb{T}_{\text{fut}} = [T_{\text{ini}} + 1, N]$,
- a symmetric positive semi-definite weighting matrix $\Phi \in \mathbb{R}^{q \times q}$,

the linear quadratic tracking problem is to find a trajectory $w_{\text{fut}} \in \mathfrak{B}|_{\mathbb{T}_{\text{fut}}}$ that solves the optimization problem

$$\begin{aligned} \underset{w_{\text{fut}} \in (\mathbb{R}^q)^{\mathbb{T}_{\text{fut}}}}{\text{minimize}} \quad & \sum_{k=T_{\text{ini}}+1}^{N} \|w_{\text{fut}}(k) - w_{\text{ref}}(k)\|^2_{I \otimes \Phi} \\ \text{subject to} \quad & w = \begin{bmatrix} w_{\text{ini}} \\ w_{\text{fut}} \end{bmatrix} \in \mathfrak{B}. \end{aligned} \tag{5.4}$$

Note. The reference trajectory w_{ref} need not be in the behavior $\mathfrak{B}$. For instance, step inputs with sudden changes can be used as reference trajectories, even if they are not achievable by the system.

If a parametric model of the system behavior is available—for example a state-space representation—then the optimization problem (5.4) is equivalent [7] to the classical MPC problem (5.2) described in Section 5.1. However, in the absence of such a model, we can utilize the behavioral framework to reformulate the problem in a data-driven manner.

5.2 Robust Least-Squares for Data-Driven Control

Predictive control methods like DeePC [7] rely on the availability of persistently exciting input-output data to construct Hankel matrices that capture the system's behavior. However, in practical scenarios, the collected data may be corrupted by noise, outliers, or other uncertainties, which can adversely affect the performance of data-driven controllers. To address these challenges, robust least-squares techniques can be employed to enhance the resilience of data-driven predictive control methods against data imperfections.

Geometric Approach to Robust Least-Squares

We consider a geometric approach to robust least-squares problem in the context of linear quadratic tracking. Building up from (5.4), we formulate:

$$\begin{aligned} \min_{w\in\mathbb{R}^n} \max_{\mathcal{S}\in\mathbb{B}_\rho(\hat{\mathcal{S}})} & \left\|w - w_{\text{ref}}\right\|^2 \\ \text{subject to} & \begin{cases} w \in \mathcal{S}, \\ w|_{\mathbb{T}_{\text{past}}} = w_{\text{ini}} \end{cases} \end{aligned} \tag{5.5}$$

Here,

- $\mathbb{B}_\rho(\hat{\mathcal{S}}) \subseteq \text{Gr}(k,n)$ is a ball of radius ρ centered at $\hat{\mathcal{S}} \equiv \mathfrak{B}|_{[1,L]}$ which is obtained via behavioral system identification.
- $n = qL, k = m(\mathfrak{B})L + n(\mathfrak{B})$ where $m(\mathfrak{B}), n(\mathfrak{B})$ are the output and state dimensions of the behavior $\mathfrak{B}$ respectively.
- $w_{\text{ref}} = \text{col}(w_{\text{ini}}, w_{\text{ref,f}}) \in (\mathbb{R}^q)^{\mathbb{T}}$ is the reference trajectory over the horizon N.
- $w_{\text{ini}} \in (\mathbb{R}^q)^{\mathbb{T}_{\text{past}}}$ is the initial trajectory over the horizon N.

Constraints

The constraints in (5.5) are briefly discussed here.

To enforce the constraint $w \in \mathcal{S}$, we utilize the orthogonal projection onto the subspace $\mathcal{S}$. Specifically, we take the transformation

$$w = P_{\mathcal{S}} x = YY^\top x, \tag{5.6}$$

where $P_{\mathcal{S}}$ is the orthogonal projection matrix onto the subspace $\mathcal{S}$, and $Y \in \mathbb{R}^{n\times k}$ is an orthonormal basis for $\mathcal{S}$. The variable $x \in \mathbb{R}^n$ is a new decision variable that allows us to express w in terms of the subspace $\mathcal{S}$.

The past constraint $w|_{\mathbb{T}_{\text{past}}} = w_{\text{ini}}$ is a linear equality constraint that can be expressed in matrix form as

$$Mw = w_{\text{ini}}, \tag{5.7}$$

where $M = [\mathbb{I}_{qT_{\text{ini}}} \quad 0] \in \mathbb{R}^{qT_{\text{ini}}\times n}$ is a selection matrix that extracts the components of w corresponding to the past horizon $\mathbb{T}_{\text{past}}$. Substituting the expression for w into this constraint gives

$$MYY^\top x = w_{\text{ini}}. \tag{5.8}$$

Table 5.9 / Reformulation of constraints in (5.5).

Reformulated Problem

Substituting the expressions for w and the constraints into the robust least-squares problem (5.5), we obtain the reformulated optimization problem:

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \max_{\mathcal{S}\in\mathbb{B}_\rho(\hat{\mathcal{S}})} & \|YY^\top x - b\|^2 \\ \text{subject to} \quad & MYY^\top x = z, \end{aligned} \tag{5.10}$$

where $b = w_{\text{ref}}$ and $z = w_{\text{ini}}$ for notational convenience. This problem is exactly in the form of the constrained robust least-squares problem discussed in Chapter 3, Subsection 3.2, and can be solved using the geometric approach outlined therein.

> **Note.** The cost function in (5.10) can be modified to include weighting matrices or regularization terms as needed (see Subsection 3.2). For instance, if we wish to penalize deviations in certain components of the trajectory more heavily, we can introduce a weighting matrix Φ into the cost function.

Thus, the algorithm for data-driven predictive control using (regularized) robust least-squares can be summarized as follows:

Algorithm 3: Data-Driven Predictive Control via Robust Least-Squares

```
Data:   Subspace estimate $\hat{\mathcal{S}} \in \mathrm{Gr}(k, n)$, System simulator $(A, B, C, D)$
Input : Initial trajectory $z = w_{\text{ini}} \in (\mathbb{R}^q)^{\mathbb{T}_{\text{ini}}}$, Reference trajectory
        $r \in (\mathbb{R}^p)^{\mathbb{T}}$
Output: Optimal trajectories $w^* \in (\mathbb{R}^q)^{\mathbb{T}}$

while $t \in [1, T]$ do
    Update $z = w_{\text{ini}} = [u_{\text{past}}^\top, y_{\text{past}}^\top]^\top$ from past data.;
    Update $w_{\text{ref,f}} = [0^\top, r^\top]^\top$ from reference data.;
    Define $b = [z^\top, w_{\text{ref,f}}^\top]^\top$.;
    Solve Algorithm 1 with inputs $(\hat{\mathcal{S}}, b, z)$ to obtain $(x^*, Y^*)$.;
    Obtain optimal trajectory $w^* = Y^* Y^{*\top} x^*$.;
    Apply control inputs $u^*(t), \ldots, u^*(t+s)$ for some $s \leq N-1$.;
    Set $t$ to $t+s$ and update past input/output data.;
end
```

5.3 Results

The proposed data-driven predictive control algorithm using robust least-squares has been evaluated on various LTI systems. The results demonstrate improved tracking performance and robustness against data imperfections compared to traditional least-squares approaches.

Double Integrator System

Consider a simple double-integrator system $\ddot{x}(t) = \dfrac{F(t)}{m}$, given in discrete time as

$$\begin{aligned} \begin{bmatrix} x_{k+1} \\ v_{k+1} \end{bmatrix} &= \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ v_k \end{bmatrix} + \begin{bmatrix} \frac{h^2}{2m} \\ \frac{h}{m} \end{bmatrix} u_k \\ y_k &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ v_k \end{bmatrix} + \eta_k \end{aligned} \tag{5.11}$$

where $x_k = x(t_k), v_k = \dot{x}(t_k)$ are the states, $h = t_{k+1} - t_k$ is the simulation time-step, and $\eta_k \sim \mathcal{N}(0, \sigma^2)$ is measurement noise. The system parameters are set to $m = 1$ kg, $h = 0.5$ s. The control objective is to track a reference trajectory $r = 1$ m over a horizon of N time steps, with a random initial condition $[x_0, v_0]$.

Offline System Identification

To identify the behavior $\mathfrak{B}$ of the double-integrator system, we collect input-output data by simulating the system with random control inputs u_k drawn from a uniform distribution over $[-1, 1]$ N. The output measurements y_k are corrupted with Gaussian noise. A total of $T = 115$ time steps of data is collected, and a persistently exciting input sequence of order $L = T_{\text{ini}} + N = 35$ is ensured. Using this data, we construct the Hankel matrix and extract the subspace estimate $\hat{\mathcal{S}}$ using the methods outlined in Chapter 4. Thus, we summarize the following parameters from system identification of the double-integrator system in Table 5.12.

Parameter	Value
Number of inputs $m(\mathfrak{B})$	1
Number of outputs $p(\mathfrak{B})$	1
Number of states $n(\mathfrak{B})$	2
Dimension of behavior k	37
Data length T_d	115
Future horizon N	25
Past horizon T_{ini}	10
Total horizon L	35
Subspace dimension $\dim(\hat{Y})$	70×37

Table 5.12 / System identification parameters for double-integrator system.

Online Predictive Control

With the identified subspace $\hat{\mathcal{S}}$, we implement the data-driven predictive control algorithm using robust least-squares. The control inputs are computed over a receding horizon of $T = 50$ time steps, with the first $s = 1$ control input applied at each iteration. The reference trajectory is set to $r_k = 1$ m for all k. The robust least-squares problem is solved using the geometric approach outlined in Chapter 3, with a ball radius ρ chosen based on the noise level in the data.

Figures 5.14 and 5.15 illustrate the system simulations and optimization convergence for the case of no measurement noise ($\sigma = 0$). The results demonstrate effective tracking of the reference trajectory and convergence of the optimization process. The parameters/results for the simulations are summarized in Table 5.13.

Parameter	**Value**
ρ	0.1°
γ	5.0
Total Run-Time	4.27 s
Tracking Time	15 s

Table 5.13 / Simulation parameters/results for $\sigma = 0$ (no noise)

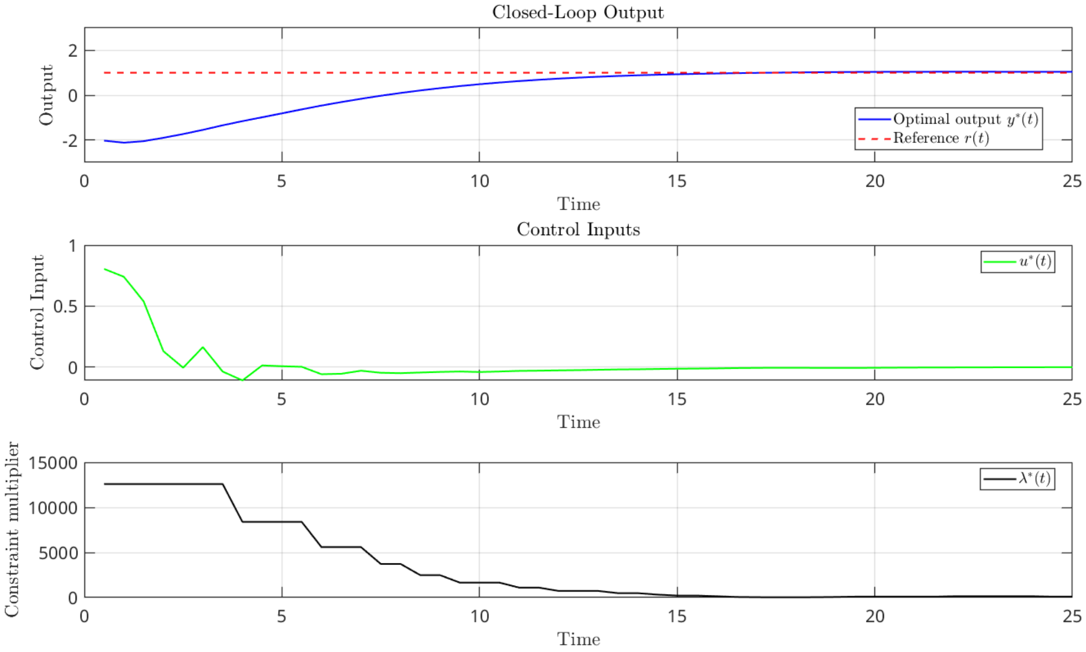


Figure 5.14 / System simulations for $\sigma = 0$ (no noise).

Observations:

- Initially, λ^* is large, indicating a significant deviation from the reference trajectory. As the optimization progresses, Y^* aligns better with the true behavior $\hat{Y}$, leading to a decrease in λ^*.
- The optimization algorithm converges within ~ 30 iterations per time step on average, demonstrating computational efficiency suitable for real-time applications.
- The choice of ball radius ρ significantly impacts the robustness of the control strategy.

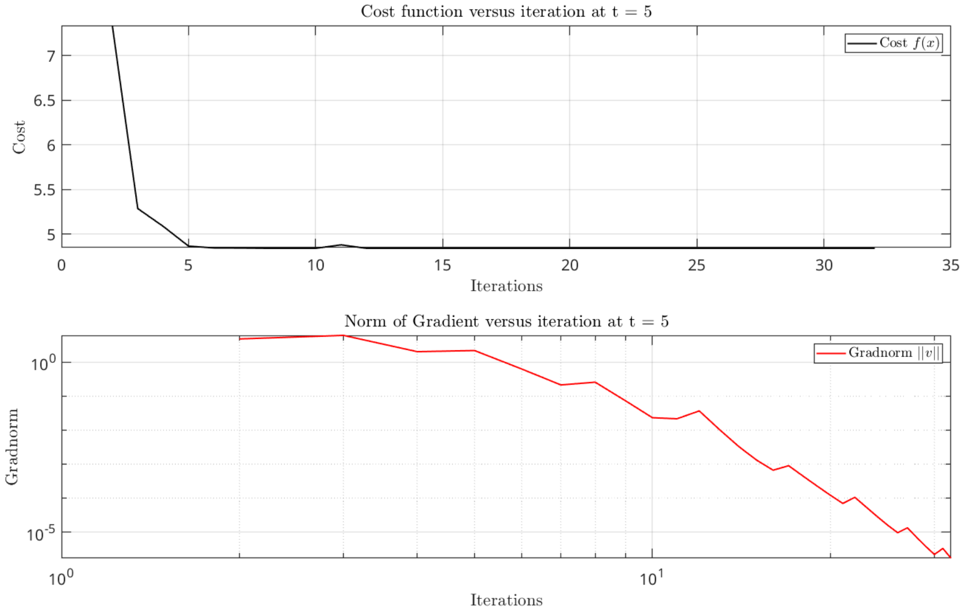


Figure 5.15 / Optimization algorithm convergence at $t = 5s$ for $\sigma = 0$ (no noise).

Next, we consider the case with measurement noise ($\sigma = 0.1$). Figures 5.16 and 5.17 illustrate the system simulations and optimization convergence for this scenario. The results show that the robust least-squares approach effectively mitigates the impact of noise, achieving satisfactory tracking performance. The parameters/results for the noisy simulations are summarized in Table 5.18.

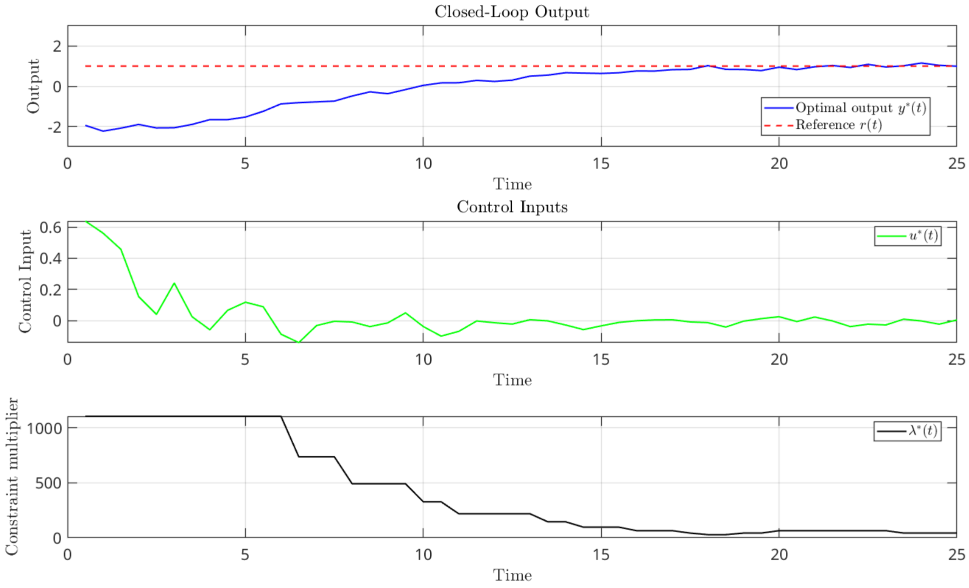


Figure 5.16 / System simulations for $\sigma = 0.1$ (with noise).

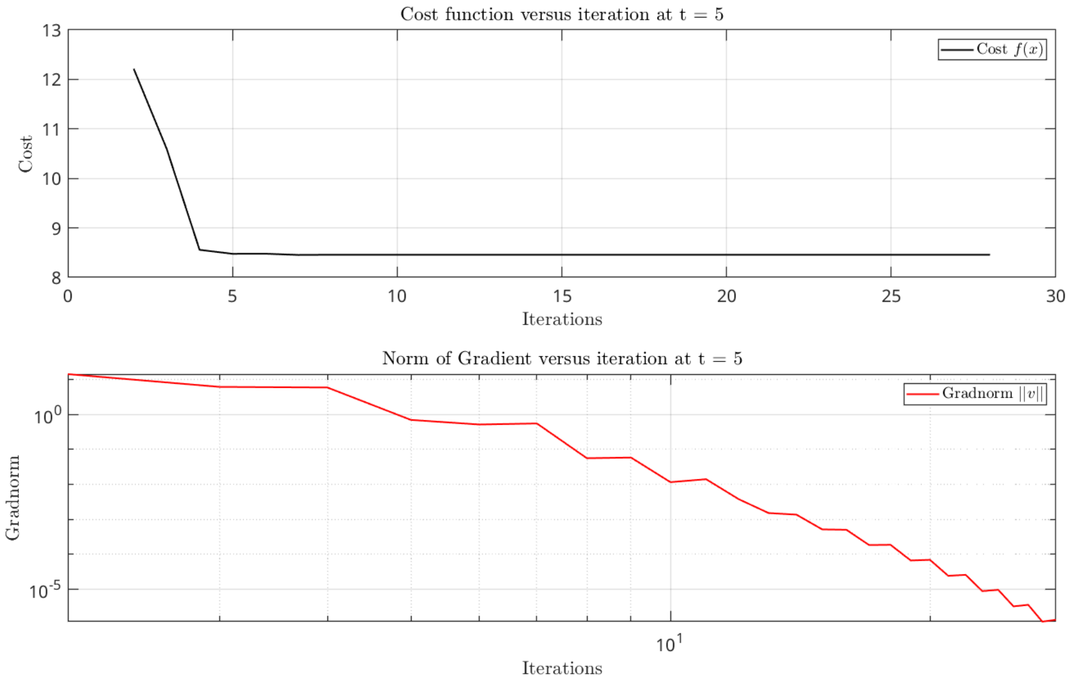


Figure 5.17 / Optimization algorithm convergence at $t = 5s$ for $\sigma = 0.1$ (with noise).

Parameter	Value
ρ	1°
γ	4.0
Total Run-Time	5.46 s
Tracking Time	18 s

Table 5.18 / Simulation parameters/results for $\sigma = 0.1$ (with noise)

Now, we further increase the measurement noise to $\sigma = 0.5$. Figures 5.19 and 5.20 illustrate the system simulations and optimization convergence for this higher noise scenario. The robust least-squares approach continues to provide effective tracking performance despite the increased noise level. The parameters/results for the high-noise simulations are summarized in Table 5.21.

Parameter	Value
ρ	3.5°
γ	5.0
Total Run-Time	10.66 s
Tracking Time	20 s

Table 5.21 / Simulation parameters/results for $\sigma = 0.5$ (with high noise)

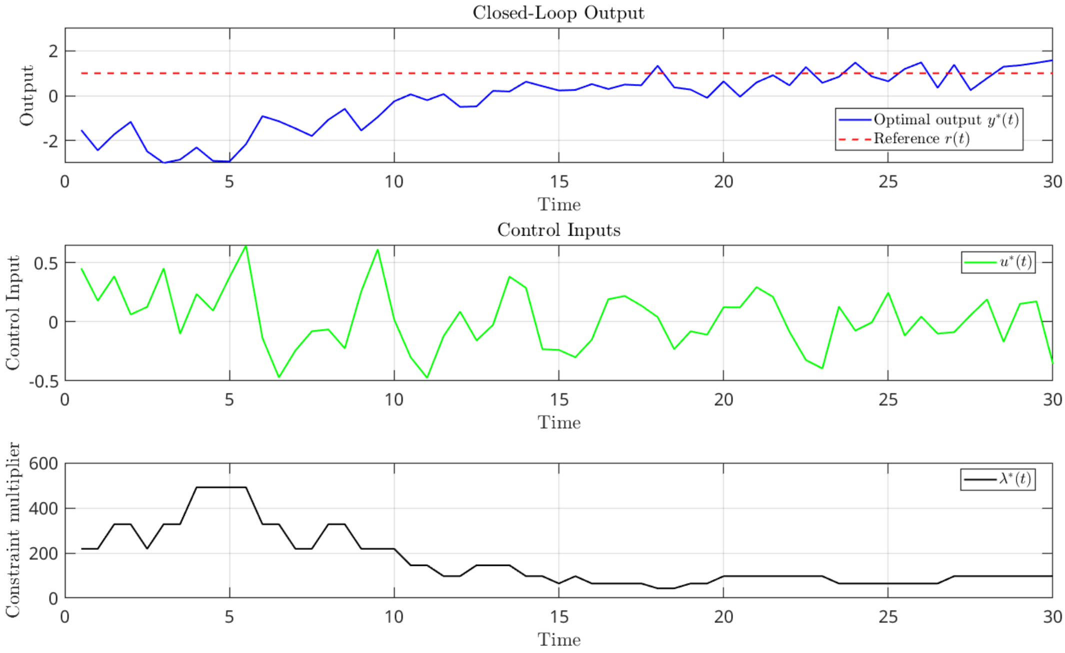


Figure 5.19 / System simulations for $\sigma = 0.5$ (with noise).

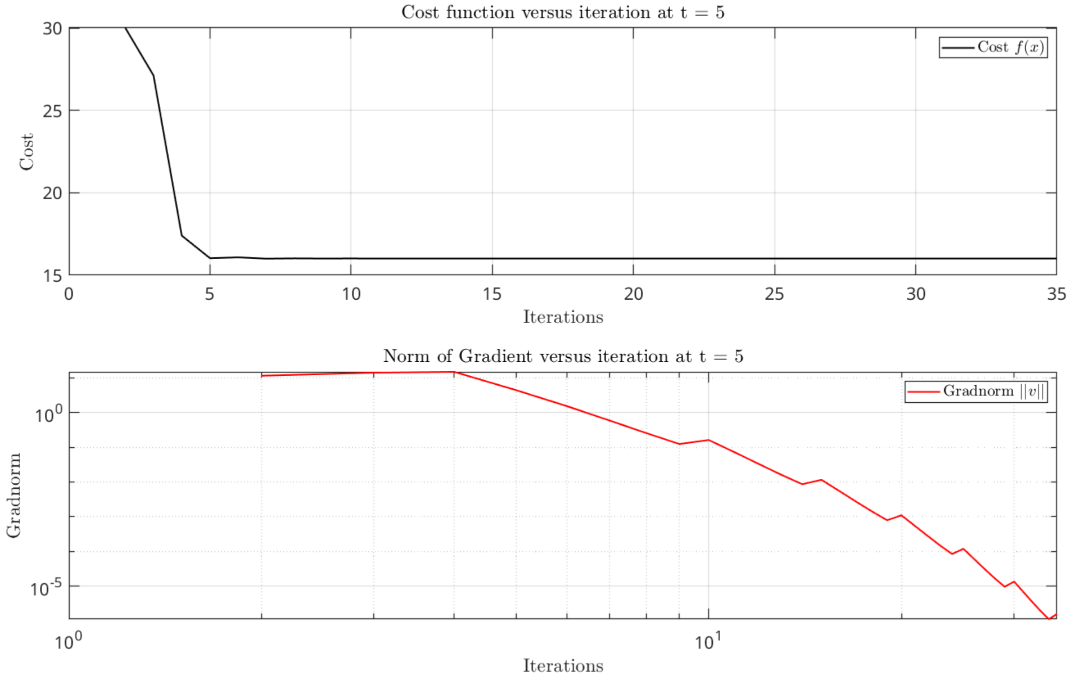


Figure 5.20 / Optimization algorithm convergence at $t = 5s$ for $\sigma = 0.5$ (with noise).

Observations:

- As the noise level increases, the choice of ρ becomes more critical to maintain robust tracking performance. Remark 3 provides insights on how ρ has been estimated in the simulations.
- The optimization algorithm remains effective, converging within a reasonable number of iterations even under high noise conditions.
- Tuning γ helps balance controller performance and algorithm convergence speed. It is yet to be understood how to perform a dynamic update of γ during the optimization process for improved results.

Remark 3. The ball radius ρ is the quantification of uncertainty in the behavioral estimate $\hat{\mathcal{S}}$ (or $\hat{Y}$). As one may expect, increase in measurement noise σ should lead to an increase in ρ. Although an exact relationship between σ and ρ is not yet established, one may estimate ρ as follows.

Assuming only measurement noise η_k, denote $\hat{w}_k \in \mathbb{R}^q$ as the noise-free trajectory at time k, and $w_k = \hat{w}_k + e_k$ as the actual trajectory, such that

$$e_k = \begin{bmatrix} 0 \\ \eta_k \end{bmatrix} \in \mathbb{R}^q.$$

Let $\hat{Y}$ be the behavioral estimate obtained from system identification (also corrupted by noise). Thus, for online control, the worst-case subspace should contain the worst-case trajectory w_k. Note that here, we can assume $\hat{w}_k \in \hat{\mathcal{S}}$. Hence,

$$\rho \approx d_\infty(\hat{Y}, Y^*) = \sin(\theta),$$

where θ is the maximum principle angle between the subspaces $\hat{\mathcal{S}}$ and $\mathcal{S}_k$ that contains the worst-case trajectory w_k. Using basic trigonometry, we thus have

$$\theta = \max_k \left\{ \tan^{-1} \left(\frac{\|P_{\hat{\mathcal{S}}}^{\perp} e_k\|}{\|\hat{w}_k + P_{\hat{\mathcal{S}}} e_k\|} \right) \right\}.$$

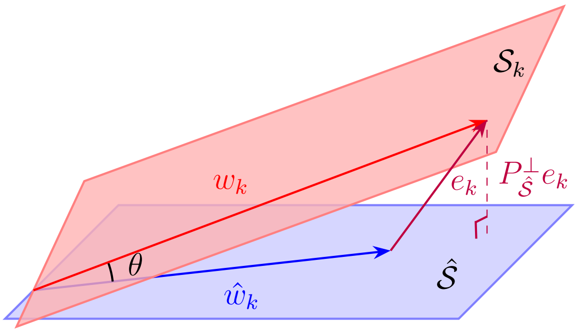


Figure 5.22 / Estimation of ball radius ρ based on trajectory deviation e_k.

Summary and Future Work

This thesis has explored the application of robust least-squares optimization in the context of data-driven predictive control, leveraging the behavioral approach to systems theory. By representing dynamical systems based on observed input-output data, we have developed a geometric framework that effectively handles uncertainties in system dynamics.

The key contributions of this work include:

(1) Development of a robust least-squares optimization framework that accounts for subspace uncertainty, enabling more reliable control strategies in the presence of data perturbations.

(2) Integration of the behavioral approach to system theory, allowing for a flexible representation of dynamical systems without explicit parametric models.

(3) Formulation of data-driven predictive control problems as constrained least-squares problems, facilitating the design of control strategies directly from observed data.

(4) Demonstration of the effectiveness of the proposed framework through numerical simulations.

Future research directions include:

- Extension of the robust least-squares framework to accommodate linear time-varying (LTV) systems and nonlinear systems.
- System-theoretic interpretation of the worst-case subspace $Y^\star$ in the context of robust or H_∞ control.
- Enhancing the computational efficiency of the proposed algorithms using advanced optimization techniques like adaptive time-stepping, etc.
- Experimental validation of the proposed framework on real-world systems to assess its practical applicability and performance.

References

Back-references to the pages where the publication was cited are given by ●.

[1] Laurent El Ghaoui and Hervé Lebret. Robust Solutions to Least-Squares Problems with Uncertain Data. *SIAM Journal on Matrix Analysis and Applications*, **1997**.
DOI: 10.1137/S0895479896298130 1, 2, 6

[2] Gene H. Golub and Charles F. van Loan. An Analysis of the Total Least Squares Problem. *SIAM Journal on Numerical Analysis*, **1980**.
DOI: 10.1137/0717073 eprint: https://doi.org/10.1137/0717073 2, 5

[3] Jan C. Willems. From time series to linear system—Part I. Finite dimensional linear time invariant systems. *Automatica*, **1986**.
DOI: https://doi.org/10.1016/0005-1098(86)90066-X 3, 22, 23

[4] Jan C. Willems. Models for dynamics. IN: *Dynamics Reported*. ed. by U. Kirchgraber and H. O. Walther. Wiley, **1989** 3, 22

[5] Alberto Padoan, Jeremy Coulson, Henk J. Van Waarde, John Lygeros, and Florian Dorfler. Behavioral uncertainty quantification for data-driven control. IN: *2022 IEEE 61st Conference on Decision and Control*. IEEE, **2022**.
DOI: 10.1109/CDC51059.2022.9993002 3, 7, 24

[6] E. F. Camacho and C. Bordons. Model Predictive Control. 2nd. Springer London, **2007**.
DOI: 10.1007/978-0-85729-398-5 3, 27

[7] Jeremy Coulson, John Lygeros, and Florian Dörfler. Data-Enabled Predictive Control: In the Shallows of the DeePC. IN: *2019 18th European Control Conference (ECC)*. **2019**.
DOI: 10.23919/ECC.2019.8795639 4, 7, 29

[8] Adrien-Marie Legendre. Nouvelles méthodes pour la détermination des orbites des comètes. F. Didot, **1805**.
URL: https://catalog.hathitrust.org/Record/008630090 5

[9] Carl Friedrich Gauss. Theoria motus corporum coelestium in sectionibus conicis solem ambientum. F. Perthes and I.H. Besser, **1809** 5

[10] Jan C. Willems. The Behavioral Approach to Open and Interconnected Systems. *IEEE Control Systems Magazine*, **2007**.
DOI: 10.1109/MCS.2007.906923 6, 21

[11] JC Willems, P Rapisarda, Markovsky, and BLM De Moor. A note on persistency of excitation. English. *Systems & Control Letters*, **2005**.

DOI: 10.1016/j.sysconle.2004.09.003 6, 24

[12] I. Markovsky and F. Dörfler. Behavioral systems theory in data-driven analysis, signal processing, and control. *Annual Reviews in Control*, **2021**.
DOI: 10.1016/j.arcontrol.2021.09.005 7, 24

[13] P.-A. Absil, R. Mahony, and R. Sepulchre. Optimization Algorithms on Matrix Manifolds. Princeton University Press, **2007** 8

[14] Nicolas Boumal. An introduction to optimization on smooth manifolds. Cambridge University Press, **2023**.
DOI: 10.1017/9781009166164 8, 11

[15] Andi Han, Bamdev Mishra, Pratik Jawanpuria, and Junbin Gao. Nonconvex-nonconcave min-max optimization on Riemannian manifolds. *Transactions on Machine Learning Research*, **2023**.
URL: https://openreview.net/forum?id=EDVIHPZhFo 13

[16] Andi Han, Bamdev Mishra, Pratik Jawanpuria, and Akiko Takeda. A Framework for Bilevel Optimization on Riemannian Manifolds. **2024**.
ARXIV: 2402.03883 [math.OC] URL: https://arxiv.org/abs/2402.03883 13

[17] Gene H. Golub, Per Christian Hansen, and Dianne P. O'Leary. Tikhonov Regularization and Total Least Squares. *SIAM Journal on Matrix Analysis and Applications*, **1999**.
DOI: 10.1137/S0895479897326432 18

[18] João P. Hespanha. Linear Systems Theory: Second Edition. NED - New edition, 2. Princeton University Press, **2018**.
URL: http://www.jstor.org/stable/j.ctvc772kp (visited on 10/21/2025) 19

[19] J.W. Polderman and J.C. Willems. Introduction to Mathematical Systems Theory: A Behavioral Approach. Springer Verlag, **1998** 22

[20] Florian Dörfler, Jeremy Coulson, and Ivan Markovsky. Bridging Direct and Indirect Data-Driven Control Formulations via Regularizations and Relaxations. *IEEE Transactions on Automatic Control*, **2023**.
DOI: 10.1109/TAC.2022.3148374 24

[21] Ivan Markovsky and Florian Dörfler. Identifiability in the Behavioral Setting. *IEEE Transactions on Automatic Control*, **2022**.
DOI: 10.1109/TAC.2022.3209954 24

[22] Brian D. O. Anderson and John B. Moore. Optimal control: linear quadratic methods. Prentice-Hall, Inc., **1990** 27

[23] A. Ferramosca, D. Limón, I. Alvarado, T. Álamo, and E. F. Camacho. MPC for tracking with optimal closed-loop performance. *Automatica*, **2009**.
DOI: 10.1016/j.automatica.2009.04.012 27