Data-Driven Predictive Control: Equivalence to Model Predictive Control Beyond Deterministic Linear Time-Invariant Systems

by

Ruiqi Li

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2024

© Ruiqi Li 2024

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	Tyler H. Summers Associate Professor, Dept. of Mechanical Engineering, University of Texas at Dallas
Supervisor:	Stephen L. Smith Professor, Dept. of Electrical and Computer Engineering, University of Waterloo
Supervisor:	John W. Simpson-Porco Assistant Professor, Dept. of Electrical and Computer Engineering, University of Toronto
Internal Member:	Christopher Nielsen Professor, Dept. of Electrical and Computer Engineering, University of Waterloo
Internal Member:	Michael Fisher Assistant Professor, Dept. of Electrical and Computer Engineering, University of Waterloo
Internal-External Member:	Kirsten Morris Professor, Dept. of Applied Mathematics, University of Waterloo

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

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

Abstract

In recent years, data-driven predictive control (DDPC) has emerged as an active research area, with well-known methods such as Data-enabled Predictive Control (DeePC) and Subspace Predictive Control (SPC) being validated through reliable experimental results. On the theoretical side, it has been established that both DeePC and SPC methods can generate equivalent control actions as one can obtain from Model Predictive Control (MPC), for deterministic linear time-invariant (LTI) systems. However, similar results do not yet exist for the application of DDPC beyond deterministic LTI systems. Therefore, the objective of our research is to generalize this theoretical equivalence between model-based and data-driven methods for more general classes of control systems.

In this thesis, we present our contributions to DDPC for linear time-varying (LTV) systems and stochastic LTI systems. In our first piece of work, we developed Periodic DeePC (P-DeePC) and Periodic SPC (P-SPC) methods, which generalize DeePC and SPC from LTI systems to linear time-periodic (LTP) systems, as a special case of LTV systems. Theoretically, we demonstrate that our P-DeePC and P-SPC methods have equivalence control actions as produced from MPC for deterministic LTP systems, under appropriate tuning conditions. As an intermediate step in our theoretical development, we extended certain aspects of behavioral systems theory from LTI systems to LTP/LTV systems. This includes extending Willems' fundamental lemma to LTP systems and the defining the concepts of order and lag for LTV systems.

In our second piece of work, we proposed a control framework for stochastic LTI systems, namely Stochastic Data-Driven Predictive Control (SDDPC). Our SDDPC method theoretically achieves equivalent control performance to model-based Stochastic MPC, under idealized conditions of appropriate tuning and noise-free offline data. This method, which applies to general linear stochastic state-space systems, serves as an alternative to the data-driven method previously proposed by Pan et al., which also achieved theoretical equivalence to Stochastic MPC but was limited to a narrower class of systems. Beyond the theoretical assumption of noise-free offline data, we performed our SDDPC method in simulations with practical noisy offline data. The simulation results demonstrated that our SDDPC method outperforms benchmark methods, achieving lower cumulative tracking cost and lower rate and amount of constraint violation.

Acknowledgements

I would like to express my deepest gratitude to my supervisors, Prof. Stephen L. Smith and Prof. John W. Simpson-Porco, for their invaluable guidance, encouragement, and support throughout the course of my doctoral journey. Their expertise, insights, and constructive feedback have been pivotal in shaping this thesis and enriching my academic experience.

I am also grateful to my committee members Prof. Christen Nielsen, Prof. Michael Fisher, Prof. Kirsten Morris and Prof. Tyler Summers for their advice and thoughtful critiques, which have helped me refine my work and deepen my understanding of data-driven predictive control.

My appreciation extends to my colleagues and friends at the University of Waterloo and the University of Toronto for their collaboration, stimulating discussions, and unwavering support during challenging times. Finally, I owe my deepest gratitude to my family, for their unconditional love, patience, and encouragement throughout this journey.

Dedication

This is dedicated to my family, for their constant love and support.

Table of Contents

List of Figures xi

List of Tables xii

1 Introduction 1

1.1 Background . . . 1

1.1.1 Model-Based Control and Data-Driven Control . . . 1

1.1.2 Model Predictive Control . . . 2

1.2 Motivation for Data-Driven Predictive Control . . . 2

1.3 An Overview of Data-Driven Predictive Control . . . 3

1.4 Relevant Works and Research Objectives . . . 4

1.4.1 DDPC for Non-linear and Time-Varying Systems . . . 5

1.4.2 DDPC for Stochastic Systems . . . 5

1.4.3 Motivation and Research Objectives . . . 6

1.5 Outline and Contributions . . . 7

1.6 Notations . . . 8

2 Preliminaries 9

2.1 Problem Statement: Deterministic LTI Case . . . 9

2.2 Deterministic Model Predictive Control . . . 11

2.3 Behavioral Systems Theory for LTI Systems . . . 12

2.3.1 Behavioral Representation of LTI Systems 13
2.3.2 State-Space Representation of Behavior 14
2.3.3 Controllability of Behavior 15
2.3.4 Structure Indices: Order and Lag 16
2.3.5 Specification of Initial Condition 17
2.3.6 Fundamental Lemma 18
2.4 Data-Driven Predictive Control for Deterministic LTI Systems 19
2.4.1 Motivation for Data-Driven Predictive Control 20
2.4.2 DDPC Methods: DeePC and SPC 23
2.5 Equivalence of MPC and DDPC for Deterministic LTI Systems 25
2.6 Regularization of DeePC and SPC 26

3 Data-Driven Predictive Control for Linear Time-Periodic Systems 28
3.1 Problem Statement: Deterministic LTP Case 28
3.2 Lifting an LTP System to an LTI System 30
3.3 Behavioral Systems Theory for LTP Systems 31
3.3.1 Behavioral Representation of LTV Systems 31
3.3.2 A Definition of Order and Lag for LTV Systems 35
3.3.3 Behavioral Systems Theory for LTP Systems 37
3.3.4 A Fundamental Lemma for LTP Systems 40
3.4 DDPC Methods for LTP Systems 41
3.4.1 Offline Data Collection 42
3.4.2 Online Warm-Up Process 44
3.4.3 Online Control Process 47
3.5 Simulations 50
3.6 Chapter Conclusions 52
3.7 Appendices 54
3.7.1 Proof of Lemma 3.4 54

3.7.2 Proof of Lemma 3.5 55
3.7.3 Proof of Lemma 3.11 56
3.7.4 Proof of Lemma 3.18 59
3.7.5 Proof of Proposition 3.20 60

4 Stochastic Data-Driven Predictive Control 62

4.1 Problem Statement: Stochastic LTI Case 62
4.1.1 Gaussian Distribution Case 64
4.1.2 Distributionally Robust Case 64
4.1.3 Our Objective: An Equivalent Data-Driven Method 65
4.2 Stochastic Model Predictive Control 66
4.2.1 Initial Condition and State Estimation 66
4.2.2 Interpolation of Initial Condition 67
4.2.3 Feedback Control Policies: Output Feedback 68
4.2.4 Feedback Control Policies: Output Error Feedback 70
4.2.5 Deterministic Approximation of Chance Constraint 72
4.2.6 Deterministic Reformulation of DR-CVaR Constraint 73
4.2.7 Terminal Condition 74
4.2.8 SMPC Optimization Problem and Implementation 75
4.2.9 Closed-Loop Properties 75
4.3 Stochastic Data-Driven Predictive Control 77
4.3.1 Use of Offline Data 77
4.3.2 Data-Driven State Estimation and Output Feedback 79
4.3.3 Optimization Problem 83
4.3.4 Online Control Algorithm 85
4.4 Simulations 88
4.4.1 Simulations on Grid-Connected Power Converter System 88
4.4.2 Simulations on Batch Reactor System 94

4.5 Chapter Conclusions 97
4.6 Appendices 98
4.6.1 Proof of (4.18) 98
4.6.2 Iterative Risk Allocation 99
4.6.3 Proof of Lemma 4.5 101
4.6.4 Proof of Lemma 4.6 102
4.6.5 Proof of Lemma 4.7 104
4.6.6 Proof of Lemma 4.8 105
4.6.7 Proof of Lemma 4.9 107
4.6.8 Proof of (4.57) and (4.58) 111
4.6.9 Proof of Claim 4.13.1 119

5 Conclusions and Outlook **122**
5.1 Contributions 122
5.2 Potential Future Topics 123

References **126**

List of Figures

1.1 Direct and indirect data-driven control. 1
1.2 State of the art in MPC and DDPC for stochastic systems. 6

2.1 Controllability of LTI behavior. 15
2.2 Order, lag, and the dimensionality of restricted behavior. 17
2.3 Output prediction in model-based and data-driven control. 21
2.4 Time horizons in Data-Driven Predictive Control. 22

3.1 The states, inputs and outputs of an LTP system and its lifted system. . . 32
3.2 Controllability of LTV behavior. 35
3.3 The column span of $\mathrm{col}(U_{\mathrm{p}}(\theta), U_{\mathrm{f}}(\theta), Y_{\mathrm{p}}(\theta), Y_{\mathrm{f}}(\theta))$ is the restricted behavior on time interval $[\mathbf{k}(\theta) - L, \mathbf{k}(\theta) + N)$. 44
3.4 The entire online process with a warm-up process. 45
3.5 Accumulating the errors δ_t^θ into Δ_t^θ. 45
3.6 A spring-mass-damper model for simulation. 50
3.7 Tracking cost of the simulated LTP system with different controllers. . . . 53

4.1 The one-line diagram of a grid-connected power converter. 89
4.2 Cumulative stage cost with different controllers, $N_{\mathrm{c}} = 10$. 92
4.3 Cumulative stage cost with different controllers, $N_{\mathrm{c}} = 1$. 92
4.4 The second output signals with SPC (light blue) and SDDPC (red) in the constraint satisfaction test. 93
4.5 The system's first output signal with DR/O-SDDPC. 96

List of Tables

1.1 A classification of some multivariate control methods. 3

3.1 Physical quantities of the simulated system. 51

3.2 Control parameters in simulation. 51

4.1 Control parameters of the grid-connected power-converter system. 91

4.2 Statistics of constraint violation of the second output channel from 0.5s to 2.0s. 93

4.3 Control parameters of the batch reactor system. 95

4.4 Simulation results of the batch reactor system. 95

Chapter 1

Introduction

1.1 Background

1.1.1 Model-Based Control and Data-Driven Control

Control design methods can broadly be classified into *model-based* methods and *data-driven* methods. Model-based design methods rely on a parametric representation (or a model) of the system, which may come from first-principles modeling or from system identification — a process of building a system model from recorded input-output data. Data-driven control, on the other hand, produces a control strategy directly from recorded historical data, with no requirement on system models. Note that the scheme of system identification followed by model-based control can also be considered as a data-driven control process,

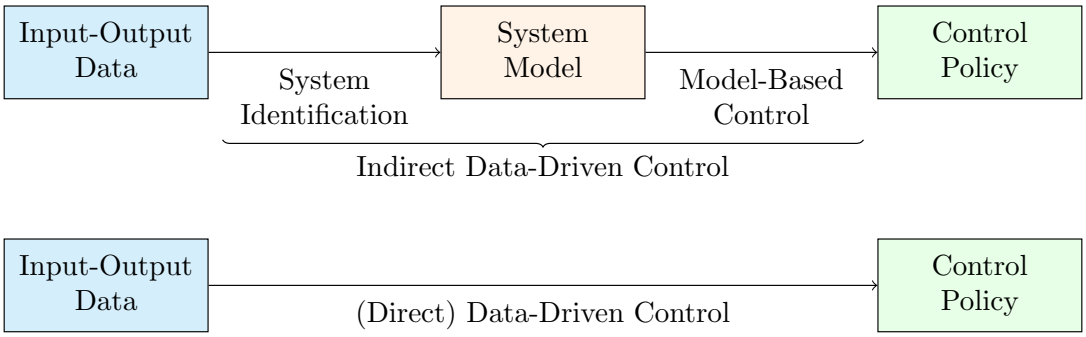

Figure 1.1: Direct and indirect data-driven control.

and is sometimes called *indirect data-driven control* [1]. *Direct* data-driven control, on the contrary, refers to data-driven methods with no model identified. See Figure 1.1 for a summary of those terms. Going forward, we use the term "data-driven control" to refer to "direct data-driven control". As modern systems of interest become increasingly complex and difficult to identify, direct data-driven control techniques become preferable, and have attracted significant research interest in recent years [2, 3]. A comprehensive survey of early data-driven control methods can be found in [4].

1.1.2 Model Predictive Control

Model Predictive Control (MPC) [5] is a particular model-based design method which has been widely used in industrial applications, such as autonomous driving [6], autonomous flight [7], mobile robots [8] and smart energy systems [9]. The MPC framework typically uses a receding horizon of prediction and planning, and at each planning step the future control decisions over the considered horizon are solved from an optimization problem. A main benefit of MPC, comparing to other classical model-based control methods such as the classical Linear-Quadratic(-Gaussian) Regulator (LQR/LQG), is the ability to incorporate input and state/output constraints, which typically model actuator saturation and safety constraints, respectively. Despite this applicational benefit, MPC requires a parametric system model for prediction of the future trajectory, and the modeling process is sometimes expensive, as discussed above.

For stochastic systems, work on *Stochastic MPC (SMPC)* [10, 11, 12] has focused on modelling the uncertainty in systems probabilistically. SMPC methods optimize over feedback control policies rather than control actions, resulting in performance benefits when compared to the naive use of deterministic MPC [13]. Additionally, SMPC allows the use of probabilistic constraints, useful for computing risk-aware controllers. Another MPC method dealing with uncertainty is *Robust MPC (RMPC)* [14], which attempts to conservatively guard against the worst-case deterministic uncertainty.

1.2 Motivation for Data-Driven Predictive Control

Although data-driven control methods show promise for complex or difficult-to-model systems, early works on data-driven control did not adequately account for constraints on inputs and outputs (see examples in [4]). This observation leads to the development of *Data-Driven Predictive Control (DDPC)*, a type of data-driven control method which can

	Data-Driven Control Methods	Model-Based Control Methods
considering constraints in control	DDPC	MPC
not considering constraints in control	early data-driven control methods [4]	classical LQR/LQG, classical H_∞ control

Table 1.1: A classification of some multivariate control methods.

incorporate input-output constraints as considered in MPC. As such, DDPC possesses both the benefit of MPC for handling constrained control and the aforementioned benefit of data-driven control over model-based control; see Table 1.1 for a comparison of DDPC, MPC and early data-driven control.

In predictive control, the future trajectory of the system should be predicted at each planning time step. In the MPC framework, the future trajectory is produced by a system model, while DDPC methods should generate the future trajectory with solely data. As such, DDPC methods can be viewed as derived from MPC where the parametric system model is replaced by a non-parametric, data-based representation of the system. Various approaches exist for constructing such data-based system representation, resulting in different types of DDPC methods, which we will introduce in the next section.

1.3 An Overview of Data-Driven Predictive Control

DDPC methods can be categorized based on different data-based representations of the dynamic system, where the data-based representations are used to predict the future trajectory in the control process. Below, we highlight some of the most common DDPC frameworks.

Subspace Predictive Control (SPC) [15, 16] is a DDPC framework, where a data-based prediction matrix is obtained through a subspace identification approach and is used to predict the future trajectory of the system, so that the need for a parametric model is eliminated. Research on SPC dates back to [17, 18] and has since been applied to control problems of complex systems such as airplanes [19] and nuclear reactors [20].

Learning-based predictive control [21, 22] is also a prominent DDPC framework, where the system dynamics is captured by a neural network trained by input-output data. This

control approach has gained significant attention due to the advancements in reinforcement learning.

Another class of DDPC methods is grounded in *behavioral systems theory* [23, 24]. In the behavioral approach, a dynamical system is characterized by the set of all possible input-output trajectories it can produce, referred to as the system's *behavior* [25, 26, 27, 28, 29, 30]. Of particular note, for a finite-dimensional discrete-time deterministic linear time-invariant (LTI) system, this behavior can be represented using a data matrix constructed from historical input-output data collected from the system — a result now known as the *fundamental lemma* [31]. By leveraging the fundamental lemma in the behavioral framework, a data matrix is sufficient to reflect the system's dynamics. This approach has led to the development of data-driven tracking controllers [32], data-driven feedback controllers [33, 34, 35, 36] and behavior-based DDPC methods. Notable examples of the latter include the work in [37] and the Data-enabled Predictive Control (DeePC) framework [38, 39, 40]. DeePC has been successfully applied to the control of power systems [41, 42], motor drives [43], quad-copters [44] and excavators [45]. Other behavior-based DDPC methods were designed to ensure closed-loop stability [46, 47, 48, 49].

Equivalence between DDPC and MPC

On the theoretical front, it has been proved that DeePC and SPC, as DDPC methods, can produce the same control actions as those obtained from MPC, for deterministic linear time-invariant (LTI) systems [38, 15, 50]. This result establishes a theoretical equivalence between the DDPC methods and the MPC framework in the idealized deterministic LTI case. For data-driven control methods, achieving performance on par with model-based control is the best possible outcome, because data-driven control cannot outperform model-based control (with an exact model). As detailed in the next section, our research seeks to extend the theoretical equivalence between model-based and data-driven control beyond deterministic LTI systems.

1.4 Relevant Works and Research Objectives

In Section 1.3, we provided an overview of DDPC methods. Many of these methods, including DeePC and SPC, are tailored for deterministic LTI systems. Regularized versions of these control methods have demonstrated robustness to data disturbances and system non-linearity [38, 39], and have been successfully applied in simulations and experiments. Meanwhile, other DDPC methods are developed to handle a wider range of systems beyond

deterministic LTI systems, which we will explore in Section 1.4.1 and Section 1.4.2. Finally, we present our research objectives in Section 1.4.3.

1.4.1 DDPC for Non-linear and Time-Varying Systems

Some DDPC methods were designed for nonlinear and time-varying systems. For non-linear systems, DDPC methods have been proposed for various types, including Hammerstein-Wiener systems [51, 52], second order discrete Volterra systems [53], flat nonlinear systems [54], polynomial time-invariant systems [55] and the worst-case scenario for general non-linear systems [56]. These methods often extend Willems' fundamental lemma to accommodate the specific characteristics of these nonlinear systems.

For time-varying systems, DDPC methods have been developed for specific types, such as linear parameter-varying systems [57, 58], linear slowly-varying systems [59] and the worst-case scenario for general linear time-varying systems [60]. Again, in these cases, Willems' fundamental lemma has sometimes been extended to address the aspects of time-varying systems. As discussed in the next section, one of our contributions will be the development of DDPC methods for time-periodic systems, an area that has not been addressed in the existing literature.

1.4.2 DDPC for Stochastic Systems

Data-driven control methods are typically designed with robustness in mind, ensuring that their performance remains insensitive to noisy data from stochastic systems. Both DeePC and SPC, as representative DDPC methods, have robust versions designed to adapt to stochastic systems. In application of SPC with noisy data, a predictor matrix is often computed with denoising methods, such as prediction error methods [41, 42] and truncated singular value decomposition [43]. Robust versions of DeePC have also been developed with stochastic systems in mind, such as norm-based regularized DeePC [38, 39] in which the regularization can be interpreted as a result of worst-case robust optimization [61, 62], as well as distributionally robust DeePC [39, 40]. While the stochastic adaptations of DeePC and SPC were validated through experiments, these stochastic data-driven control methods lack an analogous theoretical equivalence to any model-based MPC method, in contrast to the deterministic case; see Figure 1.2. Other related works include tube-based [63], sampling-based [64], innovation-based [65] and constraint-tightening [66], multiplicative-noise [67] and distributionally robust [68, 69, 70, 71, 72] stochastic DDPC schemes. These methods were validated through simulations, and most of them guarantee recursive feasibility

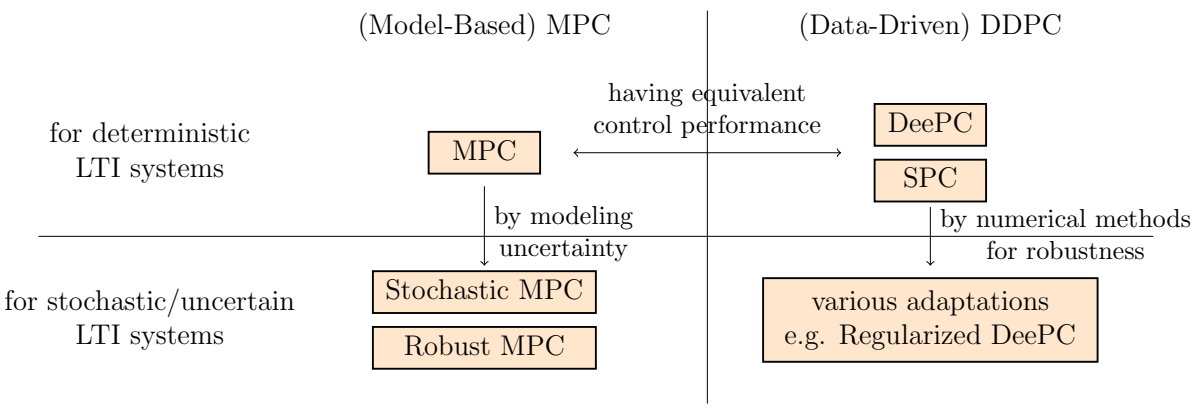

Figure 1.2: State of the art in MPC and DDPC for stochastic systems.

and closed-loop stability. Again, however, no equivalence in performance was established between these methods and model-based MPC methods.

This disconnect between data-driven and model-based methods in the stochastic case has been noticed by some researchers, and some recent DDPC methods were developed for stochastic systems that have provable equivalence to model-based MPC methods. The works in [73, 74, 75, 76] proposed data-driven control frameworks for stochastic systems applying Polynomial Chaos Expansion (PCE); the use of PCE enables modeling of arbitrary noise distributions. Their methods have equivalent performance to SMPC when disturbances are known and when stochastic signals are exactly represented by finite PCE terms [73, Thm. 1] [74, Cor. 1]. In practice, disturbances should be estimated using input-output data, which estimation requires heavier computation with larger amount of data. Their frameworks have considered systems without sensor noise and systems in the Auto-Regressive form with eXogenous input (ARX), which are special cases of systems in the state-space representation. Thus, the gap addressed here is to develop an alternative data-driven stochastic control method that has provably equivalent performance to the model-based SMPC, where we only need to estimate a fixed number of parameters regardless of the data amount and we consider general systems in the state-space form with separate process noise and measurement noise.

1.4.3 Motivation and Research Objectives

Among the numerous developed DDPC methods, we currently lack a clear understanding of which data-driven approach offers the best control performance. Due to the absence of

system model information, data-driven control methods cannot be expected to outperform model-based control methods with an exact model in any given control problem. Therefore, the best we can hope for with data-driven control is a performance equivalent to that of model-based control with an exact model. As mentioned in Section 1.3, some DDPC methods have equivalent control performance with MPC for deterministic LTI systems, which are a type of idealized dynamical systems. However, real-life dynamical systems are often stochastic, non-linear and sometimes time-varying. Motivated by this observation, our objective is to extend the theoretical equivalence between DDPC and MPC to encompass a broader range of control systems beyond deterministic LTI systems, by developing appropriately tailored DDPC methods. This extension can take three main directions: towards stochastic systems, non-linear systems and time-varying systems — each corresponding to a generalization of deterministic LTI systems. In this section, we investigate the three directions separately with review of relevant works.

As elaborated in the next section, this thesis contributes to the development of DDPC methods for a specific class of time-varying systems as well as for stochastic systems.

1.5 Outline and Contributions

The rest of the thesis is organized as follows. Chapter 2 provides an overview of the preliminaries relevant to our research. This includes a review of behavioral systems theory (Section 2.3) and and fundamental DDPC methods (Section 2.4). Chapter 3 presents our contributions to DDPC methods for linear time-periodic (LTP) systems, a specific subclass of linear time-varying (LTV) systems, based on our work in [77]. Chapter 4 details our development of a stochastic DDPC framework, referring to our works in [78, 79]. Chapter 5 concludes the thesis and summarizes the findings.

Chapter 3: Data-Driven Predictive Control for Linear Time-Periodic Systems

We consider the problem of data-driven predictive control for an unknown discrete-time linear time-periodic (LTP) system of known period. Our proposed strategy generalizes both DeePC and SPC, which are established data-driven control techniques for LTI systems. The approach is supported by an extensive theoretical development of behavioral systems theory for LTP systems, culminating in a generalization of Willems' fundamental lemma. Our algorithm produces results identical to standard MPC for deterministic LTP systems. Robustness of the algorithm to noisy data is illustrated via simulation of a regularized version of the algorithm applied to a stochastic multi-input multi-output LTP system.

Chapter 4: Stochastic Data-Driven Predictive Control

We propose a data-driven receding-horizon control method dealing with the chance constrained output-tracking problem of unknown stochastic LTI systems. The proposed method takes into account the statistics of the process noise, the measurement noise and the uncertain initial condition, following an analogous framework to Stochastic MPC, but does not rely on the use of a parametric system model. As such, our receding-horizon algorithm produces a sequence of closed-loop control policies for predicted time steps, as opposed to a sequence of open-loop control actions. Under certain conditions, we establish that our proposed data-driven control method produces identical control inputs as that produced by the associated model- based SMPC. Simulation results on a grid-connected power converter are provided to illustrate the performance benefits of our methodology.

1.6 Notations

Throughout the thesis, we apply the following notations.

$\mathbb{Z}_{[a,b]}$	the set of consecutive integers from a to b, i.e., $\mathbb{Z} \cap [a, b]$
$\mathbb{Z}_{[a,b)}$	the set of consecutive integers from a to $b-1$, i.e., $\mathbb{Z} \cap [a, b)$
$\mathbb{S}^q_+$	the set of $q \times q$ positive semi-definite symmetric matrices
$\mathbb{S}^q_{++}$	the set of $q \times q$ positive definite symmetric matrices
$M^\dagger$	the Moore-Penrose pseudo inverse of a matrix M
$\otimes$	the Kronecker product
$\mathrm{col}(M_1, \ldots, M_k)$	the column concatenation of matrices/vectors $M_1, \ldots, M_k$
$\mathrm{Diag}(M_1, \ldots, M_k)$	the block-diagonal concatenation of matrices $M_1, \ldots, M_k$

For a $\mathbb{R}^q$-valued discrete-time signal z with integer index t, we let $z_{[t_1,t_2]}$ denote either

- a vector sequence $\{z_t\}_{t=t_1}^{t_2} \subseteq \mathbb{R}^q$ or
- a concatenated vector $\mathrm{col}(z_{t_1}, \ldots, z_{t_2}) \in \mathbb{R}^{q(t_2-t_1+1)}$

where the usage is clear from the context; let $z_{[t_1,t_2)} := z_{[t_1,t_2-1]}$. A matrix sequence $\{M_t\}_{t=t_1}^{t_2}$ and a function sequence $\{\pi_t(\cdot)\}_{t=t_1}^{t_2}$ are denoted by $M_{[t_1,t_2]}$ and $\pi_{[t_1,t_2]}$ respectively.

Chapter 2

Preliminaries

In this chapter, we review some key foundational works that provide the prerequisites for our research presented in Chapter 3 and Chapter 4. Our discussion here will primarily focus on a constrained control problem of unknown deterministic linear time-invariant (LTI) systems, as formally stated in Section 2.1. The control problem for unknown systems must be addressed using data-driven control methods, whereas for known systems, Model Predictive Control (MPC) provides a sufficient solution, as outlined in Section 2.2. We provide a brief review of the behavioral systems theory in Section 2.3, which forms the theoretical foundation of the Data-enabled Predictive Control (DeePC) method. An overview of DeePC and Subspace Predictive Control (SPC) methods is in Section 2.4, which are typical Data-Driven Predictive Control (DDPC) methods. In Section 2.5, we explore the performance equivalence between the DDPC methods, DeePC and SPC, and the MPC method. Finally, in Section 2.6, we extend our discussion beyond deterministic LTI systems to show how regularization techniques are applied to adapt DeePC and SPC methods for real-world non-linear stochastic systems.

2.1 Problem Statement: Deterministic LTI Case

Throughout the chapter, we focus on the control of a deterministic LTI system described by the following state-space model $\mathcal{S}$,

$$\mathcal{S} : \begin{cases} x_{t+1} = Ax_t + Bu_t \\ y_t = Cx_t + Du_t \end{cases} \tag{2.1}$$

with state $x_t \in \mathbb{R}^n$, input $u_t \in \mathbb{R}^m$, output $y_t \in \mathbb{R}^p$ and system matrices A, B, C, D. The state x_t is *unmeasured* and the initial state x_0 is unknown. In the data-driven scenario, the system matrices A, B, C, D in (2.1) are *unknown*; we have access only to the input u_t and output y_t in in (2.1). We assume the pair (A, B) is controllable. Moreover, without loss of generality for an unknown system, we assume the pair (A, C) is observable, as justified in [23, Sec. 2.4].

In a reference tracking problem, the objective is for the output y_t to follow a specified reference signal $r_t \in \mathbb{R}^p$. The trade-off between the tracking error $y_t - r_t$ and the control effort u_t may be encoded in the quadratic instantaneous cost

$$J_t(u_t, y_t) := \|y_t - r_t\|_Q^2 + \|u_t\|_R^2 \tag{2.2}$$

at time $t \in \mathbb{N}_{\geq 0}$, where $Q \in \mathbb{S}_+^p$ and $R \in \mathbb{S}_{++}^m$ are user-selected parameters. The control objective is to minimize the accumulation of the cost (2.2) over a horizon. Other alternative convex cost functions in u_t and y_t may be considered instead of (2.2) as a different problem setup. This tracking should be achieved subject to constraints on the inputs and outputs in the form of

$$u_t \in \mathcal{U}, \qquad y_t \in \mathcal{Y} \tag{2.3}$$

for time $t \in \mathbb{N}_{\geq 0}$, where the constraint sets $\mathcal{U} \subseteq \mathbb{R}^m$ and $\mathcal{Y} \subseteq \mathbb{R}^p$ are assumed to be convex, non-empty and closed.

Notations based on System Model

For the state-space model (2.1), we define for $L \in \mathbb{N}$ the (reversed) extended controllability matrix $\mathcal{C}_L \in \mathbb{R}^{n \times mL}$, the extended observability matrix $\mathcal{O}_L \in \mathbb{R}^{pL \times n}$ and the impulse-response matrix $\mathcal{G}_L \in \mathbb{R}^{pL \times mL}$ of depth L.

$$\begin{aligned} \mathcal{C}_L &:= \begin{bmatrix} A^{L-1}B & \cdots & A^2B & AB & B \end{bmatrix} \\ \mathcal{O}_L &:= \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{L-1} \end{bmatrix}, \quad \mathcal{G}_L := \begin{bmatrix} D & & & & \\ CB & D & & & \\ CAB & CB & D & & \\ \vdots & \ddots & \ddots & \ddots & \\ CA^{L-2}B & \cdots & CAB & CB & D \end{bmatrix} \end{aligned} \tag{2.4}$$

With this notation, the unique resulting state and resulting output of (2.1) with initial state x_{t_0} at time $t = t_0$ can be expressed as follows,

$$x_{t_1} = A^{t_1 - t_0}\, x_{t_0} + \mathcal{C}_{t_1 - t_0}\, u_{[t_0, t_1)} \tag{2.5a}$$

$$y_{[t_0, t_1)} = \mathcal{O}_{t_1 - t_0}\, x_{t_0} + \mathcal{G}_{t_1 - t_0}\, u_{[t_0, t_1)} \tag{2.5b}$$

for all future time $t_1 > t_0$.

2.2 Deterministic Model Predictive Control

With the system model in (2.1) unknown, data-driven control techniques are necessary to address the problem outlined in Section 2.1. Nonetheless, in the case where the system model is known, the constrained control problem can be effectively solved using deterministic Model Predictive Control (MPC) as a standard method [5]. In this section, we review a specific framework of deterministic MPC with partial state observation.

The MPC framework operates based on a receding-horizon strategy. At each *control step* $t = k$, as a specific time step, control decisions are made through optimization for N upcoming time steps, referred to as the *prediction horizon*. Among these decisions, the control actions for the first N_c steps, known as the *control horizon*, are applied to the system. The lengths of the prediction and control horizons, $N, N_\mathrm{c} \in \mathbb{N}$ with $N_\mathrm{c} \leq N$, are fixed parameters. Once the control inputs for the current control horizon are applied, the control step is updated to $t = k + N_\mathrm{c}$, and the process is repeated, with the time horizons shifting forward. The initial control step corresponds to the starting time, $t = 0$, marking the beginning of the receding-horizon cycle.

Here we formulate an MPC optimization problem, aligning with the MPC scheme reviewed in [38]. Consider the control step $t = k$, and we are given an estimate $\hat{x}_k$ of the state x_k through a state estimator. Let $\bar{x}$, $\bar{u}$ and $\bar{y}$ denote the prediction of future state, input and output trajectories over the prediction horizon, so they should match the system model (2.1) as

$$\bar{x}_{t+1} = A\bar{x}_t + B\bar{u}_t, \qquad t \in \mathbb{Z}_{[k, k+N)} \tag{2.6a}$$

$$\bar{y}_t = C\bar{x}_t + D\bar{u}_t, \qquad t \in \mathbb{Z}_{[k, k+N)} \tag{2.6b}$$

$$\bar{x}_k = \hat{x}_k \tag{2.6c}$$

with the initial condition $\bar{x}_k$ set as $\hat{x}_k$, and satisfy the constraint (2.3) as

$$\bar{u}_t \in \mathcal{U}, \quad \bar{y}_t \in \mathcal{Y}, \qquad t \in \mathbb{Z}_{[k, k+N)} \tag{2.7}$$

over the prediction horizon. The objective function in optimization is chosen as the prediction of cost (2.2) summing over the prediction horizon,

$$\sum_{t=k}^{k+N-1} J_t(\bar{u}_t, \bar{y}_t) \tag{2.8}$$

and thus a deterministic MPC optimization problem at control step $t = k$ is

$$\underset{\bar{u}_{[k,k+N)}}{\text{minimize}} \ (2.8) \qquad \text{subject to} \quad (2.6) \text{ and } (2.7) \tag{MPC}$$

where $\bar{u}_{[k,k+N)}$ are the decision variables that determine other variables $\bar{x}_{[k,k+N]}$ and $\bar{y}_{[k,k+N)}$ through (2.6). With the receding-horizon process, our investigated MPC scheme is shown in Algorithm 1.

Algorithm 1 A Framework of Deterministic Model Predictive Control [38, Sec. III]

Input: prediction- and control-horizon lengths N, N_c, system model (A, B, C, D), stage cost function $J_t(\cdot)$, and constraint sets $\mathcal{U}, \mathcal{Y}$.

1: Initialize the control step $k \leftarrow 0$.
2: **while** `true` **do**
3: Generate state estimate $\hat{x}_k$ using past input-output data.
4: Solve control actions $\bar{u}_{[k,k+N)}$ from problem (MPC).
5: **for** t **from** k **to** $k + N_c - 1$ **do**
6: Apply input $u_t \leftarrow \bar{u}_t$ to the system (2.1).
7: Measure output y_t from the system (2.1).
8: Set $k \leftarrow k + N_c$.

The MPC framework outlined in Algorithm 1 does not inherently guarantee recursive feasibility and closed-loop stability. These properties can be ensured by augmenting the MPC problem (MPC) with terminal costs in addition to (2.8) and terminal constraints in addition to (2.7), as reviewed in [5]. Later in Section 2.5, the model-based MPC framework in Algorithm 1 will be compared to the data-driven DeePC and SPC methods introduced in Section 2.4, establishing connections between the model-based and data-driven approaches.

2.3 Behavioral Systems Theory for LTI Systems

The MPC approach described in Section 2.2 provides a model-based solution to the control problem outlined in Section 2.1, in the case where the system model is known. When

the system model is unknown, data-driven control methods are necessary to address the problem. Before presenting these data-driven control methods in Section 2.4, we first introduce key results from the behavioral systems theory [23, 24, 25, 26, 27, 28, 29, 30] as preliminary groundwork.

The concept of behavior is introduced in Section 2.3.1, followed by a review of its key properties, including state-space representation (Section 2.3.2), controllability (Section 2.3.3), and complexity indices such as order and lag (Section 2.3.4). Then, we present in Section 2.3.5 and Section 2.3.6 essential results that play pivotal roles in the formulation of DDPC methods.

2.3.1 Behavioral Representation of LTI Systems

In behavioral framework, a dynamical system is described in a different way from using a state-space model, but instead is described as a collection of possible input-output trajectories from the system, which collection is called the *behavior* of the system. In original works [25, 26, 27, 28], a behavior $\mathscr{B} \subseteq (\mathbb{R}^{m+p})^{\mathbb{N}}$ is defined as a subset of the space $(\mathbb{R}^{m+p})^{\mathbb{N}}$ of $\mathbb{R}^{m+p}$-valued semi-infinite signals, where m and p are the input dimension and output dimension, respectively. Some notions such as linearity and time-invariance are thus defined in a behavioral way based on the property of the subset $\mathscr{B}$. In this report, for simplicity of presentation, we define behavior based on the state-space representation (2.1) of systems, which definition is equivalent to the formal definition of behavior for finite-dimensional LTI systems [23, Sec. 2.3].

Definition 2.1 (Behavior)**.** For the finite-dimensional LTI system $\mathcal{S}$ in (2.1), the *behavior* $\mathscr{B}^{\mathcal{S}}$ of $\mathcal{S}$ is the set of semi-infinite trajectories

$$\mathscr{B}^{\mathcal{S}} := \left\{ \begin{bmatrix} u_{[0,\infty)} \\ y_{[0,\infty)} \end{bmatrix} \;\middle|\; \exists\, x_0 \text{ s.t. (2.1) holds for all } t \in \mathbb{N}_{\geq 0} \right\}.$$

Let $\mathscr{B}_k^{\mathcal{S}}$ be the restriction of behavior $\mathscr{B}^{\mathcal{S}}$, or the *restricted behavior*, on the first $k \in \mathbb{N}$ time steps, that is,

$$\mathscr{B}_k^{\mathcal{S}} := \left\{ \begin{bmatrix} u_{[0,k)} \\ y_{[0,k)} \end{bmatrix} \;\middle|\; \exists\, x_0 \text{ s.t. (2.1) holds for all } t \in \mathbb{Z}_{[0,k)} \right\}.$$

We can analogously define the behavior $\mathscr{B}^{\mathcal{S}}$ for a non-LTI state-space model $\mathcal{S}$ in a manner similar to Definition 2.1. We write a behavior as $\mathscr{B}$, and a restricted behavior as

$\mathscr{B}_k$, when the underlying system model is not emphasized. We say that a behavior $\mathscr{B}$ is an *LTI behavior* if it is equal to the behavior $\mathscr{B}^{\mathcal{S}}$ of some LTI system $\mathcal{S}$ by Definition 2.1.

Going forward, we discuss mainly the restricted behaviors. For linear systems, the behavior is always a vector space, where the LTI restricted case is captured as follows.

Lemma 2.2. *For LTI system $\mathcal{S}$ in (2.1) and integer $k \in \mathbb{N}$, the restricted behavior $\mathscr{B}_k^{\mathcal{S}}$ is a finite-dimensional vector space*

$$\mathscr{B}_k^{\mathcal{S}} = \mathrm{ColSpan} \begin{bmatrix} 0_{mk\times n} & I_{mk} \\ \mathcal{O}_k & \mathcal{G}_k \end{bmatrix}$$

with matrices $\mathcal{O}_k, \mathcal{G}_k$ defined in (2.4).

Corollary 2.2.1. $\dim \mathscr{B}_k^{\mathcal{S}} = \mathrm{rank}(\mathcal{O}_k) + mk$, *where m is the input dimension of $\mathcal{S}$.*

2.3.2 State-Space Representation of Behavior

The behavioral representation $\mathscr{B}^{\mathcal{S}}$ of an LTI state-space model $\mathcal{S}$ is unique, according to Definition 2.1. On the contrary, however, the state-space representation of an LTI behavior may not be unique, as we specify in Lemma 2.4, where we use the notion of equivalent LTI systems defined in Definition 2.3.

Definition 2.3 (Equivalent LTI Systems)**.** LTI state-space models $\mathcal{S}_{\mathrm{I}} : (A_{\mathrm{I}}, B_{\mathrm{I}}, C_{\mathrm{I}}, D_{\mathrm{I}})$ and $\mathcal{S}_{\mathrm{II}} : (A_{\mathrm{II}}, B_{\mathrm{II}}, C_{\mathrm{II}}, D_{\mathrm{II}})$ are said to be *equivalent* if they have the same state dimension and there exists a non-singular matrix P such that

$$A_{\mathrm{II}} = P^{-1}A_{\mathrm{I}}P, \quad B_{\mathrm{II}} = P^{-1}B_{\mathrm{I}}, \quad C_{\mathrm{II}} = C_{\mathrm{I}}P, \quad D_{\mathrm{II}} = D_{\mathrm{I}}.$$

Lemma 2.4 (State-Space Representations of Same LTI Behavior [25, Sec. 5])**.** *For LTI state-space models $\mathcal{S}_{\mathrm{I}}$ and $\mathcal{S}_{\mathrm{II}}$, their behaviors are equal, i.e.,*

$$\mathscr{B}^{\mathcal{S}_{\mathrm{I}}} = \mathscr{B}^{\mathcal{S}_{\mathrm{II}}}$$

if, and only if, the observable systems of $\mathcal{S}_{\mathrm{I}}$ and $\mathcal{S}_{\mathrm{II}}$ are equivalent in the sense of Definition 2.3.

Lemma 2.4 provides a sufficient and necessary condition for different LTI system models to possess the same behavior. Through Lemma 2.4, given an LTI state-space model $\mathcal{S}$, we are able to know all state-space models that share the same behavior with the model $\mathcal{S}$. Although a behavior has multiple state-space representations, we may focus on those representations with the smallest state dimension, which are called *minimal* representations.

Definition 2.5 (Minimal State-Space Representations [23, 24])**.** For an LTI behavior $\mathscr{B}$, an LTI state-space model $\mathcal{S}_{\mathrm{m}}$ is called a *minimal state-space representation* of $\mathscr{B}$ if

- $\mathcal{S}_{\mathrm{m}}$ is a state-space representation of $\mathscr{B}$, i.e., $\mathscr{B} = \mathscr{B}^{\mathcal{S}_{\mathrm{m}}}$, and
- among all state-space representations of $\mathscr{B}$, $\mathcal{S}_{\mathrm{m}}$ has the minimal state dimension.

Minimal state-space representations of an LTI behavior are not unique, but they are all equivalent in the sense of Definition 2.3. Moreover, for LTI behaviors, a state-space representation is minimal if, and only if, it is observable [25, Sec. 5].

2.3.3 Controllability of Behavior

Lemma 2.4 revealed the role of observability in behavioral representation. Next, we discuss controllability. In the behavioral framework, controllability is defined in a trajectory-based sense, as opposed to the more classical notion of state-controllability.

Definition 2.6 (Controllability for LTI Behaviors [31, 38])**.** An LTI behavior $\mathscr{B}$ is *controllable* if for any finite trajectory $w^{\mathrm{I}}_{[0,t_1)} \in \mathscr{B}_{t_1}$ with $t_1 \in \mathbb{N}$ and any trajectory $w^{\mathrm{II}}_{[0,\infty)} \in \mathscr{B}$, there exists a trajectory $w^{\diamond}_{[0,\infty)} \in \mathscr{B}$ and an integer $t_2 \geq t_1$ such that

$$w^{\diamond}_{[0,t_1)} = w^{\mathrm{I}}_{[0,t_1)}, \qquad w^{\diamond}_{[t_2,\infty)} = w^{\mathrm{II}}_{[0,\infty)}, \tag{2.9}$$

where we let $w_{[s_1,s_2]} := \mathrm{col}(u_{[s_1,s_2]}, y_{[s_1,s_2]})$ denote a trajectory for integers $s_1 \leq s_2$.

In other words, for a controllable behavior, two trajectories can always be patched together in finite time and form a new valid trajectory; see Fig 2.1. Relating to classical state-space concepts, the behavior is controllable if, and only if, all observable states are reachable [23, Sec. 2.4].

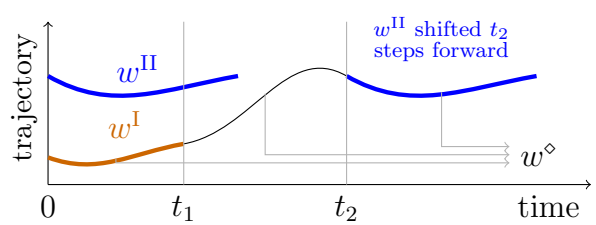

Figure 2.1: Controllability of LTI behavior.

The controllability of LTV behaviors is defined slightly different from Definition 2.6 and will be revisited in Definition 3.6 in Chapter 3. Here we focus on LTI behaviors.

2.3.4 Structure Indices: Order and Lag

The complexity of a behavior is described by integer invariants including *order* and *lag*. Order and lag are inherent properties of LTI behaviors, and they can be represented or defined in multiple equivalent ways, such as investigating dimensionality of restricted behavior [23, Sec. 2.2] [25, Sec. 7] [28, Sec. X], using kernel representation [23, Sec. 2.3] [80, Prop. 3] and using minimal representation [80, Sec. III.A]. Here we define order and lag in terms of the dimensions of restricted behaviors.

We first investigate the dimensions of restricted behaviors. For an LTI system with finite state dimension, the dimension $\dim \mathscr{B}_k$ of its restricted behavior has the following properties [25, Sec. 7] [28, Sec. X].

1. The dimension $\dim \mathscr{B}_k$ monotonically increases as k increases, while the increment in $\dim \mathscr{B}_k$ monotonically decreases as k increases, i.e.,
$$0 \leq \dim \mathscr{B}_1 \leq \dim \mathscr{B}_2 \leq \cdots, \tag{2.10a}$$
$$\dim \mathscr{B}_1 - 0 \geq \dim \mathscr{B}_2 - \dim \mathscr{B}_1 \geq \dim \mathscr{B}_3 - \dim \mathscr{B}_2 \geq \cdots. \tag{2.10b}$$

2. There exists $k_0 \in \mathbb{N}$ such that for all $k \geq k_0$, $\dim \mathscr{B}_k$ is an affine function of k,
$$\dim \mathscr{B}_k = n' + mk \tag{2.11}$$
where m is the system's input dimension and $n' \in \mathbb{N}_{\geq 0}$ is some intercept.

We consequently define order and lag based on those observations of $\dim \mathscr{B}_k$. The *order* $\mathbf{n}(\mathscr{B})$ of behavior $\mathscr{B}$ is defined as the integer offset n' in (2.11), and the *lag* $\mathbf{l}(\mathscr{B})$ is the smallest integer $k = k_0$ such that (2.11) holds [25, 28]. This definition can be written as

$$\mathbf{n}(\mathscr{B}) := \lim_{k\to\infty} \dim \mathscr{B}_k - mk, \tag{2.12a}$$
$$\mathbf{l}(\mathscr{B}) := \min \left\{ k \in \mathbb{N} : \dim \mathscr{B}_k = \mathbf{n}(\mathscr{B}) + mk \right\}. \tag{2.12b}$$

Figure 2.2 visualizes the relationship of order $\mathbf{n}(\mathscr{B})$, lag $\mathbf{l}(\mathscr{B})$ and the dimension of $\mathscr{B}_k$.

Order and lag are inherent properties of a behavior. However, given a state-space representation of the behavior, one can also represent the order and lag in terms of the state-space model. For LTI system $\mathcal{S} : (A, B, C, D)$ in (2.1), the order and lag of its behavior $\mathscr{B}^{\mathcal{S}}$ can be written as

$$\begin{aligned} \mathbf{n}(\mathscr{B}^{\mathcal{S}}) &= \lim_{k\to\infty} \operatorname{rank}(\mathcal{O}_k), \\ \mathbf{l}(\mathscr{B}^{\mathcal{S}}) &= \min \left\{ k \in \mathbb{N} : \operatorname{rank}(\mathcal{O}_k) = \mathbf{n}(\mathscr{B}^{\mathcal{S}}) \right\}, \end{aligned} \tag{2.13}$$

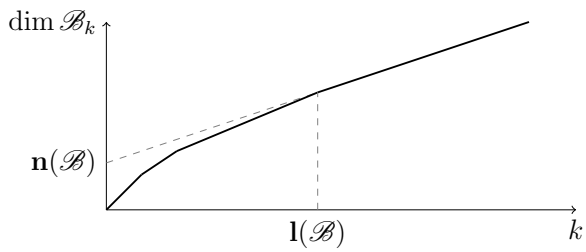

Figure 2.2: Order, lag, and the dimensionality of restricted behavior.

with matrix $\mathcal{O}_k$ defined in (2.4), according to Corollary 2.2.1 and the definitions in (2.12). Leveraging state-space terminology, the order $\mathbf{n}(\mathscr{B}^{\mathcal{S}})$ is the number of observable states, and the lag $\mathbf{l}(\mathscr{B}^{\mathcal{S}})$ is equal to the observability index, i.e., the smallest number of time steps on which all observable states are observed. Furthermore, for a minimal representation $\mathcal{S}_{\mathrm{m}} : (A_{\mathrm{m}}, B_{\mathrm{m}}, C_{\mathrm{m}}, D_{\mathrm{m}})$ of behavior $\mathscr{B}$, (2.13) is reduced as follows:

- the order $\mathbf{n}(\mathscr{B})$ is equal to the state dimension of $\mathcal{S}_{\mathrm{m}}$, and
- the lag $\mathbf{l}(\mathscr{B})$ is the smallest integer $k \in \mathbb{N}$ such that $\mathcal{O}_k$ has full column rank.

The above appears as an alternative definition of order and lag in the literature, based on minimal representation [24].

Order and lag are fundamental concepts in behavioral systems theory, serving as essential tools for characterizing system complexities. These structure indices often influence the validity of key theoretical results and their practical application. As we will see in Section 2.3.5 and Section 2.3.6, order and lag emerge as critical preconditions that guide the selection of certain user-defined parameters in the formulation of these theoretical results.

2.3.5 Specification of Initial Condition

With the causal system model (2.1), the output $y_{[t,N)}$ is uniquely determined given the input $u_{[t,t+N)}$ and the initial state x_t at time t, as demonstrated in (2.5b). However, in the model-free case where the state-space model is unknown, the initial condition cannot be specified as an initial state, since the state is not available. In the behavioral framework, we instead specify the initial condition at time t as a past input-output trajectory $u_{[t-L,t)}, y_{[t-L,t)}$ of the system. The following result shows that, if the initial-condition trajectory is sufficiently long, then the subsequence output can be uniquely determined given the corresponding input subsequence.

Lemma 2.7 (Initial-Condition Specification [32, Lemma 1] [23, Lemma 1])**.** *Consider an LTI behavior $\mathscr{B}$ and integers $L \geq \mathbf{l}(\mathscr{B})$ and $N \in \mathbb{N}$. Then, for any trajectories $\mathrm{col}(u_{[t-L,t)}, y_{[t-L,t)}) \in \mathscr{B}_L$ and any input subsequence $\bar{u}_{[t,t+N)}$, there exists a unique output subsequence $\bar{y}_{[t,t+N)}$ that satisfies*

$$\mathrm{col}(u_{[t-L,t)}, \bar{u}_{[t,t+N)}, y_{[t-L,t)}, \bar{y}_{[t,t+N)}) \in \mathscr{B}_{L+N}. \tag{2.14}$$

Lemma 2.7 ensures that the correct resulting output $\bar{y}_{[t,t+N)}$ can be identified given corresponding input $\bar{u}_{[t,t+N)}$ and the previous trajectory $u_{[t-L,t)}, y_{[t-L,t)}$ of sufficient length L. This result allows us to treat the input-output trajectory $u_{[t-L,t)}, y_{[t-L,t)}$ as an initial condition of the system at time t in the model-free case, enabling accurate prediction of the future output trajectory. As we will discuss in Section 2.4, the past input-output trajectory is commonly used as the initial condition in DDPC methods.

2.3.6 Fundamental Lemma

We have explored the behavior as an alternative representation of dynamical systems to state-space models. However, the notion of behavior cannot be directly used in data-driven control problems, as it is inherently defined based on the system dynamics which are unknown in the data-driven case. To make the behavioral systems theory applicable to data-driven control, the final building block is to represent the behavior itself in a data-driven manner.

In this section, we present a key result that enables the representation of restricted LTI behaviors using purely data from the system, known as the *Fundamental Lemma* [31, Thm. 1], which we will introduce as Lemma 2.9. We begin by defining the concept of persistent excitation.

Definition 2.8 (Persistent Excitation)**.** A sequence $z_{[t_1,t_2]}$ is *persistently exciting of order* K, for positive integer $K \leq t_2 - t_1 + 1$, if the associated block-Hankel matrix of depth K

$$\mathscr{H}_K(z_{[t_1,t_2]}) := \begin{bmatrix} z_{t_1} & z_{t_1+1} & \cdots & z_{t_2-K+1} \\ z_{t_1+1} & z_{t_1+2} & \cdots & z_{t_2-K+2} \\ \vdots & \vdots & \ddots & \vdots \\ z_{t_1+K-1} & z_{t_1+K} & \cdots & z_{t_2} \end{bmatrix}$$

has full row rank.

Lemma 2.9 (Fundamental Lemma [31])**.** *Let $\mathscr{B}$ be an LTI behavior, and let $\mathrm{col}(u^{\mathrm{d}}_{[1,T]}, y^{\mathrm{d}}_{[1,T]}) \in \mathscr{B}_T$ be a data trajectory from the system. For $K \in \mathbb{N}$, if*

(i) the behavior $\mathscr{B}$ is controllable in the sense of Definition 2.6, and
(ii) the input sequence $u^{\mathrm{d}}_{[1,T]}$ is persistently exciting of order $K + \mathbf{n}(\mathscr{B})$,

then we have

$$\operatorname{ColSpan}\begin{bmatrix}\mathscr{H}_K(u^{\mathrm{d}}_{[1,T]})\\ \mathscr{H}_K(y^{\mathrm{d}}_{[1,T]})\end{bmatrix} = \mathscr{B}_K. \tag{2.15}$$

Equality (2.15) means that an input-output sequence $\operatorname{col}(u_{[0,K)}, y_{[0,K)})$ is a trajectory of the system, i.e., $\operatorname{col}(u_{[0,K)}, y_{[0,K)}) \in \mathscr{B}_K$ if, and only if, it can be represented as

$$\begin{bmatrix}u_{[0,K)}\\ y_{[0,K)}\end{bmatrix} = \begin{bmatrix}\mathscr{H}_K(u^{\mathrm{d}}_{[1,T]})\\ \mathscr{H}_K(y^{\mathrm{d}}_{[1,T]})\end{bmatrix} g$$

for some vector $g \in \mathbb{R}^{T-K+1}$.

By Lemma 2.9, the restricted behavior can be fully represented by input-output data, with properly selected input and a controllability assumption. Some extensions of the fundamental lemma were also developed in the literature, including alleviation of the controllability requirement [80, 81], relaxations of persistent excitation [82, 81], extension to multiple dataset [83], online selection of input when recording data [84], robust versions [85] and equivalent formulations [86].

Recall from Lemma 2.7 that the future output $\bar{y}_{[t,t+N)}$ can be uniquely identified given future input $\bar{u}_{[t,t+N)}$ and past input-output data $u_{[t-L,t)}, y_{[t-L,t)}$, under the assumption that we know the restricted behavior $\mathscr{B}_{L+N}$. Through Lemma 2.9, $\mathscr{B}_{L+N}$ can be represented using data. Hence, by combination of Lemma 2.7 and Lemma 2.9, we can obtain some expression which is purely based on data and predicts the future output – playing a similar role as the constraint (2.6) in MPC. This motivates us to develop a data-driven counterpart of MPC, which will be introduced in Section 2.4.

2.4 Data-Driven Predictive Control for Deterministic LTI Systems

In this section, we introduce DDPC frameworks handling the control problem in Section 2.1. In Section 2.4.1, we start with motivation to DDPC, adapting the MPC framework in 2.2 and using the tools of the behavioral systems theory in Section 2.3. In Section 2.4.2, we introduce specific DDPC methods, including Data-enabled Predictive Control (DeePC) and Subspace Predictive Control (SPC).

2.4.1 Motivation for Data-Driven Predictive Control

As reviewed in Section 2.2, MPC is a model-based method because the output-prediction constraints (2.6) are based on an explicit system model. In data-driven scenario without a valid system model, MPC as a model-based method is no longer applicable. Because of the utility that MPC allows inherent inclusion of input and output constraints, we are interested in developing direct data-driven control methods which have the same utility. The latter type of control methods is so-called Data-Driven Predictive Control (DDPC).

To build up a data-driven counterpart of MPC, we need to replace the model-based output prediction (2.6) by some equivalent data-based expression. The output-prediction constraints (2.6) in problem (MPC) at control step $t = k$ can be compactly written as follows,

$$\boxed{\bar{y}_{[k,k+N)} = \mathscr{P}_{\mathcal{S}}\big(\bar{u}_{[k,k+N)},\, \hat{x}_k\big)} := \mathcal{O}_N\, \hat{x}_k + \mathcal{G}_N\, \bar{u}_{[k,k+N)} \tag{2.16}$$

with matrices $\mathcal{O}_N, \mathcal{G}_N$ defined in (2.4), where $\mathcal{S}$ is the label of the system model (2.1) and $\mathscr{P}_{\mathcal{S}}(\cdot)$ is the output-prediction function. That is, in MPC, the output prediction $\bar{y}_{[k,k+N)}$ is a function $\mathscr{P}_{\mathcal{S}}(\cdot)$ of future input $\bar{u}_{[k,k+N)}$ and the initial condition $\hat{x}_k$, where the function $\mathscr{P}_{\mathcal{S}}(\cdot)$ depends on the system model $\mathcal{S} : (A, B, C, D)$.

In DDPC, on the other hand, the output prediction at control step $t = k$ can be written in the following form,

$$\boxed{\bar{y}_{[k,k+N)} = \mathscr{P}_{\mathcal{D}}\Big(\bar{u}_{[k,k+N)},\, \begin{bmatrix} u_{[k-L,k)} \\ y_{[k-L,k)} \end{bmatrix}\Big)} \tag{2.17}$$

or some other expression that uniquely identifies $\bar{y}_{[k,k+N)}$ given $\bar{u}_{[k,k+N)}$ and $\begin{bmatrix} u_{[k-L,k)} \\ y_{[k-L,k)} \end{bmatrix}$ — an example is the DeePC framework to be introduced in Section 2.4.2. In (2.17), the future output $\bar{y}_{[k,k+N)}$ is uniquely determined as a function of future input $\bar{u}_{[k,k+N)}$ and recent input-output measurement $u_{[k-L,k)}, y_{[k-L,k)}$, with parameter $L \in \mathbb{N}$. The output-prediction function $\mathscr{P}_{\mathcal{D}}(\cdot)$ in (2.17) is formulated using a dataset $\mathcal{D}$ collected from the system, so that (2.17) is entirely based on data. The recent trajectory $u_{[k-L,k)}, y_{[k-L,k)}$ in (2.17) specifies the initial condition at time $t = k$, and plays the same role as $\hat{x}_k$ in (2.16). Figure 2.3 compares the output prediction in model-based and data-driven control. Therefore, the MPC framework can be adapted by replacing the model-based output-prediction constraint (2.16) by some data-based constraint in the form of (2.17), thereby formulating DDPC methods.

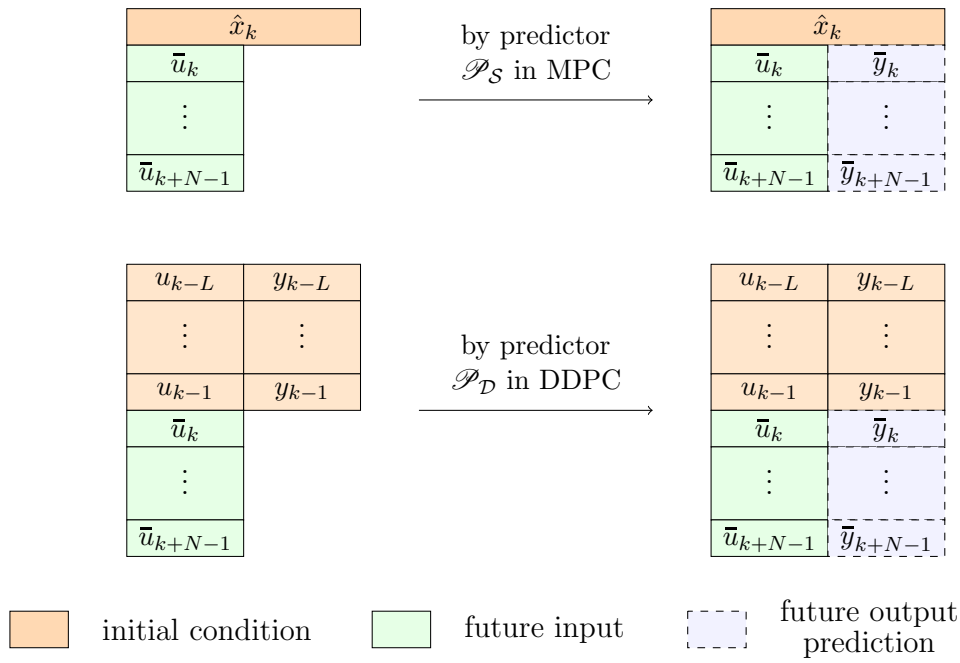

Figure 2.3: Output prediction in model-based and data-driven control.

General Formulation of DDPC

As discussed above, DDPC is a class of data-driven methods modified from MPC. Similar to MPC, DDPC is also a receding-horizon control strategy. That is, at each control step $t = k$, we solve for control inputs $\bar{u}_{[k,k+N)}$ over the prediction horizon $\mathbb{Z}_{[k,k+N)}$ by minimizing a cumulative cost over the horizon, and then apply the first N_c control actions $\bar{u}_{[k,k+N_\mathrm{c})}$ to the system on the *control horizon* $\mathbb{Z}_{[k,k+N_\mathrm{c})}$, with integer parameters $N, N_\mathrm{c} \in \mathbb{N}$ that $N_\mathrm{c} \leq N$. Moreover, in the data-driven scenario, the initial condition of the system is specified by the past input-output trajectory $u_{[k-L,k)}, y_{[k-L,k)}$ on a past interval $\mathbb{Z}_{[k-L,k)}$ called the *initial(-condition) horizon*, with parameter $L \in \mathbb{N}$. We call the union of the initial and prediction horizons the *total horizon* at control step $t = k$. Figure 2.4 summarizes the time horizons in DDPC.

The general form of DDPC optimization problem at control step $t = k$ can be written as follows,

$$\underset{\bar{u},\bar{y}}{\text{minimize}} \ (2.8) \qquad \text{subject to} \ (2.17) \text{ and } (2.7) \tag{DDPC}$$

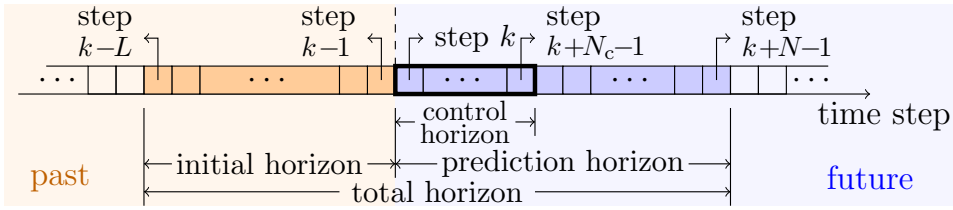

Figure 2.4: Time horizons in Data-Driven Predictive Control.

which is similar to (MPC) except that the output-prediction constraint (2.6) (equivalent to (2.16)) is replaced by a data-driven version (2.17). In (2.17), the output-prediction function $\mathscr{P}_{\mathcal{D}}(\cdot)$ is formulated based on a dataset $\mathcal{D}$ generated from the system, thus satisfying the data-driven requirement. The explicit expression of (2.17) depends on specific DDMPC algorithms, and we will see two examples in Section 2.4.2. The optimization process is repeated after we apply the solved input $u_{[k,k+N_c)}$ and then update the control step $t = k + N_c$. The entire DDPC control process is shown in Algorithm 2.

Algorithm 2 A General Framework of Data-Driven Predictive Control

Input: initial-, prediction- and control-horizon lengths L, N, N_c, stage-cost function $J_t(\cdot)$, and constraint sets $\mathcal{U}, \mathcal{Y}$, and offline dataset $\mathcal{D}$.

```
Record the initial trajectory u_{[-L,0)}, y_{[-L,0)} of the system.
Initialize the control step k ← 0.
while true do
    Solve control actions ū_{[k,k+N)} from problem (DDPC).
    for t from k to k + N_c − 1 do
        Apply input u_t ← ū_t to the system (2.1).
        Measure output from the system (2.1).
    Set k ← k + N_c.
```

DDPC preserves several features of MPC. First, DDPC (Algorithm 2) and MPC (Algorithm 1) are both receding-horizon control strategies. As such, both methods can handle reference tracking problems with time-varying reference signals. Second, DDPC frameworks can incorporate input and output constraints (2.7) as MPC does, making DDPC useful in applications of constrained control.

2.4.2 DDPC Methods: DeePC and SPC

In Section 2.4.1, we introduced a general formulation of DDPC, where the explicit form of the constraint (2.17) in problem (DDPC) varies with different DDPC methods. In this section, we exemplify two typical DDPC methods, namely Data-enabled Predictive Control (DeePC) [38, 39, 40, 61, 62] and Subspace Predictive Control (SPC) [17, 87, 15, 16].

Offline Data Collection

Both DeePC and SPC contain an offline process in addition to the (online) control process. In the offline process, we record from the system (2.1) an input-output trajectory $u^{\mathrm{d}}_{[1,T]}, y^{\mathrm{d}}_{[1,T]}$, which constitutes the dataset $\mathcal{D}$ in (2.17) and in Algorithm (DDPC). For data informativity, the input trajectory is required to be persistently exciting of order $L + N + \mathbf{n}(\mathscr{B})$. We formulate data matrices $U_{\mathrm{p}} \in \mathbb{R}^{mL\times h}$, $U_{\mathrm{f}} \in \mathbb{R}^{mN\times h}$, $Y_{\mathrm{p}} \in \mathbb{R}^{pL\times h}$ and $Y_{\mathrm{f}} \in \mathbb{R}^{pN\times h}$ as follows,

$$\begin{bmatrix} U_{\mathrm{p}} \\ U_{\mathrm{f}} \end{bmatrix} := \mathscr{H}_{L+N}(u^{\mathrm{d}}_{[1,T]}), \qquad \begin{bmatrix} Y_{\mathrm{p}} \\ Y_{\mathrm{f}} \end{bmatrix} := \mathscr{H}_{L+N}(y^{\mathrm{d}}_{[1,T]})$$

where the block-Hankel matrix $\mathscr{H}(\cdot)$ is defined in Definition 2.8, and $h := T - (L+N) + 1$ denotes the common width of $\mathscr{H}_{L+N}(u^{\mathrm{d}}_{[1,T]})$ and $\mathscr{H}_{L+N}(y^{\mathrm{d}}_{[1,T]})$. Using those data matrices, we can represent the restricted behavior $\mathscr{B}_{L+N}$ of the system,

$$\mathscr{B}_{L+N} = \mathrm{ColSpan} \begin{bmatrix} U_{\mathrm{p}} \\ U_{\mathrm{f}} \\ Y_{\mathrm{p}} \\ Y_{\mathrm{f}} \end{bmatrix}, \tag{2.18}$$

according to Lemma 2.9, under assumption that the system behavior $\mathscr{B}$ is controllable and the input data $u^{\mathrm{d}}_{[1,T]}$ is persistently exciting of order $L + N + \mathbf{n}(\mathscr{B})$.

The theoretical result in (2.18) holds once the input data is persistently exciting, which can be achieved with various types of signals. In practice, however, the data trajectory may encounter numerical issues, especially with unstable systems. To mitigate these issues, a stabilizing controller can be applied during the data generation process. Another approach to avoid numerical problems is to combine multiple short data trajectories and construct data matrices as outlined in [83].

Online Control Process

With offline data, we are able to formulate the online control algorithms. For both DeePC and SPC, the online control process follows the framework of Algorithm 2, only differing in the constraint (2.17) of problem (DDPC). The DeePC optimization problem at control step $t = k$ is what follows,

$$\underset{g,\overline{u},\overline{y}}{\text{minimize}} \ (2.8) \qquad \text{subject to} \ (2.19) \text{ and } (2.7) \tag{DeePC}$$

where $g \in \mathbb{R}^h$ is an auxiliary variable, and the output-prediction constraint (2.19) is

$$\begin{bmatrix} U_{\mathrm{p}} \\ U_{\mathrm{f}} \\ Y_{\mathrm{p}} \\ Y_{\mathrm{f}} \end{bmatrix} g = \begin{bmatrix} u_{[k-L,k)} \\ \overline{u}_{[k,k+N)} \\ y_{[k-L,k)} \\ \overline{y}_{[k,k+N)} \end{bmatrix}. \tag{2.19}$$

In (DeePC), g is the only independent variable, while $\overline{u}_{[k,k+N)}$ and $\overline{y}_{[k,k+N)}$ are both decided by g via the second and fourth block rows of (2.19). Note that (2.19) is obtained directly from (2.18) and the fact that (2.14) holds.

Remark 2.10 (**Uniqueness of Output Prediction in DeePC**). Although the output-prediction constraint (2.19) of DeePC does not follow the exact form in (2.17), it uniquely determines the future output $\overline{y}_{[k,k+N)}$ with given $\overline{u}_{[k,k+N)}$ and $\begin{bmatrix} u_{[k-L,k)} \\ y_{[k-L,k)} \end{bmatrix}$, in the case of $L \geq \mathbf{l}(\mathscr{B})$, according to Lemma 2.7. In this sense, (2.19) specifies $\overline{y}_{[k,k+N)}$ as an implicit function of $\overline{u}_{[k,k+N)}$ and $\begin{bmatrix} u_{[k-L,k)} \\ y_{[k-L,k)} \end{bmatrix}$, thereby equivalent to the form of (2.17).

The optimization problem of SPC at control step $t = k$ is the following,

$$\underset{\overline{u},\overline{y}}{\text{minimize}} \ (2.8) \qquad \text{subject to} \ (2.20) \text{ and } (2.7) \tag{SPC}$$

whose output-prediction constraint (2.20) is

$$\overline{y}_{[k,k+N)} = \underbrace{Y_{\mathrm{f}} \begin{bmatrix} U_{\mathrm{p}} \\ U_{\mathrm{f}} \\ Y_{\mathrm{p}} \end{bmatrix}^{\dagger}}_{=:\mathcal{P}_{\mathsf{spc}}} \begin{bmatrix} u_{[k-L,k)} \\ \overline{u}_{[k,k+N)} \\ y_{[k-L,k)} \end{bmatrix}. \tag{2.20}$$

The output-prediction constraints (2.19) of DeePC and (2.20) of SPC can be related by observing that

$$g = \begin{bmatrix} U_{\mathrm{p}} \\ U_{\mathrm{f}} \\ Y_{\mathrm{p}} \end{bmatrix}^{\dagger} \begin{bmatrix} u_{[k-L,k)} \\ \bar{u}_{[k,k+N)} \\ y_{[k-L,k)} \end{bmatrix}$$

is one solution for g to the first three block rows of (2.19) [50, 88].

Remark 2.11 (**Uniqueness of DeePC and SPC Solutions**). The optimal $\bar{u}_{[k,k+N)}$ and $\bar{y}_{[k,k+N)}$ are unique from both problems (DeePC) and (SPC), when the cost (2.8) is a strictly convex function of $\bar{u}_{[k,k+N)}$. This is because $\bar{y}_{[k,k+N)}$ is a function of $\bar{u}_{[k,k+N)}$ via each of the constraints (2.19) and (2.20) of DeePC and SPC, respectively; recall Remark 2.10. The optimal g from (DeePC) may not be unique, as the matrix equation (2.19) can be under-determined.

Both DeePC and SPC have their advantages over each other. Computationally, SPC is more efficient than DeePC, because of the extra variable g in DeePC. In application where DeePC and SPC are both regularized (see Section 2.6), regularized DeePC has more parameters of regularization than regularized SPC does [50] and thus is more flexible in algorithm design and parameter tuning.

There is no sufficient evidence showing which of DeePC and SPC outperforms the other. For deterministic LTI systems, both DeePC and SPC have equivalent performances with MPC, as to be shown in Section 2.5. In practical experiments where data is corrupted, regularized DeePC sometimes outperformed regularized SPC [41], while other times the former performed worse than the latter [50], and in some cases, the performances of both were comparable [43].

2.5 Equivalence of MPC and DDPC for Deterministic LTI Systems

In the following proposition, we show that for deterministic LTI systems, both DeePC and SPC yield control actions equivalent to those obtained from MPC.

Proposition 2.12 (Equivalence of DeePC, SPC and MPC [38, Cor. 5.1] [50, Thm. 1])**.** *Consider the LTI system $\mathcal{S}$ in (2.1). Let $\mathrm{col}(u^{\mathrm{d}}_{[1,T]}, y^{\mathrm{d}}_{[1,T]}) \in \mathscr{B}^{\mathcal{S}}_{T}$ be an offline data trajectory from the system. At control step $t = k$, for integers $L \geq \mathbf{l}(\mathscr{B}^{\mathcal{S}})$ and $N \in \mathbb{N}$, assume that*

(i) the behavior of $\mathcal{S}$ is controllable in the sense of Definition 2.6, and
(ii) $u^{\mathrm{d}}_{[1,T]}$ is persistently exciting of order $L + N + \mathbf{n}(\mathscr{B}^{\mathcal{S}})$.

Suppose the recent trajectory $u_{[k-L,k)}, y_{[k-L,k)}$ is known and the state is estimated exactly, i.e., $x_k = \hat{x}_k$. Then,

- *the unique optimal trajectory $\bar{u}_{[k,k+N)}, \bar{y}_{[k,k+N)}$ by (DeePC),*
- *the unique optimal trajectory $\bar{u}_{[k,k+N)}, \bar{y}_{[k,k+N)}$ by (SPC), and*
- *the unique optimal trajectory $\bar{u}_{[k,k+N)}, \bar{y}_{[k,k+N)}$ by (MPC)*

are all the same.

Proposition 2.12 demonstrates the performance equivalence of both DeePC and SPC to MPC. This performance guarantee indicates that DeePC and SPC are "perfect" data-driven substitutes of MPC in the deterministic LTI case. However, this conclusion is limited to deterministic LTI systems. For more general types of systems (e.g. stochastic, non-linear and time-varying), the theoretical support is still an open question. In Chapter 3 and Chapter 4, our research will contribute to extending these results beyond deterministic LTI systems.

2.6 Regularization of DeePC and SPC

DeePC and SPC are direct data-driven methods with equivalent performances with MPC for deterministic LTI systems, as shown in Proposition 2.12. However, both algorithms are sensitive to perturbation of data. In particular, the constraint (2.19) may be over-determined when offline data and online measurements are perturbed, and consequently the problem (DeePC) may become infeasible. The pseudo-inverse operation in (2.20) is also sensitive to small perturbations, making the solution to problem (SPC) highly sensitive to small variations in the data.

Since real-world control systems are typically stochastic and nonlinear, measured data often contain noise and linearization errors. As a result, both problems (DeePC) and (SPC) problems must be modified or robustified to account for perturbed data. In the literature, conventional robustification techniques include regularization of optimization problems and low-rank approximation of data matrices (see e.g. [38]). Here we review regularization techniques of DeePC and SPC.

Regularization of DeePC

There exist several versions of regularized DeePC [38, 39, 61, 62, 50]. Take [61] as an example, where the regularized DeePC optimization problem at control step $t = k$ is

$$\begin{aligned} \underset{\overline{u},\overline{y},g,\sigma_{\mathrm{y}}}{\text{minimize}} \quad & (2.8) + \lambda_{\mathrm{g}}\|g\|_2^2 + \lambda_{\mathrm{y}}\|\sigma_{\mathrm{y}}\|_2^2 \\ \text{subject to} \quad & (2.21) \text{ and } (2.7) \end{aligned} \tag{reg. DeePC}$$

wherein $\sigma_{\mathrm{y}} \in \mathbb{R}^{pL}$ is a slack variable, $\lambda_{\mathrm{g}} > 0$ and $\lambda_{\mathrm{y}} > 0$ are user-selected parameters, and the constraint (2.21) is a modification from (2.19) as

$$\begin{bmatrix} U_{\mathrm{p}} \\ U_{\mathrm{f}} \\ Y_{\mathrm{p}} \\ Y_{\mathrm{f}} \end{bmatrix} g = \begin{bmatrix} u_{[k-L,k)} \\ \overline{u}_{[k,k+N)} \\ y_{[k-L,k)} \\ \overline{y}_{[k,k+N)} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \sigma_{\mathrm{y}} \\ 0 \end{bmatrix}. \tag{2.21}$$

With the introduction of slack variable σ_{y}, we relax the hard constraint $Y_{\mathrm{p}}g = y_{[k-L,k)}$ in (2.19) into a soft constraint $Y_{\mathrm{p}}g = y_{[k-L,k)} + \sigma_{\mathrm{y}}$ as in (2.21), so as to improve robustness to noise-corrupted output data in both Y_{p} and $y_{[k-L,k)}$. The objective function in (reg. DeePC) is now the cost (2.8) augmented by penalization of the magnitudes of g and σ_{y}. Although the penalizing terms in (reg. DeePC) are illustrated based on L2-norms, they can also alternatively be L1-norms [38, 39] or other convex functions [62]. Other regularized DeePC formulations may also relax the constraint $U_{\mathrm{p}}g = u_{[k-L,k)}$ in (2.21) into a soft constraint $U_{\mathrm{p}}g = u_{[k-L,k)} + \sigma_{\mathrm{u}}$ with slack variable $\sigma_{\mathrm{u}} \in \mathbb{R}^{mL}$ and similarly penalize the magnitude of σ_{u} in the objective function [50].

Regularization of SPC

There are several ways to regularize the formulation of SPC. Typically, the computation of matrix $\mathcal{P}_{\mathsf{spc}}$ in (2.20) is robustified with some numerical methods. For example, in the pseudo-inverse operation in (2.20), one can treat as zero the singular values smaller than a chosen threshold σ_{thre} [41, 42]. Another way of robust the computation of $\mathcal{P}_{\mathsf{spc}}$ is to utilize Tikhonov regularization, also known as ridge regression [43], where the pseudo-inverse $Z^{\dagger}$ of $Z := \mathrm{col}(U_{\mathrm{p}}, U_{\mathrm{f}}, Y_{\mathrm{p}})^{\dagger}$ in (2.20) is replaced by $(Z^{\mathsf{T}}Z + \lambda I)^{-1}Z^{\mathsf{T}}$, with a regularization parameter $\lambda > 0$.

Chapter 3

Data-Driven Predictive Control for Linear Time-Periodic Systems

In this chapter, we focus linear time-periodic (LTP) system, as a specific type of time-varying systems. LTP systems can arise from linearization of nonlinear systems around periodic trajectories, such as in models of helicopters [89] and wind turbines [90] etc. The work in this chapter refers to our work [77]. Leveraging the standard lifting technique of LTP systems, we develop the behavioral systems theory for LTP systems, including the LTP extension of Willems' fundamental lemma. We also develop the LTP versions of the DeePC and SPC methods, namely Periodic DeePC (P-DeePC) and Periodic SPC (P-SPC), and prove that the P-DeePC and P-SPC methods produce equivalent control actions as those from MPC for deterministic LTP systems. The proposed control methods are validated through simulations.

3.1 Problem Statement: Deterministic LTP Case

Linear Time-Periodic System

In this section, we state a data-driven control problem for linear time-periodic systems. Consider a discrete-time linear time-varying (LTV) system

$$\mathcal{S} : \begin{cases} x_{t+1} = A_t x_t + B_t u_t \\ y_t = C_t x_t + D_t u_t \end{cases} \tag{3.1}$$

with initial time $t_0 \in \mathbb{Z}$, where $x_t \in \mathbb{R}^n$, $u_t \in \mathbb{R}^m$, and $y_t \in \mathbb{R}^p$ are the state, input, and output of the system. The system (3.1) is said to be *linear time-periodic (LTP)* if there exists $T \in \mathbb{N}$ (a *period*) such that

$$A_{t+T} = A_t, \quad B_{t+T} = B_t, \quad C_{t+T} = C_t, \quad D_{t+T} = D_t, \tag{3.2}$$

for all $t \in \mathbb{Z}$. The smallest T satisfying this condition is the *fundamental period*; without loss of generality, we assume going forward that T is the fundamental period. Note that when $T = 1$, the system (3.1) is linear time-invariant (LTI). A discrete-time LTP model may arise naturally in discrete time, or may have been obtained via appropriate sampling of a continuous-time LTP system.

Control Problem

Consider an LTP system (3.1) satisfying the periodicity condition (3.2) with fundamental period $T \in \mathbb{N}$. Similarly as Section 2.1, in the data-driven scenario, the system matrices A_t, B_t, C_t, D_t in (3.1) are *unknown*; we have access only to the input u_t and output y_t in (3.1). In a reference tracking problem, the objective is for the output y_t to follow a specified reference signal $r_t \in \mathbb{R}^p$. The trade-off between the tracking error $y_t - r_t$ and the control effort u_t may be encoded in the instantaneous cost (2.2), reproduced as

$$J_t(u_t, y_t) := \|y_t - r_t\|_Q^2 + \|u_t\|_R^2, \tag{2.2}$$

whose accumulation over a horizon is to be minimized, where $Q \in \mathbb{S}_+^p$ and $R \in \mathbb{S}_{++}^m$ are user-selected parameters. This tracking should be achieved subject to constraints on the inputs and outputs in the form of (2.3), reproduced as

$$u_t \in \mathcal{U}, \qquad y_t \in \mathcal{Y}, \tag{2.3}$$

for time $t \in \mathbb{N}_{\geq 0}$, where the constraint sets $\mathcal{U} \subseteq \mathbb{R}^m$ and $\mathcal{Y} \subseteq \mathbb{R}^p$ are assumed to be convex, non-empty and closed.

Notations based on System Model

For the LTV system (3.1) and integers t_1, t_2 with $t_1 \leq t_2$, the state-transition matrix $\Phi_{t_1}^{t_2} \in \mathbb{R}^{n\times n}$ and impulse response matrix $G_{t_1}^{t_2} \in \mathbb{R}^{p\times m}$ from step t_1 to t_2 are defined as

$$\Phi_{t_1}^{t_2} := \begin{cases} I, & \text{if } t_2 = t_1 \\ A_{t_2-1}A_{t_2-2}\cdots A_{t_1}, & \text{if } t_2 > t_1, \end{cases} \tag{3.3a}$$

$$G_{t_1}^{t_2} := \begin{cases} D_{t_1}, & \text{if } t_2 = t_1 \\ C_{t_2}\Phi_{t_1+1}^{t_2}B_{t_1}, & \text{if } t_2 > t_1. \end{cases} \tag{3.3b}$$

Similarly, the associated (reversed) extended controllability matrix $\mathcal{C}_{t_1}^{t_2} \in \mathbb{R}^{n\times(t_2-t_1+1)m}$, the extended observability matrix $\mathcal{O}_{t_1}^{t_2} \in \mathbb{R}^{(t_2-t_1+1)p\times n}$, and the block matrix $\mathcal{G}_{t_1}^{t_2}$ of impulse-response coefficients are defined as

$$\mathcal{C}_{t_1}^{t_2} := \left[\Phi_{t_1+1}^{t_2+1}B_{t_1},\ \Phi_{t_1+2}^{t_2+1}B_{t_1+1},\ \ldots,\ \Phi_{t_2+1}^{t_2+1}B_{t_2}\right], \tag{3.4a}$$

$$\mathcal{O}_{t_1}^{t_2} := \operatorname{col}\left(C_{t_1}\Phi_{t_1}^{t_1},\ C_{t_1+1}\Phi_{t_1}^{t_1+1},\ \ldots,\ C_{t_2}\Phi_{t_1}^{t_2}\right), \tag{3.4b}$$

$$\mathcal{G}_{t_1}^{t_2} := \begin{bmatrix} G_{t_1}^{t_1} & & & \\ G_{t_1}^{t_1+1} & G_{t_1+1}^{t_1+1} & & \\ \vdots & \vdots & \ddots & \\ G_{t_1}^{t_2} & G_{t_1+1}^{t_2} & \cdots & G_{t_2}^{t_2} \end{bmatrix}. \tag{3.4c}$$

With this notation, the unique solution of (3.1) with initial condition x_{t_1} at time $t = t_1$ can be expressed as

$$x_{t_2} = \Phi_{t_1}^{t_2}x_{t_1} + \mathcal{C}_{t_1}^{t_2-1}u_{[t_1,t_2)}, \tag{3.5a}$$

$$y_{[t_1,t_2)} = \mathcal{O}_{t_1}^{t_2-1}x_{t_1} + \mathcal{G}_{t_1}^{t_2-1}u_{[t_1,t_2)}, \tag{3.5b}$$

for any future time $t_2 > t_1$. The notations in (3.4) extend the notations in (2.4), and the result in (3.5) generalizes (2.5), from LTI systems to LTV systems.

3.2 Lifting an LTP System to an LTI System

We now recall a classical technique for "lifting" an LTP system into an LTI system [91].

Definition 3.1 (Lift of an LTP System)**.** For an LTP system $\mathcal{S}$ as in (3.1) of period T and an initial time $t_0 \in \mathbb{Z}$, the associated *lifted system* $\mathcal{S}_{\mathsf{L}}(t_0)$ *of* $\mathcal{S}$ *with initial time* t_0 is the LTI system

$$\mathcal{S}_{\mathsf{L}}(t_0) : \begin{cases} \mathbf{x}_{\tau+1} = \mathfrak{A}\mathbf{x}_\tau + \mathfrak{B}\mathbf{u}_\tau \\ \mathbf{y}_\tau = \mathfrak{C}\mathbf{x}_\tau + \mathfrak{D}\mathbf{u}_\tau \end{cases} \tag{3.6}$$

with state $\mathbf{x}_\tau \in \mathbb{R}^n$, input $\mathbf{u}_\tau \in \mathbb{R}^{mT}$, output $\mathbf{y}_\tau \in \mathbb{R}^{pT}$, and time $\tau \in \mathbb{Z}$, where

$$\mathfrak{A} := \Phi_{t_0}^{t_0+T}, \qquad \mathfrak{B} := \mathcal{C}_{t_0}^{t_0+T-1}, \qquad \mathfrak{C} := \mathcal{O}_{t_0}^{t_0+T-1}, \qquad \mathfrak{D} := \mathcal{G}_{t_0}^{t_0+T-1}. \tag{3.7}$$

The idea behind lifting is that each time step τ of the lifted system $\mathcal{S}_{\mathsf{L}}(t_0)$ corresponds to T successive time steps of the original LTP system $\mathcal{S}$. The state/input/output of $\mathcal{S}_{\mathsf{L}}(t_0)$ are related to the state/input/output of $\mathcal{S}$ via

$$\mathbf{x}_\tau = x_{t_0+\tau T}, \qquad \mathbf{u}_\tau = u_{[t_0+\tau T,\, t_0+(\tau+1)T)}, \qquad \mathbf{y}_\tau = y_{[t_0+\tau T,\, t_0+(\tau+1)T)}.$$

Each input vector $\mathbf{u}_\tau$ (or output vector $\mathbf{y}_\tau$) of $\mathcal{S}_{\mathsf{L}}(t_0)$ stacks the inputs (or outputs) of $\mathcal{S}$ over one period, and the state vector $\mathbf{x}_\tau$ is the state of $\mathcal{S}$ at the "beginning" of this period, as specified by the initial time t_0; see Fig. 3.1. Note from (3.7) that the matrices $\mathfrak{A}, \mathfrak{B}, \mathfrak{C}, \mathfrak{D}$ depend on the initial time t_0. Nonetheless, some properties of the lifted system — such as the eigenvalues of $\mathfrak{A}$ — are invariant under the choice of the initial time t_0. See [92] for more information on lifting and properties of the lifted system.

3.3 Behavioral Systems Theory for LTP Systems

We have reviewed the behavioral systems theory for LTI systems in Section 2.3. In this section, we generalize some results on behavioral systems theory for LTV systems or LTP systems.

3.3.1 Behavioral Representation of LTV Systems

In the framework of behavioral systems theory, the input-output trajectories of the system (3.1) are described independent of the state representation through the *behavior*. For a linear time-varying system $\mathcal{S}$ in (3.1), its dynamics is generally different at different time steps, so we define its behavior based on specific initial time.

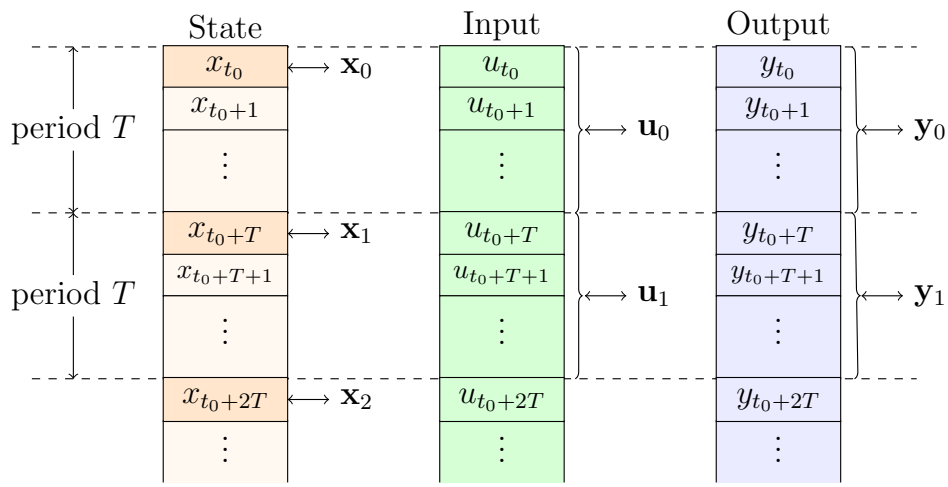

Figure 3.1: The states, inputs and outputs of an LTP system and its lifted system.

Definition 3.2 (Behavior of LTV Systems)**.** For the finite-dimensional LTV system $\mathcal{S}$ in (3.1) and $t_0 \in \mathbb{Z}$, the *behavior* $\mathscr{B}^{\mathcal{S}}_{[t_1,\infty)}$ of $\mathcal{S}$ on time interval $\mathbb{Z}_{[t_1,\infty)}$ is the set

$$\mathscr{B}^{\mathcal{S}}_{[t_1,\infty)} := \left\{ \begin{bmatrix} u_{[t_1,\infty)} \\ y_{[t_1,\infty)} \end{bmatrix} \;\middle|\; \exists\, x_{t_1} \text{ s.t. (3.1) holds for all } t \in \mathbb{Z}_{[t_1,\infty)} \right\}.$$

For integer $t_2 \geq t_1$, let $\mathscr{B}^{\mathcal{S}}_{[t_1,t_2]}$ denote the restriction of the behavior $\mathscr{B}^{\mathcal{S}}_{[t_1,\infty)}$ on time interval $\mathbb{Z}_{[t_1,t_2]}$, that is,

$$\mathscr{B}^{\mathcal{S}}_{[t_1,t_2]} := \left\{ \begin{bmatrix} u_{[t_1,t_2]} \\ y_{[t_1,t_2]} \end{bmatrix} \;\middle|\; \exists\, x_{t_1} \text{ s.t. (3.1) holds for all } t \in \mathbb{Z}_{[t_1,t_2]} \right\}.$$

In the case $t_2 > t_1$, we let $\mathscr{B}^{\mathcal{S}}_{[t_1,t_2)} := \mathscr{B}^{\mathcal{S}}_{[t_1,t_2-1]}$. We write a behavior (resp. a restricted behavior) as $\mathscr{B}_{[t_1,\infty)}$ (resp. $\mathscr{B}_{[t_1,t_2]}$ or $\mathscr{B}_{[t_1,t_2)}$) when the underlying system model is not emphasized. We say that $\mathscr{B}$ is an LTI/LTV/LTP behavior if it is equal to the behavior $\mathscr{B}^{\mathcal{S}}$ of some LTI/LTV/LTP system $\mathcal{S}$ by Definition 3.2.

The behavior defines a subspace of the vector space of semi-infinite sequences, and contains all possible input-output trajectories of the system. Going forward, we focus primarily on the restricted behavior. Similar to Lemma 2.2 and Corollary 2.2.1, the following results state that the restricted behavior is a finite-dimensional vector space.

Lemma 3.3. *The restricted behavior $\mathscr{B}^{\mathcal{S}}_{[t_1,t_2]}$ of the LTV system $\mathcal{S}$ in (3.1) is a finite-dimensional vector space and*

$$\mathscr{B}^{\mathcal{S}}_{[t_1,t_2]} = \operatorname{ColSpan}\begin{bmatrix} 0 & I \\ \mathcal{O}^{t_2}_{t_1} & \mathcal{G}^{t_2}_{t_1} \end{bmatrix}.$$

Proof. The definition of $\mathscr{B}^{\mathcal{S}}_{[t_1,t_2]}$ can be rewritten as

$$\mathscr{B}^{\mathcal{S}}_{[t_1,t_2]} = \left\{ \begin{bmatrix} u_{[t_1,t_2]} \\ y_{[t_1,t_2]} \end{bmatrix} \;\middle|\; \exists x_{t_1} \text{ s.t. (3.5b) holds} \right\}.$$

The result now follows immediately by eliminating $y_{[t_1,t_2]}$ above using (3.5b). ■

Corollary 3.3.1. $\dim \mathscr{B}^{\mathcal{S}}_{[t_1,t_2]} = \operatorname{rank}(\mathscr{O}^{t_2}_{t_1}) + m(t_2 - t_1 + 1)$.

Proof. The result follows from Lemma 3.3, with $\begin{bmatrix} 0 & I \\ \mathcal{O}^{t_2}_{t_1} & \mathcal{G}^{t_2}_{t_1} \end{bmatrix}$ block-triangular. ■

When $\mathcal{S}$ is an LTI system, the behavior is invariant under shift of the time interval, meaning that

$$\mathscr{B}^{\mathcal{S}}_{[t_1,t_2]} = \mathscr{B}^{\mathcal{S}}_{[t_1+s,t_2+s]} \quad \text{for all } s \in \mathbb{Z}.$$

This follows from Definition 2.1, due to shift-invariance of the system matrices. A similar result holds when $\mathcal{S}$ is an LTP system of period T, namely that

$$\mathscr{B}^{\mathcal{S}}_{[t_1,t_2]} = \mathscr{B}^{\mathcal{S}}_{[t_1+sT,t_2+sT]} \quad \text{for all } s \in \mathbb{Z}.$$

Unlike the case of LTI or LTP systems, the behavior of a general LTV system is not a shift-invariant subspace. However, given the behavior over an interval, the behavior on the first several steps can be easily constructed.

Lemma 3.4. *For the LTV system $\mathcal{S}$ in (3.1) and integers $t_0 \le t_1 \le t_2$, if*

$$\mathscr{B}^{\mathcal{S}}_{[t_0,t_2]} = \operatorname{ColSpan}\big(\operatorname{col}(U_{t_0},\dots,U_{t_2},Y_{t_0},\dots,Y_{t_2})\big)$$

for some matrices $U_{t_0},\dots,U_{t_2} \in \mathbb{R}^{m\times h}$ and $Y_{t_0},\dots,Y_{t_2} \in \mathbb{R}^{p\times h}$ with some $h \in \mathbb{N}$, then

$$\mathscr{B}^{\mathcal{S}}_{[t_0,t_1]} = \operatorname{ColSpan}\big(\operatorname{col}(U_{t_0},\dots,U_{t_1},Y_{t_0},\dots,Y_{t_1})\big).$$

Proof. See Section 3.7.1. ■

Different state-space models may correspond to the same behavior. The following result characterizes when different LTV systems have the same restricted behavior, being an extension of Lemma 2.4.

Lemma 3.5 (Different State-Space Representations of Same LTV Behavior)**.** *For LTV systems $\mathcal{S}_{\mathrm{I}}, \mathcal{S}_{\mathrm{II}}$ and integers $t_1 \leq t_2$, we have $\mathscr{B}^{\mathcal{S}_{\mathrm{I}}}_{[t_1,t_2]} = \mathscr{B}^{\mathcal{S}_{\mathrm{II}}}_{[t_1,t_2]}$ if, and only if,*

$$\mathrm{ColSpan}\big([\mathcal{O}_{\mathrm{I}}]_{t_1}^{t_2}\big) = \mathrm{ColSpan}\big([\mathcal{O}_{\mathrm{II}}]_{t_1}^{t_2}\big) \supseteq \mathrm{ColSpan}\big([\mathcal{G}_{\mathrm{I}}]_{t_1}^{t_2} - [\mathcal{G}_{\mathrm{II}}]_{t_1}^{t_2}\big)$$

where matrices $[\mathcal{O}_{\mathrm{I}}]_{t_1}^{t_2}$ and $[\mathcal{G}_{\mathrm{I}}]_{t_1}^{t_2}$ (resp. matrices $[\mathcal{O}_{\mathrm{II}}]_{t_1}^{t_2}$ and $[\mathcal{G}_{\mathrm{II}}]_{t_1}^{t_2}$) are defined as in (3.4b) and (3.4c) for system $\mathcal{S}_{\mathrm{I}} : (A_{\mathrm{I}}, B_{\mathrm{I}}, C_{\mathrm{I}}, D_{\mathrm{I}})$ (resp. system $\mathcal{S}_{\mathrm{II}} : (A_{\mathrm{II}}, B_{\mathrm{II}}, C_{\mathrm{II}}, D_{\mathrm{II}})$).

Proof. See Section 3.7.2. ■

Controllability of LTV Behaviors

In the behavioral framework, controllability is defined in a trajectory-based sense, as opposed to the more classical notion of state-controllability. The controllability of LTV behaviors is defined as follows, similar to Definition 2.6.

Definition 3.6 (Controllability of LTV Behaviors [23, 93])**.** An LTV behavior $\mathscr{B}$ is *controllable* if for any $t_0 \in \mathbb{Z}$, any two trajectories $w^{\mathrm{I}}_{[t_0,\infty)}$, $w^{\mathrm{II}}_{[t_0,\infty)} \in \mathscr{B}_{[t_0,\infty)}$, and any time $t_1 \geq t_0$, there exists a time $t_2 \geq t_1$ and a trajectory $w^{\diamond}_{[t_0,\infty)} \in \mathscr{B}_{[t_0,\infty)}$ such that

$$w^{\diamond}_{[t_0,t_1)} = w^{\mathrm{I}}_{[t_0,t_1)}, \qquad w^{\diamond}_{[t_2,\infty)} = w^{\mathrm{II}}_{[t_2,\infty)}, \tag{3.8}$$

where we let $w_{[s_1,s_2]} := \mathrm{col}(u_{[s_1,s_2]}, y_{[s_1,s_2]})$ for integers $s_1 \leq s_2$ denote a trajectory.

Put differently, an LTV behavior is controllable if we can "drive" from one trajectory to any other trajectory in a finite number of time steps; see Figure 3.2 for visualization. Recall that, when $\mathscr{B}$ is an LTI behavior, the controllability of $\mathscr{B}$ has been defined in Definition 2.6, which is similar to Definition 3.6 with a slight difference that the second equality in (3.8) is replaced by $w^{\diamond}_{[t_2,\infty)} = w^{\mathrm{II}}_{[t_0,\infty)}$ as in (2.9). Nonetheless, those two definitions of controllability are equivalent for LTI behaviors [93, Remark 4(i)] [23, Sec. 2.4].

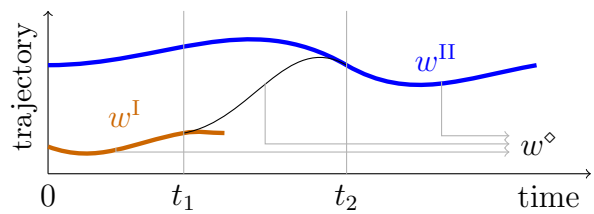

Figure 3.2: Controllability of LTV behavior.

3.3.2 A Definition of Order and Lag for LTV Systems

In behavioral systems theory, the *order* and *lag* are so-called integer invariants of an LTI system, and can be expressed using a minimal state representation of the behavior, as introduced in Section 2.3.4. In this subsection, we generalize those notions to LTV systems, and do so in a manner that avoids introducing notions of minimality for LTV models, which is considerably more complicated than LTI minimality.

First, we briefly recall the LTI definitions of order and lag [80, 23] reviewed in Section 2.3.4. For an LTI state-space model $\mathcal{S} : (A, B, C, D)$ in (2.1), the order and lag of its behavior $\mathscr{B}^{\mathcal{S}}$ can be represented as (2.13) based on the system model $\mathcal{S}$. Motivated by the this result with LTI systems, we define order and lag for LTV system as follows.

Definition 3.7 (Order and Lag for LTV State-Space Models)**.** For the LTV system $\mathcal{S} : (A_t, B_t, C_t, D_t)$ in (3.1), the *order* $\mathbf{n}(\mathcal{S}, t)$ at time t and *lag* $\mathbf{l}(\mathcal{S}, t)$ at time t are

$$\mathbf{n}(\mathcal{S}, t) := \lim_{s\to\infty} \operatorname{rank}(\mathcal{O}_t^{t+s-1}), \tag{3.9a}$$

$$\mathbf{l}(\mathcal{S}, t) := \min\{s \in \mathbb{N} : \operatorname{rank}(\mathcal{O}_t^{t+s-1}) = \mathbf{n}(\mathcal{S}, t)\}, \tag{3.9b}$$

with $\mathcal{O}_{t_1}^{t_2}$ defined in (3.4b).

Remark 3.8 (**Well-definedness**)*.* Both $\mathbf{n}(\mathcal{S}, t)$ and $\mathbf{l}(\mathcal{S}, t)$ are well-defined through (3.9). $\mathbf{n}(\mathcal{S}, t)$ is well-defined in (3.9a) since the sequence $\operatorname{rank}(\mathcal{O}_t^{t+s-1})$ of index s is bounded by the state dimension n and is non-decreasing as we increase s as $\mathcal{O}_t^{t+s-1}$ expands with extra rows. With $\mathbf{n}(\mathcal{S}, t)$ well-defined, $\mathbf{l}(\mathcal{S}, t)$ is also well-defined by (3.9b), for a finite-dimensional system $\mathcal{S}$.

When $\mathcal{S}$ is an LTI system, we can write its order and lag in the sense of (3.7) as $\mathbf{n}(\mathcal{S})$ and $\mathbf{l}(\mathcal{S})$ respectively, since they are time-independent. The definitions in (3.7) are consistent with (2.13) in the LTI case. Moreover, via Corollary 3.3.1 and (3.9a), we have

$$\dim \mathscr{B}^{\mathcal{S}}_{[t,t+L)} = \mathbf{n}(\mathcal{S}, t) + mL \tag{3.10}$$

for all integers $t \in \mathbb{Z}$ and $L \geq \mathbf{l}(\mathcal{S}, t)$, which coincides with an established result [80, Cor. 5] [23, Eq. 1] in the LTI case.

Recall from (2.12) in Section 2.3.4 that both order $\mathbf{n}(\mathscr{B})$ and lag $\mathbf{l}(\mathscr{B})$ are properties of behavior $\mathscr{B}$ in the LTI case, while in Definition 3.7 the order $\mathbf{n}(\mathcal{S}, t)$ and lag $\mathbf{l}(\mathcal{S}, t)$ are defined as properties based on the system model $\mathcal{S}$, which appears inconsistent with the original behavioral theory discussed in Section 2.3.4. In fact, the order and lag defined in Definition 3.7 are invariant over different state-space representations of a behavior, as described in the lemma below.

Lemma 3.9. *For LTV state-space models $\mathcal{S}_{\mathrm{I}}, \mathcal{S}_{\mathrm{II}}$ and integer $t \in \mathbb{Z}$, if $\mathscr{B}^{\mathcal{S}_{\mathrm{I}}}_{[t,\infty)} = \mathscr{B}^{\mathcal{S}_{\mathrm{II}}}_{[t,\infty)}$, then we have $\mathbf{n}(\mathcal{S}_{\mathrm{I}}, t) = \mathbf{n}(\mathcal{S}_{\mathrm{II}}, t)$ and $\mathbf{l}(\mathcal{S}_{\mathrm{I}}, t) = \mathbf{l}(\mathcal{S}_{\mathrm{II}}, t)$, with $\mathbf{n}(\cdot)$ and $\mathbf{l}(\cdot)$ in Definition 3.7.*

Proof. Since $\mathscr{B}^{\mathcal{S}_{\mathrm{I}}}_{[t,\infty)} = \mathscr{B}^{\mathcal{S}_{\mathrm{II}}}_{[t,\infty)}$, the restricted behaviors $\mathscr{B}^{\mathcal{S}_{\mathrm{I}}}_{[t,s)} = \mathscr{B}^{\mathcal{S}_{\mathrm{II}}}_{[t,s)}$ are also equal for all $s \in \mathbb{N}$. It follows from Corollary 3.3.1 that

$$\operatorname{rank}\left([\mathcal{O}_{\mathrm{I}}]_t^{t+s-1}\right) = \operatorname{rank}\left([\mathcal{O}_{\mathrm{II}}]_t^{t+s-1}\right),$$

where matrices $[\mathcal{O}_{\mathrm{I}}]_t^{t+s-1}$ and $[\mathcal{O}_{\mathrm{II}}]_t^{t+s-1}$ are defined as in (3.4b) for $\mathcal{S}_{\mathrm{I}} : (A_{\mathrm{I}}, B_{\mathrm{I}}, C_{\mathrm{I}}, D_{\mathrm{I}})$ and $\mathcal{S}_{\mathrm{II}} : (A_{\mathrm{II}}, B_{\mathrm{II}}, C_{\mathrm{II}}, D_{\mathrm{II}})$, respectively. With the relation above, the order and lag in the sense of Definition 3.7 are the same for systems $\mathcal{S}_{\mathrm{I}}$ and $\mathcal{S}_{\mathrm{II}}$. ■

Given Lemma 3.9, we can modify Definition 3.7 and alternatively define the order and lag as Definition 3.10, so that order and lag are properties of behavior.

Definition 3.10 (Order and Lag for LTV Behaviors)**.** For an LTV behavior $\mathscr{B}$, its *order* $\mathbf{n}(\mathscr{B}, t)$ at time t and *lag* $\mathbf{l}(\mathscr{B}, t)$ at time t are

$$\begin{aligned}
\mathbf{n}(\mathscr{B}, t) &:= \lim_{s\to\infty} \operatorname{rank}(\mathcal{O}_t^{t+s-1}),\\
\mathbf{l}(\mathscr{B}, t) &:= \min\left\{s \in \mathbb{N} : \operatorname{rank}(\mathcal{O}_t^{t+s-1}) = \mathbf{n}(\mathscr{B}, t)\right\},
\end{aligned}$$

with $\mathcal{O}_{t_1}^{t_2}$ defined in (3.4b), where $\mathcal{S} : (A_t, B_t, C_t, D_t)$ is an arbitrary state-space representation of $\mathscr{B}$.

Given Lemma 3.9, $\mathbf{n}(\mathscr{B}, t)$ and $\mathbf{l}(\mathscr{B}, t)$ in Definition 3.10 are invariant over different state-space representations $\mathcal{S}$ of the behavior $\mathscr{B}$, and thus are well-defined. Definition 3.10 is a genuine extension of the concepts of order and lag from LTI systems to LTV systems, while Definition 3.7 is an equivalent definition, with a slight abuse of notation, based on a specific state-space representation. For simplicity, going forward, we mainly apply the order $\mathbf{n}(\mathcal{S}, t)$ and lag $\mathbf{l}(\mathcal{S}, t)$ specified in Definition 3.7.

Specification of Initial Condition

As in Section 2.3.5, the lag specifies a sufficient length of a trajectory such that, with any subsequent input, the resulting output after the trajectory is uniquely determined, as captured in Lemma 3.11(ii). The following result generalizes Lemma 2.7 from the LTI case to the LTV case. Lemma 3.11(iii) gives an expression for the unique output, generalizing [50, Lemma 2] as the LTI case.

Lemma 3.11 (Initial-Condition Specification)**.** *Consider LTV system $\mathcal{S}$ in (3.1), a time step $t \in \mathbb{Z}$, and integers $L, N \in \mathbb{N}$. The following statements hold:*

(i) For any past trajectory $\mathrm{col}(u_{[t-L,t)}, y_{[t-L,t)}) \in \mathscr{B}^{\mathcal{S}}_{[t-L,t)}$ and any future input $\bar{u}_{[t,t+N)}$, there exists a future output $\bar{y}_{[t,t+N)}$ satisfying

$$\mathrm{col}(u_{[t-L,t)}, \bar{u}_{[t,t+N)}, y_{[t-L,t)}, \bar{y}_{[t,t+N)}) \in \mathscr{B}^{\mathcal{S}}_{[t-L,t+N)}. \tag{3.11}$$

(ii) If $L \geq \mathbf{l}(\mathcal{S}, t-L)$, then the output $\bar{y}_{[t,t+N)}$ from (i) is unique.

(iii) Moreover, if the restricted behavior $\mathscr{B}^{\mathcal{S}}_{[t-L,t+N)}$ can be expressed as

$$\mathscr{B}^{\mathcal{S}}_{[t-L,t+N)} = \mathrm{ColSpan}\big(\mathrm{col}(U_\mathrm{p}, U_\mathrm{f}, Y_\mathrm{p}, Y_\mathrm{f})\big) \tag{3.12}$$

for some matrices $U_\mathrm{p} \in \mathbb{R}^{mL\times h}, U_\mathrm{f} \in \mathbb{R}^{mN\times h}, Y_\mathrm{p} \in \mathbb{R}^{pL\times h}, Y_\mathrm{f} \in \mathbb{R}^{pN\times h}$ with some $h \in \mathbb{N}$, then the unique output $\bar{y}_{[t,t+N)}$ from (ii) is given as

$$\bar{y}_{[t,t+N)} = Y_\mathrm{f} \begin{bmatrix} U_\mathrm{p} \\ U_\mathrm{f} \\ Y_\mathrm{p} \end{bmatrix}^{\dagger} \begin{bmatrix} u_{[t-L,t)} \\ \bar{u}_{[t,t+N)} \\ y_{[t-L,t)} \end{bmatrix}. \tag{3.13}$$

Proof. See Section 3.7.3. ■

3.3.3 Behavioral Systems Theory for LTP Systems

Now we limit our discussion to LTP systems. We first establish the relationship between the behavior of an LTP system and the behavior of any corresponding lifted system.

Lemma 3.12. *For an LTP system $\mathcal{S}$ of period T and its lifted system $\mathcal{S}_\mathsf{L}(t_0)$ with initial step $t_0 \in \mathbb{Z}$, it holds that*

$$\mathscr{B}^{\mathcal{S}}_{[t_0,\, t_0+sT)} = \mathscr{B}^{\mathcal{S}_\mathsf{L}(t_0)}_{[0,s)} \qquad \forall s \in \mathbb{N}.$$

Proof. Let (3.1) be the model of $\mathcal{S}$, and (3.6) be the model of $\mathcal{S}_{\mathsf{L}}(t_0)$. Define matrices $\mathcal{O}_{t_1}^{t_2}$ and $\mathcal{G}_{t_1}^{t_2}$ (resp. $\widehat{\mathcal{O}}_{t_1}^{t_2}$ and $\widehat{\mathcal{G}}_{t_1}^{t_2}$) via (3.4b) and (3.4c) for system $\mathcal{S}$ (resp. $\mathcal{S}_{\mathsf{L}}(t_0)$). By Lemma 3.3, the respective behaviors can be expressed as

$$\begin{aligned} \mathscr{B}_{[t_0,\, t_0+sT)}^{\mathcal{S}} &= \operatorname{ColSpan}\begin{bmatrix} 0 & I \\ \mathcal{O}_{t_0}^{t_0+sT-1} & \mathcal{G}_{t_0}^{t_0+sT-1} \end{bmatrix}, \\ \mathscr{B}_{[0,s)}^{\mathcal{S}_{\mathsf{L}}(t_0)} &= \operatorname{ColSpan}\begin{bmatrix} 0 & I \\ \widehat{\mathcal{O}}_0^{s-1} & \widehat{\mathcal{G}}_0^{s-1} \end{bmatrix}. \end{aligned} \tag{3.14}$$

Note that $\widehat{\mathcal{O}}_0^{s-1}$ and $\widehat{\mathcal{G}}_0^{s-1}$ are defined based on $\mathfrak{A}, \mathfrak{B}, \mathfrak{C}, \mathfrak{D}$. Using the definitions of $\mathfrak{A}, \mathfrak{B}, \mathfrak{C}, \mathfrak{D}$ from (3.7) and the periodicity of $\mathcal{S}$, one can verify that for all $s \in \mathbb{N}$

$$\mathcal{O}_{t_0}^{t_0+sT-1} = \widehat{\mathcal{O}}_0^{s-1}, \qquad \mathcal{G}_{t_0}^{t_0+sT-1} = \widehat{\mathcal{G}}_0^{s-1},$$

and hence the column spans in (3.14) are equal. ■

Remark 3.13 (**Dependence on Initial Step t_0**). The lifted system and its behavior depend on the initial step t_0. For instance, consider the following single-state single-input-single-output (SISO) LTP system $\mathcal{S}$ of period $T = 2$,

$$\mathcal{S} : \begin{cases} x_{t+1} = x_t + (-1)^t u_t \\ \quad y_t = x_t \end{cases}$$

whose corresponding lifted system $\mathcal{S}_{\mathsf{L}}(t_0)$ for $t_0 \in \mathbb{Z}$.

$$\mathcal{S}_{\mathsf{L}}(t_0) : \begin{cases} \mathbf{x}_{\tau+1} = \mathbf{x}_\tau + (-1)^{t_0} \begin{bmatrix} 1 & -1 \end{bmatrix} \mathbf{u}_\tau \\ \quad \mathbf{y}_\tau = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \mathbf{x}_\tau + (-1)^{t_0} \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \mathbf{u}_\tau \end{cases}$$

It follows (via Lemma 3.3) that for $t_0 \in \mathbb{Z}$ the restricted behavior of $\mathcal{S}_{\mathsf{L}}(t_0)$ on interval $[0, 0]$ is

$$\mathscr{B}_{[0,0]}^{\mathcal{S}_{\mathsf{L}}(t_0)} = \operatorname{ColSpan}\left[\begin{array}{c|cc} 0 & 1 & 0 \\ 0 & 0 & 1 \\ \hline 1 & 0 & 0 \\ 1 & (-1)^{t_0} & 0 \end{array}\right].$$

One can now observe that $\mathscr{B}_{[0,0]}^{\mathcal{S}_{\mathsf{L}}(0)}$ and $\mathscr{B}_{[0,0]}^{\mathcal{S}_{\mathsf{L}}(1)}$ are different subspaces. Hence, it is necessary to specify the initial time t_0 when introducing the lifted system. ■

Order and Lag

Notions of order and lag for LTV systems have been introduced in Definition 3.7. The next result relates the order and lag of an LTP system to the order and lag of its lifted system.

Lemma 3.14. *For an LTP system $\mathcal{S}$ of period T, we have (i) $\mathbf{n}(\mathcal{S}_{\mathsf{L}}(t)) = \mathbf{n}(\mathcal{S}, t)$, and (ii) $\mathbf{l}(\mathcal{S}_{\mathsf{L}}(t)) = \lceil \mathbf{l}(\mathcal{S}, t)/T \rceil$.*

Proof. Let (3.1) be the model of $\mathcal{S}$, and (3.6) be the model of $\mathcal{S}_{\mathsf{L}}(t)$. Define matrix $\mathcal{O}_{t_1}^{t_2}$ (resp. $\widehat{\mathcal{O}}_{t_1}^{t_2}$) via (3.4b) for system $\mathcal{S}$ (resp. $\mathcal{S}_{\mathsf{L}}(t)$). Using the definitions of $\mathfrak{A}$ and $\mathfrak{C}$ from (3.7) and the periodicity of $\mathcal{S}$, one can verify that

$$\widehat{\mathcal{O}}_0^{s-1} = \mathcal{O}_t^{t+sT-1} \qquad \forall s \in \mathbb{N}.$$

Let $a_s := \operatorname{rank}(\mathcal{O}_t^{t+s-1})$ and $b_s := \operatorname{rank}(\widehat{\mathcal{O}}_0^{s-1})$. It follows from the above relation that $b_s = a_{sT}$. By (3.9a), we now compute that

$$\mathbf{n}(\mathcal{S}_{\mathsf{L}}(t)) = \lim_{s\to\infty} b_s = \lim_{s\to\infty} a_{sT} = \lim_{s'\to\infty} a_{s'} = \mathbf{n}(\mathcal{S}, t),$$

which shows (i). Similarly, we compute from (3.9b) that

$$\mathbf{l}(\mathcal{S}_{\mathsf{L}}(t)) = \min\{s \in \mathbb{N} : b_s = \mathbf{n}(\mathcal{S}_{\mathsf{L}}(t))\} = \min\{s \in \mathbb{N} : a_{sT} = \mathbf{n}(\mathcal{S}, t)\}, \tag{3.15a}$$

$$\mathbf{l}(\mathcal{S}, t) = \min\{s' \in \mathbb{N} : a_{s'} = \mathbf{n}(\mathcal{S}, t)\}. \tag{3.15b}$$

We know from the minimality in (3.15a) that

$$a_{[\mathbf{l}(\mathcal{S}_{\mathsf{L}}(t))-1]T} \neq \mathbf{n}(\mathcal{S}, t), \qquad a_{\mathbf{l}(\mathcal{S}_{\mathsf{L}}(t))T} = \mathbf{n}(\mathcal{S}, t).$$

By the minimality in (3.15b), the above relations imply that

$$[\mathbf{l}(\mathcal{S}_{\mathsf{L}}(t)) - 1]T < \mathbf{l}(\mathcal{S}, t) \leq \mathbf{l}(\mathcal{S}_{\mathsf{L}}(t))T,$$

which is equivalent to (ii). ■

For unknown LTP systems with known periods and state dimensions, we can establish bounds of their orders and lags.

Corollary 3.14.1. *For an LTP system $\mathcal{S}$ as in (3.1) of period T, we have (i) $\mathbf{n}(\mathcal{S}, t) \leq n$, and (ii) $\mathbf{l}(\mathcal{S}, t) \leq nT$.*

Proof. (i) follows from (3.9a), as $\operatorname{rank}(\mathcal{O}_{t_1}^{t_2})$ is bounded by the row size n of $\mathcal{O}_{t_1}^{t_2}$. (ii) is shown below:

$$\mathbf{l}(\mathcal{S}, t) \leq \mathbf{l}(\mathcal{S}_{\mathsf{L}}(t))T \leq \mathbf{n}(\mathcal{S}_{\mathsf{L}}(t))T = \mathbf{n}(\mathcal{S}, t)T \leq nT,$$

where the first inequality is by Lemma 3.14(ii), and the second inequality is because $\mathbf{l}(\mathcal{S}_{\mathsf{L}}(t)) \leq \mathbf{n}(\mathcal{S}_{\mathsf{L}}(t))$ for the LTI system $\mathcal{S}_{\mathsf{L}}(t)$ [23, Eq. 2]. ■

Controllability

The controllability of an LTP system is equivalent to the controllability of its lifted systems.

Lemma 3.15. *The behavior of an LTP system $\mathcal{S}$ is controllable if, and only if, the behavior of its lifted systems $\mathcal{S}_{\mathsf{L}}(t_0)$ are controllable for all $t_0 \in \mathbb{Z}$.*

Proof. Let T be the period of $\mathcal{S}$. Let $w_{[t_1,t_2]} := \begin{bmatrix} u_{[t_1,t_2]} \\ y_{[t_1,t_2]} \end{bmatrix}$ and $\mathbf{w}_{[\tau_1,\tau_2]} := \begin{bmatrix} \mathbf{u}_{[\tau_1,\tau_2]} \\ \mathbf{y}_{[\tau_1,\tau_2]} \end{bmatrix}$ denote a trajectory of the system $\mathcal{S}$ and the lifted system $\mathcal{S}_{\mathsf{L}}(t_0)$, respectively. Given a trajectory $w_{[t_0,\infty)}$ of $\mathcal{S}$, let $\mathbf{w}_{[0,\infty)}$ be the trajectory of $\mathcal{S}_{\mathsf{L}}(t_0)$ equal to $w_{[t_0,\infty)}$, and vise versa, where we implicitly used $\mathscr{B}^{\mathcal{S}}_{[t_0,\infty)} = \mathscr{B}^{\mathcal{S}_{\mathsf{L}}(t_0)}_{[0,\infty)}$ via Lemma 3.12.

If. For $t_0 \in \mathbb{Z}$, consider trajectories $w^{\mathrm{I}}_{[t_0,\infty)}, w^{\mathrm{II}}_{[t_0,\infty)} \in \mathscr{B}^{\mathcal{S}}_{[t_0,\infty)}$ and an arbitrary integer $t_1 \geq t_0$. Choose some $\tau_1 \in \mathbb{Z}$ such that $t_0 + \tau_1 T \geq t_1$. Since $\mathbf{w}^{\mathrm{I}}_{[0,\infty)}$ and $\mathbf{w}^{\mathrm{II}}_{[0,\infty)}$ are trajectories of $\mathcal{S}_{\mathsf{L}}(t_0)$, by controllability of $\mathcal{S}_{\mathsf{L}}(t_0)$, there exists an integer $\tau_2 \geq \tau_1$ and a trajectory $\mathbf{w}^{\diamond}_{[0,\infty)} \in \mathscr{B}^{\mathcal{S}_{\mathsf{L}}(t_0)}_{[0,\infty)}$ such that

$$\mathbf{w}^{\diamond}_{[0,\tau_1)} = \mathbf{w}^{\mathrm{I}}_{[0,\tau_1)}, \qquad \mathbf{w}^{\diamond}_{[\tau_2,\infty)} = \mathbf{w}^{\mathrm{II}}_{[\tau_2,\infty)} \tag{3.16}$$

Let $t_2 := t_0 + \tau_2 T$. From (3.16) and the equivalence of w and $\mathbf{w}$, $w^{\diamond}_{[t_0,\infty)}$ is such a trajectory that (2.9) holds. Since t_0 and t_1 are arbitrary, $\mathcal{S}$ is controllable by Definition 2.6.

Only if. Consider trajectories $\mathbf{w}^{\mathrm{I}}_{[0,\infty)}, \mathbf{w}^{\mathrm{II}}_{[0,\infty)} \in \mathscr{B}^{\mathcal{S}_{\mathsf{L}}(t_0)}_{[0,\infty)}$ and an arbitrary integer $\tau_1 \geq 0$. Since $w^{\mathrm{I}}_{[t_0,\infty)}$ and $w^{\mathrm{II}}_{[t_0,\infty)}$ are trajectories of $\mathcal{S}$, from controllability of $\mathcal{S}$ there exists an integer $t_2 \geq t_1 := t_0 + \tau_1 T$ and a trajectory $w^{\diamond}_{[t_0,\infty)} \in \mathscr{B}^{\mathcal{S}}_{[t_0,\infty)}$ such that (2.9) holds. Choose some $\tau_2 \in \mathbb{Z}$ satisfying $t_0 + \tau_2 T \geq t_2$. From (2.9) and the equivalence of w and $\mathbf{w}$, the trajectory $\mathbf{w}^{\diamond}_{[0,\infty)}$ satisfies (3.16). Since τ_1 is arbitrary, the LTI system $\mathcal{S}_{\mathsf{L}}(t_0)$ is controllable by Definition 2.6. ■

3.3.4 A Fundamental Lemma for LTP Systems

According to the so-called Fundamental Lemma, under technical conditions, the restricted behavior of an LTI system can be completely described via recorded historical data. This result has been reviewed in Lemma 2.9. Based on the lifting operation, we now define a natural extension of persistent excitation for LTP systems, and present a corresponding version of the fundamental lemma.

Definition 3.16 (Periodic Persistent Excitation)**.** A sequence $z_{[t_1,t_2]}$ is *T-periodically persistently exciting (p.p.e.) of order K*, for $K, T \in \mathbb{N}$ satisfying $K \leq t_2 - t_1 + 1$, if

$$\mathscr{H}_K^T(z_{[t_1,t_2]}) := \begin{bmatrix} z_{t_1} & z_{t_1+T} & \cdots & z_{t_1+PT} \\ z_{t_1+1} & z_{t_1+T+1} & \cdots & z_{t_1+PT+1} \\ \vdots & \vdots & \ddots & \vdots \\ z_{t_1+K-1} & z_{t_1+T+K-1} & \cdots & z_{t_1+PT+K-1} \end{bmatrix}$$

has full row rank, where $P := \lfloor (t_2 - t_1 - K + 1)/T \rfloor$.

Remark 3.17. One can observe that $\mathcal{H}_K^T(z_{[t_1,t_2]})$ is composed of every T-th column of $\mathcal{H}_K(z_{[t_1,t_2]})$.

Lemma 3.18 (Fundamental Lemma for LTP Systems)**.** *Let $\mathcal{S}$ be an LTP system of period T, and let $\mathrm{col}(u^{\mathrm{d}}_{[t_1,t_2]}, y^{\mathrm{d}}_{[t_1,t_2]}) \in \mathscr{B}^{\mathcal{S}}_{[t_1,t_2]}$ be a trajectory of $\mathcal{S}$ on interval $[t_1, t_2]$. For $K \in \mathbb{N}$, if*

(i) the behavior of $\mathcal{S}$ is controllable in the sense of Definition 3.6, and
(ii) $u^{\mathrm{d}}_{[t_1,t_2]}$ is T-p.p.e. of order $(\lceil K/T \rceil + \mathbf{n}(\mathcal{S}, t_1))T$,

then we have

$$\mathrm{ColSpan} \begin{bmatrix} \mathscr{H}_K^T(u^{\mathrm{d}}_{[t_1,t_2]}) \\ \mathscr{H}_K^T(y^{\mathrm{d}}_{[t_1,t_2]}) \end{bmatrix} = \mathscr{B}^{\mathcal{S}}_{[t_1,t_1+K)}.$$

Proof. See Section 3.7.4. ■

When $\mathbf{n}(\mathcal{S}, t_1)$ is unknown but bounded by some $n \in \mathbb{Z}$, we may obtain (ii) in Lemma 3.18 by requiring the input $u^{\mathrm{d}}_{[t_1,t_2]}$ to be T-p.p.e. of a sufficient order $(\lceil K/T \rceil + n)T$. This is because by definition a signal being T-p.p.e. of order K' is also T-p.p.e. of any smaller order $K'' \leq K'$.

3.4 DDPC Methods for LTP Systems

Based on our previous results extending behavioral systems theory to LTP systems, we develop a DDPC method for the LTP systems $\mathcal{S}$ in (3.1) of known period T.

Prediction, Control, and Initial Horizons

We consider a receding-horizon control strategy, in which at control step $t = k$ the control signal u on interval $\mathbb{Z}_{[k,k+N_c)}$ (the *control horizon*) is computed by minimizing an appropriate cost function of the predicted trajectory over a finite horizon $\mathbb{Z}_{[k,k+N)}$ (the *prediction horizon*), where $N_c, N \in \mathbb{N}$ are design parameters with $N_c \leq N$.

In the present data-driven scenario, the initial condition of the system at step k is specified by the recent trajectory in a past interval $\mathbb{Z}_{[k-L,k)}$ called the *initial horizon*, with parameter $L \in \mathbb{N}$. According to Lemma 3.11, if $L \geq \mathbf{l}(\mathcal{S}, k-L)$, we can uniquely predict the future output, given any future input. Notice via Corollary 3.14.1 that the lag $\mathbf{l}(\mathcal{S}, k-L)$ is bounded by nT, so the output prediction is always unique when we select $L \geq nT$. We call the union $\mathbb{Z}_{[k-L,k+N)}$ of the initial and prediction horizons as the *total horizon*; see Fig. 2.4 in Section 2.4.1.

3.4.1 Offline Data Collection

The restricted behavior $\mathscr{B}^{\mathcal{S}}_{[k-L,k+N)}$ on the total horizon must be known for us to predict future trajectories and compute control actions in the DDPC framework. In DDPC for LTI systems, the behavior $\mathscr{B}^{\mathcal{S}}_{[k-L,k+N)}$ can be represented using recorded offline data. We may extend this strategy to the case where $\mathcal{S}$ is an LTP system. However, since the system is periodic, its behavior $\mathscr{B}^{\mathcal{S}}_{[k-L,k+N)}$ can equal one of T different possible subspaces, depending on the time step k. Fortunately, all T possibilities for the behavior $\mathscr{B}^{\mathcal{S}}_{[k-L,k+N)}$ can be covered using collected data.

Offline Data and Data Matrices

Let $u^{\mathrm{d}}_{[t_{\mathrm{d}1},t_{\mathrm{d}2}]}, y^{\mathrm{d}}_{[t_{\mathrm{d}1},t_{\mathrm{d}2}]}$ be offline data collected from the system $\mathcal{S}$ on the interval $[t_{\mathrm{d}1}, t_{\mathrm{d}2}]$, where we require that the input signal $u^{\mathrm{d}}_{[t_{\mathrm{d}1},t_{\mathrm{d}2}]}$ is T-periodically persistently exciting of order $(\lceil K/T \rceil + \mathbf{n}(\mathcal{S}, t_{\mathrm{d}1}))T$, with $K := L + N + T - 1$. Arrange the data into the "uncropped" data matrices $U^{\mathrm{d}} \in \mathbb{R}^{mK \times h}$ and $Y^{\mathrm{d}} \in \mathbb{R}^{pK \times h}$:

$$U^{\mathrm{d}} := \mathscr{H}^T_K(u^{\mathrm{d}}_{[t_{\mathrm{d}1},t_{\mathrm{d}2}]}), \qquad Y^{\mathrm{d}} := \mathscr{H}^T_K(y^{\mathrm{d}}_{[t_{\mathrm{d}1},t_{\mathrm{d}2}]}),$$

where h denotes the common width of U^{d} and Y^{d}, given by $h := \lfloor (t_{\mathrm{d}2} - t_{\mathrm{d}1} - K + 1)/T \rfloor + 1$. We extract from U^{d} and Y^{d} the T sets of *data matrices* $U_{\mathrm{p}}(\theta) \in \mathbb{R}^{mL \times h}$, $U_{\mathrm{f}}(\theta) \in \mathbb{R}^{mN \times h}$,

$Y_{\mathrm{p}}(\theta) \in \mathbb{R}^{pL\times h}$ and $Y_{\mathrm{f}}(\theta) \in \mathbb{R}^{pN\times h}$, defined as

$$\begin{aligned} U_{\mathrm{p}}(\theta) &:= U^{\mathrm{d}}|_{[\theta,\theta+L-1]}, \qquad & U_{\mathrm{f}}(\theta) &:= U^{\mathrm{d}}|_{[\theta+L,\theta+L+N-1]}, \\ Y_{\mathrm{p}}(\theta) &:= Y^{\mathrm{d}}|_{[\theta,\theta+L-1]}, \qquad & Y_{\mathrm{f}}(\theta) &:= Y^{\mathrm{d}}|_{[\theta+L,\theta+L+N-1]}, \end{aligned} \tag{3.17}$$

where each set has an exclusive index $\theta \in \{1,\dots,T\}$. In (3.17), we let $U^{\mathrm{d}}|_{[r_1,r_2]} \in \mathbb{R}^{m(r_2-r_1+1)\times h}$ denote the sub-matrix consisting of the r_1-th, ..., r_2-th block rows of U^{d}, and similarly for $Y^{\mathrm{d}}|_{[r_1,r_2]} \in \mathbb{R}^{p(r_2-r_1+1)\times h}$.

Representation of Behavior

The matrices $U_{\mathrm{p}}(\theta)$, $U_{\mathrm{f}}(\theta)$, $Y_{\mathrm{p}}(\theta)$, $Y_{\mathrm{f}}(\theta)$ built from offline data can represent the behavior on the total horizon at control step $k = \mathbf{k}(\theta) := t_{\mathrm{d1}} + \theta + L - 1$, as said in the following lemma; see Fig. 3.3.

Lemma 3.19. *Consider an LTP system $\mathcal{S}$ as in (3.1) of period T. For $L, N \in \mathbb{N}$, construct data matrices $U_{\mathrm{p}}(\theta), U_{\mathrm{f}}(\theta), Y_{\mathrm{p}}(\theta), Y_{\mathrm{f}}(\theta)$ from (3.17) with data $w^{\mathrm{d}}_{[t_{\mathrm{d1}},t_{\mathrm{d2}}]}$. If $\mathcal{S}$ is controllable and $u^{\mathrm{d}}_{[t_{\mathrm{d1}},t_{\mathrm{d2}}]}$ is T-p.p.e. of order $(\lceil K/T\rceil + \mathbf{n}(\mathcal{S}, t_{\mathrm{d1}}))T$ with $K := L + N + T - 1$, then for $\theta \in \{1,\dots,T\}$ we have*

$$\mathrm{ColSpan}\big(\mathrm{col}(U_{\mathrm{p}}(\theta), U_{\mathrm{f}}(\theta), Y_{\mathrm{p}}(\theta), Y_{\mathrm{f}}(\theta))\big) = \mathscr{B}^{\mathcal{S}}_{[\mathbf{k}(\theta)-L,\mathbf{k}(\theta)+N)}, \tag{3.18}$$

where $\mathbf{k}(\theta) := t_{\mathrm{d1}} + \theta + L - 1$.

Proof. Let $s_1(\theta) := t_{\mathrm{d1}} + (\theta - 1)$ and $s_2(\theta) := t_{\mathrm{d2}} - (T - \theta)$, and notice that

$$\mathrm{col}(U_{\mathrm{p}}(\theta), U_{\mathrm{f}}(\theta), Y_{\mathrm{p}}(\theta), Y_{\mathrm{f}}(\theta)) = \begin{bmatrix} \mathscr{H}^T_K(u^{\mathrm{d}}_{[s_1(\theta),s_2(\theta)]}) \\ \mathscr{H}^T_K(y^{\mathrm{d}}_{[s_1(\theta),s_2(\theta)]}) \end{bmatrix}$$

Also let $H := (\lceil (L+N)/T\rceil + \mathbf{n}(\mathcal{S}, t_{\mathrm{d1}}))T$ and $\widehat{H} := (\lceil K/T\rceil + \mathbf{n}(\mathcal{S}, t_{\mathrm{d1}}))T$.

We first conclude that $u^{\mathrm{d}}_{[s_1(\theta),s_2(\theta)]}$ is T-p.p.e of order H, i.e., $\mathscr{H}^T_H(u^{\mathrm{d}}_{[s_1(\theta),s_2(\theta)]})$ has full row rank. This is because $\mathscr{H}^T_{\widehat{H}}(u^{\mathrm{d}}_{[t_{\mathrm{d1}},t_{\mathrm{d2}}]})$ has full row rank (since $u^{\mathrm{d}}_{[t_{\mathrm{d1}},t_{\mathrm{d2}}]}$ is T-p.p.e of order $\widehat{H}$) and $\mathscr{H}^T_H(u^{\mathrm{d}}_{[s_1(\theta),s_2(\theta)]})$ is a sub-matrix of $\mathscr{H}^T_{\widehat{H}}(u^{\mathrm{d}}_{[t_{\mathrm{d1}},t_{\mathrm{d2}}]})$.

Then, the result follows from Lemma 3.18, with controllability of $\mathcal{S}$ and periodic persistent excitation of $u^{\mathrm{d}}_{[s_1(\theta),s_2(\theta)]}$. ■

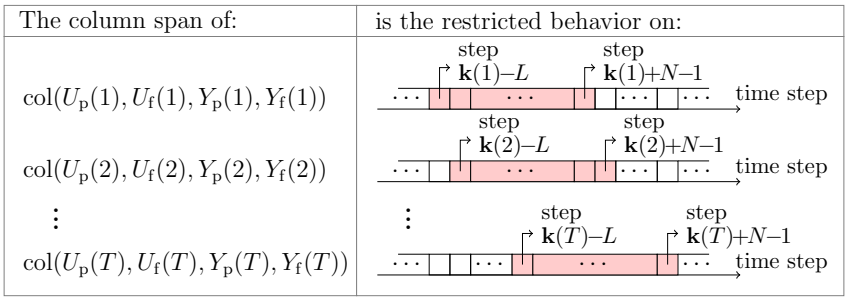

Figure 3.3: The column span of $\mathrm{col}(U_\mathrm{p}(\theta), U_\mathrm{f}(\theta), Y_\mathrm{p}(\theta), Y_\mathrm{f}(\theta))$ is the restricted behavior on time interval $[\mathbf{k}(\theta) - L, \mathbf{k}(\theta) + N)$.

Since $\{\mathbf{k}(\theta)\}_{\theta=1}^{T}$ are consecutive time steps in one period, by periodicity of $\mathcal{S}$, the subspaces $\mathscr{B}^{\mathcal{S}}_{[\mathbf{k}(\theta)-L,\mathbf{k}(\theta)+N)}$ with different selections of the index $\theta \in \{1, \ldots, T\}$ cover all T possibilities of the behavior $\mathscr{B}^{\mathcal{S}}_{[k-L,k+N)}$ for different control steps k. Define the *proper index* $\Theta(t)$ at control step k.

$$\Theta(k) := 1 + (k - t_{\mathrm{d}1} - L \mod T) \tag{3.19}$$

Thus, $\theta = \Theta(k)$ is the "correct" index θ such that the data matrices $U_\mathrm{p}(\theta)$, $U_\mathrm{f}(\theta)$, $Y_\mathrm{p}(\theta)$, $Y_\mathrm{f}(\theta)$ represent the behavior on the total horizon at control step k, i.e.,

$$\mathrm{ColSpan}\begin{bmatrix} U_\mathrm{p}(\Theta(k)) \\ U_\mathrm{f}(\Theta(k)) \\ Y_\mathrm{p}(\Theta(k)) \\ Y_\mathrm{f}(\Theta(k)) \end{bmatrix} = \mathscr{B}^{\mathcal{S}}_{[k-L,k+N)}, \tag{3.20}$$

because of (3.18), periodicity of $\mathcal{S}$ and the fact that $k - \mathbf{k}(\Theta(k))$ is a multiple of T.

3.4.2 Online Warm-Up Process

Now we introduce the online process of the algorithm. Suppose we have collected the offline data in Section 3.4.1. Before controlling the system, we start by recording an initial trajectory and then finish a so-called "index test"; see Fig. 3.4. Throughout the warm-up process, no control algorithm is executed, and a stabilizing input is applied to the system.

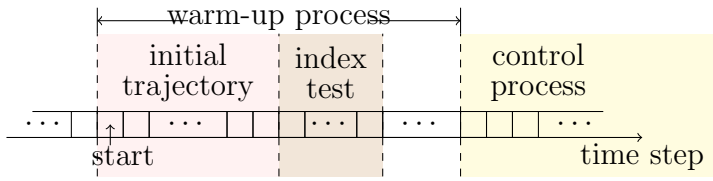

Figure 3.4: The entire online process with a warm-up process.

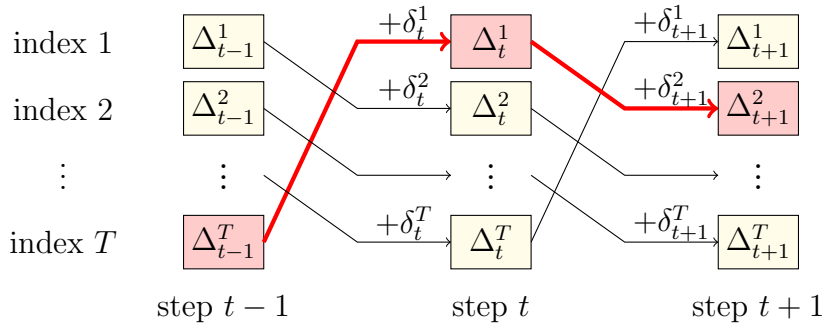

Figure 3.5: Accumulating the errors δ^θ_t into Δ^θ_t.

Initial Trajectory

At the beginning of the online process, we apply no algorithm and simply record an initial trajectory of at least L steps. This initial trajectory is used to initialize the subsequent procedures.

Index Test

After the initial trajectory is recorded, the next process is an "index test". Although the required proper index $\Theta(t)$ is clearly defined in (3.19), this value will be unknown unless t_{d1} is known during the data collection process, and will generally be unknown in a practical implementation. Due to this consideration, a process is required to identify $\Theta(t)$ at current time t. We propose a heuristic index-testing which supports both deterministic and stochastic LTP systems.

At time t, consider the recorded past trajectory $u_{[t-L,t)}, y_{[t-L,t)}$ and recall the offline data matrices $U_{\mathrm{p}}(\theta), Y_{\mathrm{p}}(\theta)$ in (3.17). We compute the SVD of the matrix $\mathrm{col}(U_{\mathrm{p}}(\theta), Y_{\mathrm{p}}(\theta))$

and the error

$$\delta_t^\theta := \left\| N_{\mathrm{p}}(\theta)^\mathsf{T} \begin{bmatrix} u_{[t-L,t)} \\ y_{[t-L,t)} \end{bmatrix} \right\|_2, \tag{3.21}$$

where the columns of the matrix $N_{\mathrm{p}}(\theta) \in \mathbb{R}^{(m+p)L \times \bullet}$ are the "non-dominant" left-singular vectors of $\mathrm{col}(U_{\mathrm{p}}(\theta), Y_{\mathrm{p}}(\theta))$ that correspond to singular values not exceeding some specified threshold $\sigma_{\mathrm{IT}} > 0$. In the deterministic case with $\sigma_{\mathrm{IT}} = 0$, we have $\delta_t^\theta = 0$ if, and only if, $\mathrm{col}(U_{\mathrm{p}}(\theta), Y_{\mathrm{p}}(\theta)) \in \mathscr{B}^{\mathcal{S}}_{[\mathbf{k}(\theta)-L,\mathbf{k}(\theta))}$ (because $\mathscr{B}^{\mathcal{S}}_{[\mathbf{k}(\theta)-L,\mathbf{k}(\theta))} = \mathrm{ColSpan}(\mathrm{col}(U_{\mathrm{p}}(\theta), Y_{\mathrm{p}}(\theta)) = \mathrm{Null}(N_{\mathrm{p}}(\theta))$, where the first equality is by (3.20) and Lemma 3.4, and second equality follows from the definition of $N_{\mathrm{p}}(\theta)$ with $\sigma_{\mathrm{IT}} = 0$), so the indices θ with $\delta_t^\theta \neq 0$ are discarded as possibilities of $\Theta(t)$. In the general case, we regard δ_t^θ as a "score" to falsify the hypothesis that the index θ is proper at time t.

We repeat the above computation on multiple time steps. Define accumulated errors $\Delta_t^\theta \in \mathbb{R}$ for $\theta \in \{1, \ldots, T\}$, which are initialized to zero and updated in the following way,

$$\Delta_t^\theta := \Delta_{t-1}^{\mathrm{prev}(\theta)} + \delta_t^\theta \tag{3.22}$$

where $\mathrm{prev}(\theta)$ denotes the "cyclically previous" index of θ.

$$\mathrm{prev}(\theta) := \begin{cases} \theta - 1, & \text{for } \theta \in \{2, \ldots, T\} \\ T, & \text{for } \theta = 1 \end{cases}$$

See Fig. 3.5: each Δ_t^θ is obtained from updating $\Delta_{t-1}^{\mathrm{prev}(\theta)}$; the red boxes and thick arrows show for example the "correct path" in which the index $\theta = \Theta(t)$ is proper at all time t; the "correct path" is expected to have the lowest values of both δ_t^θ and Δ_t^θ. The reason we obtain Δ_t^θ by updating $\Delta_{t-1}^{\mathrm{prev}(\theta)}$ in (3.22) is that an index θ is proper at time t if, and only if, $\mathrm{prev}(\theta)$ is the proper index at the last step $t-1$. Thus, Δ_t^θ is the "cumulative score" to falsify the hypothesis that $\theta = \Theta(t)$, considering all errors computed so far. The number of iterations N_{IT} of this process is user-specified. When the algorithm terminates, the index θ with the smallest Δ_t^θ is selected as the estimated proper index $\widehat{\Theta}(t)$ at the current time t. The entire index-testing process is outlined in Algorithm 3.

Once we obtain $\widehat{\Theta}(t)$ at time t, the estimated proper index $\widehat{\Theta}(t')$ for any future time $t' \geq t$ is derived according to

$$\widehat{\Theta}(t+1) := \begin{cases} \widehat{\Theta}(t) + 1, & \text{if } \widehat{\Theta}(t) \in \{1, \ldots, T-1\}, \\ 1, & \text{if } \widehat{\Theta}(t) = T, \end{cases}$$

which is the same way that $\Theta(t)$ evolves with time t.

Algorithm 3 Index Test

Input: the time step t and the data matrices $U_{\mathrm{p}}(\theta), Y_{\mathrm{p}}(\theta)$ for all $\theta \in \{1, \ldots, T\}$.

```
Compute N_p(θ) as described in Section 3.4.2 for all θ.
Initialize the accumulators Δ^θ_{t-1} = 0 for all θ.
for i from 1 to N_IT do
    Compute δ^θ_t from (3.21) for all θ.
    Update Δ^θ_t via (3.22) for all θ.
    if i < N_IT then
        Input a user-defined u_t to the system S.
        Set t ← t + 1.
```

Output: the estimated proper index $\widehat{\Theta}(t) \leftarrow \operatorname{argmin}_\theta \Delta_t^\theta$.

3.4.3 Online Control Process

With the proper index identified or known, we can start the control process. We provide for the LTP system $\mathcal{S}$ two alternative controllers, which generalize Data-enabled Predictive Control (DeePC) and Subspace Predictive Control (SPC) methods in the literature.

Let $\overline{u}$ and $\overline{y}$ denote the future input and predicted output respectively. At control step $t = k$, we consider the cost (2.8), reproduced as

$$\sum_{t=k}^{k+N-1} J_t(\overline{u}_t, \overline{y}_t)$$

with used-selected cost function J_t in the form of (2.2), and constrain the future input-output signal as (2.7), reproduced as

$$\overline{u}_t \in \mathcal{U}, \quad \overline{y}_t \in \mathcal{Y}, \qquad t \in \mathbb{Z}_{[k,k+N)},$$

with user-defined constraint sets $\mathcal{U} \subseteq \mathbb{R}^m$ and $\mathcal{Y} \subseteq \mathbb{R}^p$. The *periodic DeePC (P-DeePC) problem* at control step $t = k$ is

$$\underset{g,\overline{u},\overline{y}}{\text{minimize}} \quad (2.8) \quad \text{s.t.} \quad (3.23) \text{ and } (2.7) \tag{P-DeePC}$$

with an auxiliary variable $g \in \mathbb{R}^h$, where (3.23) is given as

$$\begin{bmatrix} U_{\mathrm{p}}(\widehat{\Theta}(k)) \\ U_{\mathrm{f}}(\widehat{\Theta}(k)) \\ Y_{\mathrm{p}}(\widehat{\Theta}(k)) \\ Y_{\mathrm{f}}(\widehat{\Theta}(k)) \end{bmatrix} g = \begin{bmatrix} u_{[k-L,k)} \\ \overline{u}_{[k,k+N)} \\ y_{[k-L,k)} \\ \overline{y}_{[k,k+N)} \end{bmatrix}. \tag{3.23}$$

The *periodic SPC (P-SPC) problem* at control step $t = k$

$$\underset{\overline{u},\overline{y}}{\text{minimize}} \quad (2.8) \quad \text{s.t.} \quad (3.24) \text{ and } (2.7) \tag{P-SPC}$$

with (3.24) given as

$$\overline{y}_{[k,k+N)} = Y_{\mathrm{f}}(\widehat{\Theta}(k)) \begin{bmatrix} U_{\mathrm{p}}(\widehat{\Theta}(k)) \\ U_{\mathrm{f}}(\widehat{\Theta}(k)) \\ Y_{\mathrm{p}}(\widehat{\Theta}(k)) \end{bmatrix}^{\dagger} \begin{bmatrix} u_{[k-L,k)} \\ \overline{u}_{[k,k+N)} \\ y_{[k-L,k)} \end{bmatrix}. \tag{3.24}$$

After solving the optimal future trajectory $\overline{u}_{[k,k+N)}$ from either (P-DeePC) or (P-SPC), we apply the first N_{c} inputs $\overline{u}_{[k,k+N_{\mathrm{c}})}$ to the system $\mathcal{S}$. The whole control process is illustrated in Algorithm 4.

Algorithm 4 Online Control Process of P-DeePC and P-SPC

Input: initial-, prediction- and control-horizon lengths L, N, N_{c}, stage-cost function $J_t(\cdot)$, and constraint sets $\mathcal{U}, \mathcal{Y}$, and offline data $u^{\mathrm{d}}, y^{\mathrm{d}}$.

```
Obtain the index estimate Θ̂(·) through Algorithm 3.
Record the initial trajectory u_[-L,0), y_[-L,0) of the system.
Initialize the control step k ← 0.
while true do
    Solve control actions ū_[k,k+N) from problem (P-DeePC) or problem (P-SPC).
    for t from k to k + N_c − 1 do
        Apply input u_t ← ū_t to the system (3.1).
        Measure output from the system (3.1).
    Set k ← k + N_c and update Θ̂(t) correspondingly.
```

Performance Guarantee

In the deterministic case, both P-DeePC and P-SPC produce the same control actions that one would obtain from traditional MPC applied to the LTP system. The MPC problem for system (3.1) at control step $t = k$,

$$\underset{\overline{x},\overline{u},\overline{y}}{\text{minimize}} \quad (2.8) \quad \text{s.t.} \quad (3.25) \text{ and } (2.7) \tag{LTV-MPC}$$

where (3.25) is the time-varying case of (2.6), given as follows.

$$\begin{aligned} \bar{x}_{t+1} &= A_t \bar{x}_t + B_t \bar{u}_t, \qquad && t \in \mathbb{Z}_{[k,k+N)} \\ \bar{y}_t &= C_t \bar{x}_t + D_t \bar{u}_t, \qquad && t \in \mathbb{Z}_{[k,k+N)} \\ \bar{x}_k &= \hat{x}_k \end{aligned} \tag{3.25}$$

Proposition 3.20. *Consider an LTP system $\mathcal{S}$ as in (3.1) of period T. Let $u^{\mathrm{d}}_{[t_{\mathrm{d}1},t_{\mathrm{d}2}]}, y^{\mathrm{d}}_{[t_{\mathrm{d}1},t_{\mathrm{d}2}]}$ be offline data from $\mathcal{S}$ on interval $[t_{\mathrm{d}1}, t_{\mathrm{d}2}]$. At control step $t = k$, for integer $L, N \in \mathbb{N}$ such that $L \geq \mathbf{l}(\mathcal{S}, k-L)$, assume that*

(i) the behavior of $\mathcal{S}$ is controllable in the sense of Definition 2.6,
(ii) $u^{\mathrm{d}}_{[t_{\mathrm{d}1},t_{\mathrm{d}2}]}$ is T-p.p.e. of order $(\lceil K/T \rceil + \mathbf{n}(\mathcal{S}, t_{\mathrm{d}1}))T$, with $K := L + N + T - 1$, and
(iii) $\widehat{\Theta}(k) = \Theta(k)$.

Suppose the recent trajectory $u_{[k-L,k)}, y_{[k-L,k)}$ is known and the state is estimated exactly, i.e., $x_k = \hat{x}_k$. Then,

- *the unique optimal trajectory $\bar{u}_{[k,k+N)}, \bar{y}_{[k,k+N)}$ by (P-DeePC),*
- *the unique optimal trajectory $\bar{u}_{[k,k+N)}, \bar{y}_{[k,k+N)}$ by (P-SPC), and*
- *the unique optimal trajectory $\bar{u}_{[k,k+N)}, \bar{y}_{[k,k+N)}$ by (LTV-MPC)*

are all the same.

Proof. See Section 3.7.5. ■

This result generalizes Proposition 2.12, which claims the equivalence of DeePC, SPC and MPC for LTI systems.

Remark 3.21. Our extension of DeePC and SPC to LTP systems is based on the insight that the data collected from an LTP system is equivalent to data collected from an appropriate LTI lifted system. In particular, after stacking LTP-system data into lifted-system data, we can apply the established LTI DDPC methods and compute control signals for the lifted system, and thereby obtain control signals for the original LTP system. A benefit of our treatment here is that discussion of lifted systems can be entirely omitted once proper behavioral systems concepts are defined directly on the LTP system, as we have done in Section 3.3. ■

Regularization

To adapt our methods for stochastic LTP systems with noisy measurements, we may regularize both P-DeePC and P-SPC. Regularizing P-DeePC is similar as regularizing

DeePC [39, 40, 61, 62]. Here we exhibit quadratic regularization, where problem (P-DeePC) is modified as follows,

$$\underset{g,\overline{u},\overline{y},\sigma_{\mathrm{y}}}{\text{minimize}} \quad (2.8) + \lambda_{\mathrm{y}}\|\sigma_{\mathrm{y}}\|_2^2 + \lambda_{\mathrm{g}}\|g\|_2^2 \ \text{ s.t. } \ (3.26) \text{ and } (2.7)$$

with a slack variable $\sigma_{\mathrm{y}} \in \mathbb{R}^{pL}$, positive parameters $\lambda_{\mathrm{y}}, \lambda_{\mathrm{g}}$, and (3.26) a modified constraint from (3.23).

$$\begin{bmatrix} U_{\mathrm{p}}(\widehat{\Theta}(k)) \\ U_{\mathrm{f}}(\widehat{\Theta}(k)) \\ Y_{\mathrm{p}}(\widehat{\Theta}(k)) \\ Y_{\mathrm{f}}(\widehat{\Theta}(k)) \end{bmatrix} g = \begin{bmatrix} u_{[k-L,k)} \\ \overline{u}_{[k,k+N)} \\ y_{[k-L,k)} \\ \overline{y}_{[k,k+N)} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \sigma_{\mathrm{y}} \\ 0 \end{bmatrix}. \tag{3.26}$$

To regularize P-SPC, in the computation of the pseudo-inverse in (3.24), we treat as zero the singular values smaller than a selected threshold σ_{SPC}; the remainder of the settings in regularized P-SPC are same as in P-SPC.

3.5 Simulations

We illustrate the algorithm proposed in Section 3.4 and its robustness to noisy data via numerical example. Consider the mass-spring-damper system in Fig. 3.6.

The control objective is reference tracking for the positions (x_1, x_2, x_3) of the three masses. There are three control inputs: the force F applied to the mass m_1, and the end positions x_4 and x_5 of the free ends of the springs k_4 and k_5. The stiffness and damping

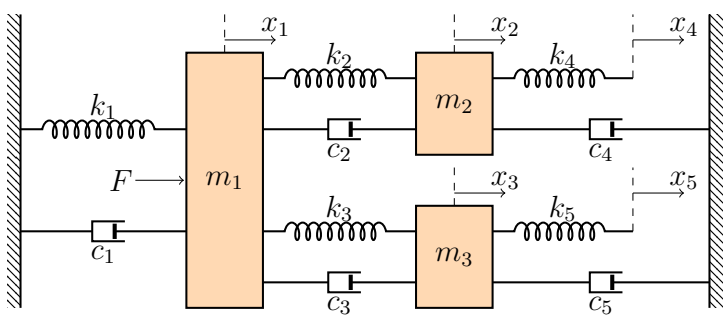

Figure 3.6: A spring-mass-damper model for simulation.

Quantity (Unit)	**Symbol**	**Value**[a]
Mass (kg)	m_1, m_2, m_3	$6, 4, 3$ respectively
Spring Stretch (N/m)	k_1	$10 - 4\sin(2\pi t) + 2\sin(4\pi t)$
	k_2, k_3	$7 - 3\cos(4\pi t)$
	k_4, k_5	$4 - 2\sin(4\pi t)$
Viscosity ($\mathrm{N} \cdot \mathrm{s/m}$)	c_1	$9 + 3\sin(2\pi t)$
	c_2, c_3	$5 + 2\cos(2\pi t)$
	c_4, c_5	15

[a]Variable t is the time in seconds.

Table 3.1: Physical quantities of the simulated system.

Parameter	**Value**
initial horizon length L	30
prediction horizon length N	30
control horizon length N_c	1
tracking cost matrix Q in (2.2)	$\mathrm{diag}(1, 1, 1)$
input cost matrix R in (2.2)	$\mathrm{diag}(10^{-6}, 10^{-4}, 10^{-4})$
input constraint set $\mathcal{U}$ in (2.3)	$[-8, 8] \times [-3, 3] \times [-3, 3]$
output constraint set $\mathcal{Y}$ in (2.3)	$[-20, 20]^3$
iteration number N_IT in the index test	12
SV threshold σ_IT in the index test	1
regularization parameter λ_y for P-DeePC	10^6
regularization parameter λ_g for P-DeePC	10^{-3}
SV threshold σ_SPC in regularized P-SPC	0.5

Table 3.2: Control parameters in simulation.

parameters k_i and c_i are periodic functions of time, given in Table 3.1, and each has a period of 1 second. We discretize the system with a sampling time 0.2s, and thus the period of the discretized system is $T = 5$. A process noise $w_t \overset{\text{i.i.d}}{\sim} \mathcal{N}(0_{6\times 1}, \sigma^2 I_6)$ and a measurement noise $v_t \overset{\text{i.i.d}}{\sim} \mathcal{N}(0_{3\times 1}, \sigma^2 I_3)$ are added to the discrete-time model, with noise amplitude $\sigma^2 = 10^{-3}$. The control and index-test parameters are selected in Table 3.2.

For collection of offline data, we apply a random input signal $u_t^{\text{d}} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0_{3\times 1}, I_3)$ and measure the resulting positions (x_1, x_2, x_3). The online process starts at time $t = 0$; in the warm-up process, the input is random $u_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0_{3\times 1}, I_3/10)$, so that heuristically the index test gives a correct result. After recording an initial trajectory on interval $[0, 29]$, we start the index test at time $t = 29$, and terminate the process at time $t = 40$ (as $N_{\text{IT}} = 12$). In our simulation, the proper index $\Theta(t)$ was identified correctly.

We start control at time $t = 40$, and apply sequential changes in the reference signals given by $r_t = [0, 0, 0]^\mathsf{T}$ for $40 \le t < 60$, $r_t = [5, 0, 0]^\mathsf{T}$ for $60 \le t < 80$, $r_t = [5, 15, 0]^\mathsf{T}$ for $80 \le t < 100$, and $r_t = [5, 15, -10]^\mathsf{T}$ for $t \ge 100$. We evaluate the control performance via the one-step cost $\|y_t - r_t\|_Q^2 + \|u_t\|_R^2$, and the results are shown in Fig. 3.7.

For comparison purposes, we also plot the closed-loop responses under (i) MPC using a perfect system model with full-state measurements, and (ii) the regularized DeePC and regularized SPC methods for LTI systems. For the latter, the settings are the same as for P-DeePC (resp. P-SPC), except that we replace the matrices $U_{\text{p}}(\widehat{\Theta}(t))$, $U_{\text{f}}(\widehat{\Theta}(t))$, $Y_{\text{p}}(\widehat{\Theta}(t))$, $Y_{\text{f}}(\widehat{\Theta}(t))$ in (3.26) (resp. (3.24)) by $U_{\text{p}}(1), U_{\text{f}}(1), Y_{\text{p}}(1), Y_{\text{f}}(1)$ respectively, i.e., we use a single set of data matrices at all time t. Around the step changes of the reference signal, all controllers have comparable performances with similar cost values. For the steady-state performance when the reference signal stays constant, the proposed regularized P-DeePC (resp. P-SPC) method outperforms the direct use of regularized DeePC (resp. SPC) of LTI systems. This significant difference indicates the necessity of using different sets of data matrices $U_{\text{p}}(\theta), U_{\text{f}}(\theta), Y_{\text{p}}(\theta), Y_{\text{f}}(\theta)$ with different indices θ as in (3.26) and (3.24) for P-DeePC and P-SPC respectively at different time steps.

3.6 Chapter Conclusions

We proposed a DDPC algorithm for unknown LTP systems with known periods. For deterministic LTP systems, the method is equivalent to classical MPC, but without the requirement of a parametric model. The approach is supported by extensions of results from behavioral systems theory to LTV and LTP systems. Simulation results provide

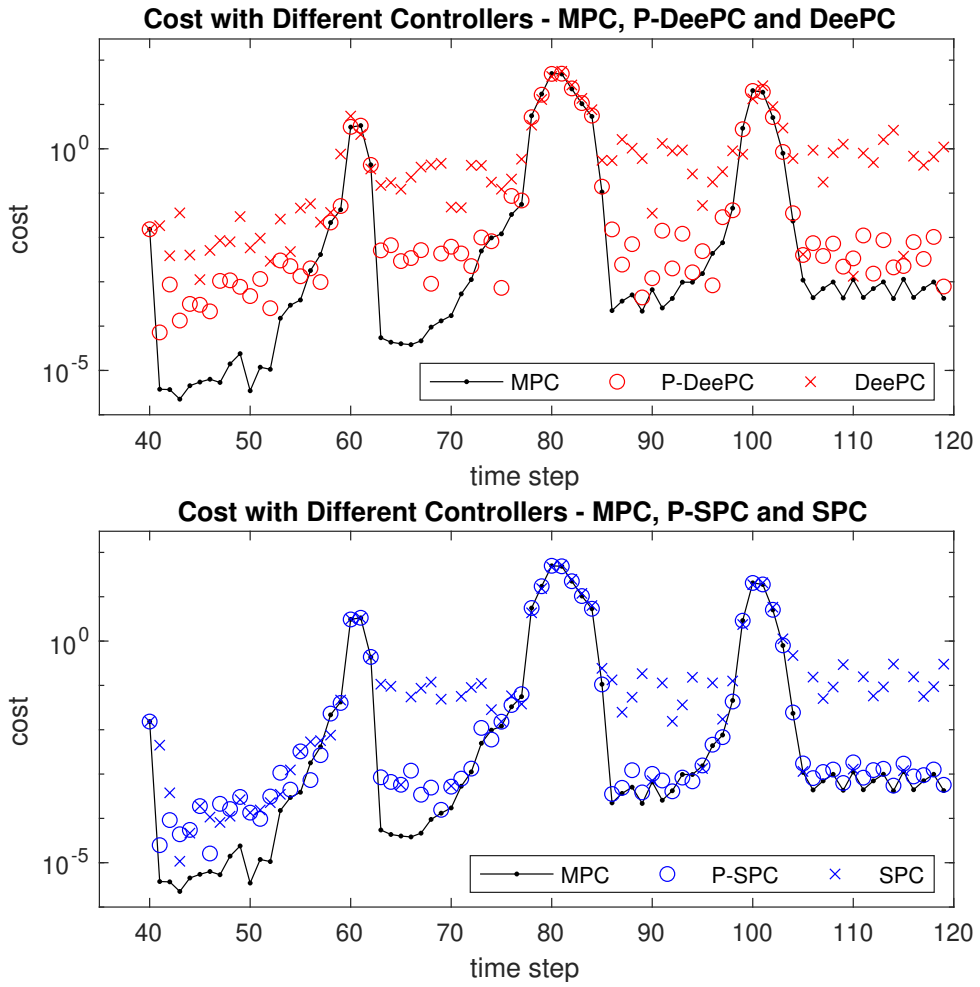

Figure 3.7: Tracking cost of the simulated LTP system with different controllers.

evidence that the approach is robust to measurement noise and stochasticity, and that it significantly outperforms a naive application of data-driven LTI control methods.

There are several open directions for future work. First, as our design requires a priori knowledge of the period T, relaxing this assumption is of interest, as is investigating the robustness of the approach to errors in the selected period. Second, we will seek to develop a rigorous performance guarantee for the "index test" outlined in Section 3.4.2. Finally, we note that there remain open questions in the behavioral theory of LTP systems, such as what relationships can be established between the behaviors of the T different lifted systems arising from a given LTP system.

3.7 Appendices

In this section, we present proofs of some results in the chapter.

3.7.1 Proof of Lemma 3.4

Proof. In this proof, let

$$\mathscr{D}_{t_0}^{t_1} := \mathrm{ColSpan}\big(\mathrm{col}(U_{t_0}, \ldots, U_{t_1}, Y_{t_0}, \ldots, Y_{t_1})\big).$$

Construct a truncation mapping $f_{\mathrm{trc}} : \mathbb{R}^{(t_2-t_0+1)(m+p)} \to \mathbb{R}^{(t_1-t_0+1)(m+p)}$

$$f_{\mathrm{trc}}\left(\begin{bmatrix} u_{[t_0,t_2]} \\ y_{[t_0,t_2]} \end{bmatrix} \right) := \begin{bmatrix} u_{[t_0,t_1]} \\ y_{[t_0,t_1]} \end{bmatrix}$$

for all input-output signals u, y. Through this mapping, $\mathscr{D}_{t_0}^{t_1}$ is the image of $\mathscr{D}_{t_0}^{t_2}$, and the image of $\mathscr{B}_{[t_0,t_2]}^{\mathcal{S}}$ is

$$\mathscr{E}_{t_0}^{t_1} := \left\{ \begin{bmatrix} u_{[t_0,t_1]} \\ y_{[t_0,t_1]} \end{bmatrix} \middle| \begin{bmatrix} u_{[t_0,t_2]} \\ y_{[t_0,t_2]} \end{bmatrix} \in \mathscr{B}_{[t_0,t_2]}^{\mathcal{S}} \right\}.$$

Now we show that $\mathscr{E}_{t_0}^{t_1} = \mathscr{B}_{[t_0,t_1]}^{\mathcal{S}}$. From Lemma 3.3, we have

$$\begin{aligned} \mathscr{B}_{[t_0,t_2]}^{\mathcal{S}} &= \mathrm{ColSpan} \left[\begin{array}{c|c} 0 & I \\ \hline \mathcal{O}_{t_0}^{t_2} & \mathcal{G}_{t_0}^{t_2} \end{array} \right] \\ &= \mathrm{ColSpan} \left[\begin{array}{c|cc} 0 & I & 0 \\ 0 & 0 & I \\ \hline \mathcal{O}_{t_0}^{t_1} & \mathcal{G}_{t_0}^{t_1} & 0 \\ \mathcal{O}_{t_1+1}^{t_2} \Phi_{t_0}^{t_1+1} & \mathcal{O}_{t_1+1}^{t_2} \mathcal{C}_{t_0}^{t_1} & \mathcal{G}_{t_1+1}^{t_2} \end{array} \right] \end{aligned}$$

where the second equality can be verified by expanding $\Phi_{t_1}^{t_2}, \mathcal{C}_{t_1}^{t_2}, \mathcal{O}_{t_1}^{t_2}, \mathcal{G}_{t_1}^{t_2}$ into system matrices via (3.3a), (3.4a), (3.4b), (3.4c) respectively. Through the truncating operation f_{trc}, the image $\mathscr{E}_{t_0}^{t_1}$ of $\mathscr{B}_{[t_0,t_2]}^{\mathcal{S}}$ is therefore

$$\mathscr{E}_{t_0}^{t_1} = \text{ColSpan}\begin{bmatrix} 0 & I \\ \mathcal{O}_{t_0}^{t_1} & \mathcal{G}_{t_0}^{t_1} \end{bmatrix}$$

which equals $\mathscr{B}_{[t_0,t_1]}^{\mathcal{S}}$ by Lemma 3.3, so $\mathscr{B}_{[t_0,t_1]}^{\mathcal{S}}$ is the image of $\mathscr{B}_{[t_0,t_2]}^{\mathcal{S}}$ by operation f_{trc}. The result $\mathscr{B}_{[t_0,t_1]}^{\mathcal{S}} = \mathscr{D}_{t_0}^{t_1}$ follows because the images of equal sets $\mathscr{B}_{[t_0,t_2]}^{\mathcal{S}} = \mathscr{D}_{t_0}^{t_2}$ are equal. ■

3.7.2 Proof of Lemma 3.5

Proof. Define the subspaces $\mathscr{O}_{\text{I}}, \mathscr{O}_{\text{II}}, \mathscr{G} \subseteq \mathbb{R}^{(t_2-t_1+1)p}$ as

$$\mathscr{O}_{\text{I}} := \text{ColSpan}\big([\mathcal{O}_{\text{I}}]_{t_1}^{t_2}\big), \quad \mathscr{O}_{\text{II}} := \text{ColSpan}\big([\mathcal{O}_{\text{II}}]_{t_1}^{t_2}\big), \quad \mathscr{G} := \text{ColSpan}\big([\mathcal{G}_{\text{I}}]_{t_1}^{t_2} - [\mathcal{G}_{\text{II}}]_{t_1}^{t_2}\big)$$

If. From $\mathscr{O}_{\text{I}} \supseteq \mathscr{G}$, there exists a matrix $\mathcal{L} \in \mathbb{R}^{n\times n}$ such that

$$[\mathcal{G}_{\text{I}}]_{t_1}^{t_2} - [\mathcal{G}_{\text{II}}]_{t_1}^{t_2} = [\mathcal{O}_{\text{I}}]_{t_1}^{t_2}\mathcal{L}.$$

Given $\mathscr{O}_{\text{I}} = \mathscr{O}_{\text{II}}$ and the relation above, we have

$$\begin{aligned} \text{ColSpan}\begin{bmatrix} 0 & I \\ [\mathcal{O}_{\text{I}}]_{t_1}^{t_2} & [\mathcal{G}_{\text{I}}]_{t_1}^{t_2} \end{bmatrix} &= \text{ColSpan}\begin{bmatrix} 0 & I \\ [\mathcal{O}_{\text{I}}]_{t_1}^{t_2} & [\mathcal{G}_{\text{II}}]_{t_1}^{t_2} + [\mathcal{O}_{\text{I}}]_{t_1}^{t_2}\mathcal{L} \end{bmatrix} \\ &\overset{\substack{\text{by column}\\ \text{operation}}}{=\!=} \text{ColSpan}\begin{bmatrix} 0 & I \\ [\mathcal{O}_{\text{I}}]_{t_1}^{t_2} & [\mathcal{G}_{\text{II}}]_{t_1}^{t_2} \end{bmatrix} \overset{\text{via } \mathscr{O}_{\text{I}} = \mathscr{O}_{\text{II}}}{=\!=} \text{ColSpan}\begin{bmatrix} 0 & I \\ [\mathcal{O}_{\text{II}}]_{t_1}^{t_2} & [\mathcal{G}_{\text{II}}]_{t_1}^{t_2} \end{bmatrix} \end{aligned}$$

which implies $\mathscr{B}_{[t_1,t_2]}^{\mathcal{S}_{\text{I}}} = \mathscr{B}_{[t_1,t_2]}^{\mathcal{S}_{\text{II}}}$ via Lemma 3.3.

Only if. Define matrices $\mathcal{V}_{\text{I}}$ and $\mathcal{V}_{\text{II}}$,

$$\mathcal{V}_{\text{I}} := \begin{bmatrix} 0 & I \\ [\mathcal{O}_{\text{I}}]_{t_1}^{t_2} & [\mathcal{G}_{\text{I}}]_{t_1}^{t_2} \end{bmatrix}, \qquad \mathcal{V}_{\text{II}} := \begin{bmatrix} 0 & I \\ [\mathcal{O}_{\text{II}}]_{t_1}^{t_2} & [\mathcal{G}_{\text{II}}]_{t_1}^{t_2} \end{bmatrix}$$

and by Lemma 3.3 $\mathcal{V}_{\text{I}}$ and $\mathcal{V}_{\text{II}}$ have the same column span. As $\text{ColSpan}(\mathcal{V}_{\text{I}}) \supseteq \text{ColSpan}(\mathcal{V}_{\text{II}})$, there exist matrices $\mathcal{M} \in \mathbb{R}^{n\times n}$, $\mathcal{N} \in \mathbb{R}^{n\times mL}$, $\mathcal{P} \in \mathbb{R}^{mL\times n}$ and $\mathcal{Q} \in \mathbb{R}^{mL\times mL}$, where $L := t_2 - t_1 + 1$, such that

$$\begin{bmatrix} 0 & I \\ [\mathcal{O}_{\text{I}}]_{t_1}^{t_2} & [\mathcal{G}_{\text{II}}]_{t_1}^{t_2} \end{bmatrix} \begin{bmatrix} \mathcal{M} & \mathcal{N} \\ \mathcal{P} & \mathcal{Q} \end{bmatrix} = \begin{bmatrix} 0 & I \\ [\mathcal{O}_{\text{I}}]_{t_1}^{t_2} & [\mathcal{G}_{\text{II}}]_{t_1}^{t_2} \end{bmatrix}.$$

Compute the left-hand side above and compare the result to the right-hand side, and then we have $\mathcal{P} = 0$, $\mathcal{Q} = I$ and

$$[\mathcal{O}_{\mathrm{I}}]_{t_1}^{t_2}\mathcal{M} = [\mathcal{O}_{\mathrm{II}}]_{t_1}^{t_2}, \qquad [\mathcal{G}_{\mathrm{I}}]_{t_1}^{t_2} - [\mathcal{G}_{\mathrm{II}}]_{t_1}^{t_2} = -[\mathcal{O}_{\mathrm{I}}]_{t_1}^{t_2}\mathcal{N}.$$

Recall the definitions of $\mathscr{O}_{\mathrm{I}}, \mathscr{O}_{\mathrm{II}}, \mathscr{G}$. Thus, the above equations imply that $\mathscr{G}$ is a subspace of $\mathscr{O}_{\mathrm{I}}$, and $\mathscr{O}_{\mathrm{II}}$ is a subspace of $\mathscr{O}_{\mathrm{I}}$. Similarly, as $\mathrm{ColSpan}(\mathcal{V}_{\mathrm{I}}) \subseteq \mathrm{ColSpan}(\mathcal{V}_{\mathrm{II}})$, we have the converse result that $\mathscr{O}_{\mathrm{I}}$ is a subspace of $\mathscr{O}_{\mathrm{II}}$. Hence, the result $\mathscr{O}_{\mathrm{I}} = \mathscr{O}_{\mathrm{II}} \supseteq \mathscr{G}$ is obtained. ■

3.7.3 Proof of Lemma 3.11

Proof. With abuse of notation, we let

$$\begin{aligned}
\Phi_{\mathrm{p}} &:= \Phi_{t-L}^{t}, & \mathcal{C}_{\mathrm{p}} &:= \mathcal{C}_{t-L}^{t-1}, & & \\
\mathcal{O}_{\mathrm{p}} &:= \mathcal{O}_{t-L}^{t-1}, & \mathcal{O}_{\mathrm{f}} &:= \mathcal{O}_{t}^{t+N-1}, & \mathcal{O}_{\mathrm{pf}} &:= \mathcal{O}_{t-L}^{t+N-1}, \\
\mathcal{G}_{\mathrm{p}} &:= \mathcal{G}_{t-L}^{t-1}, & \mathcal{G}_{\mathrm{f}} &:= \mathcal{G}_{t}^{t+N-1}, & \mathcal{G}_{\mathrm{pf}} &:= \mathcal{G}_{t-L}^{t+N-1},
\end{aligned}$$

in this proof, where the subscript "p" stands for the past interval $[t-L, t)$, "f" for the future interval $[t, t+N)$, and "pf" their union. One can verify that

$$\mathcal{O}_{\mathrm{pf}} = \begin{bmatrix} \mathcal{O}_{\mathrm{p}} \\ \mathcal{O}_{\mathrm{f}}\Phi_{\mathrm{p}} \end{bmatrix}, \qquad \mathcal{G}_{\mathrm{pf}} = \begin{bmatrix} \mathcal{G}_{\mathrm{p}} & 0 \\ \mathcal{O}_{\mathrm{f}}\mathcal{C}_{\mathrm{p}} & \mathcal{G}_{\mathrm{f}} \end{bmatrix}, \tag{3.27}$$

by expanding $\Phi_{t_1}^{t_2}, \mathcal{C}_{t_1}^{t_2}, \mathcal{O}_{t_1}^{t_2}, \mathcal{G}_{t_1}^{t_2}$ into system matrices via (3.3a), (3.4a), (3.4b), (3.4c) respectively.

(i): Since $\mathrm{col}(u_{[t-L,t)}, y_{[t-L,t)}) \in \mathscr{B}_{[t-L,t)}^{\mathcal{S}}$, by definition there exists some initial state x_{t-L} such that $y_{[t-L,t)}$ is the output resulting from input $u_{[t-L,t)}$, with the resulting state x_t at time t. From (3.5a) and (3.5b) we have

$$x_t = \Phi_{\mathrm{p}} x_{t-L} + \mathcal{C}_{\mathrm{p}} u_{[t-L,t)}, \tag{3.28}$$

$$y_{[t-L,t)} = \mathcal{O}_{\mathrm{p}} x_{t-L} + \mathcal{G}_{\mathrm{p}} u_{[t-L,t)}. \tag{3.29}$$

Thus, via (3.5b) the resulting output with the initial state x_t and input $\bar{u}_{[t,t+N)}$ is

$$\bar{y}_{[t,t+N)} = \mathcal{O}_{\mathrm{f}} x_t + \mathcal{G}_{\mathrm{f}} \bar{u}_{[t,t+N)}, \tag{3.30}$$

which is an existing $\bar{y}_{[t,t+N)}$ that satisfies (3.11).

(ii): Substituting (3.28) into (3.30), we have

$$\bar{y}_{[t,t+N)} = \mathcal{O}_{\mathrm{f}}(\Phi_{\mathrm{p}} x_{t-L} + \mathcal{C}_{\mathrm{p}} u_{[t-L,t)}) + \mathcal{G}_{\mathrm{f}} \bar{u}_{[t,t+N)}. \tag{3.31}$$

To show the uniqueness of $\bar{y}_{[t,t+N)}$, it suffices to show that the term $\mathcal{O}_{\mathrm{f}}\Phi_{\mathrm{p}} x_{t-L}$ in (3.31) is unique, although x_{t-L} may not be unique. Since $L \geq \mathbf{l}(\mathcal{S}, t-L)$, it follows from the definition of lag in (3.9b) and the definitions of $\mathcal{O}_{\mathrm{p}}, \mathcal{O}_{\mathrm{pf}}$ that

$$\mathrm{rank}(\mathcal{O}_{\mathrm{p}}) = \mathrm{rank}(\mathcal{O}_{\mathrm{pf}}).$$

Due to the rank equality above and the segmentation $\mathcal{O}_{\mathrm{pf}} = \mathrm{col}(\mathcal{O}_{\mathrm{p}}, \mathcal{O}_{\mathrm{f}}\Phi_{\mathrm{p}})$ in (3.27), we conclude that the rows in $\mathcal{O}_{\mathrm{f}}\Phi_{\mathrm{p}}$ are linearly dependent to the rows in $\mathcal{O}_{\mathrm{p}}$, so there exists a matrix $\mathcal{M} \in \mathbb{R}^{pN \times pL}$ such that

$$\mathcal{O}_{\mathrm{f}}\Phi_{\mathrm{p}} = \mathcal{M}\mathcal{O}_{\mathrm{p}}. \tag{3.32}$$

Then, we obtain that

$$\begin{aligned}
\mathcal{O}_{\mathrm{f}}\Phi_{\mathrm{p}} x_{t-L} &\overset{\text{via (3.32)}}{=\!=} \mathcal{M}\mathcal{O}_{\mathrm{p}} x_{t-L} = \mathcal{M}\mathcal{O}_{\mathrm{p}}\mathcal{O}_{\mathrm{p}}{}^{\dagger}\mathcal{O}_{\mathrm{p}} x_{t-L} \\
&\overset{\text{via (3.32)}}{=\!=} \mathcal{O}_{\mathrm{f}}\Phi_{\mathrm{p}}\mathcal{O}_{\mathrm{p}}{}^{\dagger}\mathcal{O}_{\mathrm{p}} x_{t-L} \\
&\overset{\text{via (3.29)}}{=\!=} \mathcal{O}_{\mathrm{f}}\Phi_{\mathrm{p}}\mathcal{O}_{\mathrm{p}}{}^{\dagger}(y_{[t-L,t)} - \mathcal{G}_{\mathrm{p}} u_{[t-L,t)})
\end{aligned}$$

is unique, which implies uniqueness of $\bar{y}_{[t,t+N)}$ via (3.31).

(iii): It follows from (3.11) and (3.12) that

$$\begin{bmatrix} u_{[t-L,t)} \\ \bar{u}_{[t,t+N)} \\ y_{[t-L,t)} \\ \bar{y}_{[t,t+N)} \end{bmatrix} \in \mathrm{ColSpan} \begin{bmatrix} U_{\mathrm{p}} \\ U_{\mathrm{f}} \\ Y_{\mathrm{p}} \\ Y_{\mathrm{f}} \end{bmatrix}.$$

By definition of column span, there exists $g \in \mathbb{R}^h$ such that

$$\begin{bmatrix} U_{\mathrm{p}} \\ U_{\mathrm{f}} \\ Y_{\mathrm{p}} \\ Y_{\mathrm{f}} \end{bmatrix} g = \begin{bmatrix} u_{[t-L,t)} \\ u^*_{[t,t+N)} \\ y_{[t-L,t)} \\ y^*_{[t,t+N)} \end{bmatrix}. \tag{3.33}$$

Now, we show that Y_f can be written as $\mathcal{N}[U_\mathrm{p}; U_\mathrm{f}; Y_\mathrm{p}]$ with some matrix $\mathcal{N} \in \mathbb{R}^{pN\times(mL+mN+pL)}$. From Lemma 3.3,

$$\begin{aligned}
\mathscr{B}^{\mathcal{S}}_{[t-L,t+N)} &= \mathrm{ColSpan}\left[\begin{array}{c|c} 0 & I \\ \hline \mathcal{O}_\mathrm{pf} & \mathcal{G}_\mathrm{pf} \end{array}\right] \\
&\overset{\text{via (3.27)}}{=\!=} \mathrm{ColSpan}\left[\begin{array}{c|cc} 0 & I & 0 \\ 0 & 0 & I \\ \hline \mathcal{O}_\mathrm{p} & \mathcal{G}_\mathrm{p} & 0 \\ \mathcal{O}_\mathrm{f}\Phi_\mathrm{p} & \mathcal{O}_\mathrm{f}\mathcal{C}_\mathrm{p} & \mathcal{G}_\mathrm{f} \end{array}\right]
\end{aligned}$$

Notice that the above column span and the column span in (3.12) are both equal to $\mathscr{B}^{\mathcal{S}}_{[t-L,t+N)}$. Hence, there exist matrices $\mathcal{P} \in \mathbb{R}^{n\times h}, \mathcal{Q} \in \mathbb{R}^{mL\times h}, \mathcal{R} \in \mathbb{R}^{mN\times h}$ such that

$$\begin{bmatrix} U_\mathrm{p} \\ U_\mathrm{f} \\ Y_\mathrm{p} \\ Y_\mathrm{f} \end{bmatrix} = \begin{bmatrix} 0 & I & 0 \\ 0 & 0 & I \\ \mathcal{O}_\mathrm{p} & \mathcal{G}_\mathrm{p} & 0 \\ \mathcal{O}_\mathrm{f}\Phi_\mathrm{p} & \mathcal{O}_\mathrm{f}\mathcal{C}_\mathrm{p} & \mathcal{G}_\mathrm{f} \end{bmatrix} \begin{bmatrix} \mathcal{P} \\ \mathcal{Q} \\ \mathcal{R} \end{bmatrix}.$$

Computing the right-hand side above and then comparing the result to the left-hand side, we have $\mathcal{Q} = U_\mathrm{p}$, $\mathcal{R} = U_\mathrm{f}$, and

$$Y_\mathrm{p} = \mathcal{O}_\mathrm{p}\mathcal{P} + \mathcal{G}_\mathrm{p}U_\mathrm{p}, \tag{3.34}$$
$$Y_\mathrm{f} = \mathcal{O}_\mathrm{f}\Phi_\mathrm{p}\mathcal{P} + \mathcal{O}_\mathrm{f}\mathcal{C}_\mathrm{p}U_\mathrm{p} + \mathcal{G}_\mathrm{f}U_\mathrm{f}. \tag{3.35}$$

Therefore, we can represent Y_f into the form $\mathcal{N}\,\mathrm{col}(U_\mathrm{p}, U_\mathrm{f}, Y_\mathrm{p})$,

$$\begin{aligned}
Y_\mathrm{f} &\overset{\text{via (3.35)}}{=\!=} \mathcal{O}_\mathrm{f}\Phi_\mathrm{p}\mathcal{P} + \mathcal{O}_\mathrm{f}\mathcal{C}_\mathrm{p}U_\mathrm{p} + \mathcal{G}_\mathrm{f}U_\mathrm{f} \\
&\overset{\text{via (3.32)}}{=\!=} \mathcal{M}\mathcal{O}_\mathrm{p}\mathcal{P} + \mathcal{O}_\mathrm{f}\mathcal{C}_\mathrm{p}U_\mathrm{p} + \mathcal{G}_\mathrm{f}U_\mathrm{f} \\
&\overset{\text{via (3.34)}}{=\!=} \mathcal{M}(Y_\mathrm{p} - \mathcal{G}_\mathrm{p}U_\mathrm{p}) + \mathcal{O}_\mathrm{f}\mathcal{C}_\mathrm{p}U_\mathrm{p} + \mathcal{G}_\mathrm{f}U_\mathrm{f} \\
&= \mathcal{N}\,\mathrm{col}(U_\mathrm{p}, U_\mathrm{f}, Y_\mathrm{p})
\end{aligned} \tag{3.36}$$

with matrix $\mathcal{N} := \mathrm{col}(\mathcal{O}_\mathrm{f}\mathcal{C}_\mathrm{p} - \mathcal{M}\mathcal{G}_\mathrm{p}, \mathcal{G}_\mathrm{f}, \mathcal{M})$. The result (3.13) then follows:

$$\begin{aligned}
\bar{y}_{[t,t+N)} &\overset{\text{via (3.33)}}{=\!=} Y_\mathrm{f}g \overset{\text{via (3.36)}}{=\!=} \mathcal{N}\mathcal{H}g = \mathcal{N}\mathcal{H}\mathcal{H}^\dagger\mathcal{H}g \\
&\overset{\text{via (3.36)}}{=\!=} Y_\mathrm{f}\mathcal{H}^\dagger\mathcal{H}g \overset{\text{via (3.33)}}{=\!=} Y_\mathrm{f}\mathcal{H}^\dagger \begin{bmatrix} u_{[t-L,t)} \\ \bar{u}_{[t,t+N)} \\ y_{[t-L,t)} \end{bmatrix},
\end{aligned}$$

where we let $\mathcal{H}$ denote $\mathrm{col}(U_\mathrm{p}, U_\mathrm{f}, Y_\mathrm{p})$. ■

3.7.4 Proof of Lemma 3.18

Proof. We first prove the case when K is a multiple of T, i.e., where $K = K_1 T$ for some $K_1 \in \mathbb{N}$. Let $w_{[t_1,t_2]} := \begin{bmatrix} u_{[t_1,t_2]} \\ y_{[t_1,t_2]} \end{bmatrix}$ and $\mathbf{w}_{[\tau_1,\tau_2]} := \begin{bmatrix} \mathbf{u}_{[\tau_1,\tau_2]} \\ \mathbf{y}_{[\tau_1,\tau_2]} \end{bmatrix}$ denote a trajectory of the system $\mathcal{S}$ and the lifted system $\mathcal{S}_{\mathsf{L}}(t_0)$, respectively. From Lemma 3.12, we know that $\mathscr{B}^{\mathcal{S}_{\mathsf{L}}(t_1)}_{[0,s)} = \mathscr{B}^{\mathcal{S}}_{[t_1,t_1+sT)}$ for all $s \in \mathbb{N}$, so we can establish such a trajectory $\mathbf{w}^{\mathrm{d}}_{[0,P)} \in \mathscr{B}^{\mathcal{S}_{\mathsf{L}}(t_1)}_{[0,P)}$ that

$$\mathbf{w}^{\mathrm{d}}_{[0,P)} := w^{\mathrm{d}}_{[t_1,t_1+PT)}$$

with $P := \lfloor (t_2 - t_1 + 1)/T \rfloor$, i.e., P is the number of whole periods in the interval $[t_1, t_2]$. With abuse of notation, we let $\mathbf{n} := \mathbf{n}(\mathcal{S}, t_1)$ and $\mathbf{n}_{\mathsf{L}} := \mathbf{n}(\mathcal{S}_{\mathsf{L}}(t_1))$ in this proof, then we have $\mathbf{n} = \mathbf{n}_{\mathsf{L}}$ via Lemma 3.14(i). By direct substitution, one can verify that

$$\mathscr{H}_{K_1+\mathbf{n}_{\mathsf{L}}}(\mathbf{u}^{\mathrm{d}}_{[0,P)}) = \mathscr{H}^T_{(K_1+\mathbf{n})T}(u^{\mathrm{d}}_{[t_1,t_2]}), \tag{3.37}$$

$$\mathscr{H}_{K_1}(\mathbf{w}^{\mathrm{d}}_{[0,P)}) = \mathscr{H}^T_{K}(w^{\mathrm{d}}_{[t_1,t_2]}). \tag{3.38}$$

(Note that in (3.37), the block rows on the left-hand side are of size mT, where m is the input dimension of $\mathcal{S}$, while on the right-hand side, the block rows are of size m. Similarly for (3.38).) Since $u^{\mathrm{d}}_{[t_1,t_2]}$ is T-p.p.e. of order $(K_1 + \mathbf{n})T$ (i.e., the right-hand side of (3.37) has full row rank), we know that $\mathbf{u}^{\mathrm{d}}_{[0,P)}$ is persistently exciting of order $K_1 + \mathbf{n}_{\mathsf{L}}$ (as the left-hand side of (3.37) has full row rank). We also know via Lemma 3.15 that $\mathcal{S}_{\mathsf{L}}(t_1)$ is controllable because $\mathcal{S}$ is controllable. Thus by Lemma 2.9 we have

$$\mathrm{ColSpan}(\mathscr{H}_{K_1}(\mathbf{w}^{\mathrm{d}}_{[0,P)})) = \mathscr{B}^{\mathcal{S}_{\mathsf{L}}(t_1)}_{[0,K_1)}. \tag{3.39}$$

Substitute (3.38) into the left-hand side of (3.39), and replace the right-hand side of (3.39) using $\mathscr{B}^{\mathcal{S}_{\mathsf{L}}(t_1)}_{[0,K_1)} = \mathscr{B}^{\mathcal{S}}_{[t_1,t_1+K)}$ via Lemma 3.12, and then we obtain the result.

Next, we show the result for all $K \in \mathbb{N}$. Let $K_1 := \lceil K/T \rceil$ and $\widehat{K} := K_1 T$, i.e., $\widehat{K}$ is the smallest multiple of T greater than or equal to K. Since $\lceil K/T \rceil = \lceil \widehat{K}/T \rceil$, $u^{\mathrm{d}}_{[t_1,t_2]}$ is T-p.p.e. of order $(\lceil \widehat{K}/T \rceil + \mathbf{n}(\mathcal{S}, t_1))T$, we have the condition (ii) of this lemma for the case $K \leftarrow \widehat{K}$ as a multiple of T, which case we have already proved. We therefore have

$$\mathrm{ColSpan}(\mathscr{H}^T_{\widehat{K}}(w^{\mathrm{d}}_{[t_1,t_2]})) = \mathscr{B}^{\mathcal{S}}_{[t_1,t_1+\widehat{K})}. \tag{3.40}$$

Define $\mathscr{H}_{\triangle} := \mathscr{H}^T_K(w^{\mathrm{d}}_{[t_1,t_2-(\widehat{K}-K)]})$. One can verify that $\mathscr{H}^T_K(u^{\mathrm{d}}_{[t_1,t_2-(\widehat{K}-K)]})$ consists of the first K block rows of $\mathscr{H}^T_{\widehat{K}}(u^{\mathrm{d}}_{[t_1,t_2]})$, and similarly for y^{d}. Hence, applying Lemma 3.4 for (3.40) we have

$$\mathrm{ColSpan}(\mathscr{H}_{\triangle}) = \mathscr{B}^{\mathcal{S}}_{[t_1,t_1+K)}. \tag{3.41}$$

However, note that

$$\mathrm{ColSpan}(\mathscr{H}_{\triangle}) \subseteq \mathrm{ColSpan}(\mathscr{H}_K^T(w^{\mathrm{d}}_{[t_1,t_2]})) \subseteq \mathscr{B}^{\mathcal{S}}_{[t_1,t_1+K)}, \tag{3.42}$$

where the first inclusion ($\subseteq$) above is because $\mathscr{H}_{\triangle}$ is a sub-matrix of $\mathscr{H}_K^T(w^{\mathrm{d}}_{[t_1,t_2]})$ with the same column size, and the second inclusion above is because each column of $\mathscr{H}_K^T(w^{\mathrm{d}}_{[t_1,t_2]})$ is a vector in the behavior set $\mathscr{B}^{\mathcal{S}}_{[t_1,t_1+K)}$. The result now follows by combining (3.41) and (3.42). ■

3.7.5 Proof of Proposition 3.20

Proof. **Equivalence of Optimal Sets.** We first show that the problems (P-DeePC), (P-SPC) and (LTV-MPC) have the same set of optimal trajectories $\bar{u}_{[t,t+N)}, \bar{y}_{[k,k+N)}$. Since the three problems have the same cost function (2.8) and a common constraint (2.7), it suffices to show the rest constraints, i.e. the following statements, are equivalent:

a) $\bar{u}_{[k,k+N)}$ and $\bar{y}_{[k,k+N)}$ satisfy (3.23) for some $g \in \mathbb{R}^h$;
b) $\bar{u}_{[k,k+N)}$ and $\bar{y}_{[k,k+N)}$ satisfy (3.24);
c) $\bar{u}_{[k,k+N)}$ and $\bar{y}_{[k,k+N)}$ satisfy (3.25) for some $\bar{x}_{[t,t+N]}$.

With assumptions (i) and (ii), we obtain (3.20) as discussed in Section 3.4.1, where we used Lemma 3.19 and the definition of the proper index $\Theta(t)$. It follows from assumption (iii), i.e., $\widehat{\Theta}(k) = \Theta(k)$ and (3.20) that

$$\mathrm{ColSpan}\begin{bmatrix} U_{\mathrm{p}}(\widehat{\Theta}(k)) \\ U_{\mathrm{f}}(\widehat{\Theta}(k)) \\ Y_{\mathrm{p}}(\widehat{\Theta}(k)) \\ Y_{\mathrm{f}}(\widehat{\Theta}(k)) \end{bmatrix} = \mathscr{B}^{\mathcal{S}}_{[k-L,k+N)}. \tag{3.43}$$

For showing the equivalence of a), b) and c), we introduce an auxiliary statement:

d) $\bar{u}_{[k,k+N)}$ and $\bar{y}_{[k,k+N)}$ satisfy (3.11).

a) $\iff$ d): By definition of column span, a) is same as

$$\begin{bmatrix} u_{[k-L,k)} \\ \bar{u}_{[k,k+N)} \\ y_{[k-L,k)} \\ \bar{y}_{[k,k+N)} \end{bmatrix} \in \mathrm{ColSpan}\begin{bmatrix} U_{\mathrm{p}}(\widehat{\Theta}(k)) \\ U_{\mathrm{f}}(\widehat{\Theta}(k)) \\ Y_{\mathrm{p}}(\widehat{\Theta}(k)) \\ Y_{\mathrm{f}}(\widehat{\Theta}(k)) \end{bmatrix}.$$

From (3.43), the expression above is equivalent to d).

b) $\iff$ d): Given $L \geq \mathbf{l}(\mathcal{S}, k-L)$ and (3.43), via Lemma 3.11(iii), for each $\bar{u}_{[k,k+N)}$ (3.24) specifies the unique $\bar{y}_{[k,k+N)}$ that satisfies (3.11). Therefore, b) is equivalent to d).

c) $\implies$ d): Since $\bar{u}_{[k,k+N)}, \bar{y}_{[k,k+N)}$ is a trajectory with initial state x_k via (3.25) and $u_{[k-L,k)}, y_{[k-L,k)}$ is a trajectory with final state x_k, the two trajectories can be connected into a single trajectory, i.e. (3.11) holds.

d) $\implies$ c): Define $y^*_{[k,k+N)}$ the unique output resulting from the initial state x_t and input $u^*_{[t,t+N)}$. Since $u_{[k-L,k)}, u_{[k-L,k)}$ is a trajectory with final state x_k and $\bar{u}_{[k,k+N)}, y^*_{[k,k+N)}]$ is a trajectory with initial state x_k, their connection is also a trajectory and satisfies the following.

$$\mathrm{col}(u_{[k-L,k)}, \bar{u}_{[k,k+N)}, y_{[k-L,k)}, y^*_{[t,t+N)}) \in \mathscr{B}^{\mathcal{S}}_{[k-L,k+N)}$$

Comparing (3.11) to the above, due to the uniqueness in Lemma 3.11(ii) (where we used (i) of this proposition), we conclude that $\bar{y}_{[k,k+N)} = y^*_{[k,k+N)}$. Hence, $\bar{y}_{[k,k+N)}$ is the output resulting from the initial state x_t and input $\bar{u}_{[k,k+N)}$, i.e., (3.25) holds.

Uniqueness of Future Trajectory. Finally, we show that the optimal trajectory $\bar{u}_{[k,k+N)}, \bar{y}_{[k,k+N)}$ of each problem is unique. (LTV-MPC) has a unique optimal solution $\bar{u}_{[k,k+N)}, \bar{y}_{[k,k+N)}$, because $\bar{x}_{[k,k+N]}$ and $\bar{y}_{[k,k+N)}$ are both dependent on $\bar{u}_{[k,k+N)}$ via (3.25) and hence the cost function (2.8) with $R \succ 0$ is strictly convex of the only independent variable $\bar{u}_{[k,k+N)}$. Following from the equivalence of the optimal sets, the optimal trajectories $\bar{u}_{[k,k+N)}, \bar{y}_{[k,k+N)}$ of (P-DeePC) and (P-SPC) are also unique. This completes the proof. ■

Chapter 4

Stochastic Data-Driven Predictive Control

In this chapter, we present our contributions to DDPC methods for stochastic systems, as outlined in our works [78, 79]. Section 4.1 outlines the stochastic control problem, considering both the Gaussian distribution setup and the distributionally robust setup. In Section 4.2, we examine a Stochastic MPC framework that integrates several established techniques from the literature. Our proposed Stochastic DDPC method is presented in Section 4.3, where we prove that it generates control inputs equivalent to those obtained using the investigated Stochastic MPC approach, under several tuning conditions and the assumption that offline data is noise-free. Simulation results in Section 4.4 validate our data-driven control method, showcasing improved performance compared to benchmark control methods.

4.1 Problem Statement: Stochastic LTI Case

We consider a stochastic linear time-invariant (LTI) system

$$x_{t+1} = Ax_t + Bu_t + w_t, \tag{4.1a}$$

$$y_t = Cx_t + Du_t + v_t, \tag{4.1b}$$

with input $u_t \in \mathbb{R}^m$, state $x_t \in \mathbb{R}^n$, output $y_t \in \mathbb{R}^p$, process noise $w_t \in \mathbb{R}^n$, and measurement noise $v_t \in \mathbb{R}^p$. The initial state x_0 is uncertain with a given mean $\mu^{\mathsf{x}}_{\mathsf{ini}}$ and with a

variance set to the steady-state value determined by a Kalman filter, which will be introduced in Section 4.2.1. The system matrices A, B, C, D are *unknown* and the state x_t is *unmeasured*; we have access only to the input u_t and output y_t in (4.1). The assumptions regarding the disturbances w_t and v_t will be detailed separately in Section 4.1.1 and Section 4.1.2, where we investigate two different setups of the stochastic system (4.1). We assume the system (A, B, C, D) is a minimal realization (i.e., controllable and observable), where controllability is required for control purposes and observability is assumed without loss of generality for an unknown system [23, Sec. 2.4]. Let $L \in \mathbb{N}$ be such that the extended observability matrix $\mathcal{O} := \mathrm{col}(C, CA, \ldots, CA^{L-1})$ has full column rank; such smallest L is the *lag* of the system [23, 24]. Finally, we assume the pair (A, Σ^{w}) is stabilizable, which implies that $(A, (\Sigma^{\mathsf{w}})^{1/2})$ is stabilizable and will subsequently ensure uniqueness of the state variance by the Kalman filter [94].

In a reference tracking problem, the objective is for the output y_t to follow a specified reference signal $r_t \in \mathbb{R}^p$. The trade-off between tracking error and control effort may be encoded in the cost (2.2), reproduced as

$$J_t(u_t, y_t) := \|y_t - r_t\|_Q^2 + \|u_t\|_R^2, \tag{2.2}$$

to be minimized over a horizon, where $Q \in \mathbb{S}^p_{++}$ and $R \in \mathbb{S}^m_{++}$ are user-selected parameters. This tracking should be achieved subject to constraints on the inputs and outputs. We consider a polytopic constraint in the form

$$E \begin{bmatrix} u_t \\ y_t \end{bmatrix} \leq f \tag{4.2}$$

for $t \in \mathbb{N}_{\geq 0}$, where $E \in \mathbb{R}^{q \times (m+p)}$ is a fixed matrix, $f \in \mathbb{R}^q$ is a fixed vector, with some $q \in \mathbb{N}$. In the stochastic setting, the deterministic constraint (4.2) can be adapted to probabilistic constraints, as we will again discuss separately in Section 4.1.1 and Section 4.1.2, depending on the different setups of disturbances w_t and v_t.

Remark 4.1 (**Output Constraints and Output Tracking**). State constraints and costs are commonly considered in MPC and SMPC methods [5, 10, 11, 12], being used to enforce safety conditions and quantify control performance, respectively. Our problem setup focuses on output control, with the internal state being unknown and unrealized. For this reason, we instead considered input-output constraint (4.4) for safety conditions and output-tracking cost (2.2) for performance evaluation, which are both common in DDPC methods such as [38]. ■

4.1.1 Gaussian Distribution Case

We first describe the problem formulation where all random variables are normally distributed. The disturbances w_t and v_t in (4.1) are independent of each other and of x_0, and are independently and identically distributed (i.i.d.) normally with zero mean and with variances $\Sigma^{\mathsf{w}} \in \mathbb{S}^n_+$ and $\Sigma^{\mathsf{v}} \in \mathbb{S}^p_{++}$ respectively, i.e.,

$$w_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0_{n\times 1}, \Sigma^{\mathsf{w}}), \quad v_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0_{p\times 1}, \Sigma^{\mathsf{v}}). \tag{4.3}$$

The deterministic constraint (4.2) is modelled in the stochastic setting as a probabilistic *chance constraint*

$$\mathbb{P}\Big\{E \begin{bmatrix} u_t \\ y_t \end{bmatrix} \leq f\Big\} \geq 1 - p \tag{4.4}$$

for $t \in \mathbb{N}_{\geq 0}$, where $E \in \mathbb{R}^{q\times(m+p)}$ is a fixed matrix, $f \in \mathbb{R}^q$ is a fixed vector, with some $q \in \mathbb{N}$, and $p \in (0,1)$ is a probability bound of constraint violation. One can similarly impose multiple chance constraints, e.g., separate input and output chance constraints, in the form of (4.4).

4.1.2 Distributionally Robust Case

As an alternative problem setup, we consider the case where the random variables have unknown distributions. The probability distributions of w_t and v_t are *unknown*, but we assume that w_t and v_t have zero mean and zero auto-correlation (white noise), are uncorrelated, and their variances $\Sigma^{\mathsf{w}} \in \mathbb{S}^n_+$ and $\Sigma^{\mathsf{v}} \in \mathbb{S}^p_{++}$ are known. The initial state x_0 has given mean $\mu^{\mathsf{x}}_{\mathsf{ini}}$ and variance Σ^{x} and is uncorrelated with the noise. We record these conditions as

$$\mathbb{E}\left[\begin{bmatrix} w_t \\ v_t \end{bmatrix}\right] = 0, \quad \mathbb{E}\left[\begin{bmatrix} w_t \\ v_t \end{bmatrix}\begin{bmatrix} w_s \\ v_s \end{bmatrix}^{\mathsf{T}}\right] = \begin{bmatrix} \delta_{ts}\Sigma^{\mathsf{w}} & 0 \\ 0 & \delta_{ts}\Sigma^{\mathsf{v}} \end{bmatrix}, \tag{4.5}$$

$$\mathbb{E}[x_0] = \mu^{\mathsf{x}}_{\mathsf{ini}}, \quad \mathrm{Var}[x_0] = \Sigma^{\mathsf{x}}, \quad \mathbb{E}\left[x_0 \begin{bmatrix} w_t \\ v_t \end{bmatrix}^{\mathsf{T}}\right] = 0, \tag{4.6}$$

with δ_{ts} the Kronecker delta.

We can equivalently express the constraints (4.2) as the single constraint $h(u_t, y_t) \leq 0$, where

$$h(u_t, y_t) := \max_{i\in\{1,\ldots,q\}} \; e_i^{\mathsf{T}} \begin{bmatrix} u_t \\ y_t \end{bmatrix} - f_i, \tag{4.7}$$

with $e_i \in \mathbb{R}^{m+p}$ the transposed i-th row of E and $f_i \in \mathbb{R}$ the i-th entry of f. For the system (4.1) which is subject to (possibly unbounded) stochastic disturbances, the deterministic constraint $h(u_t, y_t) \leq 0$ must be relaxed. Beyond a traditional chance constraint $\mathbb{P}[h(u_t, y_t) \leq 0] \geq 1 - \alpha$ with a violation probability $\alpha \in (0, 1)$, a *conditional value-at-risk (CVaR)* constraint is more conservative; the CVaR at level α of $h(u_t, y_t)$ is defined as the expected value of $h(u_t, y_t)$ in the $\alpha\cdot$ 100% worst cases, and takes extreme violations into account. With the noise distributions unknown, we must further guarantee satisfaction of the CVaR constraint for all possible distributions under consideration. Let $\mathbb{D}$ denote a joint distribution of all random variables in (4.1) satisfying (4.5) and (4.6), and let the *ambiguity set* $\mathcal{D}$ be the set of all such distributions. The *distributionally robust CVaR (DR-CVaR)* constraint [95, 96] is then

$$\sup_{\mathbb{D}\in\mathcal{D}} \mathbb{D}\text{-CVaR}_\alpha[h(u_t, y_t)] \leq 0, \tag{4.8}$$

where $\mathbb{D}$-CVaR$_\alpha[z]$ is the CVaR value of a random variable $z \in \mathbb{R}$ at level α given distribution $\mathbb{D}$.

4.1.3 Our Objective: An Equivalent Data-Driven Method

In a model-based setting where A, B, C, D are known, the general control problem above can be addressed by SMPC, as will be reviewed in Section 4.2. Our broad objective is to construct a direct data-driven method that addresses the same stochastic control problem and is equivalent, under certain tuning conditions, to SMPC.

In direct data-driven control methods such as DeePC and SPC for deterministic systems, a sufficiently long and sufficiently rich set of noise-free input-output data is collected. Under technical conditions, this data provides an equivalent representation of the underlying system dynamics, and is used to replace the parametric model in predictive control schemes, yielding control algorithms which are *equivalent* to model-based predictive control [38, 15]. Motivated by this equivalence, our goal here is to develop a direct data-driven control method that produces the same input-state-output sequences as produced by the SMPC method reviewed in Section 4.2 when applied to the same system (4.1) with same initial condition x_0 and same realizations of process and sensor noises w_t, v_t. Put simply, we seek a direct data-driven counterpart to SMPC.

As in the described cases of equivalence for DeePC and SPC, we will subsequently show equivalence of our data-driven method to SMPC *in the idealized case where we have access to noise-free offline data*. While this may initially seem peculiar in an explicitly stochastic control setting, we view this as the most reasonable theoretical result to aim for, given

that the prediction model must be replaced using only a finite amount of recorded data. Moreover, remark that (i) noisy offline data can be accommodated in a robust fashion through the use of regularized least-squares (Section 4.3.1), as supported by simulation results in Section 4.4, and (ii) our stochastic control approach will fully take into account process and sensor noise during the online execution of the control process.

4.2 Stochastic Model Predictive Control

Several formulations of SMPC methods have been developed in the literature [10, Table 2]. Our focus is on output-feedback SMPC [97, 98, 99, 100, 101], which is typically approached by enforcing a separation principle within the design, augmenting full-state-feedback SMPC with state estimation. Our formulation here is based on an affine feedback-policy parameterization, following e.g., [98, 99], with the modifications that we consider output tracking and output constraints, as opposed to state objectives. The SMPC method under consideration here also integrates interpolation of initial condition [102, 103], which is required for recursive feasibility with unbounded noise, and approximation of chance constraints [104], which leads to a tractable optimization problem. In this section, we assume the exact system model (A, B, C, D) in (4.1) is given, while in practice, such as the simulations in Section 4.4, an identified model should be used instead.

4.2.1 Initial Condition and State Estimation

SMPC follows a receding-horizon strategy and makes decisions for N upcoming steps at each *control step*. At control step $t = k$, the initial condition of the state x_k is modeled in the Gaussian distribution setting (Section 4.1.1) as

$$x_k \sim \mathcal{N}(\mu_k^{\mathsf{x}}, \Sigma^{\mathsf{x}}), \tag{4.9}$$

or alternatively modeled in the distributionally robust setting (Section 4.1.2) as

$$\mathbb{E}[x_k] = \mu_k^{\mathsf{x}}, \qquad \mathrm{Var}[x_k] = \Sigma^{\mathsf{x}}, \tag{4.10}$$

where the mean $\mu_k^{\mathsf{x}} \in \mathbb{R}^n$ depends on a decision variable $\theta \in [0, 1]$, according to an interpolation technique to be introduced in Section 4.2.2. The state variance $\Sigma^{\mathsf{x}} \in \mathbb{S}_+^n$ in (4.9) and (4.10) is fixed and induced by the steady-state Kalman filter. Specifically, Σ^{x} is the *unique*

positive semidefinite solution to the associated discrete-time algebraic Riccatti equation (DARE) [94]

$$\Sigma^{\mathsf{x}} = (A - L_{\mathsf{L}} C)\Sigma^{\mathsf{x}} A^{\mathsf{T}} + \Sigma^{\mathsf{w}} \tag{4.11a}$$

$$L_{\mathsf{L}} := A L_{\mathsf{K}}, \quad L_{\mathsf{K}} := \Sigma^{\mathsf{x}} C^{\mathsf{T}} (C \Sigma^{\mathsf{x}} C^{\mathsf{T}} + \Sigma^{\mathsf{v}})^{-1} \tag{4.11b}$$

given detectable (A, C) and stabilizable (A, Σ^{w}), where we let $L_{\mathsf{K}} \in \mathbb{R}^{n \times p}$ denote the steady-state Kalman gain and $L_{\mathsf{L}} \in \mathbb{R}^{n \times p}$ the associated Luenberger observer gain.

With the initial condition (4.9) or (4.10), we simulate the noise-free model for future N time steps,

$$\bar{x}_{t+1} := A\bar{x}_t + B\bar{u}_t, \qquad t \in \mathbb{Z}_{[k,k+N)} \tag{4.12a}$$

$$\bar{y}_t := C\bar{x}_t + D\bar{u}_t, \qquad t \in \mathbb{Z}_{[k,k+N)} \tag{4.12b}$$

$$\bar{x}_k := \mu_k^{\mathsf{x}} \tag{4.12c}$$

where the *nominal inputs* $\bar{u}_t \in \mathbb{R}^m$ for $t \in \mathbb{Z}_{[k,k+N)}$ will be decision variables in optimization, with resulting *nominal states* $\bar{x}_t \in \mathbb{R}^n$ and *nominal outputs* $\bar{y}_t \in \mathbb{R}^p$.

After the reveal of future measurements, estimates of the future states over the desired horizon will be computed through the steady-state Kalman filter, with L_{K} in (4.11b),

$$\nu_t := y_t - C\hat{x}_t^- - Du_t, \qquad t \in \mathbb{Z}_{[k,k+N)} \tag{4.13a}$$

$$\hat{x}_t^+ := \hat{x}_t^- + L_{\mathsf{K}}\nu_t, \qquad t \in \mathbb{Z}_{[k,k+N)} \tag{4.13b}$$

$$\hat{x}_{t+1}^- := A\hat{x}_t^+ + Bu_t, \qquad t \in \mathbb{Z}_{[k,k+N)} \tag{4.13c}$$

$$\hat{x}_k^- := \mu_k^{\mathsf{x}} \tag{4.13d}$$

where $\hat{x}_t^+$ and $\hat{x}_t^-$ denote the posterior and prior estimates of x_t, respectively, and $\nu_t \in \mathbb{R}^p$ is the *innovation*. The steady-state Kalman filter (4.13) is equivalent to a Luenberger observer as in [98, 97] with observer gain L_{L} in (4.11b), and is the stationary case of time-varying Kalman filters used in [99, 100, 101].

4.2.2 Interpolation of Initial Condition

A common choice of μ_k^{x} in (4.9) and (4.10) is the prior state estimate $\hat{x}_k^-$ produced by the estimator (4.13) in the previous control step [100, 99, 101]; we denote this choice by $\mu_k^{\hat{\mathsf{x}}}$. However, in our setting the state estimates are normally distributed and thus unbounded. The choice $\mu_k^{\mathsf{x}} = \mu_k^{\hat{\mathsf{x}}}$ may lead to an extreme value of μ_k^{x}, which in turn could render the

constraint (4.4) infeasible. A different choice of μ_k^{x} is the deterministic prediction $\bar{x}_k$ of state the x_k, obtained via (4.12) at the last control step [98]; we denote this choice by $\mu_k^{\bar{\mathsf{x}}}$. Choosing $\mu_k^{\mathsf{x}} = \mu_k^{\bar{\mathsf{x}}}$ can guarantee feasibility, with proper design of the control optimization problem; however, the value $\mu_k^{\bar{\mathsf{x}}}$ does not incorporate feedback from past measurements.

Trading off the two options, we let the initial condition μ_k^{x} in (4.9) and (4.10) *interpolate* between $\mu_k^{\hat{\mathsf{x}}}$ and $\mu_k^{\bar{\mathsf{x}}}$ [102, 103] as

$$\mu_k^{\mathsf{x}} := (1-\theta)\,\mu_k^{\hat{\mathsf{x}}} + \theta\,\mu_k^{\bar{\mathsf{x}}}, \tag{4.14}$$

where $\theta \in [0,1]$ is a decision variable, and both $\mu_k^{\hat{\mathsf{x}}}, \mu_k^{\bar{\mathsf{x}}} \in \mathbb{R}^n$ are fixed and known at time $t = k$. At initial control step $k = 0$, μ_0^{x} is equal to a given parameter $\mu_{\mathsf{ini}}^{\mathsf{x}}$, i.e., we let $\mu_0^{\hat{\mathsf{x}}} := \mu_{\mathsf{ini}}^{\mathsf{x}}$ and $\mu_0^{\bar{\mathsf{x}}} := \mu_{\mathsf{ini}}^{\mathsf{x}}$.

4.2.3 Feedback Control Policies: Output Feedback

Stochastic state-feedback control requires the determination of (causal) feedback policies π_t which map the observation history into control actions. As the space of policies is an infinite-dimensional function space, a simple affine feedback parameterization is typically used in SMPC to obtain a tractable finite-dimensional optimization problem, written as (cf. [97, 98, 99])

$$u_t = \pi_t(\hat{x}_t^-) := \bar{u}_t - K(\hat{x}_t^- - \bar{x}_t), \tag{4.15}$$

where $K \in \mathbb{R}^{m\times n}$ is a *fixed* feedback gain such that $A - BK$ is Schur stable. Through the policy (4.15), the control action u_t depends both on the decision $\bar{u}_t$ optimized at the control step, and on the state estimate $\hat{x}_t^-$ via (4.13) which is decided after the measurement of $y_{[k,t)}$ and embodies feedback from the measurements. Based on the cost (2.2), we select the gain matrix K in (4.15) as the infinite-horizon LQR gain of system (4.1) with LQR stage cost $\|Cx_t + Du_t\|_Q^2 + \|u_t\|_R^2$ (i.e., with state weight $C^\mathsf{T}QC$, input weight $R + D^\mathsf{T}QD$ and cross weight $C^\mathsf{T}QD$),

$$K := (R + B^\mathsf{T}PB + D^\mathsf{T}QD)^{-1}(B^\mathsf{T}PA + D^\mathsf{T}QC) \tag{4.16}$$

where $P \in \mathbb{S}_+^n$ is the *unique* positive semidefinite solution to the discrete-time algebraic Riccati equation (DARE) [94]

$$P = A^\mathsf{T}P(A - BK) + C^\mathsf{T}Q(C - DK), \tag{4.17}$$

given stabilizable (A, B), detectable (A, C) and $Q \succ 0$. We remark that an equivalent form $\pi_t(\hat{x}_t^-) := c_t - K\hat{x}_t^-$ of (4.15) with decision variable c_t has been used in [97] and in many SMPC examples surveyed in [10]. A time-varying-gain version of (4.15) is adopted in [98], and [99] uses $\hat{x}_t^+$ in place of $\hat{x}_t^-$ in the control policy. *Affine disturbance feedback* is sometimes considered in SMPC methods, e.g. [100], and it is shown that affine disturbance feedback control policies and affine state feedback control policies lead to equivalent control inputs [105]; here we focus on the state-feedback parameterization.

Remark 4.2 (**Input Chance Constraints**). Hard input constraints are difficult to integrate with the affine policy (4.15), as under our previous assumptions the resulting control input is normally distributed and unbounded. Chance constraint (4.4) on input is thus used in its place, as in [98]. Another option as in [101] is to use (nonlinear) saturated policies in place of (4.15), but then the resulting inputs and outputs are no longer linear in decision variables and our further analysis would be much more complicated. Ultimately in implementation of course, one can saturate input actions to satisfy hard input constraints. ■

Resulting Input-Output Distribution or Mean-Variance

In the Gaussian noise setting (Section 4.1.1), with (4.1), (4.3), (4.9), (4.12), (4.13) and (4.15), at control step $t = k$, the resulting future inputs u_t and outputs y_t for $t \in \mathbb{Z}_{[k,k+N)}$ are distributed as

$$\begin{bmatrix} u_t \\ y_t \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \bar{u}_t \\ \bar{y}_t \end{bmatrix}, \Delta_{t-k} \right), \tag{4.18}$$

where the covariance matrix $\Delta_s \in \mathbb{S}_+^{m+p}$ for $s \in \mathbb{Z}_{[0,N)}$ can be computed as (4.19a) using $\Lambda_s \in \mathbb{S}_+^n$ defined by (4.19b),

$$\Delta_s := \begin{bmatrix} -K \\ C - DK \end{bmatrix} \Lambda_s \begin{bmatrix} -K \\ C - DK \end{bmatrix}^\mathsf{T} + \mathrm{Diag}(0_{m\times m}, \Xi) \tag{4.19a}$$

$$\Lambda_s := \textstyle\sum_{r=0}^{s-1} (A - BK)^r \, L_\mathsf{L} \, \Xi \, L_\mathsf{L}^\mathsf{T} \, (A - BK)^{r\mathsf{T}} \tag{4.19b}$$

with L_L in (4.11b) and $\Xi := C\Sigma^\times C^\mathsf{T} + \Sigma^\vee \in \mathbb{S}_{++}^p$. A derivation of (4.18) can be found in Section 4.6.1. Note that the matrices $\Delta_0, \Delta_1, \ldots, \Delta_{N-1}$ are fixed and can be computed offline.

In the distributionally robust setting (Section 4.1.2), we consider (4.5) instead of (4.3) and consider (4.10) instead of (4.9), and we obtain the same input-output mean and

variance as in (4.18), i.e.,

$$\mathbb{E}\left[\begin{bmatrix} u_t \\ y_t \end{bmatrix}\right] = \begin{bmatrix} \bar{u}_t \\ \bar{y}_t \end{bmatrix}, \qquad \mathrm{Var}\left[\begin{bmatrix} u_t \\ y_t \end{bmatrix}\right] = \Delta_{t-k}. \tag{4.20}$$

Resulting Expected Cost

SMPC problems typically consider the expectation of cost (2.2) summing over the desired horizon. Given the distribution in (4.18) or the mean and variance in (4.20), the expected cost is known as a deterministic value

$$\sum_{t=k}^{k+N-1} \mathbb{E}[J_t(u_t, y_t)] = \sum_{t=k}^{k+N-1} J_t(\bar{u}_t, \bar{y}_t) + J_{\mathsf{const}}, \tag{4.21}$$

where $J_{\mathsf{const}} := \sum_{s=0}^{N-1} \mathrm{Trace}(\Delta_s \,\mathrm{Diag}(R, Q))$ is a constant independent of decision variables $\bar{u}$ and θ.

4.2.4 Feedback Control Policies: Output Error Feedback

Section 4.2.3 was based on an affine feedback policy (4.15) with a fixed feedback gain K. Here we investigate alternative control policies where the feedback gain is a time-varying decision variable. Note that the naive parameterization modified from (4.15)

$$u_t \leftarrow \bar{u}_t - K_t(\hat{x}_t - \bar{x}_t) \tag{4.22}$$

leads to non-convex bilinear terms of the decision variables $\bar{u}$ and K_t, as $\hat{x}_t, \bar{x}_t$ depend on $\bar{u}_{[k,t)}$ via (4.13), (4.12). Here we apply an *output error feedback* control policy [106]

$$u_t \leftarrow \pi_t(\nu_{[k,t)}) := \bar{u}_t + \sum_{s=k}^{t-1} M_t^s \, \nu_s \tag{4.23}$$

where the nominal input $\bar{u}_t$ and feedback gains $M_t^s \in \mathbb{R}^{m \times p}$ are both decision variables, with innovation ν in (4.13a). The policy parameterization (4.23) contains within it the policy (4.22) as a special case: indeed, for a sequence of gains $K_{[k,k+N)}$, the selection for all $s, t \in \mathbb{Z}_{[k,k+N)}, s \leq t$

$$M_t^s \leftarrow (A - BK_{t-1})(A - BK_{t-2}) \cdots (A - BK_s) L_{\mathsf{L}} \tag{4.24}$$

reduces (4.23) to (4.22). Crucially, the output-error-feedback policy (4.23) leads to jointly convex optimization in decision variables $\bar{u}$ and M_t^s, as we will see next.

Resulting Input-Output Distribution or Mean-Variance

Define $\eta_k := \mathrm{col}(x_k - \mu_k^{\mathsf{x}}, w_{[k,k+N)}, v_{[k,k+N)}) \in \mathbb{R}^{n_\eta}$ as a vector of uncorrelated zero-mean random variables of dimension $n_\eta := n + nN + pN$. Then, both input u_t and output y_t of (4.1) can be written as affine functions of the decision variables $\bar{u}$ and M_t^s through direct calculation given the estimator (4.13) and policy (4.23),

$$\begin{bmatrix} u_t \\ y_t \end{bmatrix} = \begin{bmatrix} \bar{u}_t \\ \bar{y}_t \end{bmatrix} + \Lambda_t \, \eta_k, \quad t \in \mathbb{Z}_{[k,k+N)}, \tag{4.25}$$

with $\bar{y}$ in (4.12b), where the matrix $\Lambda_t \in \mathbb{R}^{(m+p)\times n_\eta}$ in (4.25) is linearly dependent on the gain matrices M_t^s as

$$\Lambda_t := \begin{bmatrix} \Delta_{t-k}^{\mathsf{U}} \\ \Delta_{t-k}^{\mathsf{Y}} \end{bmatrix} \mathcal{M} \, \Delta^{\mathsf{M}} + \begin{bmatrix} 0_{m\times n_\eta} \\ \Delta_{t-k}^{\mathsf{A}} \end{bmatrix}, \quad t \in \mathbb{Z}_{[k,k+N)}, \tag{4.26}$$

where $\mathcal{M} \in \mathbb{R}^{mN\times pN}$ is a concatenation of M_t^s

$$\mathcal{M} := \begin{bmatrix} M_k^k & & & \\ M_{k+1}^k & M_{k+1}^{k+1} & & \\ \vdots & \vdots & \ddots & \\ M_{k+N-1}^k & M_{k+N-1}^{k+1} & \cdots & M_{k+N-1}^{k+N-1} \end{bmatrix} \tag{4.27}$$

and where $\Delta_i^{\mathsf{U}} \in \mathbb{R}^{m\times mN}, \Delta_i^{\mathsf{Y}} \in \mathbb{R}^{p\times mN}, \Delta_i^{\mathsf{A}} \in \mathbb{R}^{p\times n_\eta}$ and $\Delta^{\mathsf{M}} \in \mathbb{R}^{pN\times n_\eta}$ are independent of both decision variables $\bar{u}$ and M_t^s, defined as follows,

$$\begin{aligned} &\mathrm{col}\big(\Delta_0^{\mathsf{U}}, \ldots, \Delta_{N-1}^{\mathsf{U}}\big) := I_{mN} \\ &\mathrm{col}\big(\Delta_0^{\mathsf{Y}}, \ldots, \Delta_{N-1}^{\mathsf{Y}}\big) := \Xi(A)\,(I_N \otimes B) \\ &\mathrm{col}\big(\Delta_0^{\mathsf{A}}, \ldots, \Delta_{N-1}^{\mathsf{A}}\big) := [\Theta(A), \Xi(A), I_{pN}] \\ &\Delta^{\mathsf{M}} := [\Theta(A_{\mathsf{L}}), \Xi(A_{\mathsf{L}}), I_{pN} - \Xi(A_{\mathsf{L}})\,(I_N \otimes L_{\mathsf{L}})] \end{aligned}$$

where we define $\Theta(A) := \mathrm{col}(C, CA, \ldots, CA^{N-1}) \in \mathbb{R}^{pN\times n}$ and $\Xi(A) \in \mathbb{R}^{pN\times nN}$,

$$\Xi(A) := \begin{bmatrix} 0_{p\times n} & & & \\ C & 0_{p\times n} & & \\ \vdots & \vdots & \ddots & \\ CA^{N-2} & \cdots & C & 0_{p\times n} \end{bmatrix}$$

and similarly define $\Theta(A_{\mathsf{L}}), \Xi(A_{\mathsf{L}})$ with $A_{\mathsf{L}} := A - L_{\mathsf{L}}C$.

In the Gaussian noise setting (Section 4.1.1), η_k has the distribution $\eta_k \sim \mathcal{N}(0, \Lambda_t \Sigma^\eta \Lambda_t^\mathsf{T})$ via (4.3) and (4.9), and thus given (4.25) we have the input-output distribution as

$$\begin{bmatrix} u_t \\ y_t \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \bar{u}_t \\ \bar{y}_t \end{bmatrix}, \Lambda_t \Sigma^\eta \Lambda_t^\mathsf{T} \right). \tag{4.28}$$

In the distributionally robust setting (Section 4.1.2), η_k has zero mean $\mathbb{E}[\eta_k] = 0$ and has the variance $\mathrm{Var}[\eta_k] = \Lambda_t \Sigma^\eta \Lambda_t^\mathsf{T}$ via (4.5) and (4.10), so we have the mean and variance of the input and output via (4.25) as

$$\mathbb{E}\left[\begin{bmatrix} u_t \\ y_t \end{bmatrix} \right] = \begin{bmatrix} \bar{u}_t \\ \bar{y}_t \end{bmatrix}, \qquad \mathrm{Var}\left[\begin{bmatrix} u_t \\ y_t \end{bmatrix} \right] = \Lambda_t \Sigma^\eta \Lambda_t^\mathsf{T}. \tag{4.29}$$

Resulting Expected Cost

SMPC problems typically consider the expected cost $\sum_{t=k}^{k+N-1} \mathbb{E}[J_t(u_t, y_t)]$ summing (2.2) over the horizon, which is equal to a deterministic quadratic function of $\bar{u}$ and M_t^s,

$$\sum_{t=k}^{k+N-1} \left[J_t(\bar{u}_t, \bar{y}_t) + \|\mathrm{Diag}(R, Q)^{\frac{1}{2}} \Lambda_t (\Sigma^\eta)^{\frac{1}{2}} \|_\mathsf{F}^2 \right], \tag{4.30}$$

given the mean and variance of $\mathrm{col}(u_t, y_t)$ and given that $\mathbb{E}[\|z\|_S^2] = \|\mathbb{E}[z]\|_S^2 + \|S^{\frac{1}{2}} \mathrm{Var}[z]^{\frac{1}{2}}\|_\mathsf{F}^2$ for any random vector z and fixed matrix S; $\|\cdot\|_\mathsf{F}$ denotes the Frobenius norm.

4.2.5 Deterministic Approximation of Chance Constraint

In the Gaussian noise setting (Section 4.1.1), despite known input-output distribution (4.18) or (4.28), an exact deterministic representation of the joint chance constraint (4.4) is difficult, as it requires integration of a multivariate probability density function over a polytope and generally no analytic representation is available [12, Sec. 2.2]. For this reason, the joint constraint (4.4) is commonly approximated by, e.g., being split into individual chance constraints [104], for each time $t \in \mathbb{Z}_{[k,k+N)}$,

$$\mathbb{P}\left\{ e_i^\mathsf{T} \begin{bmatrix} u_t \\ y_t \end{bmatrix} \leq f_i \right\} \geq 1 - p_{i,t}, \quad i \in \mathbb{Z}_{[1,q]} \tag{4.31}$$

where $e_i \in \mathbb{R}^{m+p}$ is the transposed i-th row of E, and $f_i \in \mathbb{R}$ is the i-th entry of f. The allocated risk probabilities $p_{i,t} > 0$ in (4.31) are introduced as additional decision variables, such that $p_{1,t}, p_{2,t}, \ldots, p_{q,t}$ sum up to the total risk p for each time t. Note that (4.31) is a conservative approximation (or a sufficient condition) of (4.4), due to subadditivity of probabilities. Here, we focus on the distribution (4.18) resulting from output feedback, while similar results can be derived for the distribution (4.28) arising from output error feedback. The chance constraints (4.31) are converted into an equivalent deterministic form, cf. [98, 104],

$$e_i^\mathsf{T} \begin{bmatrix} \bar{u}_t \\ \bar{y}_t \end{bmatrix} \leq f_i - \sqrt{e_i^\mathsf{T} \Delta_{t-k}\, e_i}\, \mathrm{icdfn}(1 - p_{i,t}), \;\; i \in \mathbb{Z}_{[1,q]} \tag{4.32a}$$

$$\sum_{i=1}^{q} p_{i,t} = p, \qquad p_{i,t} > 0, \;\; i \in \mathbb{Z}_{[1,q]} \tag{4.32b}$$

for $t \in \mathbb{Z}_{[k,k+N)}$, where $\mathrm{icdfn}(z) := \sqrt{2}\,\mathrm{erf}^{-1}(2z-1)$ is the inverse cumulative distribution function (inverse c.d.f.) or the z-quantile of the standard normal distribution, with erf^{-1} the inverse error function. The constraints (4.32) are convex when we require $p \in (0, \frac{1}{2}]$ [104, Thm. 1].

Remark 4.3 (**Gaussian Signals**). We have assumed through (4.3) and (4.9) that random variables are normally distributed. In the case where random signals are non-Gaussian but with the same means and variances in (4.3) and (4.9), the resulting inputs u_t and outputs y_t still possess the mean and variance in (4.18), and thus the expected cost is still (4.21). However, the inverse c.d.f. in (4.32a) should change correspondingly to the actual distribution (if known), or be replaced into an upper bound $\sqrt{(1-p_{i,t})/p_{i,t}}$ via Chebyshev–Cantelli inequality that guarantees the worse case over all distributions [64, 66]. ■

4.2.6 Deterministic Reformulation of DR-CVaR Constraint

In the distributionally robust setting (Section 4.1.2), the DR-CVaR constraint (4.8) can be equivalently written as a deterministic constraint. With output error feedback (4.23) in Section 4.2.4, constraint (4.8) is reformulated as a second-order cone (SOC) constraint on the decision variables $\bar{u}$ and M_t^s in the following lemma.

Lemma 4.4 (SOC Expression of DR-CVaR Constraint)**.** *With $h(u_t, y_t)$ as in (4.7), for $t \in \mathbb{Z}_{[k,k+N)}$, (4.8) holds iff*

$$2\big(\frac{1-\alpha}{\alpha}\big)^{\frac{1}{2}} \big\|(\Sigma^\eta)^{\frac{1}{2}} \Lambda_t^\mathsf{T} e_i\big\|_2 \leq -e_i^\mathsf{T} \begin{bmatrix} \bar{u}_t \\ \bar{y}_t \end{bmatrix} + f_i, \;\; i \in \mathbb{Z}_{[1,q]}. \tag{4.33}$$

Proof. Substituting (4.25) into (4.7), $h(u_t, y_t)$ can be written as

$$h(u_t, y_t) = \max_{i \in \{1,\dots,q\}} e_i^\mathsf{T} \Lambda_t \eta_k + e_i^\mathsf{T} \mathrm{col}(\bar{u}_t, \bar{y}_t) - f_i,$$

where the random variable η_k has zero mean and variance Σ^η. According to [96, Thm. 3.3], (4.8) holds if and only if there exist $\theta_t \in \mathbb{R}$ and $\Theta_t \in \mathbb{S}_+^{n_\eta+1}$ satisfying the LMIs

$$\begin{aligned} 0 &\geq \alpha\theta_t + \mathrm{Trace}\big[\Theta_t \,\mathrm{Diag}(\Sigma^\eta,\, 1)\big] \\ \Theta_t &\succeq \begin{bmatrix} 0_{n_\eta \times n_\eta} & \Lambda_t^\mathsf{T} e_i \\ e_i^\mathsf{T} \Lambda_t & e_i^\mathsf{T} \mathrm{col}(\bar{u}_t, \bar{y}_t) - f_i - \theta_t \end{bmatrix}, \; i \in \mathbb{Z}_{[1,q]}. \end{aligned}$$

From [107, Thm. 1], these LMIs are feasible in (θ_t, Θ_t) if and only if (4.33) holds, which completes the proof. ■

With output feedback (4.22) in Section 4.2.3, one can similarly reformulate constraint (4.8) as the deterministic SOC constraint (4.33) on the decision variable $\bar{u}$, where the value of M_t^s underlying Λ_t in (4.26) is fixed as follows

$$M_t^s \leftarrow (A - BK)^{t-s} L_\mathsf{L}, \tag{4.34}$$

as similar to (4.24). This is because the output-feedback policy (4.15) is a special case of the output-error-feedback policy (4.23) where M_t^s is selected as in (4.34).

4.2.7 Terminal Condition

Terminal constraints are considered in (S)MPC frameworks to ensure recursive feasibility and closed-loop stability. Assume $N \geq L$ going forward. Here, we impose a *terminal equality constraint* [46, 47, 48, 49],

$$\bar{u}_{k+N-L} = \bar{u}_{k+N-L+1} = \cdots = \bar{u}_{k+N-1} \tag{4.35a}$$

$$\bar{y}_{k+N-L} = \bar{y}_{k+N-L+1} = \cdots = \bar{y}_{k+N-1} \tag{4.35b}$$

that requires the nominal input-output trajectory to stay at some setpoint for final L steps in the prediction horizon. *Terminal set constraints* are also leveraged in (S)MPC methods, bounding the final nominal state in a positively invariant set [97, 98, 99, 102]; here we find the input-output terminal constraint (4.35) more straightforward to adapt to the data-driven case.

4.2.8 SMPC Optimization Problem and Implementation

With the expected cost (4.21), the approximation (4.32) of the constraint (4.4), the interpolation (4.14) and the terminal constraint (4.35), the SMPC problem is formulated as

$$\begin{aligned} \underset{\bar{u},\theta,p_{i,t}}{\text{minimize}} \quad & \sum_{t=k}^{k+N-1} J_t(\bar{u}_t,\bar{y}_t) + \lambda_\theta\,\theta \\ \text{subject to} \quad & (4.32) \text{ for } t \in \mathbb{Z}_{[k,k+N)}, (4.12), (4.14), (4.35), \end{aligned} \tag{SMPC}$$

with an interpolation penalty term of parameter $\lambda_\theta > 0$ [103]. With $R \succ 0$ and $\lambda_\theta > 0$, the cost in (SMPC) is jointly strongly convex in $\bar{u}$ and θ, and thus problem (SMPC) possesses a unique optimal $(\bar{u},\theta)$ if feasible, although optimal $p_{i,t}$ may not be unique. Problem (SMPC) can be efficiently solved by the Iterative Risk Allocation method [104]; see [78, Appendix B] for more details of our implementation.

While problem (SMPC) is based on the chance constraint (4.4) from Section 4.1.1 and the fixed-gain output feedback (4.15) from Section 4.2.3, alternative setups are also available. The SMPC problem can be formulated based on the DR-CVaR constraint (4.8) from Section 4.1.2 by replacing constraint (4.32) in (SMPC) into constraint (4.33), as discussed in Section 4.2.6. For an SMPC problem with output error feedback (4.23) from Section 4.2.4, the objective function should be selected as (4.30) plus the penalty term $\lambda_\theta\,\theta$. In the remainder of Section 4.2 and in Section 4.3, we focus on the setup in (SMPC) with the chance constraint and output feedback, while the simulations in Section 4.4 will compare the performance of these different setups.

The nominal inputs $\bar{u}_{[k,k+N)}$ and interpolation variable θ determined from problem (SMPC) complete the parameterization of the control policies $\pi_{[k,k+N)}$ in (4.15). The upcoming N_c control inputs $u_{[k,k+N_\mathrm{c})}$ are decided by the first N_c policies $\pi_{[k,k+N_\mathrm{c})}$ respectively, with a parameter $N_\mathrm{c} \in \mathbb{Z}_{[1,N]}$. Then, the next control step is set as $t = k + N_\mathrm{c}$. At the new control step, the initial condition $\mu^{\mathrm{x}}_{k+N_\mathrm{c}}$ interpolates between two fixed options $\mu^{\hat{\mathrm{x}}}_{k+N_\mathrm{c}}$ and $\mu^{\bar{\mathrm{x}}}_{k+N_\mathrm{c}}$ which are decided by

$$\mu^{\hat{\mathrm{x}}}_{k+N_\mathrm{c}} := \hat{x}^{-}_{k+N_\mathrm{c}}, \qquad \mu^{\bar{\mathrm{x}}}_{k+N_\mathrm{c}} := \bar{x}_{k+N_\mathrm{c}}, \tag{4.36}$$

as described in Section 4.2.2. The entire SMPC control process is shown in Algorithm 5.

4.2.9 Closed-Loop Properties

The investigated SMPC framework possesses recursive feasibility and closed-loop stability.

Algorithm 5 A Framework of Stochastic Model Predictive Control (SMPC)

Input: horizon lengths L, N, N_c, system matrices A, B, C, D, noise variances $\Sigma^{\mathsf{w}}, \Sigma^{\mathsf{v}}$, initial state mean $\mu^{\mathsf{x}}_{\mathsf{ini}}$, cost matrices Q, R, constraint coefficients E, f, probability bound p, interpolation penalty coefficient λ_θ.

Compute Kalman gain L_{K} via (4.11b), feedback gain K via (4.16), and covariance matrices $\Delta_{[0,N)}$ via (4.19).
Initialize the control step $k \leftarrow 0$ and set the initial condition $\mu^{\hat{\mathsf{x}}}_0 \leftarrow \mu^{\mathsf{x}}_{\mathsf{ini}}$ and $\mu^{\bar{\mathsf{x}}}_0 \leftarrow \mu^{\mathsf{x}}_{\mathsf{ini}}$.
while `true` **do**
  Solve $\bar{u}_{[k,k+N)}$ and θ from problem (SMPC).
  Obtain μ^{x}_k via (4.14) and obtain $\bar{x}_{[k,k+N]}$ via (4.12).
  Obtain policies $\pi_{[k,k+N)}$ from (4.15).
  for t **from** k **to** $k + N_c - 1$ **do**
    Compute $\hat{x}^-_t$ via (4.13).
    Input $u_t \leftarrow \pi_t(\hat{x}^-_t)$ to the system (4.1).
    Measure y_t from the system (4.1).
  Set $\mu^{\hat{\mathsf{x}}}_{k+N_c} \leftarrow \hat{x}^-_{k+N_c}$ and $\mu^{\bar{\mathsf{x}}}_{k+N_c} \leftarrow \bar{x}_{k+N_c}$ as (4.36).
  Set $k \leftarrow k + N_c$.

Lemma 4.5 (SMPC Recursive Feasibility)**.** *Assume $p \in (0, \frac{1}{2}]$. In Algorithm 5, if the problem (SMPC) is feasible at control step $k = \kappa$, then it is feasible at next control step $k = \kappa + N_c$.*

Proof. See Section 4.6.3. ■

With Lemma 4.5, problem (SMPC) is feasible at all control steps if it is feasible at the initial control step, where initial feasibility can be achieved by a proper choice of parameters $\mu^{\mathsf{x}}_{\mathsf{ini}}, E, f, p$. Closed-loop stability is captured as finiteness of the asymptotic expected cost.

Lemma 4.6 (SMPC Closed-loop Stability)**.** *Consider system (4.1) with input decided by Algorithm 5, where problem (SMPC) is assumed feasible at all control steps. Let the reference signal $r_t = r$ be time-invariant. Assume $\{z \in \mathbb{R}^{m+p} \mid Ez \leq f\}$ is a bounded set. Then, the asymptotic expected cost is upper bounded by some $c \geq 0$ as*

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[J_t(u_t, y_t)] \leq c.$$

Proof. See Section 4.6.4. ■

4.3 Stochastic Data-Driven Predictive Control

This section develops a data-driven control method whose performance will be shown to be equivalent to SMPC under certain tuning conditions. In the spirit of DeePC and SPC, our proposed control method consists of an offline process, where data is collected and used for system representation, and an online process which controls the system.

At a high level, our technical approach has three key steps. First, we collect offline input-output data (Section 4.3.1), and use this offline data to parameterize an auxiliary model (Section 4.3.2-1). This auxiliary model will take the place of the original parametric system model (4.1) in the design procedure. Second, we will formulate a stochastic predictive control method using the auxiliary model (Section 4.3.2, Section 4.3.3-1, Section 4.3.4-1). Third and finally, we will establish theoretical equivalences between the model-based and data-based control methods (Section 4.3.3-2, Section 4.3.4-2).

4.3.1 Use of Offline Data

In data-driven control, sufficiently rich offline data must be collected to capture the internal dynamics of the system. In this subsection, we demonstrate how offline data is collected, and use the data to compute some quantities that are useful to formulate our control method in the rest of the section. We first develop results with data from deterministic LTI systems, and then address the case of noisy data.

Deterministic Offline Data

Consider the deterministic version of system (4.1), namely the deterministic LTI system (2.1), reproduced as follows,

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t \\ y_t &= Cx_t + Du_t \end{aligned} \tag{2.1}$$

where with a slight abuse of notation, we temporarily in this section let x_t and y_t denote the state and output of system (2.1). By assumption, (2.1) is minimal; recall $L \in \mathbb{N}$ in Section 4.1 such that $\mathcal{O} := \mathrm{col}(C, CA, \ldots, CA^{L-1})$ has full column rank. Let $u^{\mathrm{d}}_{[1,T_{\mathrm{d}}]}, y^{\mathrm{d}}_{[1,T_{\mathrm{d}}]}$ be a T_{d}-length trajectory of input-output data collected from (2.1). The input sequence $u^{\mathrm{d}}_{[1,T_{\mathrm{d}}]}$ is assumed to be *persistently exciting* of order $K_{\mathrm{d}} := L + n + 1$, i.e., its associated

K_d-depth block-Hankel matrix $\mathscr{H}_{K_\mathrm{d}}(u^\mathrm{d}_{[1,T_\mathrm{d}]})$, defined as

$$\mathscr{H}_{K_\mathrm{d}}(u^\mathrm{d}_{[1,T_\mathrm{d}]}) := \begin{bmatrix} u^\mathrm{d}_1 & u^\mathrm{d}_2 & \cdots & u^\mathrm{d}_{T_\mathrm{d}-K_\mathrm{d}+1} \\ u^\mathrm{d}_2 & u^\mathrm{d}_3 & \cdots & u^\mathrm{d}_{T_\mathrm{d}-K_\mathrm{d}+2} \\ \vdots & \vdots & \ddots & \vdots \\ u^\mathrm{d}_{K_\mathrm{d}} & u^\mathrm{d}_{K_\mathrm{d}+1} & \cdots & u^\mathrm{d}_{T_\mathrm{d}} \end{bmatrix},$$

has full row rank. To achieve persistent excitation, one must collect at least $T_\mathrm{d} \geq (m+1)K_\mathrm{d} - 1$ data samples [38]. We formulate data matrices $U_1 \in \mathbb{R}^{mL\times h}$, $U_2 \in \mathbb{R}^{m\times h}$, $Y_1 \in \mathbb{R}^{pL\times h}$ and $Y_2 \in \mathbb{R}^{p\times h}$ of a common width $h := T_\mathrm{d} - L$ by partitioning associated Hankel matrices as

$$\begin{bmatrix} U_1 \\ U_2 \end{bmatrix} := \mathscr{H}_{L+1}(u^\mathrm{d}_{[1,T_\mathrm{d}]}), \qquad \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} := \mathscr{H}_{L+1}(y^\mathrm{d}_{[1,T_\mathrm{d}]}). \tag{4.37}$$

The data matrices in (4.37) will now be used to represent some quantities related to the system (2.1). Before stating the result, we introduce some additional notation. Define a system-related matrix $\mathbf{\Gamma} \in \mathbb{R}^{p\times(m+p)L}$ as

$$\mathbf{\Gamma} = \begin{bmatrix} \mathbf{\Gamma}_\mathsf{U} & \mathbf{\Gamma}_\mathsf{Y} \end{bmatrix} := \begin{bmatrix} C\mathcal{C} & CA^L \end{bmatrix} \begin{bmatrix} I_{mL} & 0_{mL\times n} \\ \mathcal{G} & \mathcal{O} \end{bmatrix}^\dagger. \tag{4.38}$$

with sub-blocks $\mathbf{\Gamma}_\mathsf{U} \in \mathbb{R}^{p\times mL}$ and $\mathbf{\Gamma}_\mathsf{Y} \in \mathbb{R}^{p\times pL}$, where $\mathcal{C} := [A^{L-1}B, \ldots, AB, B]$ is the extended (reversed) controllability matrix, and $\mathcal{G} \in \mathbb{R}^{pL\times mL}$ is an impulse-response matrix

$$\mathcal{G} := \begin{bmatrix} D & & & \\ CB & D & & \\ \vdots & \ddots & \ddots & \\ CA^{L-2}B & \cdots & CB & D \end{bmatrix}. \tag{4.39}$$

The following result provides expressions for the quantity $\mathbf{\Gamma}$ and the system matrix D in terms of raw data.

Lemma 4.7 (Data Representation of Model Quantities). *Given the data matrices in (4.37), if system (2.1) is controllable and the input data $u^\mathrm{d}_{[1,T_\mathrm{d}]}$ is persistently exciting of order $L+n+1$, then the matrix $\mathbf{\Gamma}$ defined in (4.38) and the matrix D in the model (2.1) can be expressed as*

$$\begin{bmatrix} \mathbf{\Gamma}_\mathsf{U} & \mathbf{\Gamma}_\mathsf{Y} & D \end{bmatrix} = Y_2 \operatorname{col}(U_1, Y_1, U_2)^\dagger.$$

Proof. See Section 4.6.5 for a proof. The data-expression of the impulse response, e.g., D and $\mathcal{G}$, is present in SPC literature [15]. Our contribution here is the data-representation of $\mathbf{\Gamma}$. ■

With Lemma 4.7, the matrices $\mathbf{\Gamma}$ and D can be represented using offline data collected from system (2.1), and these matrices will be used as part of the construction for our data-driven control method.

The Case of Stochastic Offline Data

Lemma 4.7 holds for the case of noise-free data. When the measured data is corrupted by noise, as will usually be the case, the pseudoinverse computations in Lemma 4.7 are fragile and do not recover the desired matrices $\mathbf{\Gamma}$ and D. A standard technique to robustify these computations is to replace the pseudoinverse $W^\dagger$ of $W := \mathrm{col}(U_1, Y_1, U_2)$ in Lemma 4.7 with its Tikhonov regularization $W^{\mathsf{tik}} := (W^\mathsf{T} W + \lambda I_h)^{-1} W^\mathsf{T}$ where $\lambda > 0$ is the regularization parameter. To interpret this, recall that $\mathcal{P} := Y_2 W^\dagger$ is a least-square solution to $\mathrm{argmin}_{\mathcal{P}} \|Y_2 - \mathcal{P} W\|_\mathsf{F}^2$. Correspondingly, the regularization $Y_2 W^{\mathsf{tik}}$ is the solution to a ridge-regression problem $\mathrm{argmin}_{\mathcal{P}} \|Y_2 - \mathcal{P} W\|_\mathsf{F}^2 + \lambda \|\mathcal{P}\|_\mathsf{F}^2$, which gives a maximum-likelihood or worst-case robust solution to the original least-square problem $\mathrm{argmin}_{\mathcal{P}} \|Y_2 - \mathcal{P} W\|_\mathsf{F}^2$ whose multiplicative parameter W has uncertain entries; see [24] sidebar "Roles of Regularization" for more details. Hence in the stochastic case, we estimate matrices $\mathbf{\Gamma}$ and D by applying Lemma 4.7 with $\mathcal{P} = Y_2 W^\dagger$ replaced by $\widehat{\mathcal{P}} := Y_2 W^{\mathsf{tik}}$.

4.3.2 Data-Driven State Estimation and Output Feedback

The SMPC approach of Section 4.2 uses as sub-components a state estimator and an affine feedback law. We now leverage the offline data as described in Section 4.3.1 to directly design analogs of these components based on data, and without knowledge of the system matrices.

Auxiliary State-Space Model

We begin by constructing an auxiliary state-space model which has equivalent input-output behavior to (4.1), but is parameterized only by the recorded data sequences of Section 4.3.1.

Define auxiliary signals $\mathbf{x}_t, \mathbf{w}_t \in \mathbb{R}^{n_{\mathsf{aux}}}$ of dimension $n_{\mathsf{aux}} := mL + pL + pL^2$ for system (4.1) by

$$\mathbf{x}_t := \begin{bmatrix} u_{[t-L,t)} \\ \hline y^{\circ}_{[t-L,t)} \\ \hline \rho_{[t-L,t)} \end{bmatrix}, \quad \mathbf{w}_t := \begin{bmatrix} 0_{mL\times 1} \\ \hline 0_{pL\times 1} \\ \hline 0_{pL(L-1)\times 1} \\ \rho_t \end{bmatrix} \tag{4.40}$$

where $y^{\circ}_t := y_t - v_t \in \mathbb{R}^p$ is the output excluding measurement noise, and $\rho_t := \mathcal{O} w_t \in \mathbb{R}^{pL}$ stacks the system's response to process noise w_t on time interval $[t+1, t+L]$. The construction of the auxiliary state $\mathbf{x}_t$ was inspired by [108]. The auxiliary signals $\mathbf{x}_t, \mathbf{w}_t$ together with u_t, y_t, v_t then satisfy the relations given by Lemma 4.8.

Lemma 4.8 (Auxiliary Model)**.** *For system* (4.1)*, signals* u_t, y_t, v_t *and the auxiliary signals* $\mathbf{x}_t, \mathbf{w}_t$ *in* (4.40) *satisfy*

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}u_t + \mathbf{w}_t, \tag{4.41a}$$

$$y_t = \mathbf{C}\mathbf{x}_t + Du_t + v_t, \tag{4.41b}$$

with $\mathbf{A} \in \mathbb{R}^{n_{\mathsf{aux}}\times n_{\mathsf{aux}}}$, $\mathbf{B} \in \mathbb{R}^{n_{\mathsf{aux}}\times m}$, $\mathbf{C} \in \mathbb{R}^{p\times n_{\mathsf{aux}}}$ *given by*

$$\mathbf{A} := \left[\begin{array}{c|c|c} \begin{matrix} I_{m(L-1)} \\ 0_{m\times m} \end{matrix} & 0 & 0 \\ \hline \begin{matrix} 0 \\ \mathbf{\Gamma}_{\mathsf{U}} \end{matrix} & \begin{matrix} 0 \quad I_{p(L-1)} \\ \hline \mathbf{\Gamma}_{\mathsf{Y}} \end{matrix} & \begin{matrix} 0 \\ \mathbf{F} - \mathbf{\Gamma}_{\mathsf{Y}}\mathbf{E} \end{matrix} \\ \hline 0 & 0 & \begin{matrix} I_{pL(L-1)} \\ 0_{pL\times pL} \end{matrix} \end{array}\right]$$

$$\mathbf{B} := \begin{bmatrix} 0_{m(L-1)\times m} \\ I_m \\ \hline 0_{p(L-1)\times m} \\ D \\ \hline 0_{pL^2\times m} \end{bmatrix}, \quad \mathbf{C} := \left[\begin{array}{c|c|c} \mathbf{\Gamma}_{\mathsf{U}} & \mathbf{\Gamma}_{\mathsf{Y}} & \mathbf{F} - \mathbf{\Gamma}_{\mathsf{Y}}\mathbf{E} \end{array}\right]$$

with matrices $\mathbf{\Gamma}_{\mathsf{U}}, \mathbf{\Gamma}_{\mathsf{Y}}$ *in* (4.38)*, matrix* D *in* (4.1)*, and zero-one matrices* $\mathbf{E} \in \mathbb{R}^{pL\times pL^2}$ *and* $\mathbf{F} \in \mathbb{R}^{p\times pL^2}$ *composed by selection matrices* $S_j := [0_{p\times(j-1)p}, I_p, 0_{p\times(L-j)p}] \in \mathbb{R}^{p\times pL}$ *for*

$j \in \{1, \dots, L\}$ *as*

$$\left[\begin{array}{c} \mathbf{E} \\ \hline \mathbf{F} \end{array}\right] := \left[\begin{array}{cccc} 0_{p\times pL} & & & \\ S_1 & 0_{p\times pL} & & \\ \vdots & \ddots & \ddots & \\ S_{L-1} & \cdots & S_1 & 0_{p\times pL} \\ \hline S_L & \cdots & S_2 & S_L \end{array}\right].$$

Proof. See Section 4.6.6. ■

The output noise signal v_t in (4.41) is precisely the same as in (4.1); the signal $\mathbf{w}_t$ appears now as a new disturbance; $\mathbf{w}_t$ and v_t are independent and follow the i.i.d. zero-mean normal distributions

$$\mathbf{w}_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0_{n_{\mathsf{aux}}\times 1}, \mathbf{\Sigma}^{\mathsf{w}}), \quad v_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0_{p\times 1}, \Sigma^{\mathsf{v}}) \tag{4.42}$$

with variances $\mathbf{\Sigma}^{\mathsf{w}} \in \mathbb{S}_+^{n_{\mathsf{aux}}}$ and $\Sigma^{\mathsf{v}} \in \mathbb{S}_{++}^{p}$,

$$\mathbf{\Sigma}^{\mathsf{w}} := \mathrm{Diag}\big(0_{(n_{\mathsf{aux}}-pL)\times(n_{\mathsf{aux}}-pL)},\, \Sigma^{\rho}\big) \tag{4.43}$$

where $\Sigma^{\rho} := \mathcal{O}\Sigma^{\mathsf{w}}\mathcal{O}^{\mathsf{T}} \in \mathbb{S}_+^{pL}$ is the variance of ρ_t. The matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, D$ are known given offline data described in Section 4.3.1, since they by definition only depend on matrices $\mathbf{\Gamma}$ and D which are data-representable via Lemma 4.7. Hence, the auxiliary model (4.41) can be interpreted as a non-minimal data-representable realization of system (4.1). Nonetheless, the model is indeed stabilizable and detectable.

Lemma 4.9. *For the auxiliary model* (4.41) *and matrix* $\mathbf{\Sigma}^{\mathsf{w}}$ *in* (4.43)*, the pairs* $(\mathbf{A}, \mathbf{B})$ *and* $(\mathbf{A}, \mathbf{\Sigma}^{\mathsf{w}})$ *are stabilizable and the pair* $(\mathbf{A}, \mathbf{C})$ *is detectable.*

Proof. See Section 4.6.7. ■

Auxiliary State Initial Condition

The auxiliary model (4.41) will now be used for both state estimation and control purposes. Suppose we are at a control step $t = k$ in a receding-horizon process. Similar to (4.9), we model the auxiliary state $\mathbf{x}_k$ from (4.41) following a prior distribution,

$$\mathbf{x}_k \sim \mathcal{N}(\boldsymbol{\mu}_k^{\mathsf{x}}, \mathbf{\Sigma}^{\mathsf{x}}) \tag{4.44}$$

where the mean $\boldsymbol{\mu}_k^{\mathsf{x}} \in \mathbb{R}^{n_{\mathsf{aux}}}$ interpolates between two fixed vectors $\boldsymbol{\mu}_k^{\hat{\mathsf{x}}}, \boldsymbol{\mu}_k^{\bar{\mathsf{x}}} \in \mathbb{R}^{n_{\mathsf{aux}}}$ with a decision variable $\theta \in [0, 1]$,

$$\boldsymbol{\mu}_k^{\mathsf{x}} := (1-\theta)\, \boldsymbol{\mu}_k^{\hat{\mathsf{x}}} + \theta\, \boldsymbol{\mu}_k^{\bar{\mathsf{x}}} \tag{4.45}$$

wherein $\boldsymbol{\mu}_k^{\hat{\mathsf{x}}}$ and $\boldsymbol{\mu}_k^{\bar{\mathsf{x}}}$ are produced by a state estimator (see (4.48)) and a noise-free model (see (4.49)), respectively, of last control step; at initial time $k = 0$, the initial state mean $\boldsymbol{\mu}_0^{\mathsf{x}}$ is given as a parameter $\boldsymbol{\mu}_{\mathsf{ini}}^{\mathsf{x}} \in \mathbb{R}^{n_{\mathsf{aux}}}$, i.e., we let $\boldsymbol{\mu}_0^{\hat{\mathsf{x}}} := \boldsymbol{\mu}_{\mathsf{ini}}^{\mathsf{x}}$ and $\boldsymbol{\mu}_0^{\bar{\mathsf{x}}} := \boldsymbol{\mu}_{\mathsf{ini}}^{\mathsf{x}}$. The variance $\boldsymbol{\Sigma}^{\mathsf{x}} \in \mathbb{S}_+^{n_{\mathsf{aux}}}$ in (4.44) is fixed as the unique positive semidefinite solution to the DARE (4.46a),

$$\boldsymbol{\Sigma}^{\mathsf{x}} = (\mathbf{A} - \mathbf{L}_{\mathsf{L}}\mathbf{C})\boldsymbol{\Sigma}^{\mathsf{x}}\mathbf{A}^{\mathsf{T}} + \boldsymbol{\Sigma}^{\mathsf{w}} \tag{4.46a}$$

$$\mathbf{L}_{\mathsf{L}} := \mathbf{A}\mathbf{L}_{\mathsf{K}}, \quad \mathbf{L}_{\mathsf{K}} := \boldsymbol{\Sigma}^{\mathsf{x}}\mathbf{C}^{\mathsf{T}}(\mathbf{C}\boldsymbol{\Sigma}^{\mathsf{x}}\mathbf{C}^{\mathsf{T}} + \Sigma^{\mathsf{v}})^{-1} \tag{4.46b}$$

given detectable $(\mathbf{A}, \mathbf{C})$ and stabilizable $(\mathbf{A}, \boldsymbol{\Sigma}^{\mathsf{w}})$ via Lemma 4.9, where we define the Kalman gain $\mathbf{L}_{\mathsf{K}} \in \mathbb{R}^{n_{\mathsf{aux}} \times p}$ and the Luenberger observer gain $\mathbf{L}_{\mathsf{L}} \in \mathbb{R}^{n_{\mathsf{aux}} \times p}$. Not surprisingly, there is a close relationship between the distributions of x_k and $\mathbf{x}_k$, as described in the next technical result, which will be leveraged in establishing equivalence between SMPC and our proposed method.

Lemma 4.10 (Related Means of x_k and $\mathbf{x}_k$)**.** *For system* (4.1) *and auxiliary state* $\mathbf{x}_t$ *in* (4.40)*, if* μ_k *is the mean of* x_k *and* $\boldsymbol{\mu}_k$ *is the mean of* $\mathbf{x}_k$*, then we have*

$$\mu_k = \Phi_{\mathsf{orig}}\, \widetilde{\mu}_k, \qquad \boldsymbol{\mu}_k = \Phi_{\mathsf{aux}}\, \widetilde{\mu}_k, \tag{4.47}$$

for some $\widetilde{\mu}_k \in \mathbb{R}^{mL+n(L+1)}$*, with matrices* Φ_{orig} *and* Φ_{aux} *defined in Claim 4.8.1 in 4.6.6.*

Proof. Given $x_k = \Phi_{\mathsf{orig}}\, \xi_k$ and $\mathbf{x}_k = \Phi_{\mathsf{aux}}\, \xi_k$ via Claim 4.8.1, we have (4.47) by choosing $\widetilde{\mu}_k$ as the mean of ξ_k. ■

Auxiliary State Estimation and Feedback

The Kalman filter of system (4.1) was given in (4.13). Here, we analogously formulate a Kalman filter for the auxiliary model (4.41) as

$$\hat{\mathbf{x}}_t^+ := \hat{\mathbf{x}}_t^- + \mathbf{L}_{\mathsf{K}}(y_t - \mathbf{C}\hat{\mathbf{x}}_t^- - Du_t), \qquad t \in \mathbb{Z}_{[k,k+N)} \tag{4.48a}$$

$$\hat{\mathbf{x}}_{t+1}^- := \mathbf{A}\hat{\mathbf{x}}_t^+ + \mathbf{B}u_t, \qquad t \in \mathbb{Z}_{[k,k+N)} \tag{4.48b}$$

$$\hat{\mathbf{x}}_k^- := \boldsymbol{\mu}_k^{\mathsf{x}} \tag{4.48c}$$

where $\hat{\mathbf{x}}_t^+$ and $\hat{\mathbf{x}}_t^-$ are the posterior and prior estimates of $\mathbf{x}_t$, respectively, with $\mathbf{L}_\mathsf{K} \in \mathbb{R}^{n_\mathsf{aux} \times p}$ in (4.46b). A noise-free model can be formed similarly as (4.12), given initial condition (4.44),

$$\bar{\mathbf{x}}_{t+1} := \mathbf{A}\bar{\mathbf{x}}_t + \mathbf{B}\bar{u}_t, \qquad t \in \mathbb{Z}_{[k,k+N)} \tag{4.49a}$$

$$\bar{\mathbf{y}}_t := \mathbf{C}\bar{\mathbf{x}}_t + D\bar{u}_t, \qquad t \in \mathbb{Z}_{[k,k+N)} \tag{4.49b}$$

$$\bar{\mathbf{x}}_k := \boldsymbol{\mu}_k^\times, \tag{4.49c}$$

where $\bar{u}_t \in \mathbb{R}^m$ is the nominal input decided through optimization, and $\bar{\mathbf{x}}_t \in \mathbb{R}^{n_\mathsf{aux}}$ and $\bar{\mathbf{y}}_t \in \mathbb{R}^p$ are the resulting nominal state and output, respectively. The affine output feedback policy (4.15) from SMPC is now extended as $\boldsymbol{\pi}_t(\cdot)$,

$$u_t \leftarrow \boldsymbol{\pi}_t(\hat{\mathbf{x}}_t^-) := \bar{u}_t - \mathbf{K}(\hat{\mathbf{x}}_t^- - \bar{\mathbf{x}}_t) \tag{4.50}$$

where the feedback gain $\mathbf{K} \in \mathbb{R}^{m \times n_\mathsf{aux}}$ must be selected such that $\mathbf{A} - \mathbf{BK}$ is Schur stable. Given the stabilizability of $(\mathbf{A}, \mathbf{B})$ and detectability of $(\mathbf{A}, \mathbf{C})$ by Lemma 4.9, we may again use an LQR-based design as in (4.16), yielding

$$\mathbf{K} := (R + \mathbf{B}^\mathsf{T}\mathbf{P}\mathbf{B} + D^\mathsf{T}QD)^{-1}(\mathbf{B}^\mathsf{T}\mathbf{P}\mathbf{A} + D^\mathsf{T}Q\mathbf{C}), \tag{4.51}$$

where $\mathbf{P} \in \mathbb{S}_+^{n_\mathsf{aux}}$ is the unique positive semidefinite solution to the DARE

$$\mathbf{P} = \mathbf{A}^\mathsf{T}\mathbf{P}(\mathbf{A} - \mathbf{BK}) + \mathbf{C}^\mathsf{T}Q(\mathbf{C} - D\mathbf{K}). \tag{4.52}$$

Although the use of state estimation was eliminated in some data-driven methods [38, 39, 40], our data-driven controller still incorporates the state estimator to enable output feedback. Some data-driven control methods also used state estimation for denoising purposes [108].

4.3.3 Optimization Problem

SDDPC Optimization Problem

With results of Section 4.3.2, we are now ready to mirror the steps which led to (SMPC) and formulate a Stochastic Data-Driven Predictive Control (SDDPC) optimization problem.

First, following a similar process as that which led to (4.18), we may combine (4.41), (4.42), (4.44), (4.48), (4.49) and (4.50), to conclude that the input-output trajectory

$\mathrm{col}(u_t, y_t)$ for $t \in \mathbb{Z}_{[k,k+N)}$ is normally distributed as $\mathcal{N}(\mathrm{col}(\bar{u}_t, \bar{\mathbf{y}}_t), \mathbf{\Delta}_{t-k})$, where the covariance matrices $\mathbf{\Delta}_s \in \mathbb{S}_+^{m+p}$ for $s \in \mathbb{Z}_{[0,N)}$ are computed as (4.53a) using $\mathbf{\Lambda}_s \in \mathbb{S}_+^{n_{\mathsf{aux}}}$ defined as (4.53b),

$$\mathbf{\Delta}_s := \begin{bmatrix} -\mathbf{K} \\ \mathbf{C} - D\mathbf{K} \end{bmatrix} \mathbf{\Lambda}_s \begin{bmatrix} -\mathbf{K} \\ \mathbf{C} - D\mathbf{K} \end{bmatrix}^{\mathsf{T}} + \mathrm{Diag}(0_{m\times m}, \mathbf{\Xi}) \tag{4.53a}$$

$$\mathbf{\Lambda}_s := \sum_{r=0}^{s-1} (\mathbf{A} - \mathbf{BK})^r \, \mathbf{L}_{\mathsf{L}} \, \mathbf{\Xi} \, \mathbf{L}_{\mathsf{L}}^{\mathsf{T}} \, (\mathbf{A} - \mathbf{BK})^{r\mathsf{T}} \tag{4.53b}$$

with $\mathbf{L}_{\mathsf{L}}$ in (4.46b) and $\mathbf{\Xi} := \mathbf{C}\mathbf{\Sigma}^{\times}\mathbf{C}^{\mathsf{T}} + \Sigma^{\mathsf{v}} \in \mathbb{S}_{++}^{p}$. Then, the SDDPC problem for computing $\bar{u}$ and θ at control step $t = k$ is written as

$$\begin{aligned} \underset{\bar{u},\,\theta,\,p_{i,t}}{\text{minimize}} \quad & \textstyle\sum_{t=k}^{k+N-1} J_t(\bar{u}_t, \bar{\mathbf{y}}_t) + \lambda_\theta \, \theta \\ \text{subject to} \quad & (4.54) \text{ for } t \in \mathbb{Z}_{[k,k+N)},\ (4.45),\ (4.49),\ (4.55), \end{aligned} \tag{SDDPC}$$

with the safety constraint

$$\begin{aligned} & e_i^{\mathsf{T}} \begin{bmatrix} \bar{u}_t \\ \bar{\mathbf{y}}_t \end{bmatrix} \leq f_i - \sqrt{e_i^{\mathsf{T}} \mathbf{\Delta}_{t-k} \, e_i} \, \mathrm{icdfn}(1 - p_{i,t}), \quad i \in \mathbb{Z}_{[1,q]} \\ & \sum_{i=1}^{q} p_{i,t} = p, \qquad\qquad p_{i,t} > 0, \quad i \in \mathbb{Z}_{[1,q]} \end{aligned} \tag{4.54}$$

for $t \in \mathbb{Z}_{[k,k+N)}$, and with the terminal equality constraint

$$\begin{aligned} \bar{u}_{k+N-L} &= \bar{u}_{k+N-L+1} = \ldots = \bar{u}_{k+N-1}, \\ \bar{\mathbf{y}}_{k+N-L} &= \bar{\mathbf{y}}_{k+N-L+1} = \ldots = \bar{\mathbf{y}}_{k+N-1}. \end{aligned} \tag{4.55}$$

Equivalence to SMPC Optimization Problem

We now establish that the SDDPC problem (SDDPC) and the SMPC problem (SMPC) have equal feasible sets and equal optimal sets, when the initial-condition parameters are related in the form of (4.47).

Proposition 4.11 (Equivalence of Optimization Problems)**.** *If the parameters* $\mu_k^{\hat{\mathsf{x}}}, \mu_k^{\bar{\mathsf{x}}}, \boldsymbol{\mu}_k^{\hat{\mathsf{x}}}, \boldsymbol{\mu}_k^{\bar{\mathsf{x}}}$ *satisfy*

$$\begin{aligned} \mu_k^{\hat{\mathsf{x}}} &= \Phi_{\mathsf{orig}} \, \widetilde{\mu}_k^{\hat{\mathsf{x}}}, \qquad & \boldsymbol{\mu}_k^{\hat{\mathsf{x}}} &= \Phi_{\mathsf{aux}} \, \widetilde{\mu}_k^{\hat{\mathsf{x}}}, \\ \mu_k^{\bar{\mathsf{x}}} &= \Phi_{\mathsf{orig}} \, \widetilde{\mu}_k^{\bar{\mathsf{x}}}, \qquad & \boldsymbol{\mu}_k^{\bar{\mathsf{x}}} &= \Phi_{\mathsf{aux}} \, \widetilde{\mu}_k^{\bar{\mathsf{x}}}, \end{aligned} \tag{4.56}$$

for some vectors $\widetilde{\mu}_k^{\hat{\mathbf{x}}}, \widetilde{\mu}_k^{\bar{\mathbf{x}}} \in \mathbb{R}^{mL+n(L+1)}$*, then the optimal (resp. feasible) solution set of SDDPC problem* (SDDPC) *is equal to the optimal (resp. feasible) solution set of SMPC problem* (SMPC)*.*

Proof. We first claim that, for Δ_s in (4.19a) and $\mathbf{\Delta}_s$ in (4.53a),

$$\Delta_s = \mathbf{\Delta}_s, \qquad s \in \mathbb{Z}_{[0,N)}. \tag{4.57}$$

Moreover, for any $\bar{u}_{[k,k+N)}$ and θ, the resulting nominal outputs from (4.12), (4.14) and from (4.49), (4.45) are equal, i.e.,

$$\bar{y}_t = \bar{\mathbf{y}}_t, \qquad t \in \mathbb{Z}_{[k,k+N)} \tag{4.58}$$

The proof of (4.57) and (4.58) can be found in Section 4.6.8. Given (4.57) and (4.58), the objective functions of problems (SMPC) and (SDDPC) are equal, and the constraint (4.32) in problem (SMPC) and the constraint (4.54) in problem (SDDPC) are equivalent. Thus problems (SMPC) and (SDDPC) have the same objective function and constraints, and the result follows. ■

We conclude by noting that problem (SDDPC) produces a unique optimal $(\bar{u}, \theta)$ when feasible, following from Proposition 4.11 and the fact that problem (SMPC) gives a unique optimal $(\bar{u}, \theta)$ when it is feasible, as mentioned in Section 4.2.

4.3.4 Online Control Algorithm

SDDPC Control Algorithm

We now describe the online implementation of our SDDPC. At time $t = k$, the nominal input sequence $\bar{u}_{[k,k+N)}$ and the interpolation variable θ are computed from (SDDPC). We then construct the policies $\boldsymbol{\pi}_{[k,k+N)}$ via (4.50), and apply the first N_c policies to the system. Then, $t = k + N_\mathrm{c}$ is set as the next control step. The initial condition (4.44) at the new control step interpolates between two vectors $\boldsymbol{\mu}_{k+N_\mathrm{c}}^{\hat{\mathbf{x}}}$ and $\boldsymbol{\mu}_{k+N_\mathrm{c}}^{\bar{\mathbf{x}}}$ decided as

$$\boldsymbol{\mu}_{k+N_\mathrm{c}}^{\hat{\mathbf{x}}} := \hat{\mathbf{x}}_{k+N_\mathrm{c}}^{-}, \qquad \boldsymbol{\mu}_{k+N_\mathrm{c}}^{\bar{\mathbf{x}}} := \bar{\mathbf{x}}_{k+N_\mathrm{c}}, \tag{4.59}$$

which are fixed and known at time $t = k + N_\mathrm{c}$. The method is formally summarized in Algorithm 6.

Algorithm 6 Stochastic Data-Driven Predictive Control (SDDPC)

Input: horizon lengths L, N, N_c, offline data u^d, y^d, noise variances $\Sigma^\rho, \Sigma^\mathsf{v}$, initial-state mean $\boldsymbol{\mu}^\mathsf{x}_\mathsf{ini}$, cost matrices Q, R, constraint coefficients E, f, probability bound p, interpolation penalty coefficient λ_θ.

Compute matrices $\boldsymbol{\Gamma}$ and D as in Section 4.3.1 using data u^d, y^d, and formulate matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ as in Section 4.3.2.
Compute Kalman gain $\mathbf{L}_\mathsf{K}$ via (4.46b), feedback gain $\mathbf{K}$ via (4.51), and covariance matrices $\boldsymbol{\Delta}_{[0,N)}$ via (4.53).
Initialize the control step $k \leftarrow 0$ and set the initial condition $\boldsymbol{\mu}^{\hat{\mathsf{x}}}_0 \leftarrow \boldsymbol{\mu}^\mathsf{x}_\mathsf{ini}$ and $\boldsymbol{\mu}^{\bar{\mathsf{x}}}_0 \leftarrow \boldsymbol{\mu}^\mathsf{x}_\mathsf{ini}$.
while `true` **do**
  Solve $\bar{u}_{[k,k+N)}$ and θ from problem (SDDPC).
  Obtain $\boldsymbol{\mu}^\mathsf{x}_k$ via (4.45) and obtain $\bar{\mathbf{x}}_{[k,k+N]}$ via (4.49).
  Obtain policies $\boldsymbol{\pi}_{[k,k+N)}$ from (4.50).
  for t **from** k **to** $k + N_c - 1$ **do**
    Compute $\hat{\mathbf{x}}^-_t$ via (4.48).
    Input $u_t \leftarrow \boldsymbol{\pi}_t(\hat{\mathbf{x}}^-_t)$ to the system (4.1).
    Measure y_t from the system (4.1).
  Set $\boldsymbol{\mu}^{\hat{\mathsf{x}}}_{k+N_c} \leftarrow \hat{\mathbf{x}}^-_{k+N_c}$ and $\boldsymbol{\mu}^{\bar{\mathsf{x}}}_{k+N_c} \leftarrow \bar{\mathbf{x}}_{k+N_c}$ as (4.59).
  Set $k \leftarrow k + N_c$.

Closed-loop Properties of SDDPC

Similar to Lemma 4.5 and Lemma 4.6, Algorithm 6 possesses recursive feasibility and closed-loop stability, as formally stated below.

Corollary 4.5.1 (SDDPC Recursive feasibility)**.** *Assume $p \in (0, \frac{1}{2}]$. In Algorithm 6, if the problem (SDDPC) is feasible at control step $k = \kappa$, then it is feasible at next control step $k = \kappa + N_c$.*

Corollary 4.6.1 (SDDPC Closed-loop Stability)**.** *Consider system (4.1) with input decided by Algorithm 6, where problem (SDDPC) is assumed feasible at all control steps. Let the reference signal $r_t = r$ be time-invariant. Assume $\{z \in \mathbb{R}^{m+p} \mid Ez \leq f\}$ is a bounded set. Then, the asymptotic expected cost is upper bounded by some $c \geq 0$ as*

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[J_t(u_t, y_t)] \leq c.$$

The proofs of the above corollaries are analogous to the proofs of Lemma 4.5 and

Lemma 4.6, respectively, where the auxiliary model (4.41) is considered in place of model (4.1).

Equivalence to SMPC Algorithm

We present in Theorem 4.13 our main result, which says that under idealized conditions, our proposed SDDPC control method and the benchmark SMPC method will result in identical control actions.

Assumption 4.12 (SDDPC Parameter Choice w.r.t. SMPC)**.** Given the parameters in Algorithm 5, we assume the parameters in Algorithm 6 satisfy the following.

(a) L is sufficiently large so that $\mathcal{O}$ has full column rank.

(b) Data $u^{\mathrm{d}}, y^{\mathrm{d}}$ comes from the deterministic system (2.1), and input data u^{d} is persistently exciting of order $L+n+1$.

(c) Given Σ^{w} in Algorithm 5, the parameter Σ^{ρ} in Algorithm 6 is set equal to $\mathcal{O}\Sigma^{\mathsf{w}}\mathcal{O}^{\mathsf{T}}$.

(d) Given $\mu^{\mathsf{x}}_{\mathsf{ini}}$ in Algorithm 5, the parameter $\boldsymbol{\mu}^{\mathsf{x}}_{\mathsf{ini}}$ in Algorithm 6 is selected as $\Phi_{\mathsf{aux}}\widetilde{\mu}^{\mathsf{x}}_{\mathsf{ini}}$ for some $\widetilde{\mu}^{\mathsf{x}}_{\mathsf{ini}} \in \mathbb{R}^{mL+(n+1)L}$ satisfying $\mu^{\mathsf{x}}_{\mathsf{ini}} = \Phi_{\mathsf{orig}}\widetilde{\mu}^{\mathsf{x}}_{\mathsf{ini}}$, with matrices $\Phi_{\mathsf{orig}}, \Phi_{\mathsf{aux}}$ defined in 4.6.6. (Such $\widetilde{\mu}^{\mathsf{x}}_{\mathsf{ini}}$ always exists because Φ_{orig} has full row rank.)

Theorem 4.13 (**Equivalence of SMPC and SDDPC**)**.** *Consider the stochastic system (4.1) with a specific initial state x_0 and a specific noise realization $\{w_t, v_t\}_{t=0}^{\infty}$, and consider the following two control processes:*

a) decide control actions $\{u_t\}_{t=0}^{\infty}$ by executing Algorithm 5;

b) decide control actions $\{u_t\}_{t=0}^{\infty}$ by executing Algorithm 6, where the parameters satisfy Assumption 4.12.

Then, the state-input-output trajectories $\{x_t, u_t, y_t\}_{t=0}^{\infty}$ resulting from process a) and from process b) are the same.

Proof. Let $\{x^{\mathsf{a}}_t, u^{\mathsf{a}}_t, y^{\mathsf{a}}_t\}$ denote the trajectory produced by process a), and $\{x^{\mathsf{b}}_t, u^{\mathsf{b}}_t, y^{\mathsf{b}}_t\}$ the trajectory from process b). We make the following claim, whose proof can be found in 4.6.9.

Claim 4.13.1. *At control step $t = k$ in processes a) and b), if*

i) the states $x_k^{\mathsf{a}} = x_k^{\mathsf{b}}$ are equal in processes a) and b), and
ii) the parameters $\mu_k^{\hat{\mathsf{x}}}, \mu_k^{\bar{\mathsf{x}}}$ in process a) and the parameters $\boldsymbol{\mu}_k^{\hat{\mathsf{x}}}, \boldsymbol{\mu}_k^{\bar{\mathsf{x}}}$ in process b) satisfy (4.56),

then

1) the states $x_t^{\mathsf{a}} = x_t^{\mathsf{b}}$ are equal for time $t \in \mathbb{Z}_{[k,k+N_{\mathrm{c}}]}$, and the inputs $u_t^{\mathsf{a}} = u_t^{\mathsf{b}}$ and outputs $y_t^{\mathsf{a}} = y_t^{\mathsf{b}}$ are equal for time $t \in \mathbb{Z}_{[k,k+N_{\mathrm{c}})}$, and
2) parameters $\mu_{k+N_{\mathrm{c}}}^{\hat{\mathsf{x}}}, \mu_{k+N_{\mathrm{c}}}^{\bar{\mathsf{x}}}$ in process a) and parameters $\boldsymbol{\mu}_{k+N_{\mathrm{c}}}^{\hat{\mathsf{x}}}, \boldsymbol{\mu}_{k+N_{\mathrm{c}}}^{\bar{\mathsf{x}}}$ in process b) satisfy (4.56) with $k \leftarrow k + N_{\mathrm{c}}$.

We finish the proof by showing that the results 1) and 2) in Claim 4.13.1 are true for all control steps $k \in \{0, N_{\mathrm{c}}, 2N_{\mathrm{c}}, \ldots\}$, by induction on k. **Base Case.** For $k = 0$, condition i) is true given that both processes start with a common initial state x_0, and condition ii) holds due to Assumption 4.12(d) and due to the selections $(\mu_0^{\hat{\mathsf{x}}}, \mu_0^{\bar{\mathsf{x}}}) \leftarrow (\mu_{\mathsf{ini}}^{\mathsf{x}}, \mu_{\mathsf{ini}}^{\mathsf{x}})$ in Algorithm 5 and $(\boldsymbol{\mu}_0^{\hat{\mathsf{x}}}, \boldsymbol{\mu}_0^{\bar{\mathsf{x}}}) \leftarrow (\boldsymbol{\mu}_{\mathsf{ini}}^{\mathsf{x}}, \boldsymbol{\mu}_{\mathsf{ini}}^{\mathsf{x}})$ in Algorithm 6. With conditions i) and ii) satisfied, the results 1) and 2) in Claim 4.13.1 are true for $k = 0$. **Inductive Step.** For $k = \kappa$, assume results 1) and 2), which imply the conditions i) and ii) respectively for $k = \kappa + N_{\mathrm{c}}$. Thus, through Claim 4.13.1, the results 1) and 2) are true for $k = \kappa + N_{\mathrm{c}}$. By induction on k, we have results 1) and 2) for all control steps $k \in \{0, N_{\mathrm{c}}, 2N_{\mathrm{c}}, \ldots\}$. The result 1) for all k suffices to prove the theorem. ■

Theorem 4.13 should be interpreted as equivalence between SMPC and SDDPC in the idealized setting. Specifically, it establishes that if the proposed SDDPC algorithm is provided with noise-free offline data, if the initial conditions set within SMPC and SDDPC match, and if the process noise variance Σ^{ρ} in the algorithm is set in a specific idealized fashion relative to the original process noise variance Σ^{w}, then the method will produce identical results to those obtained by applying SMPC. While in practice these assumptions will not hold, noisy offline data can be accommodated as discussed in Section 4.3.1, and Σ^{ρ} becomes a tuning parameter of our SDDPC method.

4.4 Simulations

4.4.1 Simulations on Grid-Connected Power Converter System

In this section, we numerically test our proposed method on the nonlinear grid-connected power converter system from [61], shown in Fig. 4.1, and we compare the results with those of several benchmark model-based and data-based techniques.

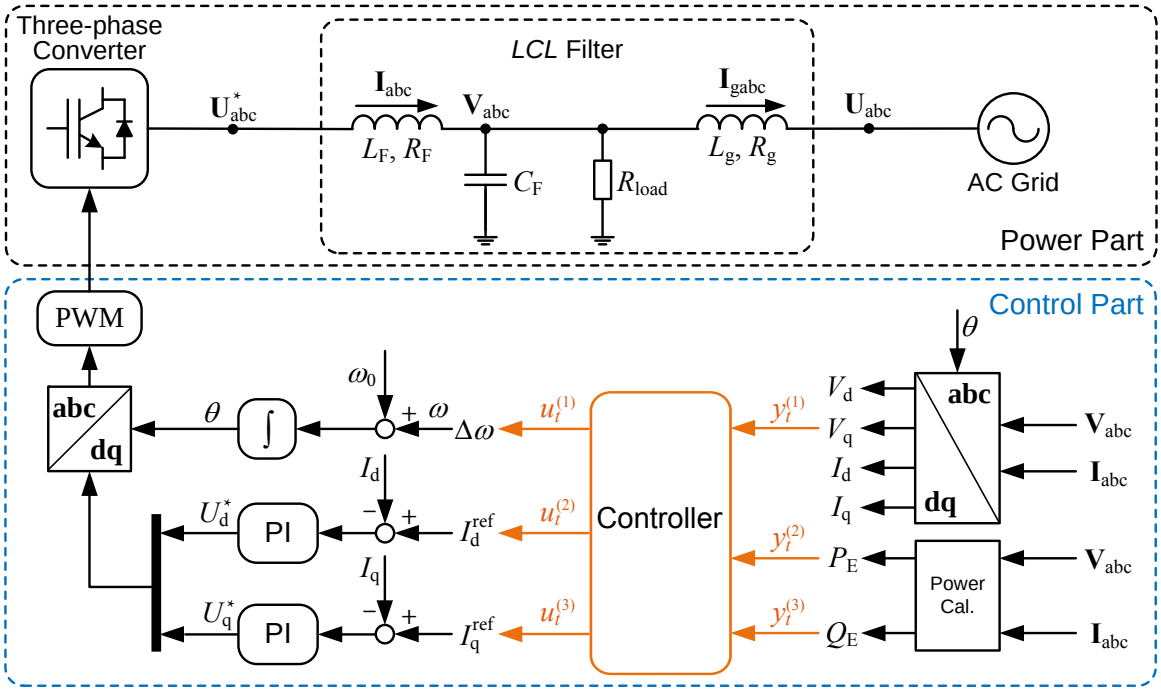

Figure 4.1: The one-line diagram of a grid-connected power converter.

The AC grid in the power part of Fig. 4.1 is modeled as an infinite bus with fixed voltage (1 p.u.) and fixed frequency (1 p.u.). This model has $n = 6$ states, $m = 3$ inputs and $p = 3$ outputs. The inputs are the angular frequency correction $\Delta\omega$ and current references $I_{\mathrm{d}}^{\mathrm{ref}}$ and $I_{\mathrm{q}}^{\mathrm{ref}}$ of d- and q-axes, respectively. The outputs to be controlled are the q-axis voltage V_{q}, the active power P_{E} and the reactive power Q_{E}. The LCL-filter parameters and the PI parameters in Fig. 4.1 are consistent with [61], whereas we choose the load resistance R_{load} as a Gaussian signal with mean 4 p.u. and noise power 10^{-3} p.u., which introduces process noise. The measurement noise on each output is Gaussian with noise power 10^{-7} p.u., consistent with [61].

Benchmark Control Methods

Several existing receding-horizon control methods are performed in our simulations and compared to our proposed SDDPC method. Here, we introduce the benchmark control methods in our simulations.

We investigate two benchmark model-based methods, namely Stochastic MPC (SMPC) (Section 4.2) and deterministic MPC (or MPC) (Section 2.2). For both SMPC and MPC, we use an identified system model in place of the true model (A, B, C, D), through N4SID system identification method [109] using offline data $u^{\mathrm{d}}, y^{\mathrm{d}}$ collected from the system. In MPC, the control action u_t is equal to the decision $\bar{u}_t$ by optimization, instead of using a feedback policy. The SMPC and MPC optimization problems at control step $t = k$ are (SMPC) and (MPC), respectively, whereas the deterministic constraint (2.7) in problem (MPC) is here replaced by

$$E \begin{bmatrix} \bar{u}_t \\ \bar{y}_t \end{bmatrix} \leq f \tag{4.60}$$

according to the constraint specified in (4.2).

We also investigate L2-regularized DeePC [61] and regularized SPC [15] as benchmark data-driven methods. In DeePC and SPC, the decisions $\bar{u}_t$ of optimization are applied as control actions u_t. The regularized DeePC and SPC optimization problems are (reg. DeePC) and (SPC), respectively, whereas the constraint (2.7) in both problems (reg. DeePC) and (SPC) is here replaced by (4.60), given the specified constraint (4.2). Moreover, the SPC predictive matrix $\mathcal{P}_{\mathsf{spc}}$ in constraint (3.24) of problem (SPC) is replaced by its Tikhonov regularization, as introduced in Section 2.6, with a regularization parameter $\lambda > 0$.

Offline Data Collection

Offline data is required in all our investigated control methods, for use in either data matrices (SDDPC, DeePC and SPC) or for system identification (MPC and SMPC). In our simulation, the data collection process lasted for 1 second and produced a data trajectory of length $T_{\mathrm{d}} = 1000$ with a sampling period of 1ms. The input data was generated as follows: $\Delta\omega$ (input 1) was set as the phase-locked loop (PLL) control action (see e.g. [41]) plus a white-noise signal, $I_{\mathrm{d}}^{\mathrm{ref}}$ (input 2) was set as 0.4 p.u. plus a white-noise signal, and $I_{\mathrm{q}}^{\mathrm{ref}}$ (input 3) was set at 0 p.u. plus a white-noise signal. Each white noise signal had noise power of 10^{-6} p.u..

Simulation Results

All controller parameters are reported in Table 4.1. Our simulation consists of two parts. In the first part, we compare the tracking performances of the different controllers. In the second part, we examine the ability of the controllers to maintain safety constraints.

Time Horizon Lengths	
Initial-condition horizon length	$L = 5$
Prediction horizon length	$N = 30$
Control horizon length	$N_{\mathrm{c}} = 10$
Problem Setup Parameters	
Sampling Period	$T_{\mathrm{s}} = 1\mathrm{ms}$
Cost matrices	$Q = 10^4 I_p,\ R = I_m$
Constraint coefficients	$E = I_{m+p} \otimes \left[\begin{smallmatrix} 1 \\ -1 \end{smallmatrix}\right]$, $f = \left[\begin{smallmatrix} 0.6\times\mathbf{1}_{2m\times 1} \\ 0.4\times\mathbf{1}_{2p\times 1} \end{smallmatrix}\right]$
Risk probability bound	$p = 0.2$
Interpolation penalty	$\lambda_\theta = 10$
Variance of v_t for SMPC/SDDPC	$\Sigma^{\mathsf{v}} = 10^{-8} I_p$
Variance of ρ_t for SDDPC	$\Sigma^{\rho} = 10^{-4} I_{pL}$
Variance of w_t for SMPC[a]	$\Sigma^{\mathsf{w}} = \mathcal{O}^{\dagger}\Sigma^{\rho}\mathcal{O}^{\dagger\mathsf{T}}$
Regularization Parameters	
Regularization in DeePC	$\lambda_{\mathrm{y}} = 10^6,\quad \lambda_{\mathrm{g}} = 10^3$
Regularization of $\mathcal{P}$ in SDDPC	$\lambda = 10^{-3}$
Regularization of $\mathcal{P}_{\mathsf{spc}}$ in SPC	$\lambda = 10^{-3}$

[a]In computation of Σ^{w}, matrix $\mathcal{O}$ is obtained given the identified system (A, B, C, D) in SMPC.

Table 4.1: Control parameters of the grid-connected power-converter system.

Tracking Performance. For each controller, we perform the following control process. From time 0s to time 0.2s, the controller is switched off, and the inputs $I_{\mathrm{d}}^{\mathrm{ref}}$ and $I_{\mathrm{q}}^{\mathrm{ref}}$ are set to zero, with $\Delta\omega$ generated from the PLL. After time 0.2s, the controller is switched on, and the output reference signal is $r_t = [0, 0, 0]^{\mathsf{T}}$ before time 0.5s and $r_t = [0, 0.3, 0]^{\mathsf{T}}$ after time 0.5s. To quantitatively compare the results, Fig. 4.2 shows the stage cost accumulated over the first two seconds for each controller. The result shows that the stochastic control methods (SMPC and SDDPC) outperformed the deterministic control methods (DeePC, SPC and MPC) in terms of their cumulative costs. This observation aligns with our expectation that stochastic control performs better with stochastic systems, since the stochastic control methods receive feedback at each time step – more frequently than the deterministic control methods which receive feedback only at each control step,

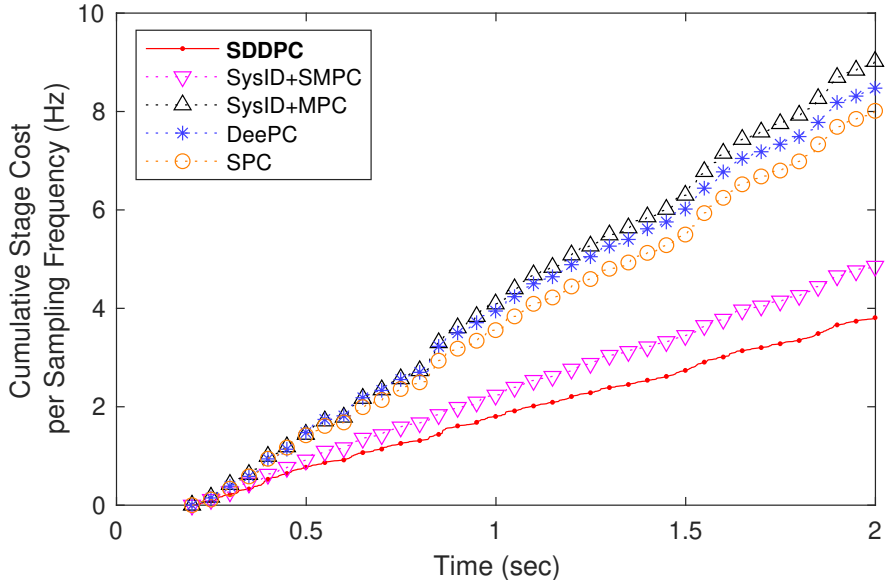

Figure 4.2: Cumulative stage cost with different controllers, $N_c = 10$.

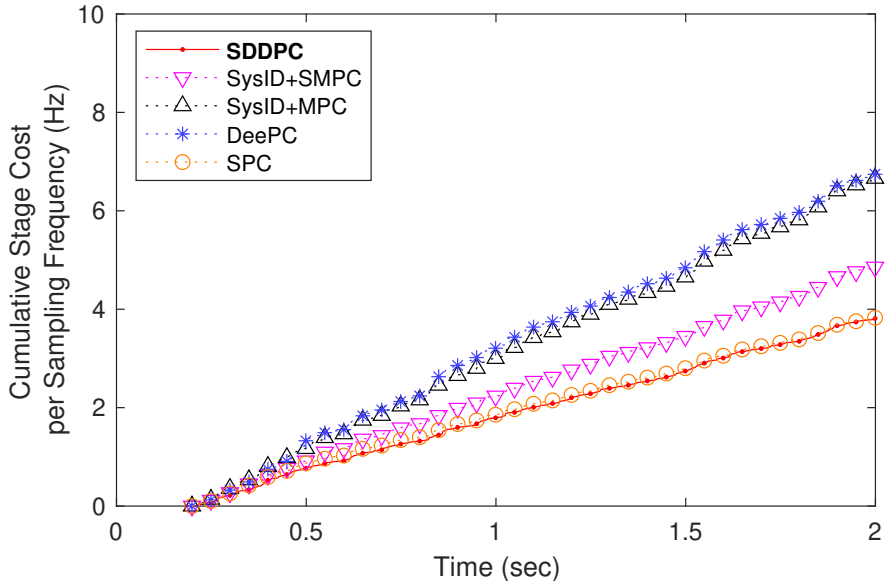

Figure 4.3: Cumulative stage cost with different controllers, $N_c = 1$.

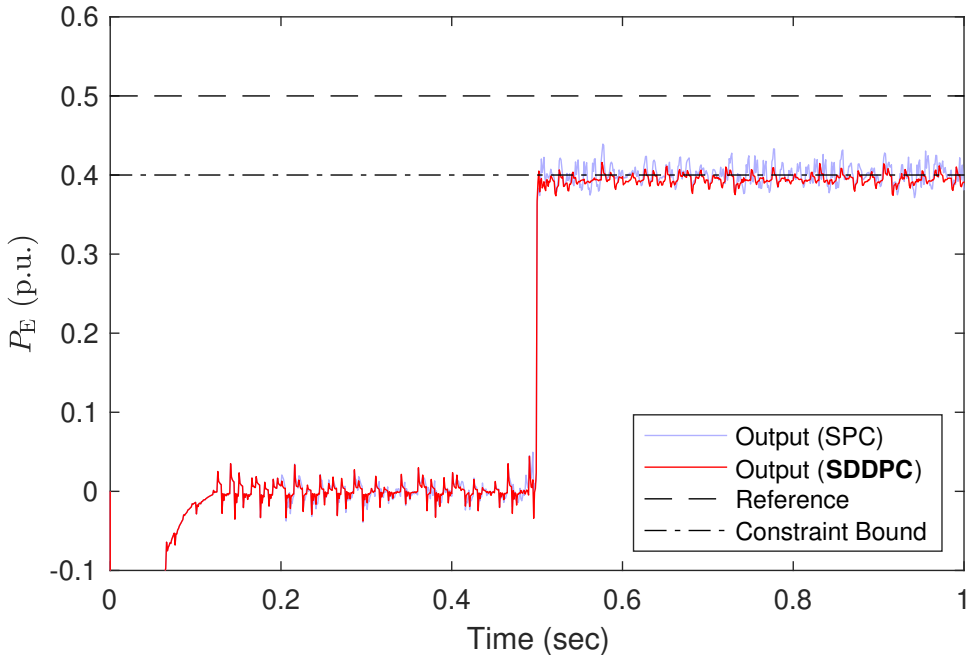

Figure 4.4: The second output signals with SPC (light blue) and SDDPC (red) in the constraint satisfaction test.

Controller	**Violation Rate**	**Total Violation Amount**
SDDPC ($p = 0.2$)	0.15	1.10
SDDPC ($p = 0.05$)	0.03	0.05
SysID+SMPC ($p = 0.2$)	0.19	1.55
SysID+SMPC ($p = 0.05$)	0.11	0.52
SysID+MPC	0.57	6.79
DeePC	0.20	1.46
SPC	0.49	8.42

Table 4.2: Statistics of constraint violation of the second output channel from 0.5s to 2.0s.

i.e., every $N_c = 10$ time steps. However, this benefit of stochastic control vanishes when we select shorter control horizons. Fig. 4.3 shows the cumulative stage costs when the control horizon has length $N_c = 1$, where we no longer observe a performance gap between all stochastic methods and all deterministic methods. SDDPC and SPC outperformed other controllers. Although we showed the results with different N_c, we emphasize significance of the $N_c = 10$ setting, which requires less computation since the optimization problems are solved less frequently.

Output Constraint Satisfaction. We next evaluate for each controller its ability to meet the output safety constraints. We repeat the control process above, but the reference signal becomes $r_t = [0, 0, 0]^\mathsf{T}$ before time 0.5s and $r_t = [0, 0.5, 0]^\mathsf{T}$ after time 0.5s. Note that the reference value 0.5 for the second output channel after time 0.5s is beyond the range of output safety constraint (with E, f in TABLE 4.1), which restricts all output channels within the range of $[-0.4, 0.4]$. As a result, in our simulations, the second output channel remained close to the upper safety bound of 0.4 after time 0.5s for all controllers; for example, the trace of the second output under SPC and SDDPC is displayed in Fig. 4.4.

To quantify the constraint satisfaction with each controller, from time 0.5s to time 2.0s (1500 time steps), we count the number and compute the rate of time steps where the measurement of the second output channel violates the safety constraint. As a second metric, we sum the amount of constraint violation that occurs between 0.5s to 2.0s for each controller. The results are displayed in TABLE 4.2, where we also displayed the results of SMPC and SDDPC with parameter p changed from 0.2 (as in TABLE 4.1) to 0.05. As the result shows, both violation rates of SMPC and SDDPC declined as we decrease p, while the violation rate of SDDPC shrank more effectively than that of SMPC. The total violation amounts of SMPC and SDDPC also reduced when we decrease p. Among the methods using deterministic safety constraint, DeePC had a lower violation rate and a smaller violation amount than MPC and SPC.

4.4.2 Simulations on Batch Reactor System

In this section, we numerically test our proposed method on a batch reactor system introduced in [110] and applied in [33, 74]. The system has $n = 4$ states, $m = 2$ inputs and

Time horizon lengths	$L = 5,\ N = 15,\ N_{\mathrm{c}} = 5$
Cost matrices	$Q = 10^3 I_p,\ R = I_m$
Safety constraint coefficients	$E = I_{m+p} \otimes \left[\begin{smallmatrix} 1 \\ -1 \end{smallmatrix}\right]$
	$f = [.1\ .1\ .5\ .1\ .4\ .4\ .4\ .4]^{\mathsf{T}}$
CVaR level[a]	$\alpha = 0.3$
Variance of v_t for SMPC/SDDPC	$\Sigma^{\mathsf{v}} = 5 \times 10^{-7} I_p$
Variance of ρ_t for SDDPC	$\Sigma^{\rho} = 10^{-7} I_{pL}$
Variance of w_t for SMPC[b]	$\Sigma^{\mathsf{w}} = \mathcal{O}^{\dagger} \Sigma^{\rho} \mathcal{O}^{\dagger\mathsf{T}}$

[a] α is used as the risk bound for chance constrained controllers.
[b] $\mathcal{O}$ is obtained given the identified model (A, B, C, D) in SMPC.

Table 4.3: Control parameters of the batch reactor system.

Controller	**Total Tracking Cost** 0s to 30s	30s to 60s	**Cumulative Violation** from 60s to 90s
DR/O-SDDPC[a]	0.02	64.2	0
DR/F-SDDPC	0.02	68.9	0
CC/F-SDDPC	0.02	64.9	0.03
DR/O-SMPC	0.02	64.2	0
DR/F-SMPC	0.02	68.0	0
CC/F-SMPC	0.02	64.9	0.01
deterministic MPC	0.09	64.6	0.20
SPC	0.18	65.5	2.23
DeePC	0.18	64.7	0.19

[a] DR – distributionally robust constrained, CC – chance constrained, O – with optimized feedback gain, F – with fixed feedback gain.

Table 4.4: Simulation results of the batch reactor system.

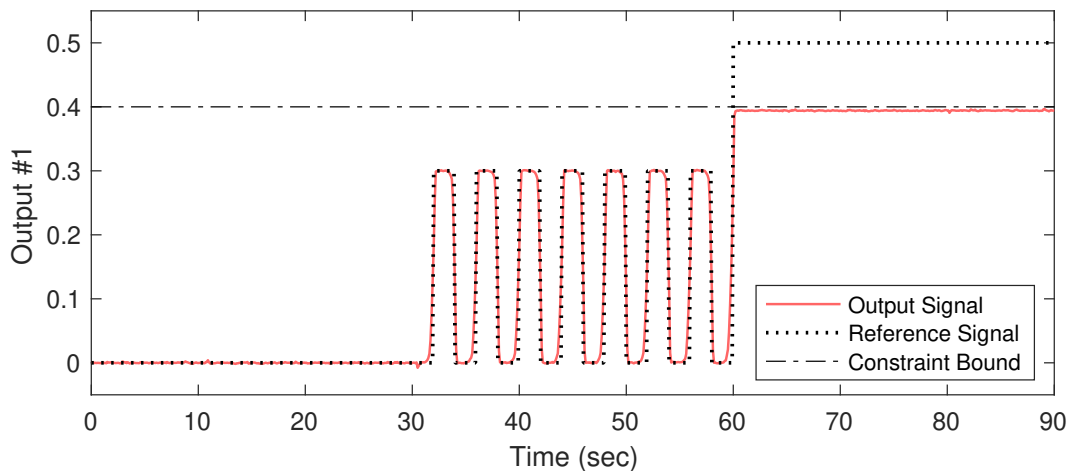

Figure 4.5: The system's first output signal with DR/O-SDDPC.

$p = 2$ outputs, and the discrete-time system matrices with sampling period 0.1s are

$$\left[\begin{array}{c|c} A & B \\ \hline C & \end{array}\right] = \left[\begin{array}{cccc|cc} 1.178 & .001 & .511 & -.403 & .004 & -.087 \\ -.051 & .661 & -.011 & .061 & .467 & .001 \\ .076 & .335 & .560 & .382 & .213 & -.235 \\ 0 & .335 & .089 & .849 & .213 & -.016 \\ \hline 1 & 0 & 1 & -1 & & \\ 0 & 1 & 0 & 0 & & \end{array}\right].$$

The process/sensor noise on each state/output follows the t-distribution of 2 DOFs scaled by 10^{-4}, which is a heavy-tailed distribution. Control parameters are reported in TABLE 4.3. We collected offline data of length $T_{\mathrm{d}} = 600$ from the noisy system, where the input data was the outcome of a PI controller $U(s) = \left[\begin{smallmatrix} 0 & -1/s \\ 2+1/s & 0 \end{smallmatrix}\right] Y(s)$ plus a white-noise signal of noise power 10^{-2}. In the online control process, the reference signal is $r_t = [0, 0]^{\mathsf{T}}$ from time 0s to time 30s, alternates between $[0, 0]^{\mathsf{T}}$ and $[0.3, 0]^{\mathsf{T}}$ from 30s to 60s, and is $r_t = [0.5, 0]^{\mathsf{T}}$ from 60s to 90s.

For comparison, we implement the simulation with different controllers. Specifically, we perform SMPC and SDDPC under different configurations of stochastic constraints and feedback gains. In addition to the SMPC and SDDPC schemes in Algorithm 5 and Algorithm 6, which utilized chance constraints in Section 4.1.1 and a fixed feedback gain in Section 4.2.3 (CC/F), we also consider analogous SMPC and SDDPC frameworks based

on distributionally robust constraints in Section 4.1.2 and an optimized feedback gain in Section 4.2.4 (DR/O). To observe separate impacts of using different stochastic constraints and different feedback policies, we also implement SMPC and SDDPC with distributionally robust constraints and a fixed feedback gain (DR/F). We also compare to DeePC, SPC and deterministic MPC as benchmarks. The model used in deterministic and stochastic MPC methods is identified from the same offline data in the data-driven controllers.

Fig. 4.5 shows the first output signal with our proposed DR/O-SDDPC method; the signal remains around 0.4 from 60s to 90s because of the safety constraint specified in TABLE 4.3. The simulation results with all examined controllers are summarized in TABLE 4.4. We evaluate (i) the controllers' tracking performance through the tracking cost from 0s to 60s and (ii) the controllers' ability to satisfy constraints according to the cumulative amount of constraint violation between 60s and 90s, when the first output signal hits the constraint margin. When the reference signal is constant (0s–30s), SMPC and SDDPC tracked better than other methods, aligning with the observation in Section 4.4.1. Comparing DR/F and CC/F methods, the controllers with DR constraints achieved lower amounts of constraint violation (60s–90s), while the tracking performance is slightly worse during 30s–60s when the reference signal has frequent step changes. Comparing DR/O and DR/F methods, we observe that the methods with optimized gain achieved lower tracking costs when the reference signal changes frequently (30s–60s).

4.5 Chapter Conclusions

We introduced a novel direct data-driven control framework named Stochastic Data-Driven Predictive Control (SDDPC). Analogous to Stochastic MPC (SMPC), SDDPC accounts for process and measurement noise in the control design, and produces closed-loop control policies through optimization. On the theoretical front, we proved that SDDPC can produce control inputs equivalent to those of SMPC under specific conditions. Simulation results indicate that the proposed approach provides benefits in terms of both cumulative stage cost and output constraint violation. Future work will seek to improve the computational efficiency of the approach, and explore the robustness with noise-corrupted offline data.

4.6 Appendices

In this section, we present proofs for several results in the chapter and introduce the Iterative Risk Allocation technique (Section 4.6.2) for solving SMPC and SDDPC optimization problems.

4.6.1 Proof of (4.18)

Proof. Define $e_t := \mathrm{col}(x_t - \hat{x}_t^-, \hat{x}_t^- - \bar{x}_t) \in \mathbb{R}^{2n}$. We first show that e_t follows the distribution

$$e_t \sim \mathcal{N}\left(0_{2n\times 1}, \begin{bmatrix} \Sigma^{\mathsf{x}} & \\ & \Lambda_{t-k} \end{bmatrix}\right) \tag{4.61}$$

for $t \in \mathbb{Z}_{[k,k+N)}$, with Λ_s in (4.19b), by induction on t. **Base Case** $t = k$**.** With $\hat{x}_k^- = \mu_k^{\mathsf{x}}$ as (4.13d) and $\bar{x}_k = \mu_k^{\mathsf{x}}$ as (4.12c), we have $e_k = \mathrm{col}(x_k - \mu_k^{\mathsf{x}}, 0_{n\times 1})$ which is distributed as $\mathcal{N}(0_{2n\times 1}, \mathrm{Diag}(\Sigma^{\mathsf{x}}, 0_{n\times n}))$ via (4.9). This shows the $t = k$ case of (4.61) given $\Lambda_0 = 0_{n\times n}$ from (4.19b). **Inductive Step.** Assume (4.61) for $t = \tau \in \mathbb{Z}_{[k,k+N-2]}$. Note the relation [98]

$$e_{\tau+1} = \Theta_0 e_\tau + \Theta_1 \begin{bmatrix} w_\tau \\ v_\tau \end{bmatrix} \tag{4.62}$$

by expressing $x_{\tau+1}, \hat{x}_{\tau+1}^-, \bar{x}_{\tau+1}$ in terms of $x_\tau, \hat{x}_\tau^-, \bar{x}_\tau, w_\tau, v_\tau$ given (4.1a), (4.12a), (4.13b), (4.13c), (4.15), where we define

$$\Theta_0 := \begin{bmatrix} A - L_{\mathsf{L}} C & 0_{n\times n} \\ L_{\mathsf{L}} C & A - BK \end{bmatrix}, \quad \Theta_1 := \begin{bmatrix} I_n & -L_{\mathsf{L}} \\ 0_{n\times n} & L_{\mathsf{L}} \end{bmatrix}. \tag{4.63}$$

Through the system (4.1) and the estimator (4.13), both w_τ and v_τ are independent of x_τ and $\hat{x}_\tau^-$ and thus independent of e_τ. It follows from the relation (4.62), the (independent) distribution of w_τ, v_τ in (4.3) and the distribution of e_τ in (4.61) that $e_{\tau+1}$ is distributed as

$$e_{\tau+1} \sim \mathcal{N}\left(0_{2n\times 1},\ \Theta_0 \begin{bmatrix} \Sigma^{\mathsf{x}} & \\ & \Lambda_{\tau-k} \end{bmatrix} \Theta_0^{\mathsf{T}} + \Theta_1 \begin{bmatrix} \Sigma^{\mathsf{w}} & \\ & \Sigma^{\mathsf{v}} \end{bmatrix} \Theta_1^{\mathsf{T}}\right). \tag{4.64}$$

The variance in (4.64) is equal to what follows, by substitution of Θ_0 and Θ_1 in (4.63) and direct matrix multiplication,

$$\begin{bmatrix} \mathcal{S}_0 - \mathcal{S}_1 - \mathcal{S}_1^{\mathsf{T}} + \mathcal{S}_2 + \Sigma^{\mathsf{x}} & \mathcal{S}_1^{\mathsf{T}} - \mathcal{S}_0 \\ \mathcal{S}_1 - \mathcal{S}_0 & \mathcal{S}_0 + (A - BK)\Lambda_{\tau-k}(A - BK)^{\mathsf{T}} \end{bmatrix} \tag{4.65}$$

where we use shortcuts $\mathcal{S}_0 := L_{\mathsf{L}}(C\Sigma^{\mathsf{x}}C^{\mathsf{T}} + \Sigma^{\mathsf{v}})L_{\mathsf{L}}^{\mathsf{T}}$, $\mathcal{S}_1 := L_{\mathsf{L}}C\Sigma^{\mathsf{x}}A^{\mathsf{T}}$ and $\mathcal{S}_2 := A\Sigma^{\mathsf{x}}A^{\mathsf{T}} + \Sigma^{\mathsf{w}} - \Sigma^{\mathsf{x}}$. Notice that $\mathcal{S}_0 = \mathcal{S}_1$ by definition of L_{L} in (4.11b), and $\mathcal{S}_1 = \mathcal{S}_2$ via (4.46a). One can also verify that $\mathcal{S}_0 + (A - BK)\Lambda_s(A - BK)^{\mathsf{T}} = \Lambda_{s+1}$ for all $s \in \mathbb{N}_{\geq 0}$, using definition (4.19b). Thus, the matrix (4.65) is equal to $\mathrm{Diag}(\Sigma^{\mathsf{x}}, \Lambda_{\tau-k+1})$, which implies that (4.64) is the $t = \tau + 1$ case of (4.61). Induction on t shows (4.61) for $t \in \mathbb{Z}_{[k,k+N)}$.

Finally, we show (4.18) for $t \in \mathbb{Z}_{[k,k+N)}$ by noting that

$$\begin{bmatrix} u_t \\ y_t \end{bmatrix} = \begin{bmatrix} \bar{u}_t \\ \bar{y}_t \end{bmatrix} + \Psi e_t + \begin{bmatrix} 0_{m\times 1} \\ v_t \end{bmatrix} \text{ with } \Psi := \begin{bmatrix} 0_{m\times n} & -K \\ C & C - DK \end{bmatrix}, \tag{4.66}$$

given (4.1b) and (4.15). With the distribution (4.61) of e_t and the distribution of v_t in (4.3), where e_t and v_t are independent, it follows from (4.66) that

$$\begin{bmatrix} u_t \\ y_t \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \bar{u}_t \\ \bar{y}_t \end{bmatrix}, \Psi \begin{bmatrix} \Sigma^{\mathsf{x}} & \\ & \Lambda_{t-k} \end{bmatrix} \Psi^{\mathsf{T}} + \begin{bmatrix} 0_{m\times m} & \\ & \Sigma^{\mathsf{v}} \end{bmatrix} \right),$$

in which the variance can be verified equal to Δ_{t-k} defined in (4.19a) through direct calculation, and thus the above distribution is equivalent to (4.18). ■

4.6.2 Iterative Risk Allocation

We record here an efficient method for solving the convex problem (SMPC), known as Iterative Risk Allocation [104], described in Algorithm 7.

To begin, note that if we fix all variables $p_{i,t}$, then problem (SMPC) is reduced into the quadratic problem

$$\begin{aligned} \underset{\bar{u},\,\theta}{\text{minimize}} \quad & \sum_{t=k}^{k+N-1} J_t(\bar{u}_t, \bar{y}_t) + \lambda_\theta\, \theta \\ \text{subject to} \quad & (4.32a) \text{ for } t \in \mathbb{Z}_{[k,k+N)}, (4.12), (4.14), (4.35), \end{aligned} \tag{4.67}$$

which can be efficiently solved. The optimal solution to (SMPC) is the infimum of the solution to (4.67) over all $p_{i,t}$ satisfying (4.32b). Hence, we solve problem (4.67) repeatedly with updated $p_{i,t}$ until the objective value converges with no significant change. The entire process shows in Algorithm 7, which extends [104, Algorithm 1] from their single chance constraint into our separate chance constraints over time steps. Newly introduced parameters are a shrinkage rate $\alpha \in (0, 1)$ and a termination threshold $\varepsilon > 0$. The initialization at line 1 ensures feasibility of problem (4.67), due to recursive feasibility.

From line 6, we obtain binary indicators $a_{i,t} \in \{0,1\}$ showing whether constraint (4.32a) is active or not for each (i,t). This indicator is utilized in the process of updating $p_{i,t}$ in lines 9-14. Note that, when the condition in line 8 is true, the update routine in lines 9-14 no longer makes change on $p_{i,t}$, so in this case the iteration terminates. In line 11, $\mathrm{cdfn}(z) := \frac{1}{2} + \frac{1}{2}\mathrm{erf}(z/\sqrt{2})$ is the cumulative density function (c.d.f.) of the standard normal distribution, with erf the error function.

Similarly, problem (SDDPC) can also be solved by Algorithm 7 with A, B, C, μ_k^{x}, $\mu_k^{\hat{\mathsf{x}}}$, $\mu_k^{\bar{\mathsf{x}}}$, Δ_s, $\bar{y}_t$ replaced by $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\boldsymbol{\mu}_k^{\mathsf{x}}$, $\boldsymbol{\mu}_k^{\hat{\mathsf{x}}}$, $\boldsymbol{\mu}_k^{\bar{\mathsf{x}}}$, $\boldsymbol{\Delta}_s$, $\bar{\mathbf{y}}_t$ respectively.

Algorithm 7 Iterative Risk Allocation for solving (SMPC)

Input: horizon lengths L, N, system matrices A, B, C, D, interpolation options $\mu_k^{\hat{\mathsf{x}}}, \mu_k^{\bar{\mathsf{x}}}$, cost matrices Q, R, constraint coefficients E, f, probability bound p, interpolation penalty coefficient λ_θ, input-output variances $\Delta_{[0,N)}$, shrinkage rate α, termination threshold ε, and the risk allocation $p_{i,t}^{\mathsf{last}}$ solved at last control step.

Output: An approximate solution $(\bar{u}, \theta, p_{i,t})$ to problem (SMPC).

1: Initialize $p_{i,t} \leftarrow p_{i,s(t)}^{\mathsf{last}}$ for $t \in \mathbb{Z}_{[k,k+N)}$ and $i \in \{1,\dots,q\}$, where $s(t) := \min(t, k+N-N_{\mathrm{c}}-1)$.

2: Initialize $J_{\mathsf{prev}}^{\star} \leftarrow +\infty$.

3: **while** `true` **do**

4: Solve $(\bar{u}, \bar{y}, \theta)$ from problem (4.67) and obtain the cost value $J^{\star}$. Record whether the constraints (4.32a) is active or not for each (i,t).

5: **if** $|J_{\mathsf{prev}}^{\star} - J^{\star}| \le \varepsilon$ **then break else** $J_{\mathsf{prev}}^{\star} \leftarrow J^{\star}$.

6: For $t \in \mathbb{Z}_{[k,k+N)}$ and $i \in \{1,\dots,q\}$, let $a_{i,t} \leftarrow 1$ if constraint (4.32a) is active for (i,t), otherwise $a_{i,t} \leftarrow 0$.

7: $a_t^{\mathsf{sum}} \leftarrow \sum_{i=1}^{q} a_{i,t}$ for all $t \in \mathbb{Z}_{[k,k+N)}$.

8: **if** $a_t^{\mathsf{sum}} \in \{0, q\}$ for all $t \in \mathbb{Z}_{[k,k+N)}$ **then break**.

9: **for** $t \in \mathbb{Z}_{[k,k+N)}$ such that $0 < a_t^{\mathsf{sum}} < q$ **do**

10: **for** all $i \in \{1,\dots,q\}$ such that $a_{i,t} = 0$ **do**

11: $p_{i,t} \leftarrow \alpha p_{i,t} + (1-\alpha)\Big(1 - \mathrm{cdfn}\Big(\dfrac{f_i - e_i^{\mathsf{T}}\,\mathrm{col}(\bar{u}_t, \bar{y}_t)}{\sqrt{e_i^{\mathsf{T}}\Delta_{t-k}e_i}}\Big)\Big)$.

12: $p_t^{\mathsf{residual}} \leftarrow p - \sum_{i=1}^{q} p_{i,t}$.

13: **for** all $i \in \{1,\dots,q\}$ such that $a_{i,t} = 1$ **do**

14: $p_{i,t} \leftarrow p_{i,t} + p_t^{\mathsf{residual}}/a_t^{\mathsf{sum}}$.

4.6.3 Proof of Lemma 4.5

Proof. Let $\kappa^+ := \kappa + N_c$, and let $|_k$ denote variables calculated at control step $k \in \{\kappa, \kappa^+\}$. Let $(\bar{u}^*, \theta^*, p^*_{i,t})|_\kappa$ be the optimal solution to problem (SMPC) at $k = \kappa$, and consider the following solution $(\bar{u}^\diamond, \theta^\diamond, p^\diamond_{i,t})|_{\kappa^+}$ at $k = \kappa^+$, cf. [102],

$$\bar{u}^\diamond_t|_{\kappa^+} := \bar{u}^*_{s(t)}|_\kappa, \quad \theta^\diamond|_{\kappa^+} := 1, \quad p^\diamond_{i,t}|_{\kappa^+} := p^*_{i,s(t)}|_\kappa, \tag{4.68}$$

for all $t \in \mathbb{Z}_{[\kappa^+, \kappa^+ + N)}$ and $i \in \mathbb{Z}_{[1,q]}$, where we let $s(t) := \min(t, \kappa + N - 1)$. In this proof, we will show that (4.68) is a feasible solution to problem (SMPC). Let $\bar{y}^*|_\kappa$ (resp. $\bar{y}^\diamond|_{\kappa^+}$) denote the resulting nominal output via (4.12), (4.14) given $(\bar{u}^*, \theta^*)|_\kappa$ (resp. $(\bar{u}^\diamond, \theta^\diamond)|_{\kappa^+}$), and we have the following.

Claim 4.5.1. *Given $(\bar{u}^\diamond, \theta^\diamond)|_{\kappa^+}$ in (4.68), the nominal output is $\bar{y}^\diamond_t|_{\kappa^+} = \bar{y}^*_{s(t)}|_\kappa$ for $t \in \mathbb{Z}_{[\kappa^+, \kappa^+ + N)}$.*

Proof. Since we choose $\theta^\diamond|_{\kappa^+} = 1$ in (4.68), the nominal states $\bar{x}_{\kappa^+}$ are the same over control steps $k \in \{\kappa, \kappa^+\}$, as

$$\bar{x}^\diamond_{\kappa^+}|_{\kappa^+} \overset{\text{via (4.12c)}}{=} \mu^{\mathsf{x}}_{\kappa^+} \overset{\text{via (4.14)}}{=} \mu^{\bar{\mathsf{x}}}_{\kappa^+} \overset{\text{via (4.36)}}{=} \bar{x}^*_{\kappa^+}|_\kappa. \tag{4.69}$$

Given the same nominal states $\bar{x}_{\kappa^+}$ in (4.69) and same nominal inputs $\bar{u}_{[\kappa^+, \kappa+N)}$ via (4.68) over control steps $k \in \{\kappa, \kappa^+\}$, the resulting nominal states and outputs are the same, i.e.,

$$\bar{x}^\diamond_t|_{\kappa^+} = \bar{x}^*_t|_\kappa, \qquad t \in \mathbb{Z}_{[\kappa^+, \kappa+N]}, \tag{4.70a}$$

$$\bar{y}^\diamond_t|_{\kappa^+} = \bar{y}^*_t|_\kappa, \qquad t \in \mathbb{Z}_{[\kappa^+, \kappa+N)}. \tag{4.70b}$$

Due to the terminal condition (4.35) at $k = \kappa$ where L is at least the system *lag* and the system is observable, the terminal state-input-output $(\bar{x}^*_{\kappa+N}, \bar{u}^*_{\kappa+N-1}, \bar{y}^*_{\kappa+N-1})|_\kappa$ is an equilibrium [48, Sec. 2.3]. (In the case where the system is unobservable, $(\bar{x}^{\mathsf{obsv}\,*}_{\kappa+N}, \bar{u}^*_{\kappa+N-1}, \bar{y}^*_{\kappa+N-1})|_\kappa$ is an equilibrium, where $\bar{x}^{\mathsf{obsv}\,*}_t$ denotes the observable component of $\bar{x}^*_t$.) Hence, with $\bar{x}^\diamond_{\kappa+N}|_{\kappa^+} = \bar{x}^*_{\kappa+N}|_\kappa$ via (4.70a) and $\bar{u}^\diamond_t|_{\kappa^+} = \bar{u}^*_{\kappa+N-1}|_\kappa$ via (4.68) for $t \in \mathbb{Z}_{[\kappa+N, \kappa^+ + N)}$, we have the nominal output

$$\bar{y}^\diamond_t|_{\kappa^+} = \bar{y}^*_{\kappa+N-1}|_\kappa, \quad t \in \mathbb{Z}_{[\kappa+N, \kappa^+ + N)},$$

which result together with (4.70b) shows the claim. ♦

We finish the proof by showing that the solution (4.68) satisfies both constraints (4.32) and (4.35). The terminal constraint (4.35) holds with solution (4.68), since we have $(\bar{u}_t^\diamond, \bar{y}_t^\diamond)|_{\kappa^+}$ for $t \in \mathbb{Z}_{[\kappa^+ + N - L, \kappa^+ + N)}$ all equal to

$$(\bar{u}_t^\diamond, \bar{y}_t^\diamond)|_{\kappa^+} = (\bar{u}_{s(t)}^*, \bar{y}_{s(t)}^*)|_\kappa = (\bar{u}_{\kappa+N-1}^*, \bar{y}_{\kappa+N-1}^*)|_\kappa,$$

where the first equality is from (4.68) and Claim 4.5.1, and the second equality is because constraint (4.35) holds at $k = \kappa$. Before showing satisfaction of (4.32), we claim a useful result.

Claim 4.5.2. *For Δ_s in (4.19), we have $\Delta_0 \preceq \Delta_1 \preceq \cdots \preceq \Delta_{N-1}$.*

Proof. Given (4.19a), the result follows from the fact $\Lambda_0 \preceq \Lambda_1 \preceq \cdots \preceq \Lambda_{N-1}$, which is clear from (4.19b). ♦

Define $\mathcal{R}(\Delta_{t-k}) \subseteq \mathbb{R}^m \times \mathbb{R}^p \times \mathbb{R}^q$ the set of all $(\bar{u}_t, \bar{y}_t, p_{\cdot,t})$ satisfying (4.32), where we let $p_{\cdot,t} := \mathrm{col}(p_{1,t}, \ldots, p_{q,t}) \in \mathbb{R}^q$. To show that constraint (4.32) is satisfied by solution (4.68), it is equivalent to show that

$$(\bar{u}_t^\diamond, \bar{y}_t^\diamond, p_{\cdot,t}^\diamond)|_{\kappa^+} = (\bar{u}_{s(t)}^*, \bar{y}_{s(t)}^*, p_{\cdot,s(t)}^*)|_\kappa \in \mathcal{R}(\Delta_{s(t)-\kappa}) \subseteq \mathcal{R}(\Delta_{t-\kappa^+})$$

for all $t \in \mathbb{Z}_{[\kappa^+, \kappa^+ + N)}$, where the first equality uses (4.68) and Claim 4.5.1, the belong sign ($\in$) is because constraint (4.32) holds at $k = \kappa$, and the final inclusion ($\subseteq$) comes from the fact $s(t) - \kappa \geq t - \kappa^+$ (implied by definition of $s(t), \kappa^+$) for $t \in \mathbb{Z}_{[\kappa^+, \kappa^+ + N)}$ and the fact $\mathcal{R}(\Delta_0) \supseteq \mathcal{R}(\Delta_1) \supseteq \ldots \supseteq \mathcal{R}(\Delta_{N-1})$, which is obtained given the definition of $\mathcal{R}(\cdot)$ referring to (4.32), given Claim 4.5.2 and given the fact $\mathrm{icdfn}(1 - p_{i,t}) > 0$ for all $p_{i,t} < p \leq \frac{1}{2}$.

Thus, the solution (4.68) at $k = \kappa^+$ satisfies both constraints (4.32) and (4.35), and the recursive feasibility is proved. ■

4.6.4 Proof of Lemma 4.6

Proof. Let $J(u_t, y_t) := \|y_t - r\|_Q^2 + \|u_t\|_R^2$ be the cost (2.2) with the constant reference. At control step k, let $(\bar{u}^*, \theta^*, p_{i,t}^*)|_k$ be the optimal solution to problem (SMPC), and the optimal value is

$$V_k^* := \sum_{t=k}^{k+N-1} J(\bar{u}_t^*, \bar{y}_t^*)|_k + \lambda_\theta \theta^*|_k, \tag{4.71}$$

where $\bar{y}^*|_k$ is the resulting nominal output given $(\bar{u}^*, \theta^*)|_k$. Through (4.68) with $(k, k+N_\mathrm{c})$ in place of (κ, κ^+), we have a feasible solution $(\bar{u}^\diamond, \theta^\diamond, p^\diamond_{i,t})|_{k+N_\mathrm{c}}$ to problem (SMPC) at the next control step $k+N_\mathrm{c}$, where the objective value is

$$\begin{aligned} V^\diamond_{k+N_\mathrm{c}} &:= \sum_{t=k+N_\mathrm{c}}^{k+N_\mathrm{c}+N-1} J(\bar{u}^\diamond_t, \bar{y}^\diamond_t)|_{k+N_\mathrm{c}} + \lambda_\theta \theta^\diamond|_{k+N_\mathrm{c}} \\ &= \sum_{t=k+N_\mathrm{c}}^{k+N_\mathrm{c}+N-1} J(\bar{u}^*_{s(t)}, \bar{y}^*_{s(t)})|_k + \lambda_\theta \\ &= \sum_{t=k+N_\mathrm{c}}^{k+N-1} J(\bar{u}^*_t, \bar{y}^*_t)|_k + N_\mathrm{c} J^{\mathsf{ter}}_k + \lambda_\theta \end{aligned} \tag{4.72}$$

with $J^{\mathsf{ter}}_k := J(\bar{u}^*_{k+N-1}, \bar{y}^*_{k+N-1})|_k$, where the second equality is via (4.68), and the third equality used the definition of $s(t)$ in (4.68). Recall the sets $\mathcal{R}(\cdot)$ defined in 4.6.3. Let J_{sup} be the supremum of $J(\bar{u}_t, \bar{y}_t)$ over all $\bar{u}_t, \bar{y}_t$ in $\mathcal{R}(\Delta_{N-1})$ with some $p_{i,t}$; such J_{sup} is finite since $\mathcal{R}(\Delta_{N-1})$ is bounded given $\{z | Ez \leq f\}$ bounded. Since $(\bar{u}^*_{k+N-1}, \bar{y}^*_{k+N-1})|_k$ is in $\mathcal{R}(\Delta_{N-1})$ by feasibility, J^{ter}_k is upper bounded by J_{sup}. We therefore have

$$\begin{aligned} &V^*_{k+N_\mathrm{c}} - V^*_k \leq V^\diamond_{k+N_\mathrm{c}} - V^*_k \\ &= N_\mathrm{c} J^{\mathsf{ter}}_k + \lambda_\theta (1 - \theta^*|_k) - \sum_{t=k}^{k+N_\mathrm{c}-1} J(\bar{u}^*_t, \bar{y}^*_t)|_k \\ &\leq N_\mathrm{c} J_{\mathsf{sup}} + \lambda_\theta - \sum_{t=k}^{k+N_\mathrm{c}-1} J(\bar{u}^*_t, \bar{y}^*_t)|_k, \end{aligned} \tag{4.73}$$

where the first inequality is due to the optimality of $V^*_{k+N_\mathrm{c}}$ (i.e., $V^*_{k+N_\mathrm{c}} \leq V^\diamond_{k+N_\mathrm{c}}$), the equality is by substituting (4.71) and (4.72) and canceling identical terms, and the final inequality is due to $\theta^*|_k \in [0,1]$ and $J^{\mathsf{ter}}_k \leq J_{\mathsf{sup}}$. Using (4.21) with N_c in place of N, the expected cost over the control horizon $[k, k+N_\mathrm{c})$ is (cf. [74, 75, 76])

$$\begin{aligned} \sum_{t=k}^{k+N_\mathrm{c}-1} \mathbb{E}[J(u_t, y_t)] &= \sum_{t=k}^{k+N_\mathrm{c}-1} J(\bar{u}^*_t, \bar{y}^*_t)|_k + J_{\mathsf{var}} \\ &\leq N_\mathrm{c} J_{\mathsf{sup}} + J_{\mathsf{var}} + \lambda_\theta + (V^*_{k+N_\mathrm{c}} - V^*_k) \end{aligned} \tag{4.74}$$

with $J_{\mathsf{var}} := \sum_{s=0}^{N_c-1} \mathrm{Trace}(\Delta_s \mathrm{Diag}(R, Q))$, and the inequality above applied (4.73). Summing (4.74) over control steps $k \in \{0, N_c, 2N_c, \ldots, (I-1)N_c\}$ with some $I \in \mathbb{N}$, we have

$$\sum_{t=0}^{IN_c-1} \mathbb{E}[J(u_t, y_t)] \leq I(N_c J_{\mathsf{sup}} + J_{\mathsf{var}} + \lambda_\theta) + (V^*_{IN_c} - V^*_0),$$

and thus the result is obtained by dividing the above by $T := IN_c$ and taking $I \to \infty$, with $c := J_{\mathsf{sup}} + (J_{\mathsf{var}} + \lambda_\theta)/N_c$. ■

4.6.5 Proof of Lemma 4.7

Proof. Let (x^d, u^d, y^d) be the state-input-output trajectory of (2.1), and define $X_1, X_2 \in \mathbb{R}^{n \times h}$ as

$$X_1 := \left[x^d_1, x^d_2, \ldots, x^d_h\right], \quad X_2 := \left[x^d_{1+L}, x^d_{2+L}, \ldots, x^d_{h+L}\right].$$

It follows by straightforward algebra that data matrices satisfy

$$X_2 = A^L X_1 + \mathcal{C} U_1, \tag{4.75a}$$
$$Y_1 = \mathcal{O} X_1 + \mathcal{G} U_1, \tag{4.75b}$$
$$Y_2 = C X_2 + D U_2. \tag{4.75c}$$

Under our assumptions of controllability and persistent excitation, it follows from [31, Corollary 2(iii)] that the matrix $\mathrm{col}(X_1, U_1, U_2)$ has full row rank. Moreover, $\begin{bmatrix} I_{mL} & \\ \mathcal{G} & \mathcal{O} \end{bmatrix}$ has full column rank, as it is block lower triangular and its diagonal blocks each has full column rank (Section 4.3.1).

First, the matrix Y_2 can be represented in terms of X_1, U_1, U_2 by combining (4.75a) and (4.75c) and eliminating X_2, i.e.,

$$Y_2 = \left[C\mathcal{C}, CA^L, D\right] \mathrm{col}(U_1, X_1, U_2). \tag{4.76}$$

We can also represent $\mathrm{col}(U_1, Y_1, U_2)$ in terms of X_1, U_1, U_2 using (4.75b) as

$$\mathrm{col}(U_1, Y_1, U_2) = \mathrm{Diag}\left(\begin{bmatrix} I_{mL} & \\ \mathcal{G} & \mathcal{O} \end{bmatrix}, I_m \right) \mathrm{col}(U_1, X_1, U_2).$$

As we know that $\mathrm{Diag}\Big(\begin{bmatrix} I_{mL} & \\ \mathcal{G} & \mathcal{O} \end{bmatrix}, I_m\Big)$ has full column rank and $\mathrm{col}(U_1, X_1, U_2)$ has full row rank, the pseudo-inverse of above is [111]

$$\mathrm{col}(U_1, Y_1, U_2)^\dagger = \mathrm{col}(U_1, X_1, U_2)^\dagger \,\mathrm{Diag}\Big(\begin{bmatrix} I_{mL} & \\ \mathcal{G} & \mathcal{O} \end{bmatrix}, I_m\Big)^\dagger.$$

By multiplying (4.76) and the relation above, we find the result

$$\begin{aligned} Y_2 \,\mathrm{col}(U_1, Y_1, U_2)^\dagger &= \big[C\mathcal{C}, CA^L, D\big] \,\mathrm{Diag}\Big(\begin{bmatrix} I_{mL} & \\ \mathcal{G} & \mathcal{O} \end{bmatrix}, I_m\Big)^\dagger \\ &= \Big[\big[C\mathcal{C}, CA^L\big] \big[\begin{smallmatrix} I_{mL} & \\ \mathcal{G} & \mathcal{O} \end{smallmatrix}\big]^\dagger \quad D\Big] \overset{\text{via (4.38)}}{=} [\mathbf{\Gamma}_{\mathsf{U}}, \mathbf{\Gamma}_{\mathsf{Y}}, D]. \end{aligned}$$

■

4.6.6 Proof of Lemma 4.8

Proof. Before proving (4.41), we start with an intermediate result Claim 4.8.1. Let $\xi_t := \mathrm{col}(u_{[t-L,t)}, x_{t-L}, w_{[t-L,t)}) \in \mathbb{R}^{n_\xi}$ with $n_\xi := mL + n(L+1)$.

Claim 4.8.1. *For system (4.1) and the auxiliary state $\mathbf{x}_t$ in (4.40), we have $x_t = \Phi_{\mathsf{orig}}\,\xi_t$ and $\mathbf{x}_t = \Phi_{\mathsf{aux}}\,\xi_t$, where we define matrices $\Phi_{\mathsf{orig}} \in \mathbb{R}^{n \times n_\xi}$ and $\Phi_{\mathsf{aux}} \in \mathbb{R}^{n_{\mathsf{aux}} \times n_\xi}$,*

$$\Phi_{\mathsf{orig}} := [\mathcal{C}, A^L, \mathcal{C}_{\mathsf{w}}], \quad \Phi_{\mathsf{aux}} := \begin{bmatrix} I_{mL} & & \\ \mathcal{G} & \mathcal{O} & \mathcal{G}_{\mathsf{w}} \\ & & I_L \otimes \mathcal{O} \end{bmatrix}$$

with matrices $\mathcal{O}, \mathcal{C}, \mathcal{G}$ in Section 4.3.1, $\mathcal{C}_{\mathsf{w}} := [A^{L-1}, \dots, A, I_n]$ and

$$\mathcal{G}_{\mathsf{w}} := \begin{bmatrix} 0_{p\times n} & & & \\ C & 0_{p\times n} & & \\ \vdots & \vdots & \vdots & \\ CA^{L-2} & \cdots & C & 0_{p\times n} \end{bmatrix}.$$

Proof. Given the system model (4.1), the state x_t and noise-free outputs $y^\circ_{[t-L,t)}$ can be expressed in terms of the previous state x_{t-L}, inputs $u_{[t-L,t)}$ and disturbances $w_{[t-L,t)}$ via

$$x_t = A^L\, x_{t-L} + \mathcal{C}\, u_{[t-L,t)} + \mathcal{C}_{\mathsf{w}}\, w_{[t-L,t)}, \tag{4.77a}$$

$$y^\circ_{[t-L,t)} = \mathcal{O}\, x_{t-L} + \mathcal{G}\, u_{[t-L,t)} + \mathcal{G}_{\mathsf{w}}\, w_{[t-L,t)}. \tag{4.77b}$$

Thus, we have $x_t = \Phi_{\mathsf{orig}}\,\xi_t$ given (4.77a) and the definitions of ξ_t and Φ_{orig}. Given the definition of $\mathbf{x}_t$ in (4.40) with $\rho_t := \mathcal{O} w_t$, we have $\mathbf{x}_t = \Phi_{\mathsf{aux}}\,\xi_t$ implied by (4.77b). ♦

Given Claim 4.8.1, we develop another intermediate result Claim 4.8.2 which directly implies (4.41b) given (4.1b).

Claim 4.8.2. *For system* (4.1) *and the auxiliary state* $\mathbf{x}_t$ *in* (4.40)*, we have* $Cx_t = \mathbf{C}\mathbf{x}_t$.

Proof. With Claim 4.8.1, it suffices to show that $C\Phi_{\mathsf{orig}} = \mathbf{C}\Phi_{\mathsf{aux}}$. Given the definitions of $\Phi_{\mathsf{orig}}, \Phi_{\mathsf{aux}}, \mathbf{C}$, we calculate $C\Phi_{\mathsf{orig}}$ as

$$C\Phi_{\mathsf{orig}} = [C\mathcal{C}, CA^L, C\mathcal{C}_{\mathsf{w}}]$$

and calculate $\mathbf{C}\Phi_{\mathsf{aux}}$ as

$$\begin{aligned}\mathbf{C}\Phi_{\mathsf{aux}} &= [\mathbf{\Gamma}_{\mathsf{U}} + \mathbf{\Gamma}_{\mathsf{Y}}\mathcal{G},\, \mathbf{\Gamma}_{\mathsf{Y}}\mathcal{O},\, \mathbf{\Gamma}_{\mathsf{Y}}\mathcal{G}_{\mathsf{w}} + (\mathbf{F} - \mathbf{\Gamma}_{\mathsf{Y}}\mathbf{E})(I_L \otimes \mathcal{O})] \\ &= [\mathbf{\Gamma}_{\mathsf{U}} + \mathbf{\Gamma}_{\mathsf{Y}}\mathcal{G},\, \mathbf{\Gamma}_{\mathsf{Y}}\mathcal{O},\, C\mathcal{C}_{\mathsf{w}}] = [C\mathcal{C}, CA^L, C\mathcal{C}_{\mathsf{w}}],\end{aligned}$$

where the second equality used the facts that $C\mathcal{C}_{\mathsf{w}} = \mathbf{F}(I_L \otimes \mathcal{O})$ and $\mathcal{G}_{\mathsf{w}} = \mathbf{E}(I_L \otimes \mathcal{O})$ which can be verified from the definitions of $\mathbf{E}, \mathbf{F}, \mathcal{C}_{\mathsf{w}}, \mathcal{G}_{\mathsf{w}}$, and the last equality above used the relation

$$[\mathbf{\Gamma}_{\mathsf{U}} + \mathbf{\Gamma}_{\mathsf{Y}}\mathcal{G},\, \mathbf{\Gamma}_{\mathsf{Y}}\mathcal{O}] = [\mathbf{\Gamma}_{\mathsf{U}},\, \mathbf{\Gamma}_{\mathsf{Y}}] \begin{bmatrix} I_{mL} & \\ \mathcal{G} & \mathcal{O} \end{bmatrix} = [C\mathcal{C}, CA^L]$$

where the last equality is due to the definition $[\mathbf{\Gamma}_{\mathsf{U}},\, \mathbf{\Gamma}_{\mathsf{Y}}] := [C\mathcal{C}, CA^L] \begin{bmatrix} I_{mL} & \\ \mathcal{G} & \mathcal{O} \end{bmatrix}^{\dagger}$ where $\begin{bmatrix} I_{mL} & \\ \mathcal{G} & \mathcal{O} \end{bmatrix}$ has full column rank. Comparing the above results of calculation, we have $C\Phi_{\mathsf{orig}} = \mathbf{C}\Phi_{\mathsf{aux}}$, and thus the result follows from Claim 4.8.1. ♦

We finally prove (4.41a). Using the definitions of $\mathbf{x}_t, \mathbf{w}_t, \mathbf{A}, \mathbf{B}$, where $\mathbf{A}$ consists of upper-shift matrices and the matrix $\mathbf{C}$, we know by direct calculation that $\mathbf{A}\mathbf{x}_t + \mathbf{B}u_t + \mathbf{w}_t$ is equal to

$$\mathrm{col}\Big(\begin{bmatrix} u_{[t-L+1,t)} \\ u_t \end{bmatrix}, \begin{bmatrix} y^{\circ}_{[t-L+1,t)} \\ \mathbf{C}\mathbf{x}_t + Du_t \end{bmatrix}, \begin{bmatrix} \rho_{[t-L+1,t)} \\ \rho_t \end{bmatrix} \Big),$$

which by definition is $\mathbf{x}_{t+1}$, given the fact $\mathbf{C}\mathbf{x}_t + Du_t = y_t - v_t = y^{\circ}_t$ using (4.41b). Thus, (4.41a) is obtained. ■

4.6.7 Proof of Lemma 4.9

Proof. The pair $(\mathbf{A}, \mathbf{C})$ is detectable by definition since there exists a matrix $\mathbf{L}^* := \mathrm{col}(0_{mL\times p}, 0_{p(L-1)\times p}, I_p, 0_{pL^2\times p})$ such that $\mathbf{A} - \mathbf{L}^*\mathbf{C}$ equal to

$$\mathrm{Diag}\Big(\begin{bmatrix} & I_{m(L-1)} \\ 0_{m\times m} & \end{bmatrix}, \begin{bmatrix} & I_{p(L-1)} \\ 0_{p\times p} & \end{bmatrix}, \begin{bmatrix} & I_{pL(L-1)} \\ 0_{pL\times pL} & \end{bmatrix}\Big)$$

is Schur stable.

We show stabilizability of $(\mathbf{A}, \mathbf{B})$ and $(\mathbf{A}, \mathbf{\Sigma}^{\mathsf{w}})$ by establishing stabilizing gains. Recall $\Phi_{\mathsf{aux}} \in \mathbb{R}^{n_{\mathsf{aux}}\times n_\xi}$ and $\Phi_{\mathsf{orig}} \in \mathbb{R}^{n\times n_\xi}$ defined in Claim 4.8.1, with $n_{\mathsf{aux}} := mL + pL + pL^2$ and $n_\xi := mL + n + nL$. We start with some basic results.

Claim 4.9.1. *For matrices $\mathbf{A}, \mathbf{B}$ in (4.41) and $\mathbf{\Sigma}^{\mathsf{w}}$ in (4.43), we have*

$$\mathbf{A}\Phi_{\mathsf{aux}} = \Phi_{\mathsf{aux}}\widetilde{A}, \quad \mathbf{B} = \Phi_{\mathsf{aux}}\widetilde{B}, \quad \mathbf{\Sigma}^{\mathsf{w}} = \Phi_{\mathsf{aux}}\widetilde{\Sigma}^{\mathsf{w}}\Phi_{\mathsf{aux}}^{\mathsf{T}},$$

with matrices $\widetilde{A} \in \mathbb{R}^{n_\xi\times n_\xi}, \widetilde{B} \in \mathbb{R}^{n_\xi\times m}, \widetilde{\Sigma}^{\mathsf{w}} \in \mathbb{S}_+^{n_\xi}$ defined as

$$\widetilde{A} := \left[\begin{array}{cc|c|cc} & I_{m(L-1)} & & & \\ 0_{m\times m} & & & & \\ \hline B & 0_{n\times m(L-1)} & A & I_n & 0_{n\times n(L-1)} \\ \hline & & & & I_{n(L-1)} \\ & & & 0_{n\times n} & \end{array}\right], \tag{4.78}$$

$$\widetilde{B} := \left[\begin{array}{c} 0_{m(L-1)\times m} \\ I_m \\ \hline 0_{n\times m} \\ \hline 0_{nL\times m} \end{array}\right], \quad \widetilde{\Sigma}^{\mathsf{w}} := \begin{bmatrix} 0_{(n_\xi-n)\times(n_\xi-n)} & \\ & \Sigma^{\mathsf{w}} \end{bmatrix}.$$

Proof. Direct calculation. ♦

Claim 4.9.2. *We have $\Phi_{\mathsf{orig}} = \Phi\Phi_{\mathsf{aux}}$ for matrices $\Phi_{\mathsf{orig}}, \Phi_{\mathsf{aux}}$ defined in Claim 4.8.1 and matrix $\Phi := [\Phi_{\mathsf{U}}, \Phi_{\mathsf{Y}}, \Phi_{\mathsf{P}}] \in \mathbb{R}^{n\times n_{\mathsf{aux}}}$ whose sub-blocks are defined as*

$$[\Phi_{\mathsf{U}}, \Phi_{\mathsf{Y}}] := [\mathcal{C}, A^L]\begin{bmatrix} I_{mL} & \\ \mathcal{G} & \mathcal{O} \end{bmatrix}^\dagger \in \mathbb{R}^{n\times(mL+pL)},$$

$$\Phi_{\mathsf{W}} := \mathcal{C}_{\mathsf{w}} - \Phi_{\mathsf{Y}}\mathcal{G}_{\mathsf{w}} \in \mathbb{R}^{n\times nL}, \ \Phi_{\mathsf{P}} := \Phi_{\mathsf{W}}(I_L \otimes \mathcal{O}^\dagger) \in \mathbb{R}^{n\times pL^2}.$$

Proof. Direct calculation, given $\mathcal{O}$ of full column rank. ♦

With matrix Φ in Claim 4.9.2, define matrices $\mathbf{K}^* \in \mathbb{R}^{m \times n_{\mathsf{aux}}}$, $\mathbf{K}^{\mathsf{w}} \in \mathbb{R}^{n_{\mathsf{aux}} \times n_{\mathsf{aux}}}$, $\widetilde{K}^* \in \mathbb{R}^{m \times n_\xi}$ and $\widetilde{K}^{\mathsf{w}} \in \mathbb{R}^{n_\xi \times n_\xi}$,

$$\mathbf{K}^* := K\Phi, \qquad \mathbf{K}^{\mathsf{w}} := \Phi_{\mathsf{aux}}^{\dagger\mathsf{T}} \operatorname{col}(0_{(n_\xi - n)\times n}, K^{\mathsf{w}})\, \Phi \tag{4.79a}$$

$$\widetilde{K}^* := K\Phi_{\mathsf{orig}}, \qquad \widetilde{K}^{\mathsf{w}} := \operatorname{col}(0_{(n_\xi - n)\times n}, K^{\mathsf{w}})\, \Phi_{\mathsf{orig}} \tag{4.79b}$$

where K is the feedback gain from (4.16) and $K^{\mathsf{w}} \in \mathbb{R}^{n \times n}$ is a matrix such that $A - \Sigma^{\mathsf{w}} K^{\mathsf{w}}$ is Schur stable. We then have another intermediate result.

Claim 4.9.3. *For matrices $\widetilde{A}, \widetilde{B}, \widetilde{\Sigma}^{\mathsf{w}}$ in (4.78) and $\widetilde{K}^*, \widetilde{K}^{\mathsf{w}}$ in (4.79b), both $\widetilde{A} - \widetilde{B}\widetilde{K}^*$ and $\widetilde{A} - \widetilde{\Sigma}^{\mathsf{w}}\widetilde{K}^{\mathsf{w}}$ are Schur stable.*

Proof. Define $\xi_t := \operatorname{col}(u_{[t-L,t)}, x_{t-L}, w_{[t-L,t)}) \in \mathbb{R}^{n_\xi}$ and $\delta_t := \operatorname{col}(0_{(n_\xi - n)\times 1}, w_t) \in \mathbb{R}^{n_\xi}$. We have the relation

$$\xi_{t+1} = \widetilde{A}\xi_t + \widetilde{B}u_t + \delta_t \tag{4.80}$$

which can be verified given the system model (4.1a) and the definition of $\widetilde{A}, \widetilde{B}$ in (4.78).

To show that $\widetilde{A} - \widetilde{B}\widetilde{K}^*$ is stable, consider the following process of system (4.1a) starting at time $t = -L$: the initial state x_{-L}, the inputs $u_{[-L,0)}$ and the noises $w_{[-L,0)}$ are arbitrarily chosen (i.e., ξ_0 is arbitrary), the noise is $w_t = 0$ for $t \geq 0$, and the inputs u_t for $t \geq 0$ are generated by state feedback $u_t = -Kx_t$. With this process, we have $x_{t+1} = (A - BK)x_t$ for $t \geq 0$, and hence $x_t \to 0$ as $t \to \infty$ because $A - BK$ is Schur stable. We therefore have $u_t, w_t \to 0$ and thus $\xi_t \to 0$ as $t \to \infty$, given the definition of ξ_t and the relations $u_t = -Kx_t$ and $w_t = 0$ for $t \geq 0$. On the other hand, with the process, we have $\delta_t = 0$ since $w_t = 0$ for $t \geq 0$, and the state feedback can be written as $u_t = -K\Phi_{\mathsf{orig}}\xi_t$ given the relation $x_t = \Phi_{\mathsf{orig}}\xi_t$ from Claim 4.8.1, and thus $u_t = -\widetilde{K}^*\xi_t$ with $\widetilde{K}^*$ defined in (4.79b). Therefore, the evolution (4.80) is reduced as $\xi_{t+1} = (\widetilde{A} - \widetilde{B}\widetilde{K}^*)\xi_t$ for $t \geq 0$, which implies that $\xi_t = (\widetilde{A} - \widetilde{B}\widetilde{K}^*)^t\xi_0$ for $t \geq 0$. Since $\xi_t \to 0$ as $t \to \infty$ and ξ_0 is arbitrarily chosen, we conclude that $(\widetilde{A} - \widetilde{B}\widetilde{K}^*)^t \to 0$ as $t \to \infty$, i.e., $\widetilde{A} - \widetilde{B}\widetilde{K}^*$ is Schur stable.

To show that $\widetilde{A} - \widetilde{\Sigma}^{\mathsf{w}}\widetilde{K}^{\mathsf{w}}$ is stable, consider a similar process of system (4.1a) from initial time $t = -L$: the initial state x_{-L}, the inputs $u_{[-L,0)}$ and the noises $w_{[-L,0)}$ are arbitrarily chosen (i.e., ξ_0 is arbitrary), the input is $u_t = 0$ for $t \geq 0$, and the disturbances w_t for $t \geq 0$ are realized as $w_t = -\Sigma^{\mathsf{w}}K^{\mathsf{w}}x_t$. With the process, we have $x_{t+1} = (A - \Sigma^{\mathsf{w}}K^{\mathsf{w}})x_t$ for $t \geq 0$, and hence $x_t \to 0$ as $t \to \infty$ because $A - \Sigma^{\mathsf{w}}K^{\mathsf{w}}$ is Schur stable. We therefore have $u_t, w_t \to 0$ and thus $\xi_t \to 0$ as $t \to \infty$, given the definition of ξ_t and the relations $u_t = 0$ and $w_t = -\Sigma^{\mathsf{w}}K^{\mathsf{w}}x_t$ for $t \geq 0$. On the other hand, with the process, we have $\delta_t = -\widetilde{\Sigma}^{\mathsf{w}}\widetilde{K}^{\mathsf{w}}\xi_t$,

given the definition of δ_t, the relation $w_t = -\Sigma^{\mathsf{w}} K^{\mathsf{w}} x_t$, the relation $x_t = \Phi_{\mathsf{orig}} \xi_t$ from Claim 4.8.1 and the definitions of $\widetilde{K}^{\mathsf{w}}, \widetilde{\Sigma}^{\mathsf{w}}$ in (4.79b), (4.78). Therefore, the evolution (4.80) is reduced as $\xi_{t+1} = (\widetilde{A} - \widetilde{\Sigma}^{\mathsf{w}} \widetilde{K}^{\mathsf{w}}) \xi_t$ for $t \geq 0$, which implies that $\xi_t = (\widetilde{A} - \widetilde{\Sigma}^{\mathsf{w}} \widetilde{K}^{\mathsf{w}})^t \xi_0$ for $t \geq 0$. Since $\xi_t \to 0$ as $t \to \infty$ and ξ_0 is arbitrarily chosen, we conclude that $(\widetilde{A} - \widetilde{\Sigma}^{\mathsf{w}} \widetilde{K}^{\mathsf{w}})^t \to 0$ as $t \to \infty$, i.e., $\widetilde{A} - \widetilde{\Sigma}^{\mathsf{w}} \widetilde{K}^{\mathsf{w}}$ is Schur stable. ♦

It follows from Claim 4.9.1 and the definitions (4.79) that

$$(\mathbf{A} - \mathbf{B}\mathbf{K}^*)\Phi_{\mathsf{aux}} = \Phi_{\mathsf{aux}}(\widetilde{A} - \widetilde{B}\widetilde{K}^*), \tag{4.81a}$$

$$(\mathbf{A} - \mathbf{\Sigma}^{\mathsf{w}}\mathbf{K}^{\mathsf{w}})\Phi_{\mathsf{aux}} = \Phi_{\mathsf{aux}}(\widetilde{A} - \widetilde{\Sigma}^{\mathsf{w}}\widetilde{K}^{\mathsf{w}}), \tag{4.81b}$$

given $\Phi_{\mathsf{orig}} = \Phi\Phi_{\mathsf{aux}}$ as Claim 4.9.2 and $\Phi_{\mathsf{aux}}^{\mathsf{T}} \Phi_{\mathsf{aux}}^{\dagger\mathsf{T}} = I_{n_\xi}$ for Φ_{aux} of full column rank. By applying (4.81a) repeatedly, we have $(\mathbf{A} - \mathbf{B}\mathbf{K}^*)^t \Phi_{\mathsf{aux}} = \Phi_{\mathsf{aux}}(\widetilde{A} - \widetilde{B}\widetilde{K}^*)^t$ for all $t \in \mathbb{N}$. Combining this relation with the fact $(\widetilde{A} - \widetilde{B}\widetilde{K}^*)^t \to 0$ as $t \to \infty$ via Schur stability in Claim 4.9.3, we have

$$(\mathbf{A} - \mathbf{B}\mathbf{K}^*)^t \Phi_{\mathsf{aux}} \to 0 \quad \text{as} \quad t \to \infty, \tag{4.82}$$

which implies Schur stability of $\mathbf{A} - \mathbf{B}\mathbf{K}^*$ through Claim 4.9.4.

Claim 4.9.4. *For matrices* $\mathbf{A}, \mathbf{B}$ *in* (4.41) *and* $\mathbf{K}^*$ *in* (4.79a)*, if* (4.82) *holds, then* $\mathbf{A} - \mathbf{B}\mathbf{K}^*$ *is Schur stable.*

Proof. We calculate $\mathbf{A} - \mathbf{B}\mathbf{K}^*$ as

$$\left[\begin{array}{c|c} \overbrace{\left[\begin{array}{c|c} \begin{matrix} 0 & I_{m(L-1)} \\ \hline \multicolumn{2}{c}{-K\Phi_{\mathsf{U}}} \end{matrix} & \begin{matrix} 0 \\ -K\Phi_{\mathsf{Y}} \end{matrix} \\ \hline \begin{matrix} 0 \\ (C - DK)\Phi_{\mathsf{U}} \end{matrix} & \begin{matrix} 0 & I_{p(L-1)} \\ \hline \multicolumn{2}{c}{(C - DK)\Phi_{\mathsf{Y}}} \end{matrix} \end{array}\right]}^{=:\mathcal{A}} & \begin{matrix} 0 \\ -K\Phi_{\mathsf{P}} \\ 0 \\ \mathbf{F} - \mathbf{\Gamma}_{\mathsf{Y}}\mathbf{E} \end{matrix} \\ \hline 0 & \begin{matrix} I_{pL(L-1)} \\ 0_{pL \times pL} \end{matrix} \end{array}\right],$$

which is Schur stable if, and only if, its sub-matrix $\mathcal{A}$ is Schur stable. Moreover, since both $\mathbf{A} - \mathbf{B}\mathbf{K}^* = \begin{bmatrix} \mathcal{A} & * \\ 0 & * \end{bmatrix}$ and $\Phi_{\mathsf{aux}} = \begin{bmatrix} \mathcal{S} & * \\ 0 & * \end{bmatrix}$ are upper block-triangular, $(\mathbf{A} - \mathbf{B}\mathbf{K}^*)^t \Phi_{\mathsf{aux}} = \begin{bmatrix} \mathcal{A}^t\mathcal{S} & * \\ 0 & * \end{bmatrix}$ is also upper block-triangular. Since $(\mathbf{A} - \mathbf{B}\mathbf{K}^*)^t \Phi_{\mathsf{aux}} \to 0$ as $t \to \infty$ via (4.82), its sub-matrix yields $\mathcal{A}^t\mathcal{S} \to 0$ as $t \to \infty$.

Let $\mathcal{L} := \lim_{t\to\infty} \mathcal{A}^t$ denote the limiting value. Given the definition $[\Phi_{\mathsf{U}}, \Phi_{\mathsf{Y}}] := [\mathcal{C}, A^L]\mathcal{S}^\dagger$ where $\mathcal{S}$ denotes $\begin{bmatrix} I_{mL} & \\ \mathcal{G} & \mathcal{O} \end{bmatrix}$, $\mathcal{A}$ can be written as $\mathcal{A} = \mathcal{D} + \mathcal{E}\mathcal{S}^\dagger$ where

$$\mathcal{D} := \mathrm{Diag}\Big(\begin{bmatrix} & I_{m(L-1)} \\ 0_{m\times m} & \end{bmatrix}, \begin{bmatrix} & I_{p(L-1)} \\ 0_{p\times p} & \end{bmatrix} \Big),$$
$$\mathcal{E} := \mathrm{col}(0_{m(L-1)\times n}, -K, 0_{p(L-1)\times n}, C - DK)\,[\mathcal{C}, A^L].$$

Define $\mathcal{P} := I - \mathcal{S}\mathcal{S}^\dagger$ as a projection matrix. With the fact $\mathcal{S}^\dagger\mathcal{P} = \mathcal{S}^\dagger(I - \mathcal{S}\mathcal{S}^\dagger) = 0$, it follows that

$$\mathcal{A}\mathcal{S}\mathcal{S}^\dagger = \mathcal{A} - \mathcal{A}\mathcal{P} = \mathcal{A} - (\mathcal{D} + \mathcal{E}\mathcal{S}^\dagger)\mathcal{P} = \mathcal{A} - \mathcal{D}\mathcal{P}.$$

Left-multiplying the above by $\mathcal{A}^{t-1}$ and taking the limit as $t \to \infty$, we find that

$$\lim_{t\to\infty} \mathcal{A}^t\mathcal{S}\mathcal{S}^\dagger = \underbrace{\lim_{t\to\infty} \mathcal{A}^t}_{=\mathcal{L}} - \underbrace{\lim_{t\to\infty} \mathcal{A}^{t-1}}_{=\mathcal{L}} \mathcal{D}\mathcal{P}$$

Since $\mathcal{A}^t\mathcal{S} \to 0$ as $t \to \infty$, the left-hand side of above is zero, so the above further reduces to $0 = \mathcal{L}(I - \mathcal{D}\mathcal{P})$. Therefore, to show $\mathcal{L} = 0$, it suffices to show that $I - \mathcal{D}\mathcal{P}$ is non-singular. Suppose a vector z in $\mathrm{Null}(I - \mathcal{D}\mathcal{P})$. If $z \notin \mathrm{Range}(\mathcal{P})$, then $\|\mathcal{P}z\|_2 < \|z\|_2$ for a projection matrix $\mathcal{P}$, and then we have

$$\|z\|_2 = \|\mathcal{D}\mathcal{P}z\|_2 \le \underbrace{\|\mathcal{D}\|_2}_{=1} \underbrace{\|\mathcal{P}z\|_2}_{<\|z\|_2} < \|z\|_2,$$

which is a contradiction. Hence, we know that $z \in \mathrm{Range}(\mathcal{P})$, which implies that $\mathcal{P}z = z$ because $\mathcal{P}$ is projection. Combining $z = \mathcal{D}\mathcal{P}z$ and $\mathcal{P}z = z$, we have $(I - \mathcal{D})z = 0$, which implies $z = 0$ since $I - \mathcal{D}$ is non-singular. Therefore, we conclude that $\mathrm{Null}(I - \mathcal{D}\mathcal{P}) = \{0\}$ and $I - \mathcal{D}\mathcal{P}$ is non-singular, so we have $\mathcal{L} = 0$, which implies that $\mathcal{A}$ is Schur stable. Thus, $\mathbf{A} - \mathbf{B}\mathbf{K}^*$ is Schur stable. ♦

Given $\mathbf{A} - \mathbf{B}\mathbf{K}^*$ Schur stable, we further have Schur stability of $\mathbf{A} - \mathbf{\Sigma}^{\mathsf{w}}\mathbf{K}^{\mathsf{w}}$ through Claim 4.9.5.

Claim 4.9.5. *For matrices* $\mathbf{A}, \mathbf{B}$ *in* (4.41), $\mathbf{\Sigma}^{\mathsf{w}}$ *in* (4.43) *and* $\mathbf{K}^*, \mathbf{K}^{\mathsf{w}}$ *in* (4.79a), $\mathbf{A} - \mathbf{B}\mathbf{K}^*$ *is Schur stable if, and only if,* $\mathbf{A} - \mathbf{\Sigma}^{\mathsf{w}}\mathbf{K}^{\mathsf{w}}$ *is Schur stable.*

Proof. Since $\Phi_{\mathsf{aux}} \in \mathbb{R}^{n_{\mathsf{aux}} \times n_\xi}$ by definition has full column rank, there exists a matrix $\Phi_{\mathsf{orth}} \in \mathbb{R}^{n_{\mathsf{aux}} \times (n_{\mathsf{aux}} - n_\xi)}$ such that $\mathrm{Range}(\Phi_{\mathsf{orth}}) = \mathrm{Null}(\Phi_{\mathsf{aux}}^{\mathsf{T}})$; it follows that

$$\Phi_{\mathsf{aux}} \Phi_{\mathsf{aux}}^{\dagger} + \Phi_{\mathsf{orth}} \Phi_{\mathsf{orth}}^{\dagger} = I_{n_{\mathsf{aux}}}. \tag{4.83}$$

Define matrices $\mathcal{S}^*, \mathcal{S}^{\mathsf{w}}, \mathcal{R}^*, \mathcal{R}^{\mathsf{w}}$,

$$\begin{aligned} \mathcal{S}^* &:= \Phi_{\mathsf{orth}}^{\dagger}(\mathbf{A} - \mathbf{B}\mathbf{K}^*)\Phi_{\mathsf{orth}}, & \mathcal{S}^{\mathsf{w}} &:= \Phi_{\mathsf{orth}}^{\dagger}(\mathbf{A} - \mathbf{\Sigma}^{\mathsf{w}}\mathbf{K}^{\mathsf{w}})\Phi_{\mathsf{orth}} \\ \mathcal{R}^* &:= \Phi_{\mathsf{aux}}^{\dagger}(\mathbf{A} - \mathbf{B}\mathbf{K}^*)\Phi_{\mathsf{orth}}, & \mathcal{R}^{\mathsf{w}} &:= \Phi_{\mathsf{aux}}^{\dagger}(\mathbf{A} - \mathbf{\Sigma}^{\mathsf{w}}\mathbf{K}^{\mathsf{w}})\Phi_{\mathsf{orth}} \end{aligned}$$

and it follows from (4.83) that

$$\begin{aligned} (\mathbf{A} - \mathbf{B}\mathbf{K}^*)\Phi_{\mathsf{orth}} &= \Phi_{\mathsf{aux}}\mathcal{R}^* + \Phi_{\mathsf{orth}}\mathcal{S}^*, \\ (\mathbf{A} - \mathbf{\Sigma}^{\mathsf{w}}\mathbf{K}^{\mathsf{w}})\Phi_{\mathsf{orth}} &= \Phi_{\mathsf{aux}}\mathcal{R}^{\mathsf{w}} + \Phi_{\mathsf{orth}}\mathcal{S}^{\mathsf{w}}. \end{aligned} \tag{4.84}$$

We moreover notice that $\mathcal{S}^* = \mathcal{S}^{\mathsf{w}} = \Phi_{\mathsf{orth}}^{\dagger}\mathbf{A}\Phi_{\mathsf{orth}}$ given the definitions of $\mathcal{S}^*, \mathcal{S}^{\mathsf{w}}$ and the facts $\Phi_{\mathsf{orth}}^{\dagger}\mathbf{B} = 0$ and $\Phi_{\mathsf{orth}}^{\dagger}\mathbf{\Sigma}^{\mathsf{w}} = 0$ which follow from the fact $\Phi_{\mathsf{orth}}^{\dagger}\Phi_{\mathsf{aux}} = 0$ via (4.83) and the relations $\mathbf{B} = \Phi_{\mathsf{aux}}\widetilde{B}$ and $\mathbf{\Sigma}^{\mathsf{w}} = \Phi_{\mathsf{aux}}\widetilde{\Sigma}^{\mathsf{w}}\Phi_{\mathsf{aux}}^{\mathsf{T}}$ from Claim 4.9.1.

Define $\Phi_{\mathsf{full}} := [\Phi_{\mathsf{aux}}, \Phi_{\mathsf{orth}}] \in \mathbb{R}^{n_{\mathsf{aux}} \times n_{\mathsf{aux}}}$ which is non-singular given (4.83). The horizontal stack of (4.81) and (4.84) yields

$$\begin{aligned} (\mathbf{A} - \mathbf{B}\mathbf{K}^*)\Phi_{\mathsf{full}} &= \Phi_{\mathsf{full}} \begin{bmatrix} \widetilde{A} - \widetilde{B}\widetilde{K}^* & \mathcal{R}^* \\ 0 & \mathcal{S}^* \end{bmatrix}, \\ (\mathbf{A} - \mathbf{\Sigma}^{\mathsf{w}}\mathbf{K}^{\mathsf{w}})\Phi_{\mathsf{full}} &= \Phi_{\mathsf{full}} \begin{bmatrix} \widetilde{A} - \widetilde{\Sigma}^{\mathsf{w}}\widetilde{K}^{\mathsf{w}} & \mathcal{R}^{\mathsf{w}} \\ 0 & \mathcal{S}^{\mathsf{w}} \end{bmatrix}. \end{aligned} \tag{4.85}$$

Since $\widetilde{A} - \widetilde{B}\widetilde{K}^*$ and $\widetilde{A} - \widetilde{\Sigma}^{\mathsf{w}}\widetilde{K}^{\mathsf{w}}$ are Schur stable through Claim 4.9.3, the matrix similarity relations (4.85) imply that $\mathbf{A} - \mathbf{B}\mathbf{K}^*$ (resp. $\mathbf{A} - \mathbf{\Sigma}^{\mathsf{w}}\mathbf{K}^{\mathsf{w}}$) is Schur stable if, and only if, $\mathcal{S}^*$ (resp. $\mathcal{S}^{\mathsf{w}}$) is Schur stable. Hence, the result follows from the fact $\mathcal{S}^* = \mathcal{S}^{\mathsf{w}}$. ♦

With both $\mathbf{A} - \mathbf{B}\mathbf{K}^*$ and $\mathbf{A} - \mathbf{\Sigma}^{\mathsf{w}}\mathbf{K}^{\mathsf{w}}$ Schur stable, the pairs $(\mathbf{A}, \mathbf{B})$ and $(\mathbf{A}, \mathbf{\Sigma}^{\mathsf{w}})$ are stabilizable. ■

4.6.8 Proof of (4.57) and (4.58)

Here, we prove (4.57) and (4.58) which are critical results supporting the proof of Proposition 4.11. The results are shown in Subsection D, while in Subsection A, Subsection B and Subsection C we establish intermediate results.

A. Preliminary Results

We begin by establishing some useful identities in Claim 4.11.1 that will be leveraged in the remainder of the proof. Recall the matrices $\Phi_{\mathsf{orig}} \in \mathbb{R}^{n \times n_\xi}$, $\Phi_{\mathsf{aux}} \in \mathbb{R}^{n_{\mathsf{aux}} \times n_\xi}$ defined in Claim 4.8.1 and matrix $\Phi = [\Phi_{\mathsf{U}}, \Phi_{\mathsf{Y}}, \Phi_{\mathsf{P}}] \in \mathbb{R}^{n \times n_{\mathsf{aux}}}$ defined in Claim 4.9.2, with $n_{\mathsf{aux}} := mL + pL + pL^2$ and $n_\xi := mL + n(L+1)$.

Claim 4.11.1. *For the system (4.1) and auxiliary model (4.41), it holds for all $t \in \mathbb{N}_{\geq 0}$ that*

$$\begin{aligned} x_t &= \Phi \mathbf{x}_t & A\Phi\Phi_{\mathsf{aux}} &= \Phi \mathbf{A} \Phi_{\mathsf{aux}} & B &= \Phi \mathbf{B} \\ w_t &= \Phi \mathbf{w}_t & C\Phi\Phi_{\mathsf{aux}} &= \mathbf{C}\Phi_{\mathsf{aux}} & \Sigma^{\mathsf{w}} &= \Phi \mathbf{\Sigma}^{\mathsf{w}} \Phi^{\mathsf{T}}. \end{aligned}$$

Proof. The relation $x_t = \Phi \mathbf{x}_t$ follows from Claim 4.8.1 and Claim 4.9.2. We have proved $C\Phi\Phi_{\mathsf{aux}} = \mathbf{C}\Phi_{\mathsf{aux}}$ in the proof of Claim 4.8.2. To show $w_t = \Phi \mathbf{w}_t$ and $\Sigma^{\mathsf{w}} = \Phi \mathbf{\Sigma}^{\mathsf{w}} \Phi^{\mathsf{T}}$, recall from the definition that $\mathbf{w}_t = J_0 w_t$ and $\mathbf{\Sigma}^{\mathsf{w}} = J_0 \Sigma^{\mathsf{w}} J_0^{\mathsf{T}}$ where $J_0 := \mathrm{col}(0_{(n_{\mathsf{aux}} - pL) \times n}, \mathcal{O})$. By direct calculation one can verify that $\Phi J_0 = I_n$, using which we obtain $w_t = \Phi \mathbf{w}_t$ given $\mathbf{w}_t = J_0 w_t$ and obtain $\Sigma^{\mathsf{w}} = \Phi \mathbf{\Sigma}^{\mathsf{w}} \Phi^{\mathsf{T}}$ given $\mathbf{\Sigma}^{\mathsf{w}} = J_0 \Sigma^{\mathsf{w}} J_0^{\mathsf{T}}$. We have $B = \Phi_{\mathsf{orig}} \widetilde{B} = \Phi \Phi_{\mathsf{aux}} \widetilde{B} = \Phi \mathbf{B}$, using $\Phi_{\mathsf{orig}} = \Phi \Phi_{\mathsf{aux}}$ as Claim 4.9.2, $\Phi_{\mathsf{aux}} \widetilde{B} = \mathbf{B}$ from Claim 4.9.1 and $B = \Phi_{\mathsf{orig}} \widetilde{B}$ which can be verified by definitions of Φ_{orig} and $\widetilde{B}$. We finally have $A\Phi\Phi_{\mathsf{aux}} = A\Phi_{\mathsf{orig}} = \Phi_{\mathsf{orig}} \widetilde{A} = \Phi \Phi_{\mathsf{aux}} \widetilde{A} = \Phi \mathbf{A} \Phi_{\mathsf{aux}}$, where we used $\Phi_{\mathsf{orig}} = \Phi \Phi_{\mathsf{aux}}$ as Claim 4.9.2, $\mathbf{A}\Phi_{\mathsf{aux}} = \Phi_{\mathsf{aux}} \widetilde{A}$ in Claim 4.9.1 and $A\Phi_{\mathsf{orig}} = \Phi_{\mathsf{orig}} \widetilde{A}$ which can be verified given the definitions of Φ_{orig} and $\widetilde{A}$. ♦

B. Relations of Feedback Gains

We relate the LQR feedback gains K and $\mathbf{K}$ as follows.

Claim 4.11.2. *For matrices K in (4.16) and $\mathbf{K}$ in (4.51), it holds that $K\Phi\Phi_{\mathsf{aux}} = \mathbf{K}\Phi_{\mathsf{aux}}$.*

Proof. Let $\widetilde{C} := C\Phi_{\mathsf{orig}}$ and let $\widetilde{A}, \widetilde{B}$ be as in (4.78). We first show the pair $(\widetilde{A}, \widetilde{C})$ is detectable. For $\lambda \in \mathbb{C}$, define $H_{\mathsf{obs}} := \mathrm{col}(\lambda I_{n_\xi} - \widetilde{A}, \widetilde{C})$, which can be permuted into the form

$$\left[\begin{array}{c|c|c} \lambda I_{mL} - \mathcal{D}_m & & \\ \hline & \lambda I_{nL} - \mathcal{D}_n & \\ \hline -B \quad 0_{n \times m(L-1)} & -I_n \quad 0_{n \times n(L-1)} & \lambda I_n - A \\ \hline \mathcal{C}\mathcal{C} & \mathcal{C}\mathcal{C}_{\mathsf{w}} & CA^L \end{array}\right], \tag{4.86}$$

wherein $\mathcal{D}_q := \begin{bmatrix} & I_{q(L-1)} \\ 0_{q\times q} & \end{bmatrix}$. Since the blocks $\lambda I_{mL} - \mathcal{D}_m$ and $\lambda I_{nL} - \mathcal{D}_n$ in (4.86) are non-singular for all $\lambda \neq 0$, to show that (4.86) has full column rank when $|\lambda| \geq 1$, we only need to verify the rank of the last block column in (4.86). Since (A, C) is observable, $\mathcal{O}_n := \mathrm{col}(C, CA, \ldots, CA^{n-1})$ has full column rank, so we have $\mathrm{Null}(\mathcal{O}_n A^L) = \mathrm{Null}(A^L)$ where Null denotes the null space. Note that $\mathcal{O}_n A^L$ is the observability matrix of the pair (A, CA^L), and thus $\mathrm{Null}(\mathcal{O}_n A^L)$ is the unobservable space of the pair (A, CA^L). Given $\mathrm{Null}(\mathcal{O}_n A^L) = \mathrm{Null}(A^L)$, all unobservable states x_{nobs} of (A, CA^L) satisfy $A^L x_{\mathsf{nobs}} = 0$ and hence are strictly stable, which implies that (A, CA^L) is detectable. From the Hautus lemma, $\mathrm{col}(\lambda I_n - A, CA^L)$ has full column rank for all λ that $|\lambda| \geq 1$. With diagonal blocks $\lambda I_{mL} - \mathcal{D}_m$, $\lambda I_{nL} - \mathcal{D}_n$ and $\mathrm{col}(\lambda I_n - A, CA^L)$ having full column rank, the matrix (4.86) has full column rank when $|\lambda| \geq 1$, and so does the pre-permutational matrix H_{obs}, which implies that $(\widetilde{A}, \widetilde{C})$ is detectable through Hautus lemma.

Next, we show that $\widetilde{P}_1 = \widetilde{P}_2$ with

$$\widetilde{P}_1 := \Phi_{\mathsf{orig}}^{\mathsf{T}} P \Phi_{\mathsf{orig}}, \qquad \widetilde{P}_2 := \Phi_{\mathsf{aux}}^{\mathsf{T}} \mathbf{P} \Phi_{\mathsf{aux}},$$

where P is the solution to (4.17) and $\mathbf{P}$ is the solution to (4.52). The equations (4.17) and (4.52) are reproduced here as

$$\begin{aligned} 0 = {} & A^{\mathsf{T}} P A + C^{\mathsf{T}} Q C - P - (A^{\mathsf{T}} P B + C^{\mathsf{T}} Q D) \\ & (R + D^{\mathsf{T}} Q D + B^{\mathsf{T}} P B)^{-1} (B^{\mathsf{T}} P A + D^{\mathsf{T}} Q C), \end{aligned} \tag{4.87a}$$

$$\begin{aligned} 0 = {} & \mathbf{A}^{\mathsf{T}} \mathbf{P} \mathbf{A} + \mathbf{C}^{\mathsf{T}} Q \mathbf{C} - \mathbf{P} - (\mathbf{A}^{\mathsf{T}} \mathbf{P} \mathbf{B} + \mathbf{C}^{\mathsf{T}} Q D) \\ & (R + D^{\mathsf{T}} Q D + \mathbf{B}^{\mathsf{T}} \mathbf{P} \mathbf{B})^{-1} (\mathbf{B}^{\mathsf{T}} \mathbf{P} \mathbf{A} + D^{\mathsf{T}} Q \mathbf{C}). \end{aligned} \tag{4.87b}$$

Left- and right-multiply (4.87a) by $\Phi_{\mathsf{orig}}^{\mathsf{T}}$ and Φ_{orig} respectively, and we obtain that

$$\begin{aligned} 0 = {} & \Phi_{\mathsf{orig}}^{\mathsf{T}} A^{\mathsf{T}} P A \Phi_{\mathsf{orig}} + \Phi_{\mathsf{orig}}^{\mathsf{T}} C^{\mathsf{T}} Q C \Phi_{\mathsf{orig}} - \Phi_{\mathsf{orig}}^{\mathsf{T}} P \Phi_{\mathsf{orig}} \\ & - (\Phi_{\mathsf{orig}}^{\mathsf{T}} A^{\mathsf{T}} P B + \Phi_{\mathsf{orig}}^{\mathsf{T}} C^{\mathsf{T}} Q D)(R + D^{\mathsf{T}} Q D + B^{\mathsf{T}} P B)^{-1} \\ & (B^{\mathsf{T}} P A \Phi_{\mathsf{orig}} + D^{\mathsf{T}} Q C \Phi_{\mathsf{orig}}) \\ = {} & \widetilde{A}^{\mathsf{T}} \widetilde{P}_1 \widetilde{A} + \widetilde{C}^{\mathsf{T}} Q \widetilde{C} - \widetilde{P}_1 - (\widetilde{A}^{\mathsf{T}} \widetilde{P}_1 \widetilde{B} + \widetilde{C}^{\mathsf{T}} Q D) \\ & (R + D^{\mathsf{T}} Q D + \widetilde{B}^{\mathsf{T}} \widetilde{P}_1 \widetilde{B})^{-1} (\widetilde{B}^{\mathsf{T}} \widetilde{P}_1 \widetilde{A} + D^{\mathsf{T}} Q \widetilde{C}), \end{aligned} \tag{4.88}$$

where the second equality used the definitions $\widetilde{C} := C \Phi_{\mathsf{orig}}$ and $\widetilde{P}_1 := \Phi_{\mathsf{orig}}^{\mathsf{T}} P \Phi_{\mathsf{orig}}$ and the

facts that

$$\Phi_{\mathsf{orig}}^{\mathsf{T}} A^{\mathsf{T}} P A \Phi_{\mathsf{orig}} = \widetilde{A}^{\mathsf{T}} \Phi_{\mathsf{orig}}^{\mathsf{T}} P \Phi_{\mathsf{orig}} \widetilde{A} = \widetilde{A}^{\mathsf{T}} \widetilde{P}_1 \widetilde{A} \tag{4.89a}$$

$$B^{\mathsf{T}} P A \Phi_{\mathsf{orig}} = \widetilde{B}^{\mathsf{T}} \Phi_{\mathsf{orig}}^{\mathsf{T}} P \Phi_{\mathsf{orig}} \widetilde{A} = \widetilde{B}^{\mathsf{T}} \widetilde{P}_1 \widetilde{A} \tag{4.89b}$$

$$B^{\mathsf{T}} P B = \widetilde{B}^{\mathsf{T}} \Phi_{\mathsf{orig}}^{\mathsf{T}} P \Phi_{\mathsf{orig}} \widetilde{B} = \widetilde{B}^{\mathsf{T}} \widetilde{P}_1 \widetilde{B} \tag{4.89c}$$

in which we used relations $A\Phi_{\mathsf{orig}} = A\Phi\Phi_{\mathsf{aux}} = \Phi\mathbf{A}\Phi_{\mathsf{aux}} = \Phi\Phi_{\mathsf{aux}}\widetilde{A} = \Phi_{\mathsf{orig}}\widetilde{A}$ and $B = \Phi\mathbf{B} = \Phi\Phi_{\mathsf{aux}}\widetilde{B} = \Phi_{\mathsf{orig}}\widetilde{B}$, given the identities $\Phi_{\mathsf{orig}} = \Phi\Phi_{\mathsf{aux}}$ as Claim 4.9.2, $A\Phi\Phi_{\mathsf{aux}} = \Phi\mathbf{A}\Phi_{\mathsf{aux}}$ and $B = \Phi\mathbf{B}$ from Claim 4.11.1 and $\mathbf{A}\Phi_{\mathsf{aux}} = \Phi_{\mathsf{aux}}\widetilde{A}$ and $\mathbf{B} = \Phi_{\mathsf{aux}}\widetilde{B}$ from Claim 4.9.1. Similarly, left- and right-multiply (4.87b) by $\Phi_{\mathsf{aux}}^{\mathsf{T}}$ and Φ_{aux} respectively, and we have

$$\begin{aligned} 0 &= \Phi_{\mathsf{aux}}^{\mathsf{T}} \mathbf{A}^{\mathsf{T}} \mathbf{P} \mathbf{A} \Phi_{\mathsf{aux}} + \Phi_{\mathsf{aux}}^{\mathsf{T}} \mathbf{C}^{\mathsf{T}} Q \mathbf{C} \Phi_{\mathsf{aux}} - \Phi_{\mathsf{aux}}^{\mathsf{T}} \mathbf{P} \Phi_{\mathsf{aux}} \\ &\quad - (\Phi_{\mathsf{aux}}^{\mathsf{T}} \mathbf{A}^{\mathsf{T}} \mathbf{P} \mathbf{B} + \Phi_{\mathsf{aux}}^{\mathsf{T}} \mathbf{C}^{\mathsf{T}} Q D)(R + D^{\mathsf{T}} Q D + \mathbf{B}^{\mathsf{T}} \mathbf{P} \mathbf{B})^{-1} \\ &\quad (\mathbf{B}^{\mathsf{T}} \mathbf{P} \mathbf{A} \Phi_{\mathsf{aux}} + D^{\mathsf{T}} Q \mathbf{C} \Phi_{\mathsf{aux}}) \\ &= \widetilde{A}^{\mathsf{T}} \widetilde{P}_2 \widetilde{A} + \widetilde{C}^{\mathsf{T}} Q \widetilde{C} - \widetilde{P}_2 - (\widetilde{A}^{\mathsf{T}} \widetilde{P}_2 \widetilde{B} + \widetilde{C}^{\mathsf{T}} Q D) \\ &\quad (R + D^{\mathsf{T}} Q D + \widetilde{B}^{\mathsf{T}} \widetilde{P}_2 \widetilde{B})^{-1} (\widetilde{B}^{\mathsf{T}} \widetilde{P}_2 \widetilde{A} + D^{\mathsf{T}} Q \widetilde{C}), \end{aligned} \tag{4.90}$$

where the second equality above used the definition $\widetilde{P}_2 := \Phi_{\mathsf{aux}}^{\mathsf{T}} \mathbf{P} \Phi_{\mathsf{aux}}$, the relation $\mathbf{C}\Phi_{\mathsf{aux}} = C\Phi\Phi_{\mathsf{aux}} = C\Phi_{\mathsf{orig}} = \widetilde{C}$ given $\Phi_{\mathsf{orig}} = \Phi\Phi_{\mathsf{aux}}$ as Claim 4.9.2 and $\mathbf{C}\Phi_{\mathsf{aux}} = C\Phi\Phi_{\mathsf{aux}}$ from Claim 4.11.1, and the facts that

$$\Phi_{\mathsf{aux}}^{\mathsf{T}} \mathbf{A}^{\mathsf{T}} \mathbf{P} \mathbf{A} \Phi_{\mathsf{aux}} = \widetilde{A}^{\mathsf{T}} \Phi_{\mathsf{aux}}^{\mathsf{T}} P \Phi_{\mathsf{aux}} \widetilde{A} = \widetilde{A}^{\mathsf{T}} \widetilde{P}_2 \widetilde{A} \tag{4.91a}$$

$$\mathbf{B}^{\mathsf{T}} \mathbf{P} \mathbf{A} \Phi_{\mathsf{aux}} = \widetilde{B}^{\mathsf{T}} \Phi_{\mathsf{aux}}^{\mathsf{T}} P \Phi_{\mathsf{aux}} \widetilde{A} = \widetilde{B}^{\mathsf{T}} \widetilde{P}_2 \widetilde{A} \tag{4.91b}$$

$$\mathbf{B}^{\mathsf{T}} \mathbf{P} \mathbf{B} = \widetilde{B}^{\mathsf{T}} \Phi_{\mathsf{aux}}^{\mathsf{T}} P \Phi_{\mathsf{aux}} \widetilde{B} = \widetilde{B}^{\mathsf{T}} \widetilde{P}_2 \widetilde{B} \tag{4.91c}$$

given $\mathbf{A}\Phi_{\mathsf{aux}} = \Phi_{\mathsf{aux}}\widetilde{A}$ and $\mathbf{B} = \Phi_{\mathsf{aux}}\widetilde{B}$ from Claim 4.9.1. Observing (4.88) and (4.90), we know that both $\widetilde{P}_1$ and $\widetilde{P}_2$ are (positive semi-definite) solutions to a similar DARE to (4.17) and (4.52), for system $(\widetilde{A}, \widetilde{B}, \widetilde{C}, D)$. In fact, this DARE has a unique positive semi-definite solution, given stabilizable $(\widetilde{A}, \widetilde{B})$ via Claim 4.9.3, detectable $(\widetilde{A}, \widetilde{C})$ as proved before and $Q \succ 0$. Hence, the solutions $\widetilde{P}_1, \widetilde{P}_2$ are equal.

Finally, we obtain the result by noting that $B^{\mathsf{T}} P A \Phi_{\mathsf{orig}} = \mathbf{B}^{\mathsf{T}} \mathbf{P} \mathbf{A} \Phi_{\mathsf{aux}}$ via (4.89b), (4.91b) and $B^{\mathsf{T}} P B = \mathbf{B}^{\mathsf{T}} \mathbf{P} \mathbf{B}$ via (4.89c), (4.91c), given $\widetilde{P}_1 = \widetilde{P}_2$. Hence, it follows from the definitions (4.16) and (4.51) of K and $\mathbf{K}$ that $K\Phi_{\mathsf{orig}} = \mathbf{K}\Phi_{\mathsf{aux}}$, which is the result given $\Phi_{\mathsf{orig}} = \Phi\Phi_{\mathsf{aux}}$ in Claim 4.9.2. ♦

We mention some useful identities in Claim 4.11.3 which follow after Claim 4.9.1, Claim 4.11.1 and Claim 4.11.2 and will be used multiple times in the rest of the proof.

Claim 4.11.3. *If $v \in \mathbb{R}^n$, $\boldsymbol{v} \in \mathbb{R}^{n_{\mathsf{aux}}}$ and $\tilde{v} \in \mathbb{R}^{n_\xi}$ are such that $v = \Phi \boldsymbol{v}$ and $\boldsymbol{v} = \Phi_{\mathsf{aux}} \tilde{v}$, then*

$$Cv = \mathbf{C}\boldsymbol{v}, \qquad Kv = \mathbf{K}\boldsymbol{v}, \qquad Av = \Phi\mathbf{A}\boldsymbol{v}, \qquad \mathbf{A}\boldsymbol{v} = \Phi_{\mathsf{aux}}\widetilde{A}\tilde{v}.$$

If $M \in \mathbb{S}^n_+$, $\mathbf{M} \in \mathbb{S}^{n_{\mathsf{aux}}}_+$ and $\widetilde{M} \in \mathbb{S}^{n_\xi}_+$ are such that $M = \Phi\mathbf{M}\Phi^\mathsf{T}$ and $\mathbf{M} = \Phi_{\mathsf{aux}}\widetilde{M}\Phi_{\mathsf{aux}}^\mathsf{T}$, then

$$\begin{aligned} &CM = \mathbf{CM}\Phi^\mathsf{T}, \;\; CMC^\mathsf{T} = \mathbf{CMC}^\mathsf{T}, \;\; CMK^\mathsf{T} = \mathbf{CMK}^\mathsf{T}, \\ &KM = \mathbf{KM}\Phi^\mathsf{T}, \;\; KMK^\mathsf{T} = \mathbf{KMK}^\mathsf{T}. \end{aligned}$$

Proof. Using $C\Phi\Phi_{\mathsf{aux}} = \mathbf{C}\Phi_{\mathsf{aux}}$ from Claim 4.11.1, we have

$$\begin{aligned} &Cv = C\Phi\boldsymbol{v} = C\Phi\Phi_{\mathsf{aux}}\tilde{v} = \mathbf{C}\Phi_{\mathsf{aux}}\tilde{v} = \mathbf{C}\boldsymbol{v}, \\ &CM = C\Phi\mathbf{M}\Phi^\mathsf{T} = C\Phi\Phi_{\mathsf{aux}}\widetilde{M}\Phi_{\mathsf{aux}}^\mathsf{T}\Phi^\mathsf{T} = \mathbf{C}\Phi_{\mathsf{aux}}\widetilde{M}\Phi_{\mathsf{aux}}^\mathsf{T}\Phi^\mathsf{T} = \mathbf{CM}\Phi^\mathsf{T}, \\ &CMC^\mathsf{T} = C\Phi\mathbf{M}\Phi^\mathsf{T}C^\mathsf{T} = C\Phi\Phi_{\mathsf{aux}}\widetilde{M}\Phi_{\mathsf{aux}}^\mathsf{T}\Phi^\mathsf{T}C^\mathsf{T} = \mathbf{C}\Phi_{\mathsf{aux}}\widetilde{M}\Phi_{\mathsf{aux}}^\mathsf{T}\mathbf{C}^\mathsf{T} = \mathbf{CMC}^\mathsf{T}. \end{aligned}$$

Similarly, using $K\Phi\Phi_{\mathsf{aux}} = \mathbf{K}\Phi_{\mathsf{aux}}$ from Claim 4.11.2, we prove $Kv = \mathbf{K}\boldsymbol{v}$, $KM = \mathbf{KM}\Phi^\mathsf{T}$ and $KMK^\mathsf{T} = \mathbf{KMK}^\mathsf{T}$ in the same way by replacing $(C, \mathbf{C})$ into $(K, \mathbf{K})$. We similarly have $CMK^\mathsf{T} = \mathbf{CMK}^\mathsf{T}$. Using $A\Phi\Phi_{\mathsf{aux}} = \Phi\mathbf{A}\Phi_{\mathsf{aux}}$ from Claim 4.11.1 and $\mathbf{A}\Phi_{\mathsf{aux}} = \Phi_{\mathsf{aux}}\widetilde{A}$ from Claim 4.9.1, we obtain that

$$\begin{aligned} &Av = A\Phi\boldsymbol{v} = A\Phi\Phi_{\mathsf{aux}}\tilde{v} = \Phi\mathbf{A}\Phi_{\mathsf{aux}}\tilde{v} = \Phi\mathbf{A}\boldsymbol{v}, \\ &\mathbf{A}\boldsymbol{v} = \mathbf{A}\Phi_{\mathsf{aux}}\tilde{v} = \Phi_{\mathsf{aux}}\widetilde{A}\tilde{v}, \end{aligned}$$ ♦

C. Relations of Observer Gains

The state variances $\Sigma^\mathsf{x}, \mathbf{\Sigma}^\mathsf{x}$, Kalman gains $L_\mathsf{K}, \mathbf{L}_\mathsf{K}$ and Luenberger gains $L_\mathsf{L}, \mathbf{L}_\mathsf{L}$ are respectively related as follows.

Claim 4.11.4. *For matrices $\Sigma^\mathsf{x}, L_\mathsf{K}, L_\mathsf{L}$ in (4.11) and $\mathbf{\Sigma}^\mathsf{x}, \mathbf{L}_\mathsf{K}, \mathbf{L}_\mathsf{L}$ in (4.46), it holds that*

(a) $\Sigma^\mathsf{x} = \Phi\mathbf{\Sigma}^\mathsf{x}\Phi^\mathsf{T}$ and $\mathbf{\Sigma}^\mathsf{x} = \Phi_{\mathsf{aux}}\widetilde{\Sigma}^\mathsf{x}\Phi_{\mathsf{aux}}^\mathsf{T}$ for some $\widetilde{\Sigma}^\mathsf{x} \in \mathbb{S}^{n_\xi}_+$;
(b) $L_\mathsf{K} = \Phi\mathbf{L}_\mathsf{K}$ and $\mathbf{L}_\mathsf{K} = \Phi_{\mathsf{aux}}\widetilde{L}_\mathsf{K}$ for some $\widetilde{L}_\mathsf{K} \in \mathbb{R}^{n_\xi \times p}$;
(c) $L_\mathsf{L} = \Phi\mathbf{L}_\mathsf{L}$ and $\mathbf{L}_\mathsf{L} = \Phi_{\mathsf{aux}}\widetilde{L}_\mathsf{L}$ for some $\widetilde{L}_\mathsf{L} \in \mathbb{R}^{n_\xi \times p}$.

Proof. We first show $\mathbf{\Sigma}^{\mathsf{x}} = \Phi_{\mathsf{aux}}\widetilde{\Sigma}^{\mathsf{x}}\Phi_{\mathsf{aux}}^{\mathsf{T}}$ in (a). Let $\widetilde{C} := C\Phi_{\mathsf{orig}}$ and let $\widetilde{A}, \widetilde{B}$ be as in (4.78). Since $(\widetilde{A}, \widetilde{\Sigma}^{\mathsf{w}})$ is stabilizable through Claim 4.9.3 and $(\widetilde{A}, \widetilde{C})$ is detectable as shown in the proof of Claim 4.11.2, the DARE

$$\widetilde{\Sigma}^{\mathsf{x}} = \widetilde{A}\widetilde{\Sigma}^{\mathsf{x}}\widetilde{A}^{\mathsf{T}} + \widetilde{\Sigma}^{\mathsf{w}} - \widetilde{A}\widetilde{\Sigma}^{\mathsf{x}}\widetilde{C}^{\mathsf{T}}(\widetilde{C}\widetilde{\Sigma}^{\mathsf{x}}\widetilde{C}^{\mathsf{T}} + \Sigma^{\mathsf{v}})^{-1}\widetilde{C}\widetilde{\Sigma}^{\mathsf{x}}\widetilde{A}^{\mathsf{T}} \tag{4.92}$$

has a unique positive semi-definite solution $\widetilde{\Sigma}^{\mathsf{x}}$. Left- and right-multiply (4.92) by Φ_{aux} and by $\Phi_{\mathsf{aux}}^{\mathsf{T}}$ respectively, and we have

$$\begin{aligned}
\Phi_{\mathsf{aux}}\widetilde{\Sigma}^{\mathsf{x}}\Phi_{\mathsf{aux}}^{\mathsf{T}} &= \Phi_{\mathsf{aux}}\widetilde{A}\widetilde{\Sigma}^{\mathsf{x}}\widetilde{A}^{\mathsf{T}}\Phi_{\mathsf{aux}}^{\mathsf{T}} + \Phi_{\mathsf{aux}}\widetilde{\Sigma}^{\mathsf{w}}\Phi_{\mathsf{aux}}^{\mathsf{T}} \\
&\qquad - \Phi_{\mathsf{aux}}\widetilde{A}\widetilde{\Sigma}^{\mathsf{x}}\widetilde{C}^{\mathsf{T}}(\widetilde{C}\widetilde{\Sigma}^{\mathsf{x}}\widetilde{C}^{\mathsf{T}} + \Sigma^{\mathsf{v}})^{-1}\widetilde{C}\widetilde{\Sigma}^{\mathsf{x}}\widetilde{A}^{\mathsf{T}}\Phi_{\mathsf{aux}}^{\mathsf{T}} \\
&= \mathbf{A}\Phi_{\mathsf{aux}}\widetilde{\Sigma}^{\mathsf{x}}\Phi_{\mathsf{aux}}^{\mathsf{T}}\mathbf{A}^{\mathsf{T}} + \Phi_{\mathsf{aux}}\widetilde{\Sigma}^{\mathsf{w}}\Phi_{\mathsf{aux}}^{\mathsf{T}} - \mathbf{A}\Phi_{\mathsf{aux}}\widetilde{\Sigma}^{\mathsf{x}}\Phi_{\mathsf{aux}}^{\mathsf{T}}\mathbf{C}^{\mathsf{T}} \\
&\quad (\mathbf{C}\Phi_{\mathsf{aux}}\widetilde{\Sigma}^{\mathsf{x}}\Phi_{\mathsf{aux}}^{\mathsf{T}}\mathbf{C}^{\mathsf{T}} + \Sigma^{\mathsf{v}})^{-1}\mathbf{C}\Phi_{\mathsf{aux}}\widetilde{\Sigma}^{\mathsf{x}}\Phi_{\mathsf{aux}}^{\mathsf{T}}\widetilde{A}^{\mathsf{T}},
\end{aligned} \tag{4.93}$$

where the second equality used $\Phi_{\mathsf{aux}}\widetilde{A} = \mathbf{A}\Phi_{\mathsf{aux}}$ in Claim 4.11.1 and used $\widetilde{C} := C\Phi_{\mathsf{orig}} = C\Phi\Phi_{\mathsf{aux}} = \mathbf{C}\Phi_{\mathsf{aux}}$ applying $\Phi_{\mathsf{orig}} = \Phi\Phi_{\mathsf{aux}}$ as Claim 4.9.2 and $C\Phi\Phi_{\mathsf{aux}} = \mathbf{C}\Phi_{\mathsf{aux}}$ in Claim 4.11.1. Due to (4.93), $\Phi_{\mathsf{aux}}\widetilde{\Sigma}^{\mathsf{x}}\Phi_{\mathsf{aux}}^{\mathsf{T}}$ is a (positive semi-definite) solution to equation (4.46a). Since the DARE (4.46a) has a unique positive semi-definite solution $\mathbf{\Sigma}^{\mathsf{x}}$, we have $\mathbf{\Sigma}^{\mathsf{x}} = \Phi_{\mathsf{aux}}\widetilde{\Sigma}^{\mathsf{x}}\Phi_{\mathsf{aux}}^{\mathsf{T}}$.

Then, we show $\Sigma^{\mathsf{x}} = \Phi\mathbf{\Sigma}^{\mathsf{x}}\Phi^{\mathsf{T}}$ in (a). Left- and right-multiply the DARE (4.46a) by Φ and by Φ^{T} respectively, and we have

$$\begin{aligned}
\Phi\mathbf{\Sigma}^{\mathsf{x}}\Phi^{\mathsf{T}} &= \Phi\mathbf{A}\mathbf{\Sigma}^{\mathsf{x}}\mathbf{A}^{\mathsf{T}}\Phi^{\mathsf{T}} + \Phi\mathbf{\Sigma}^{\mathsf{w}}\Phi^{\mathsf{T}} - \Phi\mathbf{A}\mathbf{\Sigma}^{\mathsf{x}}\mathbf{C}^{\mathsf{T}} \\
&\qquad (\mathbf{C}\mathbf{\Sigma}^{\mathsf{x}}\mathbf{C}^{\mathsf{T}} + \Sigma^{\mathsf{v}})^{-1}\mathbf{C}\mathbf{\Sigma}^{\mathsf{x}}\mathbf{A}^{\mathsf{T}}\Phi^{\mathsf{T}} \\
&= A\Phi\mathbf{\Sigma}^{\mathsf{x}}\Phi^{\mathsf{T}}A^{\mathsf{T}} + \Sigma^{\mathsf{w}} - A\Phi\mathbf{\Sigma}^{\mathsf{x}}\Phi^{\mathsf{T}}C^{\mathsf{T}} \\
&\qquad (C\Phi\mathbf{\Sigma}^{\mathsf{x}}\Phi^{\mathsf{T}}C^{\mathsf{T}} + \Sigma^{\mathsf{v}})^{-1}C\Phi\mathbf{\Sigma}^{\mathsf{x}}\Phi^{\mathsf{T}}A^{\mathsf{T}},
\end{aligned} \tag{4.94}$$

where the second equality used $\Phi\mathbf{\Sigma}^{\mathsf{w}}\Phi^{\mathsf{T}} = \Sigma^{\mathsf{w}}$ from Claim 4.11.1 and used relations $\Phi\mathbf{A}\mathbf{\Sigma}^{\mathsf{x}}\mathbf{A}^{\mathsf{T}}\Phi^{\mathsf{T}} = A\Phi\mathbf{\Sigma}^{\mathsf{x}}\Phi^{\mathsf{T}}A^{\mathsf{T}}$, $\Phi\mathbf{A}\mathbf{\Sigma}^{\mathsf{x}}\mathbf{C}^{\mathsf{T}} = A\Phi\mathbf{\Sigma}^{\mathsf{x}}\Phi^{\mathsf{T}}C^{\mathsf{T}}$ and $\mathbf{C}\mathbf{\Sigma}^{\mathsf{x}}\mathbf{C}^{\mathsf{T}} = C\Phi\mathbf{\Sigma}^{\mathsf{x}}\Phi^{\mathsf{T}}C^{\mathsf{T}}$ implied by

$$\begin{aligned}
\Phi\mathbf{A}\mathbf{\Sigma}^{\mathsf{x}} &= \Phi\mathbf{A}\Phi_{\mathsf{aux}}\widetilde{\Sigma}^{\mathsf{x}}\Phi_{\mathsf{aux}}^{\mathsf{T}} = A\Phi\Phi_{\mathsf{aux}}\widetilde{\Sigma}^{\mathsf{x}}\Phi_{\mathsf{aux}}^{\mathsf{T}} = A\Phi\mathbf{\Sigma}^{\mathsf{x}}, \\
\mathbf{C}\mathbf{\Sigma}^{\mathsf{x}} &= \mathbf{C}\Phi_{\mathsf{aux}}\widetilde{\Sigma}^{\mathsf{x}}\Phi_{\mathsf{aux}}^{\mathsf{T}} = C\Phi\Phi_{\mathsf{aux}}\widetilde{\Sigma}^{\mathsf{x}}\Phi_{\mathsf{aux}}^{\mathsf{T}} = C\Phi\mathbf{\Sigma}^{\mathsf{x}},
\end{aligned}$$

which applied $\Phi\mathbf{A}\Phi_{\mathsf{aux}} = A\Phi\Phi_{\mathsf{aux}}$ and $\mathbf{C}\Phi_{\mathsf{aux}} = C\Phi\Phi_{\mathsf{aux}}$ from Claim 4.11.1. Due to (4.94), $\Phi\mathbf{\Sigma}^{\mathsf{x}}\Phi^{\mathsf{T}}$ is a (positive semi-definite) solution to the DARE (4.11a). Since (4.11a) has a unique positive definite solution Σ^{x}, we have $\Sigma^{\mathsf{x}} = \Phi\mathbf{\Sigma}^{\mathsf{x}}\Phi^{\mathsf{T}}$.

We finally show (b) and (c). Given the definition of $\mathbf{L}_{\mathsf{K}}$ and $\mathbf{L}_{\mathsf{L}}$, we obtain $\mathbf{L}_{\mathsf{K}} = \Phi_{\mathsf{aux}}\widetilde{L}_{\mathsf{K}}$ and $\mathbf{L}_{\mathsf{L}} = \Phi_{\mathsf{aux}}\widetilde{L}_{\mathsf{L}}$

$$\begin{aligned}\mathbf{L}_{\mathsf{K}} &:= \mathbf{\Sigma}^{\mathsf{x}}\mathbf{C}^{\mathsf{T}}(\mathbf{C}\mathbf{\Sigma}^{\mathsf{x}}\mathbf{C}^{\mathsf{T}} + \Sigma^{\mathsf{v}})^{-1} = \Phi_{\mathsf{aux}}\widetilde{L}_{\mathsf{K}} \\ &= \Phi_{\mathsf{aux}}\widetilde{\Sigma}^{\mathsf{x}}\Phi_{\mathsf{aux}}^{\mathsf{T}}\mathbf{C}^{\mathsf{T}}(\mathbf{C}\mathbf{\Sigma}^{\mathsf{x}}\mathbf{C}^{\mathsf{T}} + \Sigma^{\mathsf{v}})^{-1} = \Phi_{\mathsf{aux}}\widetilde{L}_{\mathsf{K}} \\ \mathbf{L}_{\mathsf{L}} &:= \mathbf{A}\mathbf{L}_{\mathsf{K}} = \mathbf{A}\Phi_{\mathsf{aux}}\widetilde{L}_{\mathsf{K}} = \Phi_{\mathsf{aux}}\widetilde{A}\widetilde{L}_{\mathsf{K}} = \Phi_{\mathsf{aux}}\widetilde{L}_{\mathsf{L}}\end{aligned}$$

with choice $\widetilde{L}_{\mathsf{K}} := \widetilde{\Sigma}^{\mathsf{x}}\Phi_{\mathsf{aux}}^{\mathsf{T}}\mathbf{C}^{\mathsf{T}}(\mathbf{C}\mathbf{\Sigma}^{\mathsf{x}}\mathbf{C}^{\mathsf{T}} + \Sigma^{\mathsf{v}})^{-1}$ and $\widetilde{L}_{\mathsf{L}} := \widetilde{A}\widetilde{L}_{\mathsf{K}}$, where we used $\mathbf{\Sigma}^{\mathsf{x}} = \Phi_{\mathsf{aux}}\widetilde{\Sigma}^{\mathsf{x}}\Phi_{\mathsf{aux}}^{\mathsf{T}}$ and $\mathbf{A}\Phi_{\mathsf{aux}} = \Phi_{\mathsf{aux}}\widetilde{A}$ from Claim 4.9.1. With definitions of $L_{\mathsf{K}}, L_{\mathsf{L}}, \mathbf{L}_{\mathsf{K}}, \mathbf{L}_{\mathsf{L}}$, we have

$$\begin{aligned}L_{\mathsf{K}} &:= \Sigma^{\mathsf{x}}C^{\mathsf{T}}(C\Sigma^{\mathsf{x}}C^{\mathsf{T}} + \Sigma^{\mathsf{v}})^{-1} \\ &= \Phi\mathbf{\Sigma}^{\mathsf{x}}\mathbf{C}^{\mathsf{T}}(\mathbf{C}\mathbf{\Sigma}^{\mathsf{x}}\mathbf{C}^{\mathsf{T}} + \Sigma^{\mathsf{v}})^{-1} = \Phi\mathbf{L}_{\mathsf{K}}, \\ L_{\mathsf{L}} &:= AL_{\mathsf{K}} = \Phi\mathbf{A}\mathbf{L}_{\mathsf{K}} = \Phi\mathbf{L}_{\mathsf{L}},\end{aligned}$$

where we used $C\Sigma^{\mathsf{x}} = \mathbf{C}\mathbf{\Sigma}^{\mathsf{x}}\Phi^{\mathsf{T}}$ and $C\Sigma^{\mathsf{x}}C^{\mathsf{T}} = \mathbf{C}\mathbf{\Sigma}^{\mathsf{x}}\mathbf{C}^{\mathsf{T}}$ through Claim 4.11.3 with selection $(M, \mathbf{M}, \widetilde{M}) \leftarrow (\Sigma^{\mathsf{x}}, \mathbf{\Sigma}^{\mathsf{x}}, \widetilde{\Sigma}^{\mathsf{x}})$, and used $AL_{\mathsf{K}} = \Phi\mathbf{A}\mathbf{L}_{\mathsf{K}}$ implied by $Av = \Phi\mathbf{A}\boldsymbol{v}$ from Claim 4.11.3 where $v, \boldsymbol{v}, \tilde{v}$ are chosen as the i-th columns of $L_{\mathsf{K}}, \mathbf{L}_{\mathsf{K}}, \widetilde{L}_{\mathsf{K}}$, respectively, for $i \in \{1, \ldots, p\}$. ♦

D. Proof of (4.57) and (4.58)

With the results in Subsection A, Subsection B and Subsection C, we are able to prove (4.57) as Claim 4.11.6 which follows after Claim 4.11.5, and prove (4.58) as (c) in Claim 4.11.7.

Claim 4.11.5. *For matrices Ξ, Λ_s in (4.19) and $\mathbf{\Xi}, \mathbf{\Lambda}_s$ in (4.53), we have (a) $\Xi = \mathbf{\Xi}$, and (b) $\Lambda_s = \Phi\mathbf{\Lambda}_s\Phi^{\mathsf{T}}$ and $\mathbf{\Lambda}_s = \Phi_{\mathsf{aux}}\widetilde{\Lambda}_s\Phi_{\mathsf{aux}}^{\mathsf{T}}$ with some $\widetilde{\Lambda}_s \in \mathbb{S}_+^{n_\xi}$, for all $s \in \mathbb{Z}_{[0,N)}$.*

Proof. The relation (a) $\Xi = \mathbf{\Xi}$ follows from the definitions $\Xi := C\Sigma^{\mathsf{x}}C^{\mathsf{T}} + \Sigma^{\mathsf{v}}$ and $\mathbf{\Xi} := \mathbf{C}\mathbf{\Sigma}^{\mathsf{x}}\mathbf{C}^{\mathsf{T}} + \Sigma^{\mathsf{v}}$ and from the relation $C\Sigma^{\mathsf{x}}C^{\mathsf{T}} = \mathbf{C}\mathbf{\Sigma}^{\mathsf{x}}\mathbf{C}^{\mathsf{T}}$ through Claim 4.11.3 with selection $(M, \mathbf{M}, \widetilde{M}) \leftarrow (\Sigma^{\mathsf{x}}, \mathbf{\Sigma}^{\mathsf{x}}, \widetilde{\Sigma}^{\mathsf{x}})$ given (a) in Claim 4.11.4.

For an intermediate result, we show for $r \in \mathbb{N}_{\geq 0}$ that

$$(A - BK)^r L_{\mathsf{L}} = \Phi(\mathbf{A} - \mathbf{B}\mathbf{K})^r \mathbf{L}_{\mathsf{L}}, \tag{4.95a}$$

$$(\mathbf{A} - \mathbf{B}\mathbf{K})^r \mathbf{L}_{\mathsf{L}} = \Phi_{\mathsf{aux}}(\widetilde{A} - \widetilde{B}\widetilde{K})^r \widetilde{L}_{\mathsf{L}}, \tag{4.95b}$$

with $\widetilde{L}_{\mathsf{L}}$ in Claim 4.11.4 and $\widetilde{K} := \mathbf{K}\Phi_{\mathsf{aux}}$. We show (4.95b) by induction: the base case of $r = 0$ as $\mathbf{L}_{\mathsf{L}} = \Phi_{\mathsf{aux}}\widetilde{L}_{\mathsf{L}}$ is from Claim 4.11.4; given the $r = \varsigma$ case of (4.95b), we have

$$\begin{aligned}
&(\mathbf{A} - \mathbf{BK})^{\varsigma+1}\mathbf{L}_{\mathsf{L}} = (\mathbf{A}-\mathbf{BK})(\mathbf{A}-\mathbf{BK})^{\varsigma}\mathbf{L}_{\mathsf{L}} \\
&= (\mathbf{A}-\mathbf{BK})\Phi_{\mathsf{aux}}(\widetilde{A} - \widetilde{B}\widetilde{K})^{\varsigma}\widetilde{L}_{\mathsf{L}} \\
&= \Phi_{\mathsf{aux}}(\widetilde{A} - \widetilde{B}\widetilde{K})(\widetilde{A} - \widetilde{B}\widetilde{K})^{\varsigma}\widetilde{L}_{\mathsf{L}} = \Phi_{\mathsf{aux}}(\widetilde{A} - \widetilde{B}\widetilde{K})^{\varsigma+1}\widetilde{L}_{\mathsf{L}}
\end{aligned}$$

as the $r = \varsigma + 1$ case of (4.95b), where the second equality applied the $r = \varsigma$ case, and the third equality used $\mathbf{A}\Phi_{\mathsf{aux}} = \Phi_{\mathsf{aux}}\widetilde{A}$ and $\mathbf{B} = \Phi_{\mathsf{aux}}\widetilde{B}$ from Claim 4.9.1. We then show (4.95a) by induction: the base case of $r = 0$ as $L_{\mathsf{L}} = \Phi\mathbf{L}_{\mathsf{L}}$ is from Claim 4.11.4; given the $r = \varsigma$ case of (4.95a), we have

$$\begin{aligned}
&(A - BK)^{\varsigma+1}L_{\mathsf{L}} = (A - BK)(A - BK)^{\varsigma}L_{\mathsf{L}} \\
&= (A - BK)\Phi(\mathbf{A} - \mathbf{BK})^{\varsigma}\mathbf{L}_{\mathsf{L}} \\
&= (A - BK)\Phi\Phi_{\mathsf{aux}}(\widetilde{A} - \widetilde{B}\widetilde{K})^{\varsigma}\widetilde{L}_{\mathsf{L}} \\
&= \Phi(\mathbf{A} - \mathbf{BK})\Phi_{\mathsf{aux}}(\widetilde{A} - \widetilde{B}\widetilde{K})^{\varsigma}\widetilde{L}_{\mathsf{L}} \\
&= \Phi(\mathbf{A} - \mathbf{BK})(\mathbf{A} - \mathbf{BK})^{\varsigma}\mathbf{L}_{\mathsf{L}} = \Phi(\mathbf{A} - \mathbf{BK})^{\varsigma+1}\mathbf{L}_{\mathsf{L}},
\end{aligned}$$

as the $r = \varsigma + 1$ case of (4.95a), where the second equality applied the $r = \varsigma$ case, the third and fifth equalities used (4.95b), and the fourth equality used $A\Phi\Phi_{\mathsf{aux}} = \Phi\mathbf{A}\Phi_{\mathsf{aux}}$ and $B = \Phi\mathbf{B}$ in Claim 4.11.1 and used $K\Phi\Phi_{\mathsf{aux}} = \mathbf{K}\Phi_{\mathsf{aux}}$ as Claim 4.11.2.

Through the relations (4.95a), (4.95b) and $\Xi = \boldsymbol{\Xi}$, we obtain (b) $\Lambda_s = \Phi\boldsymbol{\Lambda}_s\Phi^{\mathsf{T}}$ and $\boldsymbol{\Lambda}_s = \Phi_{\mathsf{aux}}\widetilde{\Lambda}_s\Phi_{\mathsf{aux}}^{\mathsf{T}}$ by choosing

$$\widetilde{\Lambda}_s := \textstyle\sum_{r=0}^{s}(\widetilde{A} - \widetilde{B}\widetilde{K})^r\,\widetilde{L}_{\mathsf{L}}\boldsymbol{\Xi}\widetilde{L}_{\mathsf{L}}(\widetilde{A} - \widetilde{B}\widetilde{K})^{r\mathsf{T}}. \qquad \blacklozenge$$

Claim 4.11.6. *For matrices Δ_s in (4.19) and $\boldsymbol{\Delta}_s$ in (4.53), we have $\Delta_s = \boldsymbol{\Delta}_s$ for $s \in \mathbb{Z}_{[0,N)}$, i.e., relation (4.57) holds.*

Proof. Given definitions (4.19a), (4.53a) of $\Delta_s, \boldsymbol{\Delta}_s$ and the relation $\Xi = \boldsymbol{\Xi}$ as (a) in Claim 4.11.5, it suffices to show $C\Lambda_sC^{\mathsf{T}} = \mathbf{C}\boldsymbol{\Lambda}_s\mathbf{C}^{\mathsf{T}}$, $K\Lambda_sK^{\mathsf{T}} = \mathbf{K}\boldsymbol{\Lambda}_s\mathbf{K}^{\mathsf{T}}$ and $C\Lambda_sK^{\mathsf{T}} = \mathbf{C}\boldsymbol{\Lambda}_s\mathbf{K}^{\mathsf{T}}$, which relations are obtained through Claim 4.11.3 with selection $(M, \mathbf{M}, \widetilde{M}) \leftarrow (\Lambda_s, \boldsymbol{\Lambda}_s, \widetilde{\Lambda}_s)$ given (b) in Claim 4.11.5. $\blacklozenge$

Claim 4.11.7. *With given $\bar{u}_{[k,k+N)}$ and given θ, if $\mu_k^{\hat{\mathsf{x}}}, \mu_k^{\bar{\mathsf{x}}}, \boldsymbol{\mu}_k^{\bar{\mathsf{x}}}, \boldsymbol{\mu}_k^{\hat{\mathsf{x}}}$ satisfy (4.56) for some $\widetilde{\mu}_k^{\hat{\mathsf{x}}}, \widetilde{\mu}_k^{\bar{\mathsf{x}}}$, then we have*

(a) $\mu_k^{\mathsf{x}} = \Phi\boldsymbol{\mu}_k^{\mathsf{x}}$ and $\boldsymbol{\mu}_k^{\mathsf{x}} = \Phi_{\mathsf{aux}}\widetilde{\mu}_k^{\mathsf{x}}$ with some $\widetilde{\mu}_k^{\mathsf{x}} \in \mathbb{R}^{n_\xi}$,

(b) $\bar{x}_t = \Phi\,\bar{\mathbf{x}}_t$ *and* $\bar{\mathbf{x}}_t = \Phi_{\mathsf{aux}}\,\widetilde{\bar{x}}_t$ *with some* $\widetilde{\bar{x}}_t \in \mathbb{R}^{n_\xi}$*, for all* $t \in \mathbb{Z}_{[k,k+N)}$*, and*
(c) $\bar{y}_t = \bar{\mathbf{y}}_t$ *for all* $t \in \mathbb{Z}_{[k,k+N)}$*, i.e., relation* (4.58) *holds.*

Proof. (a) We obtain $\mu_k^{\mathsf{x}} = \Phi_{\mathsf{orig}}\widetilde{\mu}_k^{\mathsf{x}}$ by combining (4.14) and (4.56), and obtain $\boldsymbol{\mu}_k^{\mathsf{x}} = \Phi_{\mathsf{aux}}\widetilde{\mu}_k^{\mathsf{x}}$ by combining (4.45) and (4.56), where we let $\widetilde{\mu}_k^{\mathsf{x}} := (1-\theta)\widetilde{\mu}_k^{\hat{\mathsf{x}}} + \theta\widetilde{\mu}_k^{\bar{\mathsf{x}}}$. Then, $\mu_k^{\mathsf{x}} = \Phi\boldsymbol{\mu}_k^{\mathsf{x}}$ follows from $\Phi_{\mathsf{orig}} = \Phi\Phi_{\mathsf{aux}}$, and thus (a) is proved.

(b) Prove by induction. **Base Case.** Select $\widetilde{\bar{x}}_k := \widetilde{\mu}_k^{\mathsf{x}}$. The $t = k$ case of (b) follows from (a) and relations $\bar{x}_k := \mu_k^{\mathsf{x}}$ as (4.12c) and $\bar{\mathbf{x}}_k := \boldsymbol{\mu}_k^{\mathsf{x}}$ as (4.49c). **Inductive Step.** Assume the $t = \tau$ case of (b) for some $\tau \in \mathbb{Z}_{[k,k+N-2]}$, and thus we have

$$\bar{x}_{\tau+1} \overset{\text{via (4.12a)}}{=} A\bar{x}_\tau + B\bar{u}_\tau = \Phi\mathbf{A}\bar{\mathbf{x}}_\tau + \Phi\mathbf{B}\bar{u}_\tau \overset{\text{via (4.49a)}}{=} \Phi\bar{\mathbf{x}}_{\tau+1},$$

where the second equality used $B = \Phi\mathbf{B}$ in Claim 4.11.1 and $A\bar{x}_\tau = \Phi\mathbf{A}\bar{\mathbf{x}}_\tau$ through Claim 4.11.3 with selection $(v, \boldsymbol{v}, \tilde{v}) \leftarrow (\bar{x}_\tau, \bar{\mathbf{x}}_\tau, \widetilde{\bar{x}}_\tau)$ given (b) of $t = \tau$. Moreover, we have

$$\bar{\mathbf{x}}_{\tau+1} \overset{\text{via (4.49a)}}{=} \mathbf{A}\bar{\mathbf{x}}_\tau + \mathbf{B}\bar{u}_\tau = \Phi_{\mathsf{aux}}\widetilde{A}\,\widetilde{\bar{x}}_\tau + \Phi_{\mathsf{aux}}\widetilde{B}\bar{u}_\tau = \Phi_{\mathsf{aux}}\widetilde{\bar{x}}_{\tau+1}$$

by choosing $\widetilde{\bar{x}}_{\tau+1} := \widetilde{A}\,\widetilde{\bar{x}}_\tau + \widetilde{B}\bar{u}_\tau$, where the second equality used $\mathbf{B} = \Phi_{\mathsf{aux}}\widetilde{B}$ in Claim 4.9.1 and $\mathbf{A}\bar{\mathbf{x}}_\tau = \Phi_{\mathsf{aux}}\widetilde{A}\,\widetilde{\bar{x}}_\tau$ through Claim 4.11.3 with $(v, \boldsymbol{v}, \tilde{v}) \leftarrow (\bar{x}_\tau, \bar{\mathbf{x}}_\tau, \widetilde{\bar{x}}_\tau)$ given (b) of $t = \tau$. Thus, we have the $t = \tau + 1$ case of (b). This shows (b).

(c) We have $C\bar{x}_t = \mathbf{C}\bar{\mathbf{x}}_t$ applying Claim 4.11.3 with selection $(v, \boldsymbol{v}, \tilde{v}) \leftarrow (\bar{x}_t, \bar{\mathbf{x}}_t, \widetilde{\bar{x}}_t)$ given (b). Thus, (c) is obtained as

$$\bar{y}_t \overset{\text{via (4.12b)}}{=} C\bar{x}_t + D\bar{u}_t = \mathbf{C}\bar{\mathbf{x}}_t + D\bar{u}_t \overset{\text{via (4.49b)}}{=} \bar{\mathbf{y}}_t. \qquad \blacklozenge$$

4.6.9 Proof of Claim 4.13.1

Proof. We first show an extended result Claim 4.13.2 which implies Claim 4.13.1. Recall the matrices $\Phi \in \mathbb{R}^{n\times n_{\mathsf{aux}}}$, $\Phi_{\mathsf{orig}} \in \mathbb{R}^{n\times n_\xi}$ and $\Phi_{\mathsf{aux}} \in \mathbb{R}^{n_{\mathsf{aux}}\times n_\xi}$ used in Section 4.6.8.

Claim 4.13.2. *At control step* $t = k$ *in processes a) and b), if*

i) the states $x_k^{\mathsf{a}} = x_k^{\mathsf{b}}$ *are equal in processes a) and b), and*
ii) the parameters $\mu_k^{\hat{\mathsf{x}}}, \mu_k^{\bar{\mathsf{x}}}$ *in process a) and the parameters* $\boldsymbol{\mu}_k^{\hat{\mathsf{x}}}, \boldsymbol{\mu}_k^{\bar{\mathsf{x}}}$ *in process b) satisfy* (4.56)*,*

then, for $t \in \mathbb{Z}_{[k,k+N_{\mathrm{c}}]}$*, we have*

(a) the states $x_t^{\mathsf{a}} = x_t^{\mathsf{b}}$ *are equal in processes a) and b),*

(b) the variable $\hat{x}_t^-$ in process a) and the variable $\hat{\mathbf{x}}_t^-$ in process b) satisfy $\hat{x}_t^- = \Phi\hat{\mathbf{x}}_t^-$ and $\hat{\mathbf{x}}_t^- = \Phi_{\mathsf{aux}}\widetilde{\hat{x}_t^-}$ for some $\widetilde{\hat{x}_t^-} \in \mathbb{R}^{n_\xi}$,

and, for $t \in \mathbb{Z}_{[k,k+N_c)}$, we have

(c) the inputs $u_t^{\mathsf{a}} = u_t^{\mathsf{b}}$ are equal in processes a) and b).

(d) the outputs $y_t^{\mathsf{a}} = y_t^{\mathsf{b}}$ are equal in processes a) and b),

Proof. We prove by induction. **Base Case.** We show (a) and (b) for $t = k$. Result (a) of $t = k$ is exactly as condition i). Through Proposition 4.11 and the fact that both problems (SMPC) and (SDDPC) produce unique optimal θ, the values of θ are the same in processes a) and b). Given condition ii), μ_k^{x} in process a) and $\boldsymbol{\mu}_k^{\mathsf{x}}$ in process b) satisfy $\mu_k^{\mathsf{x}} = \Phi\boldsymbol{\mu}_k^{\mathsf{x}}$ and $\boldsymbol{\mu}_k^{\mathsf{x}} = \Phi_{\mathsf{aux}}\widetilde{\mu}_k^{\mathsf{x}}$ for some $\widetilde{\mu}_k^{\mathsf{x}}$ according to Claim 4.11.7. Combining these relations with $\hat{x}_k^- := \mu_k^{\mathsf{x}}$ as (4.13d) and $\hat{\mathbf{x}}_k^- := \boldsymbol{\mu}_k^{\mathsf{x}}$ as (4.48c), we obtain (b) of $t = k$ by choosing $\widetilde{\hat{x}_k^-} := \widetilde{\mu}_k^{\mathsf{x}}$, as

$$\hat{x}_k^- = \mu_k^{\mathsf{x}} = \Phi\boldsymbol{\mu}_k^{\mathsf{x}} = \Phi\hat{\mathbf{x}}_k^-, \qquad \hat{\mathbf{x}}_k^- = \boldsymbol{\mu}_k^{\mathsf{x}} = \Phi_{\mathsf{aux}}\widetilde{\mu}_k^{\mathsf{x}} = \Phi_{\mathsf{aux}}\widetilde{\hat{x}_k^-}.$$

Inductive Step. We assume (a) and (b) for $t = \tau \in \mathbb{Z}_{[k,k+N_c)}$, and then prove (c), (d) for $t = \tau$ and (a), (b) for $t = \tau + 1$. The control inputs u_τ^{a}, u_τ^{b} are obtained through (4.15) and (4.50) respectively, where the nominal inputs $\bar{u}_\tau$ are the same according to Proposition 4.11 and the fact that both problems (SMPC), (SDDPC) produce a unique optimal $\bar{u}$, i.e.,

$$u_\tau^{\mathsf{a}} = \bar{u}_\tau - K(\hat{x}_\tau - \bar{x}_\tau), \quad u_\tau^{\mathsf{b}} = \bar{u}_\tau - \mathbf{K}(\hat{\mathbf{x}}_\tau - \bar{\mathbf{x}}_\tau).$$

Thus, we have (c) $u_\tau^{\mathsf{a}} = u_\tau^{\mathsf{b}}$ of $t = \tau$, because of $K\hat{x}_\tau = \mathbf{K}\hat{\mathbf{x}}_\tau$ by applying Claim 4.11.3 with $(v, \boldsymbol{v}, \tilde{v}) \leftarrow (\hat{x}_\tau^-, \hat{\mathbf{x}}_\tau^-, \widetilde{\hat{x}_\tau^-})$ where $\hat{x}_\tau^-, \hat{\mathbf{x}}_\tau^-, \widetilde{\hat{x}_\tau^-}$ satisfy (b) of $t = \tau$, and because of $K\bar{x}_\tau = \mathbf{K}\bar{\mathbf{x}}_\tau$ by applying Claim 4.11.3 with $(v, \boldsymbol{v}, \tilde{v}) \leftarrow (\bar{x}_\tau, \bar{\mathbf{x}}_\tau, \widetilde{\bar{x}}_\tau)$ given $\bar{x}_\tau = \Phi\bar{\mathbf{x}}_\tau$ and $\bar{\mathbf{x}}_\tau = \Phi_{\mathsf{aux}}\widetilde{\bar{x}}_\tau$ via Claim 4.11.7. It follows that (d) $y_\tau^{\mathsf{a}} = y_\tau^{\mathsf{b}}$ holds for $t = \tau$ and (a) $x_{\tau+1}^{\mathsf{a}} = x_{\tau+1}^{\mathsf{b}}$ holds for $t = \tau + 1$, according to the system dynamics $y_\tau^{\mathsf{z}} = Cx_\tau^{\mathsf{z}} + Du_\tau^{\mathsf{z}} + v_t$ as (4.1b) and $x_{\tau+1}^{\mathsf{z}} = Ax_\tau^{\mathsf{z}} + Bu_\tau^{\mathsf{z}} + w_t$ as (4.1a), for $\mathsf{z} \in \{\mathsf{a}, \mathsf{b}\}$. Finally, we prove (b) for $t = \tau + 1$ as follows,

$$\begin{aligned}
&\hat{x}_{\tau+1}^- \overset{\text{via (4.13)}}{=} A\hat{x}_\tau^- + Bu_\tau^{\mathsf{a}} + L_{\mathsf{L}}(y_\tau^{\mathsf{a}} - C\hat{x}_\tau^-) \\
&= \Phi\mathbf{A}\hat{\mathbf{x}}_\tau + \Phi\mathbf{B}u_\tau^{\mathsf{b}} + \Phi\mathbf{L}_{\mathsf{L}}(y_\tau^{\mathsf{b}} - \mathbf{C}\hat{\mathbf{x}}_\tau^-) \overset{\text{via (4.48)}}{=} \Phi\hat{\mathbf{x}}_{\tau+1}^- \\
&\hat{\mathbf{x}}_{\tau+1}^- \overset{\text{via (4.48)}}{=} \mathbf{A}\hat{\mathbf{x}}_\tau + \mathbf{B}u_\tau^{\mathsf{b}} + \mathbf{L}_{\mathsf{L}}(y_\tau^{\mathsf{b}} - \mathbf{C}\hat{\mathbf{x}}_\tau^-) \\
&= \Phi_{\mathsf{aux}}\widetilde{A}\,\widetilde{\hat{x}}_\tau + \Phi_{\mathsf{aux}}\widetilde{B}u_\tau^{\mathsf{b}} + \Phi_{\mathsf{aux}}\widetilde{L}_{\mathsf{L}}(y_\tau^{\mathsf{b}} - \mathbf{C}\hat{\mathbf{x}}_\tau^-) = \Phi_{\mathsf{aux}}\widetilde{\hat{x}}_{\tau+1}^-
\end{aligned}$$

by choosing $\widetilde{\hat{x}}^-_{\tau+1} := \widetilde{A}\,\widetilde{\hat{x}}_\tau + \widetilde{B}u^{\mathsf{b}}_\tau + \widetilde{L}_{\mathsf{L}}(y^{\mathsf{b}}_\tau - \mathbf{C}\hat{\mathbf{x}}^-_\tau)$, where we used $B = \Phi\mathbf{B}$ in Claim 4.11.1, $\mathbf{B} = \Phi_{\mathsf{aux}}\widetilde{B}$ in Claim 4.9.1, $L_{\mathsf{L}} = \Phi\mathbf{L}_{\mathsf{L}}$ and $\mathbf{L}_{\mathsf{L}} = \Phi\widetilde{L}_{\mathsf{L}}$ in Claim 4.11.4, and $A\hat{x}^-_\tau = \Phi\mathbf{A}\hat{\mathbf{x}}_\tau$ and $\mathbf{A}\hat{\mathbf{x}}^-_\tau = \Phi_{\mathsf{aux}}\widetilde{A}\,\widetilde{\hat{x}}^-_\tau$ by applying Claim 4.11.3 with $(v, \boldsymbol{v}, \tilde{v}) \leftarrow (x^-_\tau, \mathbf{x}^-_\tau, \widetilde{x}^-_\tau)$ where $x^-_\tau, \mathbf{x}^-_\tau, \widetilde{x}^-_\tau$ satisfy (b) of $t = \tau$. Hence, we proved (c), (d) for $t = \tau$ and (a), (b) for $t = \tau + 1$. The result follows by induction. ♦

The result 1) in Claim 4.13.1 is covered by (a), (c), (d) of Claim 4.13.2. The rest of the proof shows the result 2) in Claim 4.13.1. From (b) of Claim 4.13.2 with $t = k + N_{\mathrm{c}}$, we have

$$\hat{x}^-_{k+N_{\mathrm{c}}} = \Phi\Phi_{\mathsf{aux}}\widetilde{\hat{x}}^-_{k+N_{\mathrm{c}}}, \qquad \hat{\mathbf{x}}^-_{k+N_{\mathrm{c}}} = \Phi_{\mathsf{aux}}\widetilde{\hat{x}}^-_{k+N_{\mathrm{c}}}. \tag{4.96}$$

From Claim 4.11.7 with $t = k + N_{\mathrm{c}}$, we have

$$\bar{x}_{k+N_{\mathrm{c}}} = \Phi\Phi_{\mathsf{aux}}\widetilde{\bar{x}}_{k+N_{\mathrm{c}}}, \qquad \bar{\mathbf{x}}_{k+N_{\mathrm{c}}} = \Phi_{\mathsf{aux}}\widetilde{\bar{x}}_{k+N_{\mathrm{c}}}. \tag{4.97}$$

Recall that $\mu^{\hat{x}}_{k+N_{\mathrm{c}}}, \mu^{\bar{x}}_{k+N_{\mathrm{c}}}$ in Algorithm 5 and $\boldsymbol{\mu}^{\hat{x}}_{k+N_{\mathrm{c}}}, \boldsymbol{\mu}^{\bar{x}}_{k+N_{\mathrm{c}}}$ in Algorithm 6 are obtained through (4.36) and (4.59) respectively. Combine (4.96) with (4.36) and combine (4.97) with (4.59), where we note $\Phi_{\mathsf{orig}} = \Phi\Phi_{\mathsf{aux}}$, and then we obtain relation (4.56) with $k + N_{\mathrm{c}}$ in place of k, where we select $\widetilde{\mu}^{\hat{x}}_{k+N_{\mathrm{c}}} := \widetilde{\hat{x}}^-_{k+N_{\mathrm{c}}}$ and $\widetilde{\mu}^{\bar{x}}_{k+N_{\mathrm{c}}} := \widetilde{\bar{x}}_{k+N_{\mathrm{c}}}$; this is the result 2) in Claim 4.13.1. ■

Chapter 5

Conclusions and Outlook

Data-driven predictive control (DDPC), as a subfield of data-driven control, has emerged as a prominent research topic in recent years. DDPC methods such as Data-enabled Predictive Control (DeePC) and Subspace Predictive Control (SPC) have demonstrated practical utility through various experimental applications. On the theoretical side, it has been established that both DeePC and SPC can yield control actions identical to those produced by Model Predictive Control (MPC) for deterministic linear time-invariant (LTI) systems. This theoretical equivalence between data-driven and model-based control has captured our interest and forms the foundation of our research. Our overarching research objective is to extend this equivalence to broader classes of systems beyond deterministic LTI systems.

5.1 Contributions

In this thesis, we contributed to the field by developing DDPC methods for linear time-periodic (LTP) systems, as a type of linear-time varying (LTV) systems (Chapter 3), and proposing a novel DDPC framework for stochastic LTI systems (Chapter 4). Both frameworks were shown to achieve equivalence with MPC or Stochastic MPC under specific conditions, providing theoretical advancements of DDPC to more complex system types beyond deterministic LTI systems. Our proposed DDPC methods were also validated through simulations, demonstrating practical advantages over benchmark control methods.

In Chapter 3, we focused on LTP systems of known periods, as a specific type of LTV systems. The classical lifting technique of LTP systems is reviewed as a preliminary.

We reviewed some established results in the behavioral systems theory for LTV systems and then developed a definition of order and lag for LTV systems, which extends the notions of order and lag which were defined only for LTI systems in the previous behavioral framework. For LTP systems in particular, we constructed the relation of the behaviors of LTP systems and their lifted systems, and established a fundamental lemma for LTP systems, which generalizes Willems' fundamental lemma [31] from LTI systems to LTP systems. We finally proposed Periodic DeePC (P-DeePC) and Periodic SPC (P-SPC) methods for LTP systems, which DDPC methods produce equivalent control actions with classical MPC for deterministic LTP systems, under tuning conditions. Simulation results provide evidence that regularized versions of the approach is robust to noise-corrupted data.

In Chapter 4, we proposed a Stochastic Data-Driven Predictive Control (SDDPC) framework, accounting for either chance constraints or distributionally robust Conditional Value-at-Risk (DR-CVaR) constraints. As an intermediate step, we introduced an auxiliary model of a stochastic LTI system, as a specific (non-minimal) realization of the system. The auxiliary model can be accurately represented by noise-free offline data and be approximated by noisy offline data from the system. Using the auxiliary model, we constructed our SDDPC framework with components analogous to those in Stochastic MPC (SMPC), such as data-driven feedback policies and data-driven state estimation. Theoretically, we demonstrated that our SDDPC method can generate control inputs equivalent to those of SMPC under tuning conditions and the assumption of noise-free offline data. While in practice offline data is noisy, our data-driven control method achieved reliable tracking performances in simulations. Simulation results further highlight the practical advantages of our approach compared to benchmark control methods, showing both lower cumulative tracking costs and lower rates and amounts of constraint violations.

5.2 Potential Future Topics

Based on our broad objective to extend the equivalence between model-based and data-driven control beyond deterministic LTI systems, there are other relevant research topics beyond the scope of the thesis. As mentioned in Section 1.4, the extension can be pursued for stochastic system, non-linear systems and time-varying systems. Here, we enumerate several relevant topics of the author's interest.

Behavioral Systems Theory for Stochastic Systems

The behavioral systems theory was originally developed for deterministic systems [23, 24, 25, 26, 27, 28, 29, 30], and extending it to stochastic systems remains an open and active area, with some initial attempts [112, 113, 114]. Due to the close relationship between the behavioral theory and DDPC methods, extension of behavioral systems theory for stochastic systems would facilitate the further development of stochastic DDPC. Currently, while several stochastic DDPC methods [78, 79, 73, 74, 75, 76] have been established with theoretical performance equivalence with SMPC, those theoretical results rely on strong and impractical assumptions on offline data, such as noise-free offline data [78, 79] and knowledge of the offline disturbance signal [73, 74, 75, 76]. Establishing a stochastic behavioral theory could pave the way for the development of stochastic DDPC methods that achieve equivalence to SMPC, while relying on more practical and less restrictive assumptions about offline data.

The author envisions that developing a stochastic behavioral theory could address two key challenges: capturing the stochasticity in online data and accounting for the uncertainties in offline data. First, the behavior of stochastic systems could be formulated in a distributional manner, offering a more complex representation than the single behavior set used for deterministic systems. Unlike in the deterministic case, where a trajectory either belongs to or lies outside the behavior set, in the stochastic case, a trajectory may belong to the behavior with a probability. This probabilistic formulation reflects the inherent stochasticity of online data trajectories in stochastic systems.

Second, one may consider a distribution or an ambiguity set of possible behaviors, since the behavior identified using Willems' fundamental identified is based on offline data, which is inherently noisy in stochastic systems. This topic can be framed as the development of a stochastic version of Willems' fundamental lemma, addressing the impact of uncertainty of offline data.

Data-Driven Predictive Control for a Variety of Time-Varying Systems

Several DDPC methods have been established for specific types of time-varying systems, as reviewed in Section 1.4. DDPC methods for linear time-periodic (LTP) systems [77] and linear parameter-varying (LPV) systems [57, 58] have been established with theoretical equivalence to MPC methods, supported by associated extensions of Willems' fundamental lemma. However, for other types of time-varying systems, existing DDPC methods (e.g. [59, 60]) were developed without a theoretical connection to model-based control methods, although they provided performance guarantees such as closed-loop stability. Therefore, an

open and active research topic is to develop DDPC methods with theoretical equivalence to model-based methods for a broader range of time-varying systems, such as slowly-varying systems. This topic is closely related to further extensions of Willems' fundamental lemma and novel developments in behavioral systems theory for time-varying cases.

References

[1] Florian Dörfler. Data-driven control: Part two of two: Hot take: Why not go with models? *IEEE Control Syst. Mag.*, 43(6):27–31, 2023.

[2] Håkan Hjalmarsson. From experiment design to closed-loop control. *Automatica*, 41(3):393–438, 2005.

[3] Babatunde A Ogunnaike. A contemporary industrial perspective on process control theory and practice. *Annu Rev Control*, 20:1–8, 1996.

[4] Zhong-Sheng Hou and Zhuo Wang. From model-based control to data-driven control: Survey, classification and perspective. *Inf. Sci.*, 235:3–35, 2013.

[5] David Q Mayne. Model predictive control: Recent developments and future promise. *Automatica*, 50(12):2967–2986, 2014.

[6] Matthew Brown, Joseph Funke, Stephen Erlien, and J Christian Gerdes. Safe driving envelopes for path tracking in autonomous vehicles. *Control Eng. Pract.*, 61:307–316, 2017.

[7] Ionela Prodan, Sorin Olaru, Ricardo Bencatel, Joao Borges de Sousa, Cristina Stoica, and Silviu-Iulian Niculescu. Receding horizon flight control for trajectory tracking of autonomous aerial vehicles. *Control Eng. Pract.*, 21(10):1334–1349, 2013.

[8] Francesco Bullo, Emilio Frazzoli, Marco Pavone, Ketan Savla, and Stephen L Smith. Dynamic vehicle routing for robotic systems. *Proc. IEEE*, 99(9):1482–1504, 2011.

[9] Georgios Darivianakis, Angelos Georghiou, Roy S Smith, and John Lygeros. The power of diversity: Data-driven robust predictive control for energy-efficient buildings and districts. *IEEE Trans. Control Syst. Tech.*, 27(1):132–145, 2017.

[10] Ali Mesbah. Stochastic model predictive control: An overview and perspectives for future research. *IEEE Control Syst. Mag.*, 36(6):30–44, 2016.

[11] Tor Aksel N Heirung, Joel A Paulson, Jared O'Leary, and Ali Mesbah. Stochastic model predictive control–how does it work? *Comput Chem Eng*, 114:158–170, 2018.

[12] Marcello Farina, Luca Giulioni, and Riccardo Scattolini. Stochastic linear model predictive control with chance constraints–a review. *J. Process Control*, 44:53–67, 2016.

[13] Ranjeet Kumar, Jordan Jalving, Michael J Wenzel, Matthew J Ellis, Mohammad N ElBsat, Kirk H Drees, and Victor M Zavala. Benchmarking stochastic and deterministic MPC: A case study in stationary battery systems. *AIChE Journal*, 65(7):e16551, 2019.

[14] Alberto Bemporad and Manfred Morari. Robust model predictive control: A survey. In *Robustness in identification and control*, pages 207–226. Springer, 2007.

[15] Biao Huang and Ramesh Kadali. *Dynamic modeling, predictive control and performance monitoring: a data-driven subspace approach*. Springer, 2008.

[16] Peter Van Overschee and Bart De Moor. *Subspace identification for linear systems: Theory—Implementation—Applications*. Springer Science & Business Media, 2012.

[17] Wouter Favoreel, Bart De Moor, and Michel Gevers. SPC: Subspace predictive control. *IFAC Proceedings Volumes*, 32(2):4004–4009, 1999.

[18] Wouter Favoreel, Bart De Moor, P Van Overschee, and M Gevers. Model-free subspace-based LQG-design. In *Proc. ACC*, volume 5, pages 3372–3376. IEEE, 1999.

[19] Redouane Hallouzi and Michel Verhaegen. Fault-tolerant subspace predictive control applied to a boeing 747 model. *Journal of guidance, control, and dynamics*, 31(4):873–883, 2008.

[20] Vineet Vajpayee, Siddhartha Mukhopadhyay, and Akhilanand Pati Tiwari. Data-driven subspace predictive control of a nuclear reactor. *IEEE Trans. Nucl. Sci.*, 65(2):666–679, 2017.

[21] Lukas Hewing, Kim P Wabersich, Marcel Menner, and Melanie N Zeilinger. Learning-based model predictive control: Toward safe learning in control. *Annual Review of Control, Robotics, and Autonomous Systems*, 3(1):269–296, 2020.

[22] Kanghua Zhang, Jixin Wang, Xueting Xin, Xiang Li, Chuanwen Sun, Jianfei Huang, and Weikang Kong. A survey on learning-based model predictive control: Toward path tracking control of mobile platforms. *Applied Sciences*, 12(4):1995, 2022.

[23] Ivan Markovsky and Florian Dörfler. Behavioral systems theory in data-driven analysis, signal processing, and control. *Annu Rev Control*, 52:42–64, 2021.

[24] Ivan Markovsky, Linbin Huang, and Florian Dörfler. Data-driven control based on behavioral approach: From theory to applications in power systems. Technical report, Technical report, 2022.

[25] Jan C Willems. From time series to linear system—Part I. Finite dimensional linear time invariant systems. *Automatica*, 22(5):561–580, 1986.

[26] Jan C Willems. From time series to linear system—Part II. Exact modelling. *Automatica*, 22(6):675–694, 1986.

[27] Jan C Willems. From time series to linear system—Part III: Approximate modelling. *Automatica*, 23(1):87–115, 1987.

[28] Jan C Willems. Paradigms and puzzles in the theory of dynamical systems. *IEEE Trans. Autom. Control*, 36(3):259–294, 1991.

[29] Jan C Willems. The behavioral approach to open and interconnected systems. *IEEE Control Syst. Mag.*, 27(6):46–99, 2007.

[30] Jan C Willems. In control, almost from the beginning until the day after tomorrow. *Eur. J. Control*, 13(1):71, 2007.

[31] Jan C Willems, Paolo Rapisarda, Ivan Markovsky, and Bart LM De Moor. A note on persistency of excitation. *IEEE Control Syst. Let.*, 54(4):325–329, 2005.

[32] Ivan Markovsky and Paolo Rapisarda. Data-driven simulation and control. *Int. J. Control*, 81(12):1946–1959, 2008.

[33] Claudio De Persis and Pietro Tesi. Formulas for data-driven control: Stabilization, optimality, and robustness. *IEEE Trans. Autom. Control*, 65(3):909–924, 2019.

[34] Claudio De Persis and Pietro Tesi. On persistency of excitation and formulas for data-driven control. In *Proc. IEEE CDC*, pages 873–878. IEEE, 2019.

[35] Julian Berberich, Anne Koch, Carsten W Scherer, and Frank Allgöwer. Robust data-driven state-feedback design. In *Proc. ACC*, pages 1532–1538. IEEE, 2020.

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

[37] Hua Yang and Shaoyuan Li. A data-driven predictive controller design based on reduced hankel matrix. In *Asian Ctrl. Conf.*, pages 1–7. IEEE, 2015.

[38] Jeremy Coulson, John Lygeros, and Florian Dörfler. Data-enabled predictive control: In the shallows of the deepc. In *Proc. ECC*, pages 307–312. IEEE, 2019.

[39] Jeremy Coulson, John Lygeros, and Florian Dörfler. Regularized and distributionally robust data-enabled predictive control. In *Proc. IEEE CDC*, pages 2696–2701. IEEE, 2019.

[40] Jeremy Coulson, John Lygeros, and Florian Dörfler. Distributionally robust chance constrained data-enabled predictive control. *IEEE Trans. Autom. Control*, 2021.

[41] Linbin Huang, Jeremy Coulson, John Lygeros, and Florian Dörfler. Data-enabled predictive control for grid-connected power converters. In *Proc. IEEE CDC*, pages 8130–8135. IEEE, 2019.

[42] Linbin Huang, Jeremy Coulson, John Lygeros, and Florian Dörfler. Decentralized data-enabled predictive control for power system oscillation damping. *IEEE Trans. Control Syst. Tech.*, 2021.

[43] Paolo Gherardo Carlet, Andrea Favato, Saverio Bolognani, and Florian Dörfler. Data-driven predictive current control for synchronous motor drives. In *ECCE*, pages 5148–5154. IEEE, 2020.

[44] Ezzat Elokda, Jeremy Coulson, Paul N Beuchat, John Lygeros, and Florian Dörfler. Data-enabled predictive control for quadcopters. *Int. J. Robust & Nonlinear Control*, 31(18):8916–8936, 2021.

[45] Felix Wegner. Data-enabled predictive control of robotic systems. Master's thesis, ETH Zurich, 2021.

[46] Julian Berberich, Johannes Köhler, Matthias A Müller, and Frank Allgöwer. Data-driven model predictive control with stability and robustness guarantees. *IEEE Trans. Autom. Control*, 66(4):1702–1717, 2020.

[47] Julian Berberich, Johannes Köhler, Matthias A Müller, and Frank Allgöwer. Data-driven tracking mpc for changing setpoints. *IFAC-PapersOnLine*, 53(2):6923–6930, 2020.

[48] Julian Berberich, Johannes Köhler, Matthias A Müller, and Frank Allgöwer. Robust constraint satisfaction in data-driven MPC. In *Proc. IEEE CDC*, pages 1260–1267. IEEE, 2020.

[49] Julian Berberich, Johannes Köhler, Matthias A Müller, and Frank Allgöwer. Data-driven model predictive control: closed-loop guarantees and experimental results. *AT-AUTOM*, 69(7):608–618, 2021.

[50] Felix Fiedler and Sergio Lucia. On the relationship between data-enabled predictive control and subspace predictive control. In *Proc. ECC*, pages 222–229. IEEE, 2021.

[51] Xiao-Suo Luo and Yong-Duan Song. Data-driven predictive control of hammerstein–wiener systems based on subspace identification. *Inf. Sci.*, 422:447–461, 2018.

[52] Julian Berberich and Frank Allgöwer. A trajectory-based framework for data-driven system analysis and control. In *Proc. ECC*, pages 1365–1370. IEEE, 2020.

[53] Juan G Rueda-Escobedo and Johannes Schiffer. Data-driven internal model control of second-order discrete volterra systems. In *Proc. IEEE CDC*, pages 4572–4579. IEEE, 2020.

[54] Mohammad Alsalti, Julian Berberich, Victor G Lopez, Frank Allgöwer, and Matthias A Müller. Data-based system analysis and control of flat nonlinear systems. In *Proc. IEEE CDC*, pages 1484–1489. IEEE, 2021.

[55] Ivan Markovsky. Data-driven simulation of generalized bilinear systems via linear time-invariant embedding. *IEEE Trans. Autom. Control*, 68(2):1101–1106, 2022.

[56] Linbin Huang, John Lygeros, and Florian Dörfler. Robust and kernelized data-enabled predictive control for nonlinear systems. *IEEE Trans. Control Syst. Tech.*, 2023.

[57] Chris Verhoek, Roland Tóth, Sofie Haesaert, and Anne Koch. Fundamental lemma for data-driven analysis of linear parameter-varying systems. In *Proc. IEEE CDC*, pages 5040–5046. IEEE, 2021.

[58] Chris Verhoek, Hossam S Abbas, Roland Tóth, and Sofie Haesaert. Data-driven predictive control for linear parameter-varying systems. *IFAC-PapersOnLine*, 54(8):101–108, 2021.

[59] Stefanos Baros, Chin-Yao Chang, Gabriel E Colon-Reyes, and Andrey Bernstein. Online data-enabled predictive control. *Automatica*, 138:109926, 2022.

[60] Kaijian Hu and Tao Liu. Robust data-driven predictive control for linear time-varying systems. *IEEE Control Syst. Let.*, 2024.

[61] Linbin Huang, Jianzhe Zhen, John Lygeros, and Florian Dörfler. Quadratic regularization of data-enabled predictive control: Theory and application to power converter experiments. *IFAC-PapersOnLine*, 54(7):192–197, 2021.

[62] Linbin Huang, Jianzhe Zhen, John Lygeros, and Florian Dörfler. Robust data-enabled predictive control: Tractable formulations and performance guarantees. *IEEE Trans. Autom. Control*, 68(5):3163–3170, 2023.

[63] Sebastian Kerz, Johannes Teutsch, Tim Brüdigam, Marion Leibold, and Dirk Wollherr. Data-driven tube-based stochastic predictive control. *IEEE Open J. Control Syst.*, 2023.

[64] Johannes Teutsch, Sebastian Kerz, Dirk Wollherr, and Marion Leibold. Sampling-based stochastic data-driven predictive control under data uncertainty. *arXiv preprint arXiv:2402.00681*, 2024.

[65] Yibo Wang, Keyou You, Dexian Huang, and Chao Shang. Data-driven output prediction and control of stochastic systems: An innovation-based approach. *Automatica*, 171:111897, 2025.

[66] Mingzhou Yin, Andrea Iannelli, and Roy S Smith. Stochastic data-driven predictive control: Regularization, estimation, and constraint tightening. *IFAC-PapersOnLine*, 58(15):79–84, 2024.

[67] Benjamin Gravell, Iman Shames, and Tyler Summers. Robust data-driven output feedback control via bootstrapped multiplicative noise. In *L4DC*, pages 650–662. PMLR, 2022.

[68] Yi Guo, Kyri Baker, Emiliano Dall'Anese, Zechun Hu, and Tyler Holt Summers. Data-based distributionally robust stochastic optimal power flow—part I: Methodologies. *IEEE Trans. Power Syst.*, 34(2):1483–1492, 2018.

[69] Yi Guo, Kyri Baker, Emiliano Dall'Anese, Zechun Hu, and Tyler Holt Summers. Data-based distributionally robust stochastic optimal power flow—part II: Case studies. *IEEE Trans. Power Syst.*, 34(2):1493–1503, 2018.

[70] Yi Guo, Kyri Baker, Emiliano Dall'Anese, Zechun Hu, and Tyler Summers. Stochastic optimal power flow based on data-driven distributionally robust optimization. In *Proc. ACC*, pages 3840–3846. IEEE, 2018.

[71] Francesco Micheli, Tyler Summers, and John Lygeros. Data-driven distributionally robust MPC for systems with uncertain dynamics. In *Proc. IEEE CDC*, pages 4788–4793. IEEE, 2022.

[72] Kyri Baker, Emiliano Dall'Anese, and Tyler Summers. Distribution-agnostic stochastic optimal power flow for distribution grids. In *Proc. NAPS*, pages 1–6. IEEE, 2016.

[73] Guanru Pan, Ruchuan Ou, and Timm Faulwasser. On a stochastic fundamental lemma and its use for data-driven optimal control. *IEEE Trans. Autom. Control*, 2022.

[74] Guanru Pan, Ruchuan Ou, and Timm Faulwasser. Towards data-driven stochastic predictive control. *Int. J. Robust Nonlinear Control*, 2022.

[75] Guanru Pan, Ruchuan Ou, and Timm Faulwasser. On data-driven stochastic output-feedback predictive control. *arXiv preprint arXiv:2211.17074*, 2023.

[76] Guanru Pan, Ruchuan Ou, and Timm Faulwasser. Data-driven stochastic output-feedback predictive control: Recursive feasibility through interpolated initial conditions. In *L4DC*, pages 980–992. PMLR, 2023.

[77] Ruiqi Li, John W. Simpson-Porco, and Stephen L. Smith. Data-driven model predictive control for linear time-periodic systems. In *Proc. IEEE CDC*, pages 3661–3668, 2022.

[78] Ruiqi Li, John W. Simpson-Porco, and Stephen L. Smith. Stochastic data-driven predictive control with equivalence to stochastic MPC. *arXiv preprint arXiv:2312.15177*, 2024.

[79] Ruiqi Li, John W. Simpson-Porco, and Stephen L. Smith. Distributionally robust stochastic data-driven predictive control with optimized feedback gain. *arXiv preprint arXiv:2409.05727*, 2024.

[80] Ivan Markovsky and Florian Dörfler. Identifiability in the behavioral setting. *VUB*, 2020.

[81] Yue Yu, Shahriar Talebi, Henk J van Waarde, Ufuk Topcu, Mehran Mesbahi, and Behçet Açıkmeşe. On controllability and persistency of excitation in data-driven control: Extensions of willems' fundamental lemma. In *Proc. IEEE CDC*, pages 6485–6490. IEEE, 2021.

[82] Henk J Van Waarde, Jaap Eising, Harry L Trentelman, and M Kanat Camlibel. Data informativity: a new perspective on data-driven analysis and control. *IEEE Trans. Autom. Control*, 65(11):4753–4768, 2020.

[83] Henk J van Waarde, Claudio De Persis, M Kanat Camlibel, and Pietro Tesi. Willems' fundamental lemma for state-space systems and its extension to multiple datasets. *IEEE Control Syst. Let.*, 4(3):602–607, 2020.

[84] Henk J van Waarde. Beyond persistent excitation: Online experiment design for data-driven modeling and control. *IEEE Control Syst. Let.*, 6:319–324, 2021.

[85] Jeremy Coulson, Henk van Waarde, and Florian Dörfler. Robust fundamental lemma for data-driven control. *arXiv preprint arXiv:2205.06636*, 2022.

[86] Mina Ferizbegovic, Håkan Hjalmarsson, Per Mattsson, and Thomas B Schön. Willems' fundamental lemma based on second-order moments. In *Proc. IEEE CDC*, pages 396–401. IEEE, 2021.

[87] Ramesh Kadali, Biao Huang, and Anthony Rossiter. A data driven subspace approach to predictive controller design. *Control Eng. Pract.*, 11(3):261–278, 2003.

[88] Jan-Willem van Wingerden, Sebastiaan P Mulders, Rogier Dinkla, Tom Oomen, and Michel Verhaegen. Data-enabled predictive control with instrumental variables: the direct equivalence with subspace predictive control. In *Proc. IEEE CDC*, pages 2111–2116. IEEE, 2022.

[89] Vincent Verdult, Marco Lovera, and Michel Verhaegen. Identification of linear parameter-varying state-space models with application to helicopter rotor dynamics. *Int. J. Control*, 77(13):1149–1159, 2004.

[90] Matthew S Allen, Michael W Sracic, Shashank Chauhan, and Morten Hartvig Hansen. Output-only modal analysis of linear time-periodic systems with application to wind turbine simulation data. *MSSP*, 25(4):1174–1191, 2011.

[91] Sergio Bittanti and Patrizio Colaneri. Invariant representations of discrete-time periodic systems. *Automatica*, 36(12):1777–1793, 2000.

[92] Lamia Ben Jemaa and Edward Joseph Davison. Performance limitations in the robust servomechanism problem for discrete time periodic systems. *Automatica*, 39(6):1053–1059, 2003.

[93] Achim Ilchmann and Volker Mehrmann. A behavioral approach to time-varying linear systems. part 1: General theory. *SIAM J Ctrl Optm*, 44(5):1725–1747, 2005.

[94] Vladimír Kučera. The discrete Riccati equation of optimal control. *Kybernetika*, 8(5):430–447, 1972.

[95] Bart PG Van Parys, Daniel Kuhn, Paul J Goulart, and Manfred Morari. Distributionally robust control of constrained stochastic systems. *IEEE Trans. Autom. Control*, 61(2):430–442, 2015.

[96] Steve Zymler, Daniel Kuhn, and Berç Rustem. Distributionally robust joint chance constraints with second-order moment information. *Math. Program.*, 137:167–198, 2013.

[97] Mark Cannon, Qifeng Cheng, Basil Kouvaritakis, and Saša V Raković. Stochastic tube MPC with state estimation. *Automatica*, 48(3):536–541, 2012.

[98] Marcello Farina, Luca Giulioni, Lalo Magni, and Riccardo Scattolini. An approach to output-feedback MPC of stochastic linear discrete-time systems. *Automatica*, 55:140–149, 2015.

[99] Eunhyek Joa, Monimoy Bujarbaruah, and Francesco Borrelli. Output feedback stochastic MPC with hard input constraints. In *Proc. ACC*, pages 2034–2039. IEEE, 2023.

[100] Jack Ridderhof, Kazuhide Okamoto, and Panagiotis Tsiotras. Chance constrained covariance control for linear stochastic systems with output feedback. In *Proc. IEEE CDC*, pages 1758–1763. IEEE, 2020.

[101] Peter Hokayem, Eugenio Cinquemani, Debasish Chatterjee, Federico Ramponi, and John Lygeros. Stochastic receding horizon control with output feedback and bounded controls. *Automatica*, 48(1):77–88, 2012.

[102] Johannes Köhler and Melanie N Zeilinger. Recursively feasible stochastic predictive control using an interpolating initial state constraint. *IEEE Control Syst. Let.*, 6:2743–2748, 2022.

[103] Henning Schlüter and Frank Allgöwer. Stochastic model predictive control using initial state optimization. *IFAC-PapersOnLine*, 55(30):454–459, 2022.

[104] Masahiro Ono and Brian C Williams. Iterative risk allocation: A new approach to robust model predictive control with a joint chance constraint. In *Proc. IEEE CDC*, pages 3427–3432. IEEE, 2008.

[105] Paul J Goulart, Eric C Kerrigan, and Jan M Maciejowski. Optimization over state feedback policies for robust control with constraints. *Automatica*, 42(4):523–533, 2006.

[106] Paul J Goulart and Eric C Kerrigan. Output feedback receding horizon control of constrained systems. *Int J Control*, 80(1):8–20, 2007.

[107] Laurent El Ghaoui, Maksim Oks, and Francois Oustry. Worst-case value-at-risk and robust portfolio optimization: A conic programming approach. *Oper. Res.*, 51(4):543–556, 2003.

[108] Daniele Alpago, Florian Dörfler, and John Lygeros. An extended Kalman filter for data-enabled predictive control. *IEEE Control Syst. Let.*, 4(4):994–999, 2020.

[109] Peter Van Overschee and Bart De Moor. N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30(1):75–93, 1994.

[110] Hong Ye. Scheduling of networked control systems. *IEEE Control Syst.*, 21(1):57–65, 2001.

[111] Thomas Nall Eden Greville. Note on the generalized inverse of a matrix product. *Siam Review*, 8(4):518–521, 1966.

[112] Jan C Willems. Open stochastic systems. *IEEE Trans. Autom. Control*, 58(2):406–421, 2012.

[113] Alberto Padoan, Jeremy Coulson, Henk J Van Waarde, John Lygeros, and Florian Dörfler. Behavioral uncertainty quantification for data-driven control. In *Proc. IEEE CDC*, pages 4726–4731. IEEE, 2022.

[114] Timm Faulwasser, Ruchuan Ou, Guanru Pan, Philipp Schmitz, and Karl Worthmann. Behavioral theory for stochastic systems? a data-driven journey from willems to wiener and back again. *Annu Rev Control*, 55:92–117, 2023.