Thèse n° 9798

Data-driven Methods for Control: from Linear to Lifting

Présentée le 23 juin 2023

Faculté des sciences et techniques de l'ingénieur
Laboratoire d'automatique 3
Programme doctoral en robotique, contrôle et systèmes intelligents

pour l'obtention du grade de Docteur ès Sciences

par

Yingzhao LIAN

Acceptée sur proposition du jury

Prof. G. Ferrari Trecate, président du jury
Prof. C. N. Jones, directeur de thèse
Prof. C. De Persis, rapporteur
Prof. T. Keviczky, rapporteur
Prof. J. Lygeros, rapporteur

École
polytechnique
fédérale
de Lausanne

2023

Still water runs deep

To my parents and my wife Jiayi. . .

Acknowledgements

With utmost sincerity and a deep sense of gratitude, I must express my heartfelt thanks to my esteemed advisor, Prof. Colin Jones, for giving me the opportunity to pursue my PhD at Laboratoire d'Automatique. His unwavering support and encouragement have been the guiding force behind my research journey. The level of freedom he has granted me has allowed me to explore and develop my own unique rhythm. Meanwhile, his constructive thinking and benevolent support, particularly in tackling abstract mathematical concepts, have been nothing short of exceptional. He has always been active in our meetings, sharing his innovative ideas and challenging me to dig deeper into each topic. His guidance and mentorship have been instrumental in shaping my research career.

I cannot express how much I appreciate the members of our lab. Working with them on diverse topics has been a source of inspiration for me. Unfortunately, almost half of my Ph.D. period was impacted by the pandemic, but during these tough times, our virtual and occasional in-person events gave me warmth and comfort. I want to express my gratitude to Dr. Harsh Ambarishkumar Shukla, who has always been a great friend and mentor to me. We met when he was my teaching assistant at EPFL during my master's period. He has since guided my research interests and shown me the practical value of control science. I am also grateful to Emilio T. Maddalena, who joined the group around the same time as me. Having someone in the lab working on a similar topic was beneficial, as we could discuss and brainstorm together. These two colleagues have left a profound impact on me, and I will never forget the time we spent together and the crazy things we did. I also want to express my appreciation for Jicheng Shi, whom I supervised during his student period and later on as a fellow Ph.D. student in our lab. We spent countless hours making things work, and I am grateful for the time we spent together. Last but not least, I am indebted to Dr. Christophe Salzmann, who bridged the gap between me and the building operations team at EPFL. Without his support, this thesis would have been incomplete.

The pandemic time was incredibly challenging, but I am grateful to have met Prof. Joel Rosenfeld, Prof. Ben Russo, and Prof. Rushikesh Kamalapurkar in a virtual session at ACC 2020. They kindly invited me to a discussion about the RKHS and their research. Despite not being their student, they were open to sharing their insights about the RKHS, and we even had regular Zoom calls during that term. Around the same time, I virtually met Dr. Jia Guo, who offered me insightful instructions about the RKHS research from a different mathematical viewpoint. These two groups of people showed me why the control community thrives, proving that the passion for exploring new knowledge will never be stopped by the pandemic.

After the pandemic period, my Ph.D. life returned to normal mode, and I had the opportunity to visit Imperial College London. I want to express my gratitude to Prof. E.C. Kerrigan for hosting my stay, and the students in the 11th floor of the EEEC building who welcomed me with open arms. Eduardo, Lorenzo, Ali, Lucien, and all of you, thank you for showing me around and making my stay even more enjoyable. It was during this visit that my mood, which had been dampened by the pandemic, was finally lifted. I am especially thankful to Yuwen Chen for our great idea on SDP and the three-day discussion we had over ECC 2022, which showed me the beauty of conic programming.

Lastly, I am grateful to my family for their unconditional support and care. My parents' upbringing instilled in me a strong work attitude and patience to face. Jiayi Lan, my lovely wife, my Ph.D life would lose half of its colour without your accompany and support.

Lausanne, May 9, 2023 YZ. Lian

Abstract

The progress towards intelligent systems and digitalization relies heavily on the use of automation technology. However, the growing diversity of control objects presents significant challenges for traditional control approaches, as they are highly dependent on expert knowledge and require substantial commissioning effort. In response to this challenge, data-driven methods have emerged as a promising alternative that reduces human involvement by incorporating knowledge extracted from data. This thesis follows a conventional control research path and investigates the application of data-driven methods to linear time-invariant dynamics and nonlinear dynamics.

The first part of the thesis focuses on predictive control based on Willems' fundamental lemma. A tractable robust formulation based on the data-enabled predictive control (DeePC) framework is introduced, followed by a bi-level approach that aims to improve robustness and adaptivity. The focus then shifts to nonlinear dynamics, where reproducing kernel Hilbert space (RKHS) and Koopman operator-based heuristics are utilized to extend the applicability of Willems' fundamental lemma.

The second part of the thesis concentrates on stability analysis, which is a fundamental aspect of control science. Stability analysis must be robust enough to account for the infinitely many possible realizations of underlying dynamics based on a fixed finite set of data. To this end, a robust stability guarantee for a piece-wise affine (PWA) Lyapunov function is provided, which is a generalization of the classical Lyapunov-Massera local asymptotic stability theorem. Additionally, a convex second-order cone program (SOCP) is proposed to learn a robust PWA Lyapunov function assuming the underlying dynamics are Lipschitz. This approach provides a new means of designing stable control systems without requiring significant human intervention.

The last part of this thesis presents additional research on self-triggered control and real-time optimization algorithm design. These studies complement the primary investigation and provide a complete exposition of the research carried out during the Ph.D. program.

Résumé

Les progrès vers des systèmes intelligents et la numérisation reposent largement sur l'utilisation de la technologie d'automatisation. Cependant, la diversité croissante des objets de contrôle présente des défis importants pour les approches de contrôle traditionnelles, car elles dépendent fortement des connaissances d'experts et nécessitent des efforts considérables de mise en service. En réponse à ce défi, des méthodes axées sur les données ont émergé comme une alternative prometteuse qui réduit l'implication humaine en incorporant des connaissances extraites des données. Cette thèse suit un chemin de recherche conventionnel en matière de contrôle et examine l'application de méthodes axées sur les données aux dynamiques linéaires invariantes dans le temps et aux dynamiques non linéaires.

La première partie de la thèse se concentre sur le contrôle prédictif basé sur le lemme fondamental de Willems. Une formulation robuste traitable basée sur le cadre de contrôle prédictif activé par les données (DeePC) est introduite, suivie d'une approche bi-niveau qui vise à améliorer la robustesse et l'adaptabilité. L'accent est ensuite mis sur les dynamiques non linéaires, où des heuristiques basées sur l'espace de Hilbert à noyau reproduisant (RKHS) et l'opérateur de Koopman sont utilisées pour étendre l'applicabilité du lemme fondamental de Willems.

La deuxième partie de la thèse se concentre sur l'analyse de stabilité, qui est un aspect fondamental de la science du contrôle. L'analyse de stabilité doit être suffisamment robuste pour tenir compte des nombreuses réalisations possibles des dynamiques sous-jacentes basées sur un ensemble fini de données fixes. À cette fin, une garantie de stabilité robuste pour une fonction de Lyapunov piece-wise affine (PWA) est fournie, qui est une généralisation du théorème de stabilité asymptotique locale de Lyapunov-Massera classique. De plus, un programme convexe de cône de second ordre (SOCP) est proposé pour apprendre une fonction de Lyapunov PWA robuste en supposant que les dynamiques sous-jacentes sont Lipschitz. Cette approche fournit un nouveau moyen de concevoir des systèmes de contrôle stables sans nécessiter une intervention humaine significative.

La dernière partie de cette thèse présente des recherches supplémentaires sur le contrôle déclenché par soi-même et la conception d'algorithmes d'optimisation en temps réel. Ces études complètent l'enquête principale et fournissent une exposition complète de la recherche menée pendant le programme de doctorat.

Contents

Introduction

Controller design and stability analysis are two primary topics in control engineering. Classic control techniques heavily rely on expert knowledge, making it challenging to handle the increasing diversity and complexity of control objects. To overcome this challenge, a collection of control techniques called *data-driven methods* has emerged. These techniques aim to automate the workflow in control engineering, including controller design and stability analysis, through the use of data or high fidelity simulators. While simulators have proven successful in reinforcement learning [1] and digital twin [2] applications, the implementation of a high fidelity simulator still requires significant effort. Therefore, this thesis focuses on control techniques that involve only data.

The first part of this thesis focuses on controller design, where the characterization of system dynamics based on measured data serves as the main enabler of this procedure. Instead of running through a parametric modelling procedure [3–6], non-parametric methods distinguish themselves by directly representing system dynamics with data [7–10]. In particular, there is a recent spark of interest in behavioural theory [11] where system dynamics are characterized by trajectories. This viewpoint concludes a simple and closed representation for linear time invariant systems [12], coined Willems' fundamental lemma, and has been successfully applied to predictive control [13], named data-enabled predictive control (DeePC). The first half of the first part study the refinement of this framework, with a specific focus on building applications. Particularly, building controllers manipulate the heating, ventilation, and air conditioning (HVAC) units to control in the indoor climate. The main investigation in Chapter 1 is the robustness aspect, where the indoor comfort should be guaranteed robustly, even when disturbed by outdoor climates and occupants' behaviour. As a result, this chapter will provide the readers a systematic data-driven approach to handle unknown measurement noise and uncertain but predictable process noise in linear systems.

Several works have attempted to map a nonlinear control problem into a linear one, motivated by the success of linear control theory. Chapter 3 follows this track via the idea of lifting, where nonlinear mappings are used to warp a nonlinear system to a linear one. Wiener system and Hammerstein systems are standing examples applying this idea, and recent work also applies this idea to the Willems' fundamental lemma [14–17]. Without a direct interaction with nonlinear dynamics in the state space, the techniques studied in Chapter 3 models the behaviour of our observation. This idea originates from the Koopman operator theory, where the Koopman operator is a linear composite operator [18] acting on an autonomous system. Even though this operator and its generator is only well-defined in forward-complete systems [19], it still serves as a successful heuristic motivating applications in system identification and controller design [20]. The first half of Chapter 3 is devoted to the parametric finite approximation of a Koopman operator. Meanwhile, its heuristic extension to a system with input motivates a nonlinear extension of the Willems' fundamental lemma. Different from the Koopman operator based approach, another framework that models the function operation (i.e. functional) is investigated in the second half of Chapter 3, where an infinite dimensional extension of the Willems' fundamental lemma is studied. The results in this chapter provides a systematic viewpoint of how lifting can be used to simplify the analysis of nonlinear system, albeit of its possibly infinite dimension.

The second part of this thesis shifts the focus to stability analysis, which is another major research topic in control science. Among various stability criteria, Lyapunov analysis [21] plays a key role in this field. In this framework, stability analysis is reformulated into the search for a Lyapunov function. This has been widely studied in the model-based or sampling-based setup, where the user is assumed to have direct access to the model or its high-fidelity simulator. Note that the system model or its simulator may not be available all the time, it is usually possible and much easier to measure a finite number of system responses offline. Therefore, stability analysis based on a fixed measured data set poses a new challenge and becomes desirable. Motivated by this need, we will study the Lyapunov analysis based on a given finite set of measurements of system response in Chapter 3. The proposed approach can learn a piece-wise affine (PWA) Lyapunov function on a compact set without access to the system model/simulator. The learning scheme only resorts to the solution of a convex second order cone programming (SOCP) even when the underlying dynamics is nonlinear. To arrive at this major contribution, fundamental contributions in stability analysis are made in this chapter. Firstly, we generalized the Lyapunov stability analysis of a set around an equilibrium point to a more general forward invariant set, which follows a robust stability condition that guarantee stability with respect to an uncertain set of functions.

After addressing the aforementioned aspects, this thesis aims to demonstrate the power of learning-based methods. By directly using data, rather than going through an explicit modelling or system identification process, one can generate desired controllers or synthesize solutions end-to-end. Despite covering technically distinct topics, the key message of this thesis is:

Robustness/uncertainty quantification is the key enabler of learning based methods.

This message follows an implicit story line throughout the thesis and encompasses at least two key aspects. Firstly, we are only able to acquire limited knowledge from finitely many data, and this in the end limits our capability of predicting the systems' behaviours. This holds true even when the system is deterministic, as there may exist multiple models that are consistent with our measurements. Thus, it is important to mathematically quantify and systematically consider such uncertainty in the design of controllers or stability analyses. In the field of Bayesian modelling, this kind of uncertainty is referred to as *aleatoric* uncertainty [22], and its consideration is typically reflected in Section 2.1 and Chapter 3.

Secondly, uncertainty also arises from the interaction between the dynamical system, the environment, and the users. This includes the measurement noise that contaminates the sensor read-out, the disturbance that perturb the system responses, e.t.c. In Bayesian modelling, this type of uncertainty is referred to as *epistemic* uncertainty. Such inherent uncertainty limits our ability to fully understand the underlying system and may result in mismatches between expected and actual system responses. Chapter 2 sets the tone for the consideration of this aspect, where output predictions that are used for control decision respects such uncertainty. Beyond the use of in control engineering, aleatoric and epistemic uncertainty play critical roles in computer vision [23] and the training of deep neural networks [24].

About the Appendix

In the appendix, we present an alternative parametrization of the feedback control law based on the bi-level approach discussed in Section 1.3. This additional result offers a deeper theoretical understanding of how the null space and range of the Hankel matrices play a role in causality and feedback. While this contribution is informative and enhances the comprehensiveness of the thesis, the resulting algorithm is less stable compared to the one utilized in Section 1.3. Thus, it is not advisable for practitioners to rely on the algorithm presented in the appendix.

About Other Contributions

As you will notice later in my list of publications, my Ph.D is composed of multiple research topics: learning, optimization algorithm and self-triggered control. The first two parts of this thesis structure a self-contained story, where selected results related to the data-driven methods are included. In order to briefly cover the major results in another two tracks, an independent and self-contained chapter is assigned to each topic respectively in Part III. For the sake of compactness, each chapter presents one selected result, while other results are summarized at the end of the corresponding chapter. In particular, the results from the following two papers are respectively presented in the chapters in Part III:

- **Lian, Y.**, Jiang, Y., Stricker, N., Thiele, L. and Jones, C.N., 2021. *Resource-aware stochastic self-triggered model predictive control. IEEE Control Systems Letters, 6, pp.1262-1267.*
- **Lian, Y.**, Jiang, Y., Opila, D.F. and Jones, C.N., 2023.A Proximal-Point Lagrangian Based Parallelizable Nonconvex Solver for Bilinear Model Predictive Control. *Conditionally accepted by Transactions on Control Systems Technology*

Notation

$\mathbb{S}^+$, $\mathbb{S}^{++}$	Positive definite/semi-definite cone
$\mathcal{N}(\mu,\Sigma)$	Gaussian distribution with mean μ and covariance Σ
$\times$	Set product operation: $A\times B := \{(x,y)\mid x\in A,\ y\in B\}$
$\oplus$	Minkowski sum: $A\oplus B := \{x\mid \exists\, x_1\in A,\ x_2\in B,\ x=x_1+x_2\}$
$\ominus$	Pontryagin difference: $A\ominus B := \{x\mid x+y\in A,\ \forall\ y\in B\}$
$+$	when A, B are sets, + denotes direct sum
$\mathbf{O}$, $\mathbf{0}$	are zero matrix/vector
I_n	identity matrix in $\mathbb{R}^{n\times n}$
$\lVert\cdot\rVert$	Euclidean norm
$\{x_i\}_{i=1}^T$	a set/sequence of size T indexed by i
x_i	the measurement of x at time i
$x_{1:L}$	$x_{1:L} := [x_1^\top, x_2^\top \dots x_L^\top]^\top$ denotes a concatenated sequence of x_i ranging from x_1 to x_L, the index is dropped to improve clarity if the intention is clear from the context
$\langle\cdot,\cdot\rangle$	inner product
$\text{span}(A)$	column space(i.e. range) of matrix A
$T_x\mathcal{M}$	tangent space of manifold $\mathcal{M}$ evaluated at x
id	identity map
$\lVert\cdot\rVert_p$	p-norm
$\text{diag}(c)$	diagonal matrix whose diagonal entries stacks to vector c
E_x	evaluation functional at x
C^∞	smooth function
A^*	when $A: H_x \to H_y$ is a linear operator, A^* denotes its adjoint
$X\setminus Y$	$X\setminus Y := \{x\in X \mid x\notin Y\}$ for all $Y\subset X$
$\lvert C\rvert$	cardinality of set C
$\text{int}(\mathcal{X})$	interior of topological space $\mathcal{X}$
$\overline{\mathcal{X}}$	closure of topological space $\mathcal{X}$
$\partial\mathcal{X}$	boundary of topological space $\mathcal{X}$
$\mathbb{Z}_a^b$	the set of integers $\{a, a+1\dots, b\}$
$\mathbb{P}(x)$	probability of stochastic event x
$\text{vec}(A)$	stacking all columns of A into one long vector
mat	reverse operation of $\text{vec}(\cdot)$, $\text{mat}(\text{vec}(A)) = A$

Table 1: Notation

Collaborations

The bi-level DeePC result in Section 1.3 is done in collaboration with Jicheng Shi. Our joint effort in relaunching the experiment platform Polydome provided the foundation for this collaboration. After that, Jicheng carries out most experiments to prove the efficacy of my bi-level framework during his stay as research assistant in our lab.

Renzi Wang's contribution to the implementation of the semi-parametric algorithm in Section 2.1.3 was indispensable. Renzi completed her master project with me, and her technical skills and dedication were critical to the success of the project.

I am particularly proud of the results in Chapter 3 due to their elegant mathematical foundation. Matteo Tacchi is the one who helps me to arrive at such mathematical elegance. His insight into stability theory helped to sort out the logic behind the proof and ensure its rigour by covering all the corner cases.

Although the results are not included in this thesis, the collaboration with Ugo Rosolia in the proof of the convergence of the iterative learning MPC scheme was fruitful. I am also grateful for the help and guidance of Prof. Daniel Opila during his visit to our lab, particularly in conducting the motor experiment with our proximal-point Lagrangian MPC solver. His expertise in power electronics and electric machines provided valuable insight.

Regarding the projects in self-triggered control, the collaboration with Naomi Stricker, Lothar Thiele, Daniel Opila, and Stefan Wildhagen on self-triggered control led to fruitful results. Their insight helped to frame the mathematical problem appropriately.

I would like to express my gratitude to Jia Guo, Prof. Joel Rosenfeld, Prof.Ben Russo and Prof. Rushikesh Kamalapurkar, who are great experts in RKHS. Although our discussions and collaborations on occupation kernel and partially persistent excitation condition have not yet resulted in a publishable result, our time spent together has brought me closer to the core of research in RKHS.

Finally, the collaboration with Lorenzo and Eduardo during my visit to Imperial College leave me a great memory. Their work on numerical integration with adaptive grids and global optimization with set-membership method pushed me out of my comfort zone, and I am looking forward to seeing our results published one day.

Publications

Articles related to the first part:

- **Lian, Y.** and Jones, C.N., 2021, December. From system level synthesis to robust closed-loop data-enabled predictive control. *In 2021 60th IEEE Conference on Decision and Control (CDC) (pp. 1478-1483). IEEE.*

- **Lian, Y.**, Shi, J., Koch, M. and Jones, C.N., 2021. Adaptive robust data-driven building control via bi-level reformulation: an experimental result. *To appear in Transactions on Control Systems Technology*

- **Lian, Y**. and Jones, C.N., 2020. On Gaussian process based koopman operators.*IFAC-PapersOnLine, 53(2), pp.449-455.*

- **Lian, Y.** and Jones, C.N., 2019, December. Learning feature maps of the koopman operator: A subspace viewpoint. *In 2019 IEEE 58th Conference on Decision and Control (CDC) (pp. 860-866). IEEE.*

- **Lian, Y.** and Jones, C.N., 2021, May. Nonlinear data-enabled prediction and control. *In Learning for Dynamics and Control (pp. 523-534). PMLR.*

- Maddalena, E.T., **Lian, Y**. and Jones, C.N., 2020. Data-driven methods for building control—A review and promising future directions. Control Engineering Practice, 95, p.104211.

- Di Natale, L., **Lian, Y.**, Maddalena, E.T., Shi, J. and Jones, C.N., 2022. Lessons Learned from Data-Driven Building Control Experiments: Contrasting Gaussian Process-based MPC, Bilevel DeePC, and Deep Reinforcement Learning. *In 2022 61st IEEE Conference on Decision and Control (CDC) (pp. 1478-1483). IEEE.*

- Shi, J., **Lian, Y**. and Jones, C.N., 2022. Data-driven input reconstruction and experimental validation. *In IEEE Control Systems Letters, vol. 6, pp. 3259-3264, 2022*

Articles related to the second part:

- **Lian, Y.**, Tacchi, M., and Jones, C.N., 2023. Robust Data-driven Lyapunov Analysis with Finite Data. *To be submit to Transactions on Automatic Control.*

Articles related to self-triggered control:

- Stricker, N., **Lian, Y**., Jiang, Y., Jones, C.N. and Thiele, L., 2021, November. Joint energy management for distributed energy harvesting systems. *In Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems (pp. 575-577).*

- **Lian, Y.**, Jiang, Y., Stricker, N., Thiele, L. and Jones, C.N., 2021. *Resource-aware stochastic self-triggered model predictive control. IEEE Control Systems Letters, 6, pp.1262-1267.*

- **Lian, Y.**, Jiang, Y., Stricker, N., Thiele, L. and Jones, C.N., 2021. Robust Resource-Aware Self-Triggered Model Predictive Control. *IEEE Control Systems Letters, 6, pp.1724-1729.*

- **Lian, Y.**, Wildhagen, S., Jiang, Y., Houska, B., Allgöwer, F. and Jones, C.N., 2020, December. Resource-aware asynchronous multi-agent coordination via self-triggered MPC. *In 2020 59th IEEE Conference on Decision and Control (CDC) (pp. 685-690). IEEE.*

- **Lian, Y.**, Jiang, Y., Jones, C.N. and Opila, D.F., 2022. Scheduling Delays and Curtailment for Household Appliances With Deterministic Load Profiles Using MPC. *IEEE Control Systems Letters, 6, pp.3301-3306.*
- Stricker, N., **Lian, Y**., Jiang, Y., Jones, C.N. and Thiele, L., 2021, November. Self-triggered Control with Energy Harvesting Sensor Nodes. *To appear in Transactions on Cyber-Physical Systems. IEEE.*

Articles related to iterative Learning MPC:

- Rosolia, U., **Lian, Y.**, Maddalena, E.T., Ferrari-Trecate, G. and Jones, C.N., 2022. On the Optimality and Convergence Properties of the Iterative Learning Model Predictive Controller. *IEEE Transactions on Automatic Control, 68(1), pp.556-563.*
- Shi, J., **Lian, Y**. and Jones, C.N., 2021, June. Robust Learning Model Predictive Control for Periodically Correlated Building Control. *In 2021 European Control Conference (ECC) (pp. 192-197). IEEE.*

Articles related to optimization solver:

- **Lian, Y.**, Jiang, Y., Opila, D.F. and Jones, C.N., 2023.A Proximal-Point Lagrangian Based Parallelizable Nonconvex Solver for Bilinear Model Predictive Control. *To appear in Transactions on Control Systems Technology*

Part I

Data-driven Predictive Control

1 Linear Predictive Control

Although linear control theory is well-developed, the Willems' fundamental lemma [12] and the system level synthesis (SLS) [25] still spark significant research interest. While the SLS, as a model-based framework, has proven its efficacy in large-scale distributed control, the Willems' fundamental lemma plays a key role in learning-based methods. Following the main topic of this thesis, the Willems' fundamental lemma is the central piece of this chapter. In this part, we will develop a robust data-driven controller. Even though we are able to target at more general applications, this chapter still specifically focus on building applications, in order to have a more intuitive interpretation of the engineering details. Hence, the underlying dynamics are assumed to be linear, and they are disturbed by measurable process noise. Beyond building control, systems with measurable disturbances are ubiquitous, especially in energy-related applications: solar radiation in photovoltaic power systems, electricity demand in power grids, and power generation in airborne wind energy systems, to name a few.

Previous Work

Data can be used to model building dynamics [26] or to directly generate/improve control policies. Due to seasonal variations [27] and component wear, building dynamics are usually slowly time-varying, and adaptive model predictive control (MPC) has been introduced to combine online parameter estimation and control in [28]. For example, the experiment in [29] adaptively estimates the model of an evaporator in the HVAC system, which is then used to control the valve set-points. [30] runs an extended Kalman filter to achieve online parameter adaptation before the estimated model is used in an MPC controller. However, these parameter-estimation-based adaptive methods usually require a-priori knowledge about the structure of the building dynamics and/or the HVAC systems.

Beyond running through a modelling/estimation procedure, data can be used to refine a control policy. The main approaches in this direction include reinforcement learning (RL) [1] and iterative learning control (ILC) [31]. In particular, ILC has been used for buildings with fixed heating/occupant schedules [32, 33] and RL for learning a building control policy that is not necessarily iterative [34]. To run these learning schemes, a high-fidelity building simulator is usually required, and therefore publications on successful experiments with HVAC systems are rare, with a few exceptions being [35–39].

Beyond RL and ILC, data can also be used to directly characterize the system's responses from a behavioural theoretic viewpoint [11]. Willems' fundamental lemma is such a tool that provides a characterization of linear time invariant (LTI) systems from measured input and output trajectories. Such a characterization offers a convenient interface to data-driven controller design [13, 40–42]. Motivated by the simplicity and effectiveness of the fundamental lemma, its extension to a more general setting is attracting broad attention, including nonlinear extensions [14, 16, 17, 43], data informativity [44], descriptor system [45], e.t.c [46].

Even for LTI systems, the absence of output measurement noise in the standard Willems' fundamental lemma limits its practicality. To accommodate this issue, a wide range of researchers are studying this challenge. In [47–49], classic robust control design tools such as linear fractional transforms have been proposed to design robust linear feedback controllers. Parallel to the studies in linear feedback control laws, robustness in predictive control schemes has also been studied [50–52], where regularization is the main tool applied to deal with measurement noise. [50] shows that the regularization is related to the distributional robustness of the system uncertainty (including measurement noise). [51] studies a different viewpoint, where the regularization is linked to the loss function used in the system identification. When convex relaxation is applied, the loss function in the system identification procedure results in a regularization term [51, 53].

Building on this regularization viewpoint, data-driven controllers based on the Willems' fundamental lemma have been successfully deployed on different real-world systems, such as a quadrotor [54], a four-tank system [55], etc. However, tuning the regularization weight still poses a non-trivial challenge, and an exhaustive search based on simulation is commonly used [53].

Outline

The last result of this chapter is to present a robust adaptive data-driven predictive controller. To achieve this goal, we first introduce the single-level framework, also known as data-enabled predictive control (DeePC), in order to provide a comprehensive understanding of this data-driven controller. This will serve as a basis for the subsequent bi-level framework. After recapitulating the background knowledge in Section 1.1, we state the problem setup. Initially, we consider LTI systems with measurement-noise-free I/O data in Section 1.2, where we introduce a basic framework of DeePC and develop its corresponding robust tractable form. We then extend this formulation to the bi-level framework for LTI systems and present its heuristic extension to linear-time-varying (LTV) systems in Section 1.3. To validate the efficacy of both the single-level and bi-level frameworks, we use numerical examples to demonstrate their effectiveness in building models. Additionally, at the end of Section 1.3, we conduct a real-world experiment based on the bi-level framework.

1.1 Preliminary

1.1.1 Willems' Fundamental Lemma and DeePC

Definition 1. *A Hankel matrix of depth L associated with a vector-valued signal sequence $s := \{s_i\}_{i=1}^{T}$, $s_i \in \mathbb{R}^{n_s}$ is*

$$\mathfrak{H}_L(s) := \begin{bmatrix} s_1 & s_2 & \dots & s_{T-L+1} \\ s_2 & s_3 & \dots & s_{T-L+2} \\ \vdots & \vdots & & \vdots \\ s_L & s_{L+1} & \dots & s_T \end{bmatrix} .$$

A deterministic linear time-invariant (LTI) system, dubbed $\mathfrak{B}(A,B,C,D)$, is defined as

$$x_{i+1} = Ax_i + Bu_i \;,\; y_i = Cx_i + Du_i \;, \tag{1.1}$$

whose order is n_x. n_u, n_y denote its input and output dimensions respectively. An L-step trajectory generated by this system is

$$\begin{bmatrix} u_{1:L} & y_{1:L} \end{bmatrix} := \begin{bmatrix} u_1^\top & \dots & u_L^\top & y_1^\top & \dots & y_L^\top \end{bmatrix}^\top .$$

The set of all possible L-step trajectories generated by $\mathfrak{B}(A,B,C,D)$ is denoted by $\mathfrak{B}_L(A,B,C,D)$.

For the sake of consistency, a datapoint coming from the historical dataset is marked by boldface subscript $_\mathbf{d}$. Given a sequence of input-output measurements $\{u_{\mathbf{d},i}, y_{\mathbf{d},i}\}_{i=1}^{T}$, we call the input sequence persistently exciting of order L if $\mathfrak{H}_L(u_\mathbf{d})$ is full row rank. By building the following n_c-column stacked Hankel matrix

$$\mathfrak{H}_L(u_\mathbf{d}, y_\mathbf{d}) := \begin{bmatrix} \mathfrak{H}_L(u_\mathbf{d}) \\ \mathfrak{H}_L(y_\mathbf{d}) \end{bmatrix} , \tag{1.2}$$

we state **Willems' Fundamental Lemma** as

Lemma 1. *[12, Theorem 1] Consider a controllable linear system and assume $\{u_\mathbf{d}\}_{i=1}^{T}$ is persistently exciting of order $L + n_x$. The condition colspan($\mathfrak{H}_L(u_\mathbf{d}, y_\mathbf{d})$) $= \mathfrak{B}_L(A,B,C,D)$ holds.*

For the sake of consistency, L is reserved for the length of the system responses and n_c denotes the number of columns in a Hankel matrix.

A data-driven control scheme has been proposed in [13, 40], called data-enabled predictive control (DeePC), where Lemma 1 generates a trajectory prediction. To distinguish the difference between this framework and the one investigated in Section 1.3, we call the DeePC and its regularized variants single-level approaches. In particular, for an LTI system without process noise and measurement noise, the single-level approach solves the following optimal control problem in a receding horizon scheme:

$$\min_{\substack{y_{pred},u_{pred}\\ g,\sigma}} \quad J(y_{pred}, u_{pred}) \tag{1.3a}$$

$$\text{s.t.} \begin{bmatrix} \mathfrak{H}_{L,init}(y_{\mathbf{d}}) \\ \mathfrak{H}_{L,init}(u_{\mathbf{d}}) \\ \mathfrak{H}_{L,pred}(u_{\mathbf{d}}) \\ \mathfrak{H}_{L,pred}(y_{\mathbf{d}}) \end{bmatrix} g = \begin{bmatrix} y_{init} \\ u_{init} \\ u_{pred} \\ y_{pred} \end{bmatrix} \tag{1.3b}$$

$$u_{pred} \in \mathcal{U} \ , \ y_{pred} \in \mathcal{Y} \ ,$$

where $J(\cdot,\cdot)$ is a convex objective function, $\mathcal{U}$ and $\mathcal{Y}$ denote compact convex constraint sets on the input and the output. u_{init}, y_{init} are t_{init}-step sequences of the measured inputs and outputs preceding the current point in time. Accordingly, u_{pred}, y_{pred} are the corresponding n_h-step predictive sequences viewed from the current time step. The matrix $\mathfrak{H}_L(y_{\mathbf{d}})$ is split into two sub-Hankel matrices:

$$\mathfrak{H}_L(y_{\mathbf{d}}) = \begin{bmatrix} \mathfrak{H}_{L,init}(y_{\mathbf{d}}) \\ \mathfrak{H}_{L,pred}(y_{\mathbf{d}}) \end{bmatrix} .$$

The matrix $\mathfrak{H}_{L,init}(y_{\mathbf{d}})$ is of depth t_{init} and the depth of $\mathfrak{H}_{L,pred}(y_{\mathbf{d}})$ is the prediction horizon n_h such that $t_{init} + n_h = L$. The matrices $\mathfrak{H}_{L,init}(u_{\mathbf{d}})$, $\mathfrak{H}_{L,pred}(u_{\mathbf{d}})$ are defined similarly. The choice of t_{init} is made to ensure a unique estimation of the initial state; please refer to [41] for more details. Intuitively speaking, for a noise-free and disturbance-free LTI system, the prediction made by (1.3b) is exact due to the Willems' fundamental lemma 1. With this capability of generating exact prediction, we can enforce the constraints $\mathcal{U}$, $\mathcal{Y}$ and optimize the predictive performance for finite steps. The beauty of this framework comes from its trajecotry predictor (1.3b). This predictor is uniquely characterized by historical I/O data, whose prediction is implicitly defined by g.

However, real-world measurements are always noisy, the single-level approach can adapt to this case by adopting regularization in its objective. The detailed discussion about the regularized single-level approaches and the bi-level approach is assigned in Section 1.3.1, in order to better articulate the idea behind the existing methods and to better compare the difference between the bi-level approach and the single-level approaches.

1.1.2 Wasserstein Distance

Wasserstein distance is central to the field of optimal transport, which studies the distance between probability distribution by mass transportation of minimal effort. This idea of optimal transport was first hinted in the seminal work by Monge [56], and it is later proved to have concrete connection to convexity, partial differential equations and statistics [57]. Optimal transport is of great practical importance, as any logistics, economic and network routing problems involve moving products/mass [58–60]. Over the last two decades, scientist from computer, imaging start to apply this theory to study distribution in more abstract context, where comparing distribution between data and unknown distribution is of the greatest interest [61, 62]. In this chapter, we only need to use the 2-Wasserstein distance, and interested readers are refered to [63, 64] for more details.

The 2-Wasserstein distance between two distributions $\mathbb{P}_x$ and $\mathbb{P}_y$ is defined by

$$W(\mathbb{P}_x, \mathbb{P}_y) := \left(\inf_{\gamma \in \Gamma(\mathbb{P}_x, \mathbb{P}_y)} \mathbb{E}_{(x,y)\sim\gamma} \|x - y\|^2 \right)^{\frac{1}{2}} ,$$

where $\Gamma(\mathbb{P}_x, \mathbb{P}_y)$ is the family of joint distributions whose marginals are $\mathbb{P}_x$ and $\mathbb{P}_y$. The 2-Wasserstein distance models the optimal transport between $\mathbb{P}_x$ and $\mathbb{P}_y$ in terms of the Euclidean distance. If $\mathbb{P}_x \sim \mathcal{N}(\mu_x, \Sigma_x)$ and $\mathbb{P}_y \sim \mathcal{N}(\mu_y, \Sigma_y)$, then the squared 2-Wasserstein distance has a closed-form [**givens1984class**, 65]

$$W(\mathbb{P}_x, \mathbb{P}_y)^2 = \|\mu_x - \mu_y\|^2 + \operatorname{tr}\left(\Sigma_x + \Sigma_y - 2(\Sigma_x^{\frac{1}{2}} \Sigma_y \Sigma_x^{\frac{1}{2}})^{\frac{1}{2}} \right). \tag{1.4}$$

1.1.3 Setting the Stage

Recall the main properties in building applications:

- The system dynamics are slowly time-varying.
- The building dynamics evolve under strong **measurable** process noise, such as solar radiation and outdoor temperature.
- Because of the activity of the occupants, the output measurements, particularly the measurements of the indoor temperature, are noisy.

Given these properties, this work focuses on the following uncertain linear time-varying (LTV) system

$$\begin{aligned} x_{i+1} &= A_i x_i + B_i u_i + E_i w_i \\ \overline{y}_i &= C_i x_i + D_i u_i \\ y_i &= \overline{y}_i + v_i \end{aligned} \tag{1.5}$$

where $w_i \in \mathbb{R}^{n_w}$ is bounded measurable process noise and $v_i \sim \mathcal{N}(0, \Sigma_v)$ with $v_i \in \mathbb{R}^{n_y}$ independent and identically distributed (i.i.d) unknown measurement noise. In particular, $\overline{y}$ is the system output, which is unknown, and y is the measurement read out from the sensors. In particular, in the building control problem, w mostly reflects the external temperature, solar radiation, occupancy, etc. Note that LTI systems are special case of the LTV systems (1.5).

In a similar manner to [66, 67], by viewing w as uncontrolled inputs, Lemma 1 can be readily generalized and the corresponding L-step trajectory is augmented to

$$[u_{1:L}\ \ w_{1:L}\ \ y_{1:L}] := \begin{bmatrix} u_1^\top & \dots & u_L^\top & w_1^\top & \dots & w_L^\top & y_1^\top & \dots & y_L^\top \end{bmatrix}^\top .$$

1.2 Robust Single-level DeePC

In this part, we will study how the robustness to process noise can be considered in the single-level framework (1.3). We first resitrict our discussion to measurement noise free LTI systems, they satisfy:

$$A_i = A_j,\ B_i = B_j,\ C_i = C_j,\ D_i = D_j,\ E_i = E_j, v_i = \mathbf{0},\ \forall\ i, j\ .$$

Recall that the main enabler of the single-level approach is its data-driven predictor (1.3b). This predictor is uniquely characterized by the historical I/O data, whose prediction is implicitly determined by g (see (1.3b)). When a robust predictive control problem is considered, such implicit link between prediction and the solution to the predictor (1.3b)(i.e.g) gives rises to a new parametrization of the feedback control law. Before we state the problem formulation, we assume:

Assumption 1. $\{u_\mathbf{d}, \tilde{w}_\mathbf{d}\}$ *is persistently excited of order* $L + n_x$.

This assumption is made to enable the application of the Willems' fundamental lemma. On top of that, we denote the set of predicted future process noise realizations by $\tilde{w}_{pred} \in \overline{w} \oplus \tilde{\mathscr{W}}$, where $\overline{w}$ is the nominal prediction and $\tilde{\mathscr{W}}$ denotes the uncertainty tube[1]. And we state our assumption:

Assumption 2. $\overline{w}$ *and* $\tilde{\mathscr{W}}$ *are known.*

Based on the assumptions above, we can adapt the single-level problem (1.3) to the following

[1] The use of $\tilde{\cdot}$ will be clear in Section 1.3, we stick to this notation for the sake of consistency.

robust closed-loop predictive control problem:

$$\min_{\substack{\overline{g},K,\\ u_{pred},y_{pred}}} \max_{\tilde{w}_{pred}\in\mathscr{W}} J(\overline{y}_{pred},\overline{u}_{pred}) \tag{1.6a}$$

$$\text{s.t}\ \forall\ \tilde{w}_{pred} \in \overline{w} \oplus \tilde{W} \tag{1.6b}$$

$$\begin{bmatrix} \mathfrak{H}_{L,init}(y_{\mathbf{d}}) \\ \mathfrak{H}_{L,init}(u_{\mathbf{d}}) \\ \mathfrak{H}_{L,init}(w_{\mathbf{d}}) \\ \mathfrak{H}_{L,pred}(u_{\mathbf{d}}) \\ \mathfrak{H}_{L,pred}(w_{\mathbf{d}}) \\ \mathfrak{H}_{L,pred}(y_{\mathbf{d}}) \end{bmatrix} (\overline{g}+K\tilde{w}_{pred}) = \begin{bmatrix} y_{init} \\ u_{init} \\ w_{init} \\ u_{pred} \\ \tilde{w}_{pred} \\ y_{pred} \end{bmatrix} \tag{1.6c}$$

$$u_{pred} \in \mathscr{U}\ ,\ y_{pred} \in \mathscr{Y}\ .$$

Note that in the non-robust problem (1.3), u_{pred} and y_{pred} are uniquely defined by g, but the mapping between them becomes implicit. If the uncertain process noise is viewed as uncontrolled input, Problem (1.6) needs to solve infinitely many linear equations due to the implicit link between input and output requires. To resolve this numerical tractability issue, we first parametrize the solution to (1.6c) by $\overline{g}+K\tilde{w}_{pred}$. This is a parametrization of a closed-loop prediction scheme, and it is different from the one commonly used in model-based linear MPC and the one in Section 1.3. At the first glance, this closed-loop parametrization does not resolve the tractability issue, as it still need to solve infinitely many linear equations. However, it allows us to recast this problem to a tractable form, which requires the following characterization of the robust equation systems (1.6c):

Lemma 2. *If $\overline{g}$ and K satisfy the following constraints, then the control law* (1.6c) *guarantees n_h-step robust feasibility.*

$$\begin{aligned} &\begin{bmatrix} \mathscr{H}_{init} \\ \mathfrak{H}_{L,pred}(w_{\boldsymbol{d}}) \end{bmatrix} \overline{g} = \begin{bmatrix} h_{init} \\ \mathbf{0} \end{bmatrix}, \\ &\mathscr{H}_{init}K = \mathbf{O}\ ,\ \mathfrak{H}_{L,pred}(w_{\boldsymbol{d}})K = I\ , \\ &\forall\ \tilde{w}_{pred} \in \overline{w} \oplus \tilde{W}\ , \\ &\begin{bmatrix} \mathfrak{H}_{L,pred}(u_{\boldsymbol{d}}) \\ \mathfrak{H}_{L,pred}(y_{\boldsymbol{d}}) \end{bmatrix} (\overline{g}+K\tilde{w}_{pred}) = \begin{bmatrix} u_{pred} \\ y_{pred} \end{bmatrix} \\ &u_{pred} \in \mathscr{U}\ ,\ y_{pred} \in \mathscr{Y}\ , \end{aligned} \tag{1.7}$$

where

$$\mathscr{H}_{init} := \begin{bmatrix} \mathfrak{H}_{L,init}(y_{\boldsymbol{d}}) \\ \mathfrak{H}_{L,init}(u_{\boldsymbol{d}}) \\ \mathfrak{H}_{L,init}(w_{\boldsymbol{d}}) \end{bmatrix},\ h_{init} := \begin{bmatrix} y_{init} \\ u_{init} \\ w_{init} \end{bmatrix}$$

Proof. In the control law (1.6c), the nominal $\overline{g}$ generates a disturbance-free n_h step prediction.

Hence, based on the prediction equation (1.6c), we enforce

$$\begin{aligned}&\forall\ \tilde{w}_{pred} \in \overline{w} \oplus \tilde{W} \\ &\begin{bmatrix} \mathscr{H}_{init} \\ \mathfrak{H}_{L,pred}(u_{\mathbf{d}}) \\ \mathfrak{H}_{L,pred}(w_{\mathbf{d}}) \\ \mathfrak{H}_{L,pred}(y_{\mathbf{d}}) \end{bmatrix} (\underbrace{\overline{g}}_{(a)} + \underbrace{K\tilde{w}_{pred}}_{(b)}) = \begin{bmatrix} h_{init} \\ \overline{u}_{pred} \\ \mathbf{0} \\ \overline{y}_{pred} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ u_{fb} \\ \tilde{w}_{pred} \\ y_{fb} \end{bmatrix},\end{aligned} \tag{1.8}$$

where the matrix products of (a) and (b) correspond to the components on the right-hand side accordingly. Addtionally, $\overline{u}_{pred}$ and $\overline{y}_{pred}$ are the nominal input/output sequence corresponding to the nominal predictive process noise $\overline{w}$. As the future disturbance $\tilde{w}_{pred}$ is unknown and arbitrary within the polytope $\overline{w} \oplus \tilde{\mathscr{W}}$, the matrix product of term (b) in (1.8) implies

$$\mathscr{H}_{init}K = \mathbf{O}\,,\ \mathfrak{H}_{L,pred}(w_{\mathbf{d}})K = I. \tag{1.9}$$

Due to the perturbation of the unknown future disturbance, the actual input and the actual output under the control law (1.6c) are $u_{pred} := \overline{u}_{pred} + u_{fb}$ and $y_{pred} := \overline{y}_{pred} + y_{fb}$ respectively, which summarizes the proof with the robust constraints (1.7). ■

Lemma 2 enables us to rewrite Problem (1.6) to the following tractable form:

$$\begin{aligned}\min_{\substack{\overline{g},K,\\ u_{pred},y_{pred}}} \ \max_{\tilde{w}_{pred}\in\mathscr{W}} \ & J(y_{pred}, u_{pred}) \\ \text{s.t} & \begin{bmatrix} \mathscr{H}_{init} \\ \mathfrak{H}_{L,pred}(w_{\mathbf{d}}) \end{bmatrix} \overline{g} = \begin{bmatrix} h_{init} \\ \mathbf{0} \end{bmatrix}, \\ & \mathscr{H}_{init}K = \mathbf{O}\,,\ \mathfrak{H}_{L,pred}(w_{\mathbf{d}})K = I\,, \\ & \forall\ \tilde{w}_{pred} \in \overline{w} \oplus \tilde{\mathscr{W}}\,, \\ & \begin{bmatrix} \mathfrak{H}_{L,pred}(u_{\mathbf{d}}) \\ \mathfrak{H}_{L,pred}(y_{\mathbf{d}}) \end{bmatrix} (\overline{g} + K\tilde{w}_{pred}) = \begin{bmatrix} u_{pred} \\ y_{pred} \end{bmatrix} \\ & u_{pred} \in \mathscr{U}\,,\ y_{pred} \in \mathscr{Y}\,. \end{aligned} \tag{1.10}$$

This formulation is numerically tractable, as only one linear equation needs to be solved. It is worth noting that, when the constraints $\mathscr{U}$, $\mathscr{Y}$ are polyhedral or ellipsoidal, this problem can be efficiently solved by a convex reformulation based on the standard dualization technique [68].

1.2.1 Tractable Causual Feedback K

Before discussing the details of this section, we first recall some notations used in the previous sections, i-th row block of $\mathfrak{H}_{L,pred}(x_{\mathbf{d}})$ is

$$\mathfrak{H}_{L,pred,i}(x_{\mathbf{d}}) := [x_{\mathbf{d},i+t_{init}}, x_{\mathbf{d},i+1+t_{init}} \dots, x_{\mathbf{d},i+n_c-1+t_{init}}]\,.$$

And we define $K_{:,j}$ as the j-th block column of feedback law K, which corresponds to the feedback generated by $\tilde{w}_{pred,i}$ and that is the $(n_w \times (i-1))+1$-th to $n_w \times i$-th entry of $\tilde{w}_{pred}$

Here starts the main result of this section. If the feedback matrix K is arbitrary, then the feedback control law is not necessarily causal. In particular, the feedback computed from $\tilde{w}_{pred,i}$ should not be able to change $u_{pred,1:i}$ and $y_{pred,1:i-1}$, because those events happen no later than $\tilde{w}_{pred,i}$. To construct the causal data-driven control law, we define

$$H_{rb} = \begin{bmatrix} \mathfrak{H}_{L,init}(u_{\mathbf{d}}) \\ \mathfrak{H}_{L,init}(w_{\mathbf{d}}) \\ \mathfrak{H}_{L,init}(y_{\mathbf{d}}) \\ \mathfrak{H}_{L,pred,1}(u_{\mathbf{d}}) \\ \mathfrak{H}_{L,pred,1}(y_{\mathbf{d}}) \\ \vdots \\ \mathfrak{H}_{L,pred,n_h}(u_{\mathbf{d}}) \\ \mathfrak{H}_{L,pred,n_h}(y_{\mathbf{d}}) \end{bmatrix}, \tag{1.11}$$

with $H_{rb} \in \mathbb{R}^{n_r \times n_c}$ and n_r the number of rows. The QR decomposition [69] of its transpose is

$$H_{rb}^{\top} = \begin{bmatrix} Q_a & Q_b \end{bmatrix} \begin{bmatrix} R \\ \mathbf{O} \end{bmatrix},$$

Theorem 1. *The causal feedback robust control law in* (1.10) *is identical to the following predictive control law,*

$$\begin{aligned} \min_{\substack{\overline{g},K_p,\\ u_{pred},y_{pred}}} \ \max_{\tilde{w}_{pred}} \ & J(y_{pred}, u_{pred}) \\ s.t. \ & \begin{bmatrix} \mathscr{H}_{init} \\ \mathfrak{H}_{L,pred}(w_{\boldsymbol{d}}) \end{bmatrix} \overline{g} = \begin{bmatrix} h_{init} \\ \mathbf{0} \end{bmatrix} \\ & K = \begin{bmatrix} Q_{a,:,n_{init}+1:n_r} & Q_b \end{bmatrix} K_p \,, \\ & \mathscr{H}_{init} K = \mathbf{O} \,, \ \mathfrak{H}_{L,pred}(w_{\boldsymbol{d}}) K = I \\ & \forall \ \tilde{w}_{pred} \in \overline{w} \oplus \tilde{\mathscr{W}} \\ & \begin{bmatrix} \mathfrak{H}_{L,pred}(u_{\boldsymbol{d}}) \\ \mathfrak{H}_{L,pred}(y_{\boldsymbol{d}}) \end{bmatrix} (\overline{g} + K\tilde{w}_{pred}) = \begin{bmatrix} u_{pred} \\ y_{pred} \end{bmatrix} \\ & u_{pred} \in \mathscr{U} \,, \ y_{pred} \in \mathscr{Y} \end{aligned} \tag{1.12}$$

where $n_{init} := t_{init} \times (n_u + n_y) + n_u$ and K_p has a lower block triangular structure as

$$K_p = \begin{bmatrix} K_{p,1,1} & \mathbf{O} & \mathbf{O} & \dots & \mathbf{O} \\ K_{p,2,1} & K_{p,2,2} & \mathbf{O} & \dots & \mathbf{O} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ K_{p,n_h,1} & K_{p,n_h,2} & K_{p,n_h,3} & \dots & K_{p,n_h,n_h} \end{bmatrix} .$$

$K_{p,i,j}$ are dense matrix blocks, whose sizes are $(n_u + n_y) \times n_w$ for $\forall\ i < n_h$ and are $[n_y + (n_c - n_r)] \times n_w$ when $i = n_h$.

Proof. First, it is observed that the data-driven formulation is based on the robust controller (1.10), where $\overline{y}_{pred}$ and $\overline{u}_{pred}$ are both well-defined.

We recall a useful property of QR decomposition [69]: the range of the first n rows of H is spanned by the first n columns of Q_a. Because $[Q_a, Q_b]$ is an unitary matrix, the remaining $n_r - n$ columns in Q_a and the matrix Q_b forms the null space of the first n rows in matrix H_{rb}. Considering the constraint $\mathscr{H}_{init}K = \mathbf{O}$ in problem (1.10), each column of matrix K must lie within the null space of $\mathscr{H}_{init}$. Meanwhile, the feedback ingredients from $\tilde{w}_{pred}$ cannot change the value of the first input $u_{pred,0}$. In conclusion, we enforce

$$\mathrm{colspan}(K) \subset \mathrm{colspan}([Q_{a,:,n_{init}+1:n_r}, Q_b]) ,$$

with $n_{init} := t_{init} \times (n_u + n_y) + n_u$. The j-th block column in matrix K, $K_{:,j}$, defines the feedback ingredient with respect to $\tilde{w}_{pred,j}$. By causality, the feedback from $\tilde{w}_{pred,j}$ should not be able to change the inputs and outputs that occur before $\tilde{w}_{pred,j}$. In particular, $K_{:,j}$ should further lie in the null space of the matrices $\mathfrak{H}_{L,pred,1:i}(u_\mathbf{d}), \mathfrak{H}_{L,pred,1:i-1}(y_\mathbf{d})$, we therefore enforce

$$K_{:,j} \subset \mathrm{colspan}([Q_{a,:,n_{init}+(j-1)(n_u+n_y)+1:n_r}, Q_b]) .$$

All the constraints on the null spaces can be reformulated as

$$K = \begin{bmatrix} Q_{a,:,n_{init}+1:n_r} & Q_b \end{bmatrix} K_p ,$$

which concludes the proof. ■

Remark 1. *In comparison with a model-based robust controller, the robust single-level scheme has the same scale of computational cost. In particular, the size of the optimization problem* (1.12) *only differs in the formulation of the feedback, where the number of decision variables in the feedback control law is $O(n_h \times (n_u + n_y))$ due to the causal reformulation. Hence, the computational cost of the robust single-level DeePC control scheme is similar to the robust MPC.*

As we will soon discuss in Section 1.3.2, this single-level robust scheme has low computational cost even when the Hankel matrices are updated online. In particular, the computational cost of the QR decomposition update by adding or removing a column scales linearly with respect to *the size of the Hankel matrix [70, Section 6.5].*

1.2.2 Numerical Results

In this section, numrical experiments are carried out to validate the proposed robust single-level DeePC. First, a second order system is used to show the equivalence between the robust single-level DeePC and the MPC with full state measurement. After that, we test the robust single-level scheme in a building control problem to adapt power consumption with respect to the occupation patterns. The code is implemented with YALMIP [71] interfacing the GUROBI solver [72]

Second Order System

The robust single-level scheme is compared against a robust MPC controller with state space model and full state measurement. The comparison is done on a second order system:

$$\begin{aligned} x_{i+1} &= \begin{bmatrix} 0.9535 & 0.0761 \\ -0.8454 & 0.5478 \end{bmatrix} x_i + \begin{bmatrix} 0.0465 \\ 0.8454 \end{bmatrix} u_i + \begin{bmatrix} 0.0465 \\ 0.8454 \end{bmatrix} w_i \\ y_i &= \begin{bmatrix} 1 & 0 \end{bmatrix} x_i \,, \end{aligned}$$

the process noise w is bounded within $[-0.1, 0.1]$, the inputs and outputs are constrained by $u \in [-5, 5]$, $y \in [-0.5, 0.5]$. A quadratic stage cost is used

$$\begin{aligned} J(y_{pred}, u_{pred}) = \sum_{i=1}^{n_h} &(y_{pred,i+1} - r)^\top Q (y_{pred,i+1} - r) \\ &+ u_{pred,i}^\top R u_{pred,i} \,, \end{aligned}$$

where $Q = 10$, $R = 0.1$ and r is the reference. The Hankel matrices in the robust single-level DeePC are built with a sequence of length 100. The tracking performance of the proposed controller is shown in Figure 1.1, where the robust MPC has full state measurement. Both controllers can guarantee safe operation within the constraints. Meanwhile, we can observe that the response of the robust single-level DeePC is identical to the robust MPC with full state measurement. Their equivalence are proved in our paper [67]. To avoid the introduction of SLS, we skip those details in this thesis in order to make the story line clear.

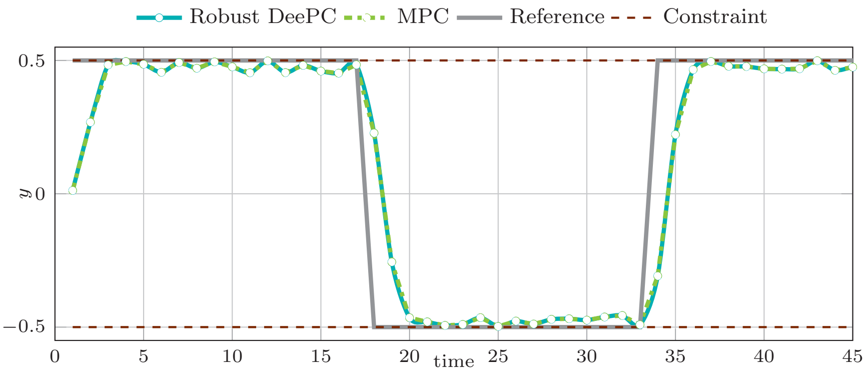


Figure 1.1: Comparison of the Robust DeePC and the MPC controller. Note that the two responses are the same.

Building Control

We consider a single zone building model [73], which is disturbed by internal heat gain, solar radiation and external temperature. The model used to generate the data is

$$\begin{aligned} x_{i+1} &= \begin{bmatrix} 0.8511 & 0.0541 & 0.0707 \\ 0.1293 & 0.8635 & 0.0055 \\ 0.0989 & 0.0032 & 0.7541 \end{bmatrix} x_i + \begin{bmatrix} 0.0035 \\ 0.0003 \\ 0.0002 \end{bmatrix} u_i \\ &\quad + 10^{-3} \begin{bmatrix} 22.2170 & 1.7912 & 42.2123 \\ 1.5376 & 0.6944 & 2.29214 \\ 103.1813 & 0.1032 & 196.0444 \end{bmatrix} w_i \, , \\ y_i &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x_i \, , \end{aligned}$$

where x models the indoor temperature, wall temperature and the corridor temperature respectively. In this building control application, the controller is designed to maintain occupant comfort while minimizing energy consumption. During winter, the indoor temperature is kept above $23°C$ to maintain occupant comfort during the day. When the room is not used at night, the room temperature is only required to stay above $17°C$. Beyond the control requirements, the disturbances, especially the internal heat gain, also show a time dependent pattern. Without loss of generality, we assume that during the day, the solar radiation and the internal heat gain are bounded within $[4,6]$ with an external temperature fluctuating between $[6°C, 8°C]$. During the night, the solar radiation is 0 with much lower internal heat gain, ranging between $[0,2]$. Meanwhile, the external temperature is also lower at around $[2°C, 4°C]$. As the controller is designed to minimize power consumption, the loss function is

$$J(y_{pred}, u_{pred}) = \|u_{pred,i}\|_1 \, .$$

The Hankel matrices used in the single-level DeePC are built by a 100-step I/O sequence. The result its operation for one day is shown in Figure 1.2, where the operation starts from 6 A.M.

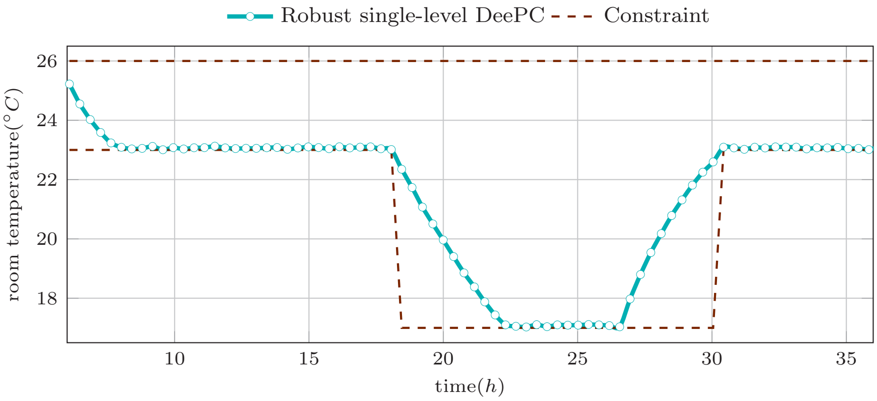


Figure 1.2: Temperature control with robust single-level DeePC

We can see that the room was over the indoor temperature lower bound at the beginning, and the controller can effectively lower the indoor temperature to save energy. It is also observed that the controller pre-heats the room to slightly above $23^\circ C$ before the 6 A.M before the second morning. This operation cycle shows the effectiveness of the controller.

1.3 Robust bi-level DeePC

In the last Section 1.2, we have developed the robust predictive controller based on the single-level DeePC framework, where a numerical tractable reformulation of closed-loop prediction is introduced. However, this controller is limits to noise-free data set. When the data is contaminated by measurement noise, regularization is used (see Section 1.3.1). In the application of robust control, the regularization approaches have at least two limitations. Firstly, the technique introduced in the last section is not applicable to avoid solving infinitely many linear equations. Secondly, the predictive trajectory is not unique given a predictive input/disturbance sequence, where the concept of robustness becomes unclear. If we refer to the bi-level optimization literature [74], it could be that, for a specific input sequence, there exists a predictive trajectory that is safe. Or it could also be that, for a specific input sequence, any predictive trajectory is safe. While the latter viewpoint seems to be more reasonable under the robust optimization framework, the resulting optimization problem will be infeasible as the set of its prediction can span an affine set. The main issue leading to the aforementioned problem is that, for the single-level DeePC, the set of predictive trajectory is not unique given a specific predictive input sequence. To resolve this issue, we develop and analyse the following bi-level predictive control problem in this part:

$$
\begin{aligned}
\min_{\substack{y_{pred}\\ \overline{u}_{pred},K}} \quad & J(y_{pred}, u_{pred}) \\
\text{s.t. } \forall\ & \tilde{w}_{pred} \in \overline{w} \oplus \tilde{\mathscr{W}} && \text{(1.13a)}\\
& u_{pred} = \overline{u}_{pred} + K\tilde{w}_{pred} \in \tilde{\mathscr{U}}\ , && \text{(1.13b)}\\
& y_{pred} = \mathfrak{H}_{L,pred}(y_{\mathbf{d}})g \in \mathscr{Y}\ , && \text{(1.13c)}\\
& g \in \arg\min_{g_l,\sigma_l} \frac{1}{2}\|\sigma_l\|^2 + \frac{1}{2}g_l^\top \mathscr{E}_g g_l && \text{(1.13d)}\\
& \text{s.t.} \begin{bmatrix} \mathfrak{H}_{L,init}(y_{\mathbf{d}}) \\ \mathfrak{H}_{L,init}(u_{\mathbf{d}}) \\ \mathfrak{H}_{L,init}(\tilde{w}_{\mathbf{d}}) \\ \mathfrak{H}_{L,pred}(u_{\mathbf{d}}) \\ \mathfrak{H}_{L,pred}(\tilde{w}_{\mathbf{d}}) \end{bmatrix} g_l = \begin{bmatrix} y_{init} + \sigma_l \\ u_{init} \\ \tilde{w}_{init} \\ u_{pred} \\ \tilde{w}_{pred} \end{bmatrix} . && \text{(1.13e)}
\end{aligned}
$$

Recall the definition used in (1.6c), $J(\cdot,\cdot)$ is a convex objective function, $\tilde{\mathscr{U}}$ and $\mathscr{Y}$ denote compact convex constraint sets on the input and the output. $\overline{w} \oplus \tilde{\mathscr{W}}$ models the set of possible future process noise with $\overline{w}$ the nominal future process noise and $\tilde{\mathscr{W}}$ the quantified uncertainty. Additionally, the tight input constraint is $\mathscr{U}$, and $\tilde{\mathscr{U}}$ is its tightening (i.e. $\tilde{\mathscr{U}} \subseteq \mathscr{U}$). $\tilde{\mathscr{W}}$ is an augmented set of actual process noise, and the set of the actual process noise is $\mathscr{W}$ (i.e. $\exists\, n \geq 0$, such that $\tilde{\mathscr{W}} \subset \mathscr{W} \times \mathbb{R}^n$). The selection of both $\tilde{\mathscr{U}}$ and $\tilde{\mathscr{W}}$ will be elaborated in the following sections. $u_{init}, \tilde{w}_{init}, y_{init}$ are t_{init}-step sequences of the measured inputs, process noise and outputs preceding the current point in time. Accordingly, $u_{pred}, \tilde{w}_{pred}, y_{pred}$ are the corresponding n_h-step predictive sequences viewed from the current time step. The matrix $\mathfrak{H}_L(y_{\mathbf{d}})$ is split into two sub-Hankel matrices:

$$
\mathfrak{H}_L(y_{\mathbf{d}}) = \begin{bmatrix} \mathfrak{H}_{L,init}(y_{\mathbf{d}}) \\ \mathfrak{H}_{L,pred}(y_{\mathbf{d}}) \end{bmatrix} .
$$

The matrix $\mathfrak{H}_{L,init}(y_{\mathbf{d}})$ is of depth t_{init} and the depth of $\mathfrak{H}_{L,pred}(y_{\mathbf{d}})$ is the prediction horizon n_h such that $t_{init} + n_h = L$. The matrices $\mathfrak{H}_{L,init}(u_{\mathbf{d}}), \mathfrak{H}_{L,pred}(u_{\mathbf{d}}), \mathfrak{H}_{L,init}(\tilde{w}_{\mathbf{d}}), \mathfrak{H}_{L,pred}(\tilde{w}_{\mathbf{d}})$ are defined similarly.

Scheme (1.13) is bi-level, where the penalty weight $\mathscr{E}_g$, the set of disturbances $\tilde{\mathscr{W}}$, and the tightened input constraints $\tilde{\mathscr{U}}$ vary with respect to the working conditions. The selections of these terms will be elaborated in the following sections. In Section 1.3.1, a rigorous setup for LTI systems will first be developed to show the logic behind the bi-level formulation. The extension to LTV system and the computational details will then be discussed in Section 1.3.2.

1.3.1 Robust Bi-level Data-driven Control for LTI Systems

In this part, we will establish a rigorous mathematical framework for the bi-level scheme (1.13) by considering the LTI version of the targeted time-varying dynamics (1.5). The LTI dynamics satisfy:

$$A_i = A_j,\ B_i = B_j,\ C_i = C_j,\ D_i = D_j,\ E_i = E_j, \forall\ i, j\ .$$

Note that the output trajectory prediction with respect to a given control input sequence u_{pred} is the key component for predictive control. In this section, we will show that this can be done by minimizing a Wasserstein distance upper bound between the trajectories determined by the fundamental lemma and noisy measurement sequences (Lemma 3). This trajectory prediction problem defines the lower level problem in our bi-level scheme (1.13), whose single-level reformulation is summarized in Lemma 4. In the end, Subsection 1.3.1 concludes this Section 1.3.1 by comparing the bi-level scheme with the existing single-level schemes.

Wasserstein Prediction Upper Bound

Regarding (1.5), the system output $\overline{y}_{init}$ is contaminated by measurement noise, giving a noisy measurement vector y_{init}. The output $\overline{y}_{init}$ thus follows the distribution (Section 1.1.3):

$$\overline{y}_{init} \sim \mathcal{N}(y_{init}, \Sigma_{init})\ ,$$

where $\Sigma_{init} = I_{t_{init}} \otimes \Sigma_v$. Similarly, the measurement Hankel matrix $\mathfrak{H}_{L,init}(y_{\mathbf{d}})$ is subject to measurement noise, and we denote the uncertain system output Hankel matrix by $\mathfrak{H}_{L,init}(\overline{y_{\mathbf{d}}})$. Then, for arbitrary $g \in \mathbb{R}^{n_c}$, the following lemma quantifies the distribution distance between an uncertain trajectory generated by the fundamental lemma, $\mathfrak{H}_{L,init}(\overline{y}_{\mathbf{d}})g$, and the uncertain system output sequence $\overline{y}_{init}$

Lemma 3. *$\forall\ g \in \mathbb{R}^{n_c}$, the squared Wasserstein distance $W(\mathfrak{H}_{L,init}\left(\overline{y_d}\right) g, \overline{y}_{init})^2$ is upper bounded by*

$$\|\mathfrak{H}_{L,init}(y_d)g - y_{init}\|^2 + \left(\sqrt{t_{init}}\|g\| - 1\right)^2 tr(\Sigma_v)\ .$$

Proof. Similar to the distribution of $\overline{y}_{init}$, the distribution of the i-th column of the output Hankel matrix $\mathfrak{H}_{L,init}(\overline{y})$ follows a Gaussian distribution $\mathcal{N}(y_{\mathbf{d},i:i+t_{init-1}}, \Sigma_{init})$ and the adjacent columns are correlated. By the basic properties of Gaussian distributions, we have

$$\mathfrak{H}_{L,init}(\overline{y_{\mathbf{d}}})g \sim \mathcal{N}(\mathfrak{H}_{L,init}(y_{\mathbf{d}})g, \tilde{\Sigma}_{init})\ ,$$

where $\tilde{\Sigma}_{init} = \tilde{G} \otimes \Sigma_v$ and the entry of $\tilde{G} \in \mathbb{R}^{t_{init} \times t_{init}}$ is

$$\tilde{G}_{i,j} = \sum_{n=1}^{n_c-|i-j|} g_n g_{n+|i-j|}\ .$$

Hence, the squared Wasserstein distance $W(\mathfrak{H}_{L,init}(\overline{y_{\mathbf{d}}})g, \overline{y}_{init})^2$ is

$$W(\mathfrak{H}_{L,init}(\overline{y_{\mathbf{d}}})g, \overline{y}_{init})^2 = \|\mathfrak{H}_{L,init}(y_{\mathbf{d}})g - y_{init}\|^2 \\ + \underbrace{\mathrm{tr}(\tilde{\Sigma}_{init} + \Sigma_{init} - 2(\Sigma_{init}^{\frac{1}{2}}\tilde{\Sigma}_{init}\Sigma_{init}^{\frac{1}{2}})^{\frac{1}{2}})}_{(a)} \,.$$

In the rest of this proof, we will upper bound the term (a) above,

$$\begin{aligned}
(a) &= \mathrm{tr}(\tilde{\Sigma}_{init}) + \mathrm{tr}(\Sigma_{init}) + 2\,\mathrm{tr}\left((\Sigma_{init}^{\frac{1}{2}}\tilde{\Sigma}_{init}\Sigma_{init}^{\frac{1}{2}})^{\frac{1}{2}}\right) \\
&= \left(\mathrm{tr}(\tilde{G}) + \mathrm{tr}(I_{t_{init}})\right)\mathrm{tr}(\Sigma_v) - 2\,\underbrace{\mathrm{tr}\left(((\tilde{G}I_{t_{init}}) \otimes \Sigma_v^2)^{\frac{1}{2}}\right)}_{(b)} \\
&= t_{init}(\|g\|^2 + 1)\,\mathrm{tr}(\Sigma_v) - 2\,\underbrace{\mathrm{tr}(\tilde{G}^{\frac{1}{2}} \otimes \Sigma_v)}_{(c)} \\
&= \left(t_{init}(\|g\|^2 + 1) - 2\,\mathrm{tr}(\tilde{G}^{\frac{1}{2}})\right)\mathrm{tr}(\Sigma_v) \\
&\overset{(d)}{\le} \left(t_{init}(\|g\|^2 + 1) - 2\,\mathrm{tr}(\tilde{G})^{\frac{1}{2}}\right)\mathrm{tr}(\Sigma_v) \\
&= \left(t_{init}(\|g\|^2 + 1) - 2\sqrt{t_{init}}\|g\|\right)\mathrm{tr}(\Sigma_v) \\
&\overset{(e)}{\le} \left(\sqrt{t_{init}}\|g\| - 1\right)^2 \mathrm{tr}(\Sigma_v)\,,
\end{aligned} \tag{1.14}$$

where terms (b), (c) follow the same fact that $(A \otimes B)(C \otimes D) = AC \otimes BD$. The inequality (d) follows [75, Theorem 1 (ii)]

$$\mathrm{tr}(\tilde{G})^{\frac{1}{2}} \le \mathrm{tr}(\tilde{G}^{\frac{1}{2}})\,.$$

The inequality (e) uses the fact that $t_{init} \ge 1$. We conclude the proof with

$$W(\mathfrak{H}_{L,init}\left(\overline{y_{\mathbf{d}}}\right)g, \overline{y}_{init})^2 \le \\ \|\mathfrak{H}_{L,init}(y_{\mathbf{d}})g - y_{init}\|^2 + \left(\sqrt{t_{init}}\|g\| - 1\right)^2 \mathrm{tr}(\Sigma_v)\,.$$

■

Based on the upper bound in Lemma 3, we can generate a trajectory prediction via the following optimization problem:

$$y_{pred} = \mathfrak{H}_{L,pred}(y_{\mathbf{d}})g(\tilde{w}_{pred}) \tag{1.15a}$$

$$g(\tilde{w}_{pred}) \in \underset{g_l,\sigma_l}{\arg\min} \frac{1}{2}\|\sigma_l\|^2 + \frac{1}{2}\left(\sqrt{t_{init}}\|g_l\| - 1\right)^2 \mathrm{tr}(\Sigma_v)$$

$$\text{s.t.} \begin{bmatrix} \mathfrak{H}_{L,init}(y_{\mathbf{d}}) \\ \mathfrak{H}_{L,init}(u_{\mathbf{d}}) \\ \mathfrak{H}_{L,init}(\tilde{w}_{\mathbf{d}}) \\ \mathfrak{H}_{L,pred}(u_{\mathbf{d}}) \\ \mathfrak{H}_{L,pred}(\tilde{w}_{\mathbf{d}}) \end{bmatrix} g_l = \begin{bmatrix} y_{init} + \sigma_l \\ u_{init} \\ \tilde{w}_{init} \\ u_{pred} \\ \tilde{w}_{pred} \end{bmatrix}. \tag{1.15b}$$

Equation (1.15b) gives the trajectory expectation generated by the Willems' fundamental lemma. Meanwhile, $\left[y_{init}^\top + \sigma_l^\top \quad u_{init}^\top \quad \tilde{w}_{init}^\top \quad u_{pred}^\top \quad \tilde{w}_{pred}^\top\right]^\top$ is the trajectory composed by noisy measurements read out from the sensors, the planned future input and a predicted disturbance trajectory. Thus, optimization problem (1.15) minimizes the discrepancy upper bound between an I/O sequence estimated by measurement and an I/O trajectory estimated by the fundamental lemma.

Remark 2. *Note that the Hankel matrix $\mathfrak{H}_{L,pred}(y_{\boldsymbol{d}})$ is also subject to measurement noise (v in* (1.5)*), which means that the estimate of a fundamental lemma based predictive trajectory $\mathfrak{H}_{L,pred}(\overline{y}_{\boldsymbol{d}})g$ is also uncertain and Gaussian with expectation $\mathfrak{H}_{L,pred}(y_{\boldsymbol{d}})g$. The prediction given by* (1.15a) *can therefore be considered as a certainty equivalent prediction. Meanwhile, even though it is possible to consider the uncertainty in y_{pred}, for the sake of a clean layout, we only use the certainty equivalence prediction* (1.15) *in this thesis. It is noteworthy to mention that, based on our experience, the quantified predictive uncertainty $\overline{w}_{pred} \oplus \tilde{\mathscr{W}}$ is highly conservative in building applications, using certainty equivalence is empirically sufficient to give desirable robust control performance (see Section 1.3.1 for more details).*

Tractable Bi-level Reformulation

In this section, we will develop the bi-level scheme (1.13) by integrating the prediction problem (1.15) into a predictive control problem. This controller should maintain the system performance while ensuring robust constraint satisfaction regardless of future realizations of the process noise, and so we assume that the future process noise $\tilde{w}_{pred}$ can be predicted with uncertainty quantification. Buildings are systems satisfying this assumption. For example, the weather forecast can provide a future temperature prediction with an uncertainty tube centered around a nominal prediction, such that the actual future temperature realization fluctuates within this tube. Identical to the set up given in Section 1.2, we denote the set of predicted future process noise realizations by $\tilde{w}_{pred} \in \overline{w} \oplus \tilde{\mathscr{W}}$, where $\overline{w}$ is the nominal prediction and $\tilde{\mathscr{W}}$ denotes the uncertainty tube.

To make the controller less conservative, we consider a predictive control input with a linear feedback from process noise

$$u_{pred} = \overline{u}_{pred} + K\tilde{w}_{pred}\,, \tag{1.16}$$

where the feedback control K is a decision variable in our predictive control problem. In particular, $\overline{u}$ reflects the nominal control inputs while feedback K adapts the control input based on the actual realization of the process noise.

By replacing the lower level problem in (1.13) (i.e. Equation (1.13d) and (1.13c)) with the prediction problem (1.15), we can state a bi-level robust predictive control problem for LTI systems:

$$\min_{\substack{\overline{y}_{pred}\\ \overline{u}_{pred},K}} \quad J(y_{pred}, u_{pred}) \tag{1.17a}$$

$$\begin{aligned}
&\text{s.t.}\ \forall\ \tilde{w}_{pred} \in \overline{w} \oplus \tilde{\mathcal{W}}\\
&\qquad u_{pred} = \overline{u}_{pred} + K\tilde{w}_{pred} \in \tilde{\mathcal{U}}\,,\\
&\qquad y_{pred} = \mathfrak{H}_{L,pred}(y_{\mathbf{d}})g(\tilde{w}_{pred}) \in \mathcal{Y}\,,
\end{aligned} \tag{1.17b}$$

$$g(\tilde{w}_{pred}) \in \operatorname*{argmin}_{g_l,\sigma_l} \frac{1}{2}\|\sigma_l\|^2 + \frac{1}{2}\left(\sqrt{t_{init}}\|g_l\| - 1\right)^2 \operatorname{tr}(\Sigma_v)$$

$$\text{s.t.} \begin{bmatrix} \mathfrak{H}_{L,init}(y_{\mathbf{d}}) \\ \mathfrak{H}_{L,init}(u_{\mathbf{d}}) \\ \mathfrak{H}_{L,init}(\tilde{w}_{\mathbf{d}}) \\ \mathfrak{H}_{L,pred}(u_{\mathbf{d}}) \\ \mathfrak{H}_{L,pred}(\tilde{w}_{\mathbf{d}}) \end{bmatrix} g_l = \begin{bmatrix} y_{init} + \sigma_l \\ u_{init} \\ \tilde{w}_{init} \\ u_{pred} \\ \tilde{w}_{pred} \end{bmatrix},$$

where we set $\tilde{\mathcal{W}} = \mathcal{W}$ and $\tilde{\mathcal{U}} = \mathcal{U}$. Problem (1.17) is a bi-level optimization problem, whose upper level decides the optimal control and whose lower level generates the corresponding predictive output trajectory. In particular, the upper level problem sends $\overline{u}_{pred}$ and K to the lower level problem. Then for each specific $\tilde{w}_{pred}$, the lower level problem returns the corresponding predictor $g(\tilde{w}_{pred})$, which accordingly defines the output trajectory predictions y_{pred} for that $\tilde{w}_{pred}$ (Equation (1.17b)). In turn, the robust constraint in the upper level ensures that input and output constraints are satisfied for all $\tilde{w}_{pred}$ in the considered set $\overline{w} \oplus \mathcal{W}$. In conclusion, the upper level problem optimizes $\overline{u}_{pred}$ and K based on the prediction given by the lower level problem.

However, this bi-level problem (1.17) is hard to solve numerically, because the objective in the lower level problem is non-convex.[2] To address this issue, we state a looser, but convex, Wasserstein upper bound in the following corollary.

Corollary 1. *$\forall\ g \in \mathbb{R}^{n_c}$, the squared Wasserstein distance $W(\mathfrak{H}_{L,init}\left(\overline{y_{\mathbf{d}}}\right)g, \overline{y}_{init})^2$ is upper*

[2] To ensure a composition of two convex functions is convex, the outer convex function needs be non-decreasing [76, Chapter 3.2.4]. To see the non-convexity in (1.17), one can plot the following function $f(x) = (|x|-1)^2, x \in \mathbb{R}$.

bounded by

$$\|\mathfrak{H}_{L,init}(y_{\boldsymbol{d}})g - y_{init}\|^2 + t_{init}\, tr(\Sigma_v)\left(\|g\|^2+1\right).$$

Proof. In the inequality (d) of Equation (1.14), we have $\mathrm{tr}(\tilde{G}) \geq 0$ as $\tilde{G}$ is positive semi-definite. Therefore, we give a convex Wasserstein distance upper bound as

$$\begin{aligned}&W\left(\mathfrak{H}_{L,init}\left(\overline{y_{\mathbf{d}}}\right)g, \overline{y}_{init}\right)^2\\ &\quad \leq \|\mathfrak{H}_{L,init}(y_{\mathbf{d}})g - y_{init}\|^2 + t_{init}^2\,\mathrm{tr}(\Sigma_v)\left(\|g\|^2+1\right)\ .\end{aligned}$$

■

Bringing everything together, we eliminate the solution of the nonconvex problem (1.17), and arrive at a convex tractable version (i.e. the bi-level scheme (1.13)). This numerically tractable predictive control problem (1.13) will be used throughout the paper. The terms $\tilde{\mathcal{U}}$, $\tilde{\mathcal{W}}$ and $\mathcal{E}_g$ used in the LTI case are:

- $\mathcal{E}_g = t_{init}^2\,\mathrm{tr}(\Sigma_v)I_{n_c}$: The lower level problem minimizes the convex Wasserstein upper bound given in Corollary 1. For the sake of clarity, the constant term $t_{init}^2\,\mathrm{tr}(\Sigma_v)$ is dropped in its objective.
- $\tilde{\mathcal{W}} = \mathcal{W}$, $\tilde{\mathcal{U}} = \mathcal{U}$: The set of predictive process noise is directly defined by the process noise forecast (Assumption 2), and the input constraint is not tightened.

Up to this point, we have derived the bi-level structure used in the bi-level scheme (1.13), and we will show that it has a tractable single-level reformulation via the following lemma:

Lemma 4. *The following single-level robust optimization problem is equivalent to the bi-level problem* (1.13).

$$\begin{aligned}\min_{\substack{\overline{y}_{pred}\\ \overline{u}_{pred},K}}\quad & J(y_{pred}, u_{pred})\\ s.t.\ \forall\ & \tilde{w}_{pred} \in \overline{w} \oplus \tilde{\mathcal{W}}\\ & u_{pred} = \overline{u}_{pred} + K\tilde{w}_{pred} \in \tilde{\mathcal{U}}\ ,\\ & y_{pred} = \mathfrak{H}_{L,pred}(y_{\boldsymbol{d}})g(\tilde{w}_{pred}) \in \mathcal{Y}\ ,\\ & \begin{bmatrix} g(\tilde{w}_{pred}) \\ \kappa \end{bmatrix} = M^{-1}\begin{bmatrix} \mathfrak{H}_{L,init}(y_{\boldsymbol{d}})^\top y_{init} \\ u_{init} \\ \tilde{w}_{init} \\ u_{pred} \\ \tilde{w}_{pred}\end{bmatrix}\ ,\end{aligned} \tag{1.18a}$$

where κ is the dual variable of (1.13e) *and*

$$H := \begin{bmatrix} \mathfrak{H}_{L,init}(u_{\boldsymbol{d}}) \\ \mathfrak{H}_{L,init}(\tilde{w}_{\boldsymbol{d}}) \\ \mathfrak{H}_{L,pred}(u_{\boldsymbol{d}}) \\ \mathfrak{H}_{L,pred}(\tilde{w}_{\boldsymbol{d}}) \end{bmatrix} \tag{1.19a}$$

$$M := \begin{bmatrix} \mathfrak{H}_{L,init}^{\top}(y_{\boldsymbol{d}})\mathfrak{H}_{L,init}(y_{\boldsymbol{d}}) + \mathscr{E}_g & H^{\top} \\ H & \boldsymbol{O} \end{bmatrix}. \tag{1.19b}$$

Proof. Note that the uncertain lower level problem in (1.13) is strongly convex and therefore can be equivalently represented by its KKT system [74, Chapter 4]. By replacing σ_l by $\mathfrak{H}_{L,init}(y_{\mathbf{d}})g_l - y_{init}$, the Lagrangian of the lower level problem is

$$\mathscr{L}(g) = \frac{1}{2}\|\mathfrak{H}_{L,init}(y_{\mathbf{d}})g - y_{init}\|^2 + \frac{1}{2}g^{\top}\mathscr{E}_g g + \kappa^{\top}(Hg - \begin{bmatrix} u_{init} \\ \tilde{w}_{init} \\ u_{pred} \\ \tilde{w}_{pred} \end{bmatrix}),$$

where κ is the dual variable of the equality constraint. Based on this, we have the stationary condition of the KKT system

$$\frac{\partial \mathscr{L}(g)}{\partial g}^{\top} = (\mathfrak{H}_{L,init}(y_{\mathbf{d}})^{\top}\mathfrak{H}_{L,init}(y_{\mathbf{d}}) + \mathscr{E}_g)g + H^{\top}\kappa - \mathfrak{H}_{L,init}(y_{\mathbf{d}})^{\top}y_{init} = 0\,.$$

By recalling the primal feasibility condition

$$Hg = \begin{bmatrix} u_{init} \\ \tilde{w}_{init} \\ u_{pred} \\ \tilde{w}_{pred} \end{bmatrix},$$

we get the uncertain KKT matrix M in (1.19b). Finally, by Assumption 1, M is full-rank and hence invertible [77, Chapter 16]. This leads to the uncertaint KKT equation (1.18a), which concludes the proof. ■

On top of the single-level reformulation (1.18), we further enforce causality on the decision variable K through a lower-block triangular structure [78, Chapter 5.1]:

$$K = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \vdots \\ K_{2,1} & \mathbf{0} & \mathbf{0} & \dots & \vdots \\ K_{3,1} & K_{3,2} & \mathbf{0} & \dots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ K_{n_h,1} & K_{n_h,2} & K_{n_h,3} & \dots & \mathbf{0} \end{bmatrix}. \tag{1.20}$$

In particular, causality means that the i-th step of the future process noise can only change the events happening later than it, which only includes the $i+1$-th to the n_h-th components in u_{pred}.

Remark 3. *When the feasible sets* $\tilde{\mathscr{U}}, \mathscr{Y}$ *are polytopic, the robust optimization problem* (1.18) *can be solved by a standard dualization procedure [68].*

Discussion

We would wrap up this Section 1.3.1 by a comparison between the bi-level scheme and existing single-level schemes. When the historical I/O data is contaminated by measurement noise, the noise-free single-level problem (1.3) is adapted to the following regularized form [13, 79]:

$$\min_{\substack{y_{pred}, u_{pred} \\ g, \sigma}} \quad \underbrace{J(y_{pred}, u_{pred})}_{(a)} + \underbrace{\eta_g \|g\| + \eta_\sigma \|\sigma\|}_{(b)} \tag{1.21a}$$

$$\text{s.t.} \quad \begin{bmatrix} \mathfrak{H}_{L,init}(y_\mathbf{d}) \\ \mathfrak{H}_{L,init}(u_\mathbf{d}) \\ \mathfrak{H}_{L,pred}(u_\mathbf{d}) \\ \mathfrak{H}_{L,pred}(y_\mathbf{d}) \end{bmatrix} g = \begin{bmatrix} y_{init} + \sigma \\ u_{init} \\ u_{pred} \\ y_{pred} \end{bmatrix} \tag{1.21b}$$

$$u_{pred} \in \mathscr{U}, \; y_{pred} \in \mathscr{Y},$$

where η_g and η_σ are user-defined parameters and other components are similar to those defined in the bi-level scheme (1.13). For the sake of clarity, we neglect the process noise in this subsection. By comparing its objective (1.21a) with the objective function in the lower level problem (i.e. Equation (1.13d)), one can see that the last two terms (b) can be understood as the penalty on prediction error. These regularization terms are studied in [50, 51]. In particular, [51] shows that the first term in (b) is linked to the objective function used in the standard system identification procedure. Therefore, Problem (1.21) can be understood as a bi-objective optimization problem, whose loss function tries to balance the prediction accuracy and control performance. Such a trade-off between these two objectives is modelled into the user-defined weights η_g and η_σ. However, based on our experiments and the results reported in [13, 54, 66, 80], the tunning of η_g and η_σ usually requires exhaustive search and is in general non-trivial. Instead of balancing the prediction accuracy and the control performance in a single objective function (1.21a), the bi-level scheme couples them hierarchically in a bi-level optimization problem (1.13). This follows an intuitive logic applied in predictive control: the control is decided based on an accurate output prediction. In the bi-level scheme (1.13), the prediction accuracy is optimized directly by the lower level problem. Thus, the prediction accuracy of the output trajectory is always guaranteed without sacrificing the control optimality (i.e. upper level objective). On the other hand, the prediction accuracy of the single-level scheme (1.21) might be compromised due to the trade-off between control performance and the prediction accuracy (i.e. terms (a) and (b) in objective (1.21a) respectively). To better see why the bi-level scheme is preferable, one can consider a special case where $\mathscr{Y} = \mathbf{0}$, $\mathscr{U} = \mathbf{0}$ and $u_{init} = \mathbf{0}$. In this case, problem (1.21) has a non-empty solution set as $g = 0$ is feasible regardless of the value of y_{init}. This solution corresponds to an output trajectory that can jump to the origin in one step with zero input regardless of the initial state. If we do not consider the case where $A_i = \mathbf{O} \ \forall \ i$ (see dynamics (1.5)), the underlying dynamics are not able to follow this predicted trajectory for arbitrary y_{init}, and hence the prediction can be inconsistent in the single-level scheme (1.21). In comparison, because the prediction accuracy is independently ensured by the lower level problem, the bi-level scheme (1.13) will be infeasible unless the initial state is at the origin. In practice, the corner case given above barely happens, as reported by [79, 80], it is always possible to tune the η_g and η_σ to achieve a desirable closed loop performance. Note that, when the regularization term (b) in (1.21a) is not quadratic, its corresponding bi-level will be more difficult to solve numerically, as there is no convex single level reformulation. Hence, whether to use the proposed bi-level structure depends on the specific application.

Remark 4. *The idea of bi-level optimization is also presented in [51] and in subspace predictive control (SPC) [81]. In these previous works, the lower level problem defines a system identification problem based on the historical data (i.e. $\{u_{\mathbf{d}}, y_{\mathbf{d}}\}$). These approaches try to identify one single model, which is a preprocessed historical dataset in [51] (see e.g. Problem (5), (20) in [51]) and an ARX model in SPC. Thus, in this identification setup, the treatment of the noise presented in the online measurements (e.g. noise in y_{init}) is independent of the treatment of the measurement noise presented in the historical data (i.e. the identification problem). On the contrary, our bi-level scheme deals with these two sources of measurement noise in a more unified way. Firstly, because of the noise in the historical data, the representation of the underlying dynamics is uncertain. In the bi-level scheme, all the possible representations are considered (i.e. the uncertain Hankel matrices considered in Section 1.3.1). Secondly, both the uncertain representations and the online measurement noise are treated in one single problem via a Wasserstein distance upper bound (Corollary 1). Due to these differences, we intentionally call the lower level problem in the bi-level scheme a “trajectory prediction" problem but not an “identification" problem.*

Remark 5. *Different from our analysis, [50] studies a general setup that further includes unknown process noise. [50] applies the Wasserstein distance to show that the regularization on g (first term of* (b) *in* (1.21a)*) is related to the minimization of the conditional value at risk (CVaR) from a distributionally robust viewpoint.*

Remark 6. *The objective function $J(y_{pred}, u_{pred})$ has different choices, such as the robust objective $J(y_{pred}, u_{pred}) = \max_{\tilde{w}_{pred}} \tilde{J}(y_{pred}, u_{pred})$. However, unless $\tilde{J}(\cdot,\cdot)$ is linear, this robust* *objective is non-convex, and therefore does not meet our requirement (Section 1.1.3). Notice that the certainty equivalence is widely adopted in building applications, it is therefore reasonable to optimize the nominal performance (i.e. $J(y_{pred}, u_{pred}) = \tilde{J}(\overline{y}_{pred}, \overline{u}_{pred})$).*

1.3.2 Heuristic Time-varying Extension

In this section, we will adapt the controller (1.13) to LTV dynamics (1.5) by two heuristics. The modifications are summarized as follows:

1. **Modification of the lower level objective:** $\mathscr{E}_g$ is a diagonal matrix $\mathrm{diag}(\eta_{g,1}, \ldots, \eta_{g,n_c})$ and the weight sequence $\{\eta_{g,i}\}_{i=1}^{n_c}$ is decreasing. Recall that the i-th entry of g, g_i, is the weight of the i-th column in H and $\mathfrak{H}_{L,init}(y_\mathbf{d})$ used for prediction. Thus, a non-uniform penalty on g can be used to model our preference of using recent data for trajectory prediction, and the decreasing diagonal elements in $\mathscr{E}_g$ reflect this preference.
2. **Adaptation of the measured dataset:** The data are updated online to capture the latest dynamics from the system. In particular, Hankel matrices are updated by appending new input/output measurements on the right side of the Hankel matrices and by discarding old data on the left side.

Note that the second heuristic is tailored for slowly time-varying systems. This is the case for building applications, whose variations are usually seasonal. This section will discuss how to update the dataset in a way that can robustly guarantee the data quality (Section 1.3.2), and that has a scalable update computation (Section 1.3.2). It is worth mentioning that updating dataset online is also used in [82] to learn the linearized model around different operating points online.

Active Excitation

In this part, we will discuss the selection of the $\tilde{\mathscr{U}}$ and $\tilde{\mathscr{W}}$ in (1.13). Recall that persistent excitation is the key assumption required for Willems' fundamental lemma to apply. As the persistent excitation condition is not explicitly considered in the predictive control problem (1.13), thus the control input excitation may become impersistent in closed loop. In this case, updating Hankel matrices with these latest I/O sequence is not reasonable. For example, in building control, if the outdoor temperature and/or solar irradiation are near the building's equilibrium point, no extra heating/cooling is needed when the energy consumption is aimed to be minimized. In this case, the long-term zero-valued control input will not excite the system persistently. To accommodate this issue, we introduce a robust active excitation scheme, which perturbs the control input applied at time i by a random excitation signal

$$u_i = \underbrace{\overline{u}_{pred,1|i}}_{(a)} + \underbrace{u_{e,i}}_{(b)}\ , \tag{1.22}$$

where $\overline{u}_{pred,1|i}$ is the first element of $\overline{u}_{pred}$ determined by a predictive control problem (1.13) solved at time i. In this decomposition, the term (a) is determined by the predictive control problem (1.13) with some specific choice of $\tilde{\mathcal{U}}$ and $\tilde{\mathcal{W}}$ (see Algorithm (1) below), and the term (b) is a bounded excitation input, which is unknown to the decision process of $\overline{u}_{pred,1|i}$. More specifically, from the viewpoint of u_{pred}, u_e is an uncontrolled, but measurable, process noise and the underlying time-varying linear dynamics therefore becomes

$$x_{i+1} = A_i x_i + B_i u_{pred,1|i} + \begin{bmatrix} E_i & B_i \end{bmatrix} \begin{bmatrix} w_i \\ u_{e,i} \end{bmatrix},$$

where the excitation input is randomly sampled from a user-defined compact set as $u_e \in \mathcal{U}_e \subset \mathcal{U}$. The process noise $\tilde{w}$ is accordingly augmented to $\begin{bmatrix} w_{pred}^\top & u_e^\top \end{bmatrix}^\top$, and the uncertainty set $\tilde{\mathcal{W}}$ in (1.13a) is augmented to $\left(\overline{w} \oplus \mathcal{W}\right) \times \mathcal{U}_e$. As a result, $\tilde{w}_{init}$ is set to $\begin{bmatrix} w_{init}^\top & \mathbf{0}^\top \end{bmatrix}^\top$. Meanwhile, due to an extra excitation signal in (1.22), the feasible set of the control input $\tilde{\mathcal{U}}$ is tightened to $\mathcal{U} \ominus \mathcal{U}_e$.

In practice, this active excitation mechanism sacrifices the flexibility of the control input for data quality. If not necessary, we should set $\tilde{\mathcal{U}} = \mathcal{U}$ and $\tilde{\mathcal{W}} = \mathcal{W}$ to generate control inputs and deactivate the active excitation scheme. We summarize the general algorithm of the proposed controller in Algorithm 1, which automatically selects the uncertainty set $\tilde{\mathcal{W}}$ and input feasible set $\tilde{\mathcal{U}}$.

Algorithm 1

```
while true do
    Measure $y_i, w_i$ and update Hankel matrices.
    $\tilde{\mathcal{U}} \leftarrow \mathcal{U}, \tilde{\mathcal{W}} \leftarrow \overline{w} \oplus \mathcal{W}$
    Solve problem (1.13) to get $\overline{u}_{pred|i}$
    if input excitation $\{u_{\mathbf{d}}, \overline{u}_{pred|i}\}$ is not persistently exciting then
        $\tilde{\mathcal{U}} \leftarrow \mathcal{U} \ominus \mathcal{U}_e$, $\tilde{\mathcal{W}} \leftarrow \left(\overline{w} \oplus \mathcal{W}\right) \times \mathcal{U}_e$
        Solve problem (1.13) to get $\overline{u}_{pred|i}$
        Sample $u_{e,i}$ from $\mathcal{U}_e$
        Apply $u_i = \overline{u}_{pred,1|i} + u_{e,i}$
    else
        Apply $u_i = \overline{u}_{pred,1|i}$
    end if
    $i \leftarrow i + 1$
end while
```

In this algorithm, the problem (1.18) is first solved without active excitation. If its nominal solution $\overline{u}_{pred}$ is not expected to be persistently excited, the problem (1.18) is re-solved considering the active excitation scheme. Note that the computational cost of checking the rank condition is $O(n_c^3)$ [69], and may not be affordable for online computation. This can be replaced by some effective heuristics. In building control, the control input excitation becomes impersistent mainly when the control input is zero (e.g. when the cooling/heating is turned off), and thus the persistence excitation condition can be heuristically replaced by checking whether the nominal predictive input is near to a zero valued sequence up to some user-defined tolerance. This heuristic is particularly useful in building control. Because when input is not zero-valued, the stochastic property of the process noise (e.g. solar radiation and outdoor weather) will cause random fluctuation in the closed-loop input trajectory, and the persistent excitation condition is in turn satisfied. On top of this aspect, due to the stochastic property of the process noise, the process noise is usually persitently excited.

Remark 7. *In general, generating a persistently excited control input while considering control performance is challenging, as the persistent excitation condition depends on the rank of $\mathfrak{H}_L(u_{\mathbf{d}})$, which turns the optimization problem into a challenging non-smooth non-convex optimization [83]. It is noteworthy that [84] also perturbs the nominal control input to guarantee persistent excitation, however, their result has no guarantee of robust constraint satisfaction.*

Remark 8. *Due to the causality constraints in* (1.20), *the matrix K in* (1.13b) *cannot instantaneously counteract the excitation signal u_e with a $K = \begin{bmatrix} K_w & -I \end{bmatrix}$, which is non-causal.*

Numerical Details

In our proposed adaptive robust controller, the Hankel matrices are updated online with the real-time measurement of u_i, y_i, w_i (Section 1.3.2 and Algorithm 1). Meanwhile, a numerically efficient reformulation of the robust problem (1.18) requires an explicit evaluation of matrix inversion M^{-1} in (1.18a) at each update. More specifically, when the feasible sets $\tilde{\mathcal{U}}, \mathcal{Y}$ and the uncertainty set $\tilde{\mathcal{W}}$ are polytopic or ellipsoidal, the dualization/explicit upper bound of the robust inequality constraint depends on the matrix inversion M^{-1}. However, the computational cost of M^{-1} is $O((n_c+n_r)^3)$ [69] with n_r and n_c the number of rows and columns in the matrix H, which is roughly cubic in the size of the data set. Thus, direct inverse of M online is not scalable. This is particularly important for building applications, because the computing unit in building applications is usually of lower performance [85]. We therefore propose to apply two linear algebraic techniques to resolve this computational bottleneck.

Notice that the dual variable κ in the lower problem does not affect the upper level problem. For the sake of compactness, we denote

$$M_{1,1} := \mathfrak{H}_{L,init}^{\top}(y_{\mathbf{d}})\mathfrak{H}_{L,init}(y_{\mathbf{d}}) + \mathscr{E}_g$$

and $M_{\text{sch}} := (HM_{1,1}^{-1}H^{\top})^{-1}HM_{1,1}^{-1}$. By matrix inversion of a block-structured matrix, we have

$$M^{-1} = \begin{bmatrix} M_{1,1}^{-1} - M_{1,1}^{-1}H^{\top}M_{\text{sch}} & M_{\text{sch}}^{\top} \\ M_{\text{sch}} & -(HM_{1,1}^{-1}H^{\top})^{-1} \end{bmatrix}.$$

We can therefore replace the constraint (1.18a) by

$$g(\tilde{w}_{pred}) = M_{\mathrm{top}} \begin{bmatrix} \mathfrak{H}_{L,init}(y_{\mathbf{d}})^\top y_{init} \\ u_{init} \\ \tilde{w}_{init} \\ u_{pred} \\ \tilde{w}_{pred} \end{bmatrix},$$

where

$$M_{\mathrm{top}} := \begin{bmatrix} M_{1,1}^{-1} - M_{1,1}^{-1} H^\top M_{\mathrm{sch}} & M_{\mathrm{sch}} \end{bmatrix}.$$

With the aforementioned modification, the computational cost is lowered to $O(n_c^3)$, whose computational bottleneck lies at the inversion of $M_{1,1}$. We further lower the computational cost by the Woodbury matrix identity,

$$\begin{aligned} &M_{1,1}^{-1} \\ &= (\mathfrak{H}_{L,init}^\top(y_{\mathbf{d}})\mathfrak{H}_{L,init}(y_{\mathbf{d}}) + \mathscr{E}_g)^{-1} \\ &= \mathscr{E}_g^{-1} - \mathscr{E}_g^{-1}\mathfrak{H}_{L,init}^\top(y_{\mathbf{d}}) M_{\mathrm{mid}}{}^{-1}\mathfrak{H}_{L,init}(y_{\mathbf{d}})\mathscr{E}_g^{-1}, \end{aligned} \tag{1.23}$$

where $M_{\mathrm{mid}} := I_m + \mathfrak{H}_{L,init}(y_{\mathbf{d}})\mathscr{E}_g^{-1}\mathfrak{H}_{L,init}^\top(y_{\mathbf{d}})$. As $\mathscr{E}_g$ is diagonal with a simple and explicit inversion, the major computation cost happens at the inversion of a size m matrix M_{mid}, where $m := t_{init} n_y$. Thus, the computational cost of $M_{1,1}^{-1}$ and M_{top} is lowered to $O(m^3)$, which is fixed and independent of the number of columns in the Hankel matrices (*i.e.* roughly the size of the data-set).

Remark 9. *Note that the KKT matrix is usually ill-conditioned [77, Chapter 16]. Replacing the full matrix inversion in* (1.18a) *with the proposed techniques can improve the numerical stability. Because only the matrix inversion of $\mathscr{E}_g$ and M_{mid} in* (1.23) *are evaluated, and these two matrices are well-conditioned.*

1.3.3 Numerical Results

In this part, we will demonstrate that the bi-level scheme shows comparable performance against some model-based methods in both LTI and LTV systems.

Multi-zone Building Model

It should be noted that we are not claiming that the data-driven approach outperforms all model-based approaches, as it is definitely possible to tune a better model based method, such as considering the uncertainty of the identified parameters or using a more complex estimator/controller. We only aim to show that the bi-level approach has comparable performance against a model-based method, but without requiring a model. We therefore select a standard model-based controller design pipeline that we believe is reasonable, and we compare this standard scheme against our bi-level scheme in this example. We considered an LTI multi-zone building model reported in [86] (index of the rooms are shown in schematic diagram Figure. 1.3. Due to the space limit, the parameters of the model (i.e. A, B, C, D, E matrices) are included in the supplementary material on Github[3]. In this multi-zone building (Figure 1.3), room 4 is the corridor linking a large warehouse (room 1) and two offices (room 2 and 3). The indoor temperature of room 1 is controlled by an independent HVAC system, while another HVAC controls the temperature of all other rooms (i.e. $u \in \mathbb{R}^2$). Only the indoor temperature of these four rooms are measured (i.e. $y \in \mathbb{R}^4$), while the underlying model is 13 dimensional including the wall temperature (i.e. $x \in \mathbb{R}^{13}$). Process noise are outdoor temperature and solar radiation (i.e. $w \in \mathbb{R}^2$), and real weather data is used for the closed-loop simulation. The sampling time of this discrete time model is 15 minutes. Recall the system dynamics (1.5), two different level of unknown measurement noise v are considered to show the reliability of the bi-level scheme. We remind the reader that only the measurement y_i is available to both schemes and $\overline{y}_i$ is unknown to both schemes (See the dynamics (1.5)).

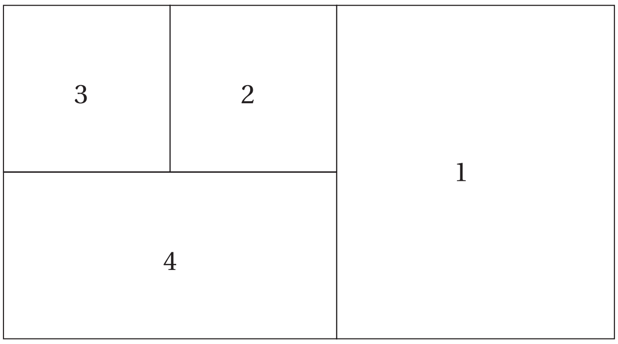


Figure 1.3: Schematic diagram of the multi-zone building

In the “standard" scheme, the model is identified by a subspace identification algorithm and the state estimation is done by a Kalman filter[4]. Following this, a robust model predictive controller with linear feedback [78] is used in the “standard" scheme to generate control

[3] https://github.com/YingZhaoleo/Building_results

[4] We use the commands N4SID and KALMAN in MATLAB to do subspace identification and Kalman filter design respectively

inputs[5].

$$
\begin{aligned}
\min_{\substack{u_{pred},K_w\\ y_{pred}}} \; & J(\overline{y}_{pred},\overline{u}_{pred}) \\
\text{s.t. } & x_{pred,0} = x_0 \\
& \forall\; i = 0,1\dots,n_h \\
& x_{pred,i+1} = A_{id}x_{pred,i} + B_{id}u_{pred,i} \\
& \qquad\qquad + E_{id}w_{pred,i} \\
& y_{pred,i} = C_{id}x_{pred,i} + D_{id}u_{pred,i} \\
& \forall\; w_{pred} \in \mathscr{W}\;,\; u_{pred} = \overline{u}_{pred} + K_w w_{pred} \\
& u_{pred} \in \mathscr{U}\;,\; y_{pred} \in \mathscr{Y}\;,
\end{aligned}
\tag{1.24a}
$$

where the parameters of A_{id} , B_{id} , E_{id} , C_{id} , D_{id} come from system identification and x_0 is estimated with a Kalman filter. Note that the feedback control law K_w is optimized by (1.24). Other components, such as $J(\cdot,\cdot)$, n_h and $\mathscr{U},\mathscr{Y},\mathscr{W}$, are identical to those used in the proposed robust data-driven scheme. In particular, the heating power (control input) $0\ kW \leq u_{pred} \leq 4.5\ kW$, the indoor temperature for all the rooms are bounded within $20°C \leq y_{pred} \leq 26°C$ and the uncertainty of the solar radiation, outdoor temperature prediction is modelled by a tube around their nominal prediction. The radius of the tube are respectively $2\ kW/m^2$ and $2°C$, and the nominal prediction $\overline{w}_{pred,i}$ comes from the weather forecast. An indoor temperature control problem is considered:

$$
J(y_{pred},u_{pred}) = \sum_{i=0}^{n_h} 10(\overline{y}_{pred,i+1} - y_{ref})^2 + 0.1\overline{u}^2_{pred,i}\,,
$$

where y_{ref} is the set point of the indoor temperature and the prediction horizon is set to $n_h = 8$, and the sampling time of this model 15 minutes. Additionally, we set $t_{init} = 5$ in the proposed robust scheme. To ensure a fair comparison, the same dataset is used for system identification, and for defining the Hankel matrices in the LTI case as well as the initial Hankel matrices in the LTV case. Four days of historical data is used, which includes 384 data points.

Two experiments with different levels of measurement noise are considered, whose standard deviation are respectively $0.05°C$ and $0.3°C$. These two cases correspond to a measurement error roughly bounded by $0.15°C$ and $1°C$ respectively. The results are shown in Figure 1.4[6] and Table 1.1. Each experiment carries out 50 Monte-Carlo runs. In each Monte-Carlo run, a new dataset is generated for system identification and for the definition of the Hankel matrices. The averaged fitting accuracy[7] are summarized in Table 1.1 as well, which shows good modelling accuracy in the “standard" scheme. Additionally, the constraint violation is

[5] We used the code template of “Approximate closed-loop minimax solution” in https://yalmip.github.io/example/robustmpc

[6] To be fair, both controllers are subject to the same measurement noise and process noise during their online operation.

[7] The fitting accuracy is generated by the COMPARE command in MATLAB.

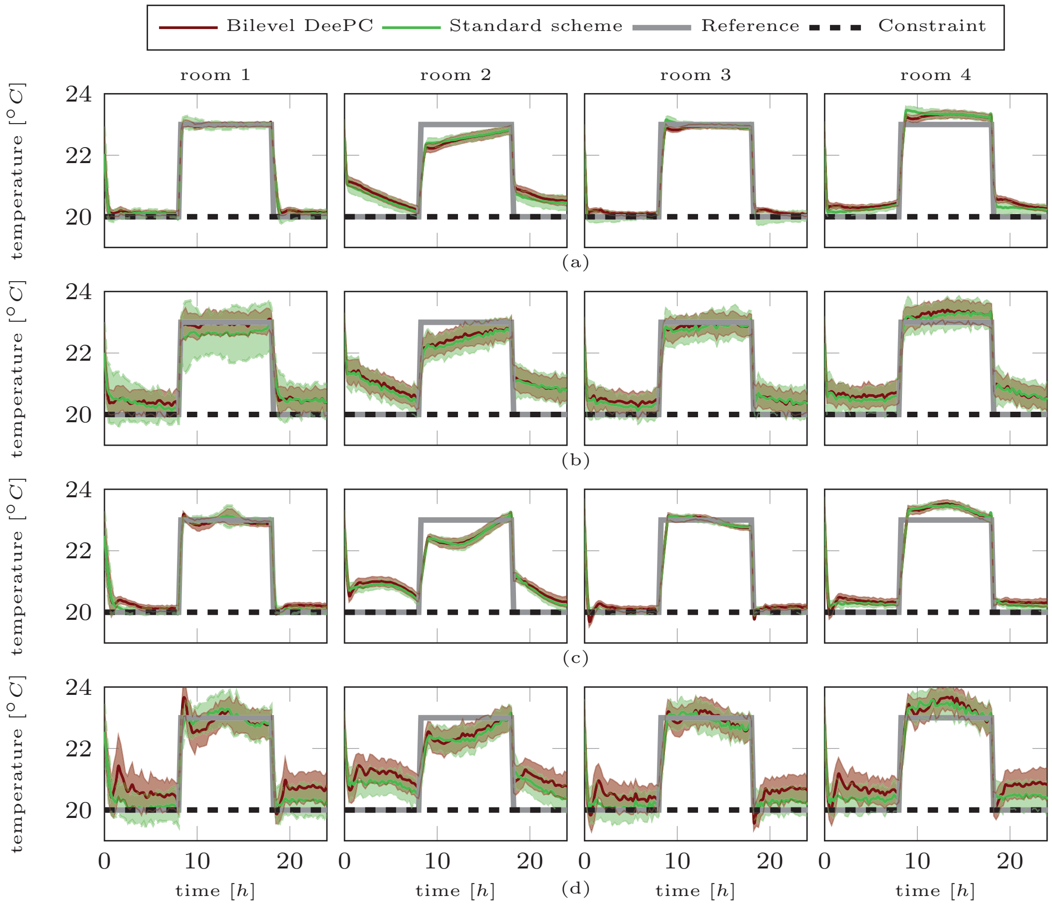


Figure 1.4: Comparison of the proposed robust data-driven control and model based controller. The shaded region is the mean plus/minus two standard deviation of all MonteCarlo runs. Standard deviations of measurement noise: row (a): LTI model, measurement noise std: 0.05 $^\circ C$; row (b): LTI model, measurement noise std: 0.3 $^\circ C$; row (c): LTV model, measurement noise std: 0.05 $^\circ C$; row (d): LTV model, measurement noise std: 0.3 $^\circ C$

Room index		1	2	3	4
Constraint violation	Proposed scheme	0.67% (3.12%)	0% (0.14%)	0.40% (2.76%)	0% (0.078%)
	Standard scheme	2.78% (11.46%)	0% (0.99%)	3.28% (10.27%)	0.42% (3.77%)
Averaged fitting accuracy		93.17% (77.60%)	93.07% (73.43%)	90.72% (71.32%)	92.58% (83.93%)

Table 1.1: Statistics of the constraint violation and the fitting accuracy of the identified models. The bracket number in each entry corresponds to the tests whose standard deviation of measurment noise is 0.3$^\circ C$, the unbracketed ones are of 0.05$^\circ C$ measurement noise standard deviation.

calculated by

$$\frac{\text{number of steps where a constraint violation occurs}}{(\text{number of simulation steps} \times \text{number of Monte-Carlo runs})}.$$

In the second test, we consider that the building dynamics varies slowly on a weekly basis:

$$\begin{aligned} A_i &= A + 0.02 \sin\left(\frac{i\pi}{336}\right) I_{13} \\ B_i &= B,\ C_i = C,\ D_i = D,\ E_i = E,\ \forall\ i\ . \end{aligned}$$

Our bi-level scheme is compared against a model based adaptive method, where a recursive least square (RLS) estimator updates the parameter of an ARX model:

$$y_i = \sum_{j=1}^{t_{init}} \theta_{y,j}^\top y_{i-j} + \theta_{u,j}^\top u_{i-j} + \theta_{w,j}^\top w_{i-j}\ ,$$

where $\theta_{y,j} \in \mathbb{R}^4$, $\theta_{u,j} \in \mathbb{R}^2$, $\theta_{w,j} \in \mathbb{R}^2$. The model estimated by RLS is used in the following robust MPC problem:

$$\begin{aligned} \min_{\substack{u_{pred}, K_w \\ y_{pred}}} \ & \max_{w_{pred}} J(y_{pred}, u_{pred}) \\ \text{s.t. } & x_{pred,0} = x_0 \\ & \forall\ i = 0, 1 \ldots, n_h \\ & y_{pred,i} = \sum_{j=1}^{t_{init}} \theta_{y,j} y_{pred,i-j} + \theta_{u,j} u_{pred,i-j} \\ & \qquad\quad + \theta_{w,j} w_{pred,i-j} \\ & \forall\ w_{pred} \in \mathcal{W}\ ,\ u_{pred} = \overline{u}_{pred} + K_w w_{pred} \\ & u_{pred} \in \mathcal{U}\ ,\ y_{pred} \in \mathcal{Y}\ , \end{aligned} \tag{1.25}$$

where the parameters θ_y, θ_u and θ_w are updated by the RLS estimator. Other settings are the same as the previous experiments. In particular, the RLS estimator has a forgetting factor of 0.98 and it is initialized by the data used to build the initial Hankel matrices of the bi-level scheme. Two experiments with different levels of measurement noise standard deviation are conducted. The results are shown in rows (c) and (d) in Figure 1.4 and the constraint violation statistics are given in Table 1.2. From rows (c) and (d) in Figure 1.4, we can see that both approaches perform the tracking task properly, and the bi-level scheme shows comparable performance against the RLS-MPC approach. One major observation is that the bi-level scheme shows better constraint satisfaction against the RLS-MPC method. Indeed, it is possible to consider the uncertainty generated by the RLS estimator to improve the robustness of the model-based approach, however, it turns out to be a non-convex robust optimization problem and there is no standard approach to solve this problem.

1.3.4 Experiment

This part presents our real-world experiments conducted at a conference building on the EPFL campus.

room index		1	2	3	4
Constraint violation	Proposed scheme	1.64% (6.99%)	0% (0.49%)	2.70% (7.65%)	0.47% (2.25%)
	RLS-MPC scheme	8.23% (22.54%)	0.37% (3.08%)	10.16% (22.21%)	0.69% (11.11%)

Table 1.2: Statistic of the constriant violation. The bracket number in each entry corresponds to the tests whose standard deviation of measurment noise is $0.3^\circ C$, the unbracketed ones are of $0.05^\circ C$ measurement noise standard deviation.

Experimental Setup

The building control experiment is conducted on an entire building, which is a freestanding $600m^2$ single-zone building on the EPFL campus, called the *Polydome*. It is regularly used for lectures/exams and accommodates up to 200 people (Figure 1.5).

Figure 1.5: The Polydome

In the presented experiments, the indoor temperature measurement is the noisy output measurement y, the active electrical energy consumption of the HVAC system is the input u, and the weather conditions (outdoor temperature and solar radiation) are the process noise $w = \begin{bmatrix} w_1 & w_2 \end{bmatrix}$. The sampling period T_s is 15 minutes and the structure of the control system is depicted in Figure 1.6, where the arrows indicate the direction of data transmission. The system consists of five main components

- **Sensors**: Four Z-wave Fibaro Door/Window Sensor v2 are put in different locations in the *Polydome* to measure the indoor temperature (path (a)). Every five minutes, the temperature measurements are sent to the database through a wireless Z-wave network (path (c)). The average value of the four measurements is used as the indoor temperature. The active power consumption of the HVAC system is measured via an EMU 3-phase power meter [87] (path (b)).

- **Database**: We use INFLUXDB 1.3.7 [88] to log the time-series data, which records the measurements from sensors (path (c)), the control input computed from the PC (path (d)) and the historical weather (path (f)).
- **Weather API**: We use TOMORROW.IO [89], which provides both historical and current measurements of solar radiation and outdoor temperature to the database (path (f)). It also provides the forecast of solar radiation and outdoor temperature to the PC to solve the predictive control problem (path (g)).
- **Controller**: The controller is implemented in MATLAB, interfacing YALMIP [90], which fetches historical data from the database to build/update the Hankel matrices (path (d)), and acquires weather forecasts from the weather API (path (g)). It runs Algorithm 1 to generate the control input. This control signal is transmitted to the HVAC system via the serial communication protocol, Modbus [91].
- **HVAC**: A roof-top HVAC unit (series No: AERMEC RTY-04 heat pump) is used for heating, cooling and ventilation. Its heating and cooling units are different, consisting of two compressors for heating and one compressor for cooling respectively.

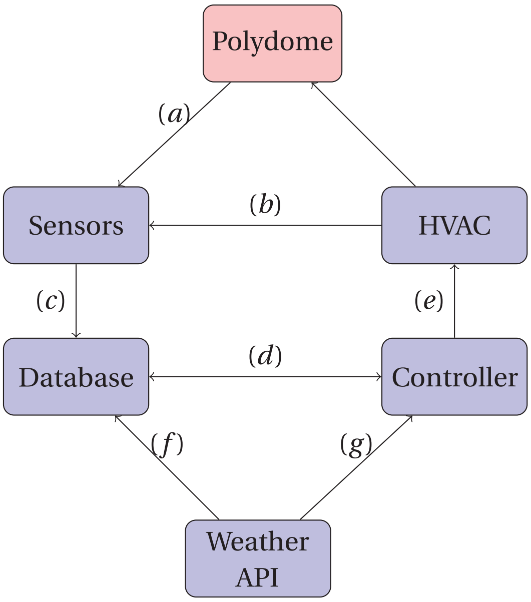


Figure 1.6: Structure of the building control system

The HVAC system is shipped with an internal hierarchical controller, which includes:

- **Mode scheduler:** The scheduler determines whether the HVAC is in heating or cooling mode, and we are **not** authorized to access this scheduler.

- **Temperature controller:** The indoor temperature is controlled by a bang-bang controller that compares the set-point temperature and the return-air (indoor) temperature with a dead-band of 1°C. For example, in the heating mode, if the return-air temperature is 1°C lower than the set-point temperature, the heating will turn on and run at full power of 8.4 kW until the set-point is reached. Vice versa for cooling, except that the electric power is 7 kW.

To map this controller to our proposed controller, we applied the following strategies:

- As the cooling and heating modes of the HVAC system show different responses, two different I/O datasets for different modes are maintained/updated independently. The controller monitors the mode of the HVAC system and deploys the corresponding I/O dataset to build the proposed controller.

- Recall that the input used in our bi-level scheme is electrical energy consumption. To achieve a desired power consumption, we convert this desired consumption to a set-point sequence: For example, if the HVAC is in heating mode, and a non-zero desired power consumption is planned, the controller will turn on the heating by giving a set-point that is 2°C above the return-air temperature, until the desired energy ($P_{el}T_s$) is reached (path (b) in Figure 1.6). Then, the heating is turned off by setting the set-point to the return-air temperature.

In our bi-level predictive control scheme, the Hankel matrices for both heating and cooling modes are built from 200 data-points, with $t_{init} = 10$ and a prediction horizon $n_h = 10$. The controller minimizes electrical power consumption with the following loss function

$$J(y_{pred}, u_{pred}) = \sum_{i}^{N_{pred}} |\overline{u}_i|$$

To better distinguish the heating and cooling modes in our plots, we use a positive input value for the heating mode and a negative input for the cooling mode. We further enforce the following input constraint to model the maximal energy consumption of the heating/cooling.

$$\begin{cases} 0\,kWh \le u_i \le 1.5\,kWh, & \text{heating mode} \\ -1.15\,kWh \le u_i \le 0\,kWh, & \text{cooling mode} \end{cases}$$

Note that the HVAC unit consumes a constant 2.4 kWh of energy for ventilation, even without heating or cooling. The aforementioned input constraint excludes this basic ventilation power. The parameter $\mathcal{E}_g$ in (1.18) is set by MATLAB command DIAG(LINSPACE(0.2,0.02,n_c)). The uncertainty set of the weather forecasts is estimated from an analysis of historical data as

$$\mathscr{W} := \left\{ w_i \middle| \begin{bmatrix} -1°C \\ -50W/m^2 \end{bmatrix} \leq w_i \leq \begin{bmatrix} 1°C \\ 50W/m^2 \end{bmatrix} \right\} \tag{1.26}$$

Experimental Results

In this section, we describe four experiments that were conducted from May 2021 to June 2021. In particular, the first experiment shows the necessity of robust control and the second experiment shows the adaptivity to mode switching. The third one runs a 20-day experiment to show the adaptivity and reliability of the bi-level scheme and the fourth one runs a 4-day experiment to compare the bi-level scheme with the default controller. Meanwhile, recall that we use a negative control input to represent cooling and a positive control input for heating. Accordingly, and show the control input and system output (indoor temperature) within the same plot to better show the response from input to output.

Experiment 1

The first experiment includes two parts: a non-robust version of the bi-level scheme (*i.e.* $\mathscr{W} = \{\mathbf{0}\}$, $K = \mathbf{O}$) and a robust version with the uncertainty tube given in (1.26). In this test, we consider a time-varying indoor temperature constraint with respect to office hours, which is relaxed during the night and is tightened to ensure occupant comfort during office hours.

$$\begin{cases} 21°C \leq y_i \leq 26°C, & \text{from 8 a.m. to 6 p.m.} \\ 19°C \leq y_i \leq 30°C, & \text{otherwise} \end{cases}$$

The non-robust experiment was conducted on 14^{th} May 2021 and the result is plotted in Figure 1.7. The HVAC system was in heating mode throughout this experiment, and we can observe that the input was 0 until the indoor temperature hit the lower bound at around 4 A.M.. Later, it started pre-heating the room to satisfy the office hours temperature constraint at around 6 A.M. However, we can still observe frequent but small constraint violations from 8 A.M. to 10 A.M., which then lead to overheating after 10 A.M.

In comparison, the robust controller effectively handled these issues in an experiment conducted on 25^{th} May 2021. The result is shown in Figure 1.8, where we used the same time varying indoor temperature constraint. We can observe that the robust controller safely protects the system from violating the lower bound through the whole test, and it also successfully pre-heated the building to fit the time-varying indoor temperature constraint. The performance deterioration that occurred to the non-robust controller after 10 A.M. was avoided as well, where the controller smoothly turned off the heating without unnecessary overheating.

Experiment 2

The second experiment is a pilot test to validate the adaptivity to the mode switching and

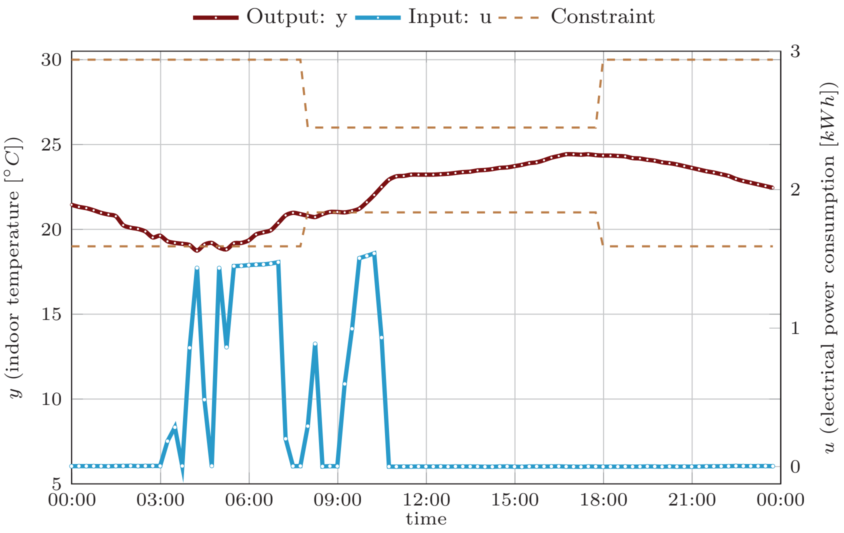


Figure 1.7: First experiment in *Polydome*: one-day heating-mode running by the bi-level data-driven control without robust optimization

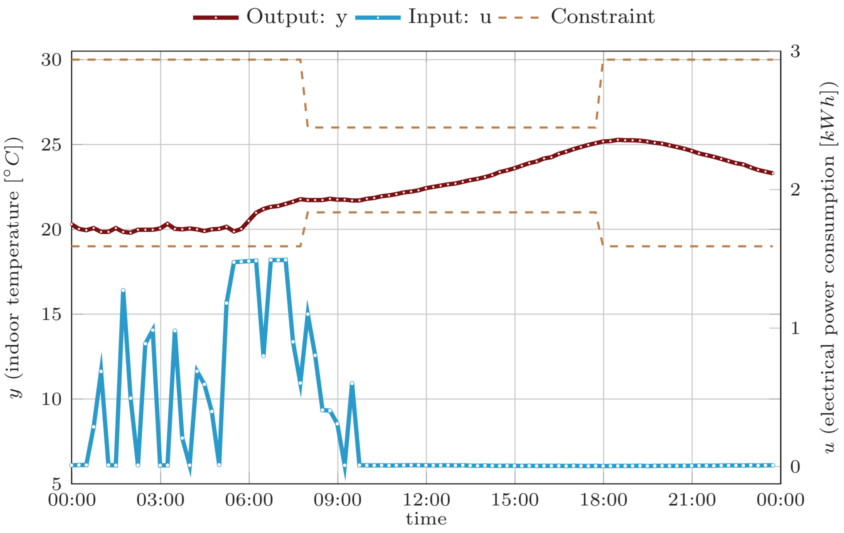


Figure 1.8: First experiment in *Polydome*: one-day heating-mode running by the proposed robust data-driven control

the necessity of active excitation (Algorithm 1). The experiment was conducted from 28^{th} to 29^{th} May 2021, with the result shown in Figure 1.9, where the indoor temperature of this experiment is bounded within

$$21°C \leq y_i \leq 25°C,$$

This change of heating/cooling mode is depicted as a positive/negative control input and a red/blue shaded region in Figure 1.9. However, it is noteworthy that the system was in cooling mode on the second evening. If there was no active excitation, the cooling should be off through the night to minimize energy consumption, and the Hankel matrices used for the controller would have lost persistent excitation. Instead, the active excitation, which is depicted as small fluctuations from 0:00 to 8:00 on the second day, maintained the persistency of excitation. The excitation signal is randomly selected as follows:

$$\begin{cases} 0\,kWh \leq u_{e,i} \leq 0.1\,kWh, & \text{heating mode} \\ -0.075\,kWh \leq u_{e,i} \leq 0\,kWh, & \text{cooling mode} \end{cases}$$

In conclusion, the controller successfully carried out the task of energy minimization in this experiment. In particular, when there is no need for heating/cooling, such as during the second evening, only active excitation took effect to maintain persistency of excitation. The heating/cooling also takes effect to robustly maintain the indoor temperature within the constraints.

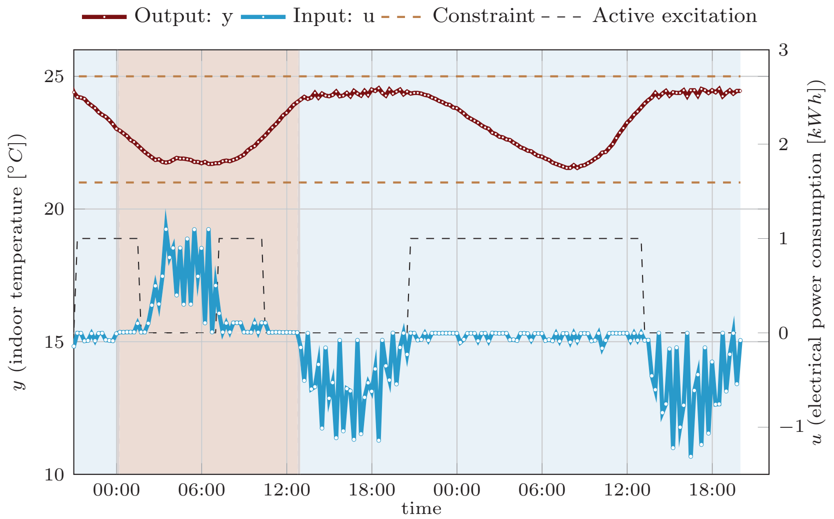


Figure 1.9: Second experiment in *Polydome*: two-day running by the proposed robust data-driven control. The black line indicates the time interval within which the active extication is active: high level: active, low level: inactive.

Experiment 3

This experiment was planned to validate the long-term reliability and adaptivity of the proposed controller. The experiment ran continuously for 20 days from 10^{th} June 2021 to 30^{th} June 2021 (Figure 1.10 and 1.11). The statistics of this experiment is summarized in Table 1.3

Note that the weather varies a lot throughout this experiment, with the outdoor temperature even once surpassing 29°C on the 20^{th} and also once dropping below 15°C on the 26^{th}. Therefore, the experiment shows the adaptivity of the proposed controller to weather variation. Moreover, the cooling mode dominated the whole experiment, with only a few days of heating at night. The proposed controller gives a long-term guarantee of temperature constraint satisfaction (see Table 1.3), while updating the Hankel matrices constantly. Regarding the data update, the active excitation scheme (Algorithm 1) also occasionally took effect to ensure persistency of excitation. The time intervals within which the active excitation are active are plotted by value 1 along the black dashed line in Figure 1.11.

average indoor temperature	average outdoor temperature	average solar radiation	duration of constraint violation	average hourly energy comsumption
$23.7°C$	$20.8°C$	$0.21W/m^2$	$0h$	$1.6kWh$
min/max indoor temperature	min/max outdoor temperature	min/max solar radiation	hours of heating	hours of cooling
$21.5°C$ $24.88°C$	$14.6°C$ $29.5°C$	$0W/m^2$ $0.89W/m^2$	$48h$	$432h$

Table 1.3: Statistics of the 20-day experiment

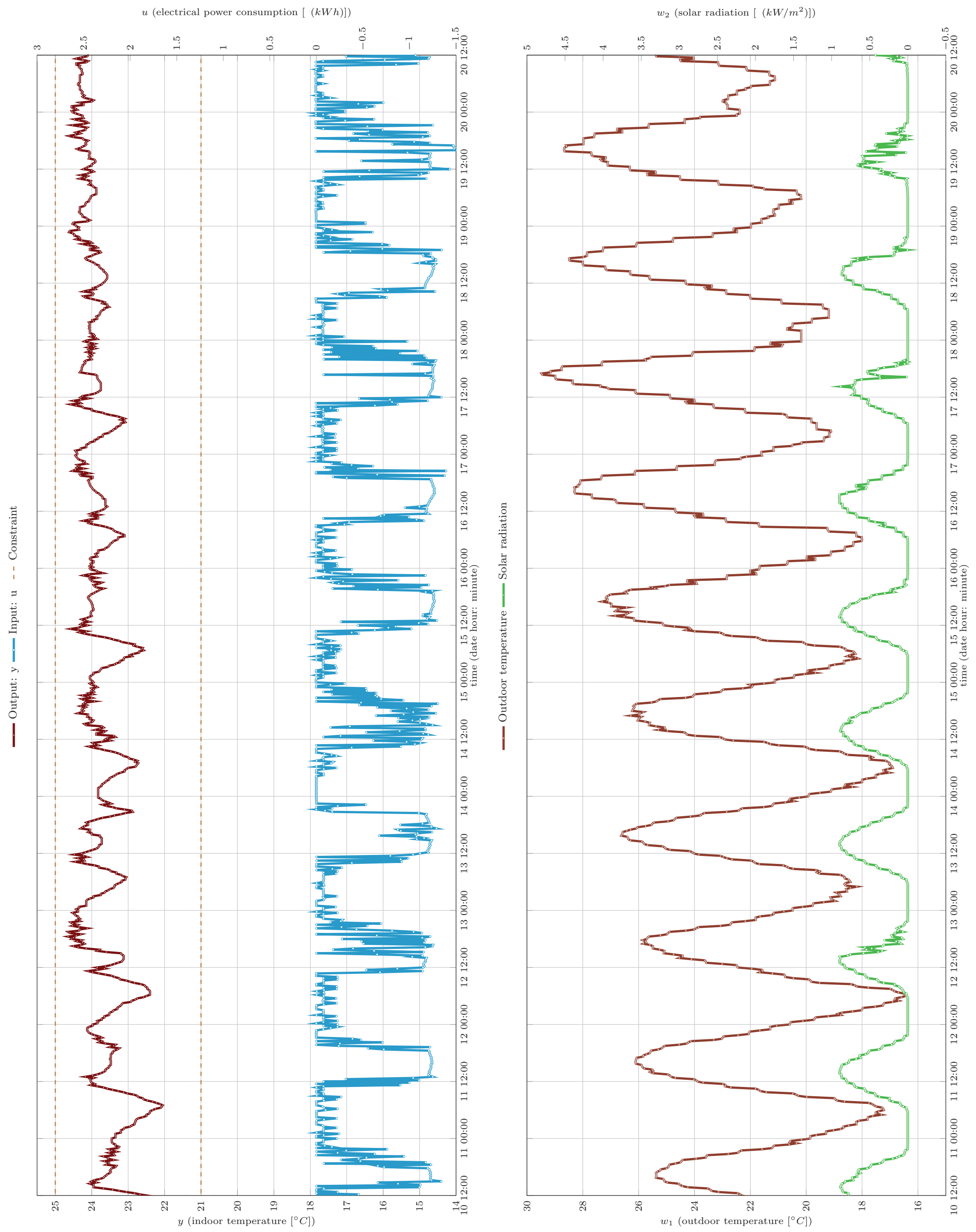


Figure 1.10: Experimental results from noon 10^{th} June 2021 to noon 20^{th} June 2021. Left:indoor temperature and electrical power consumption, right: outdoor temperature and solar radiation

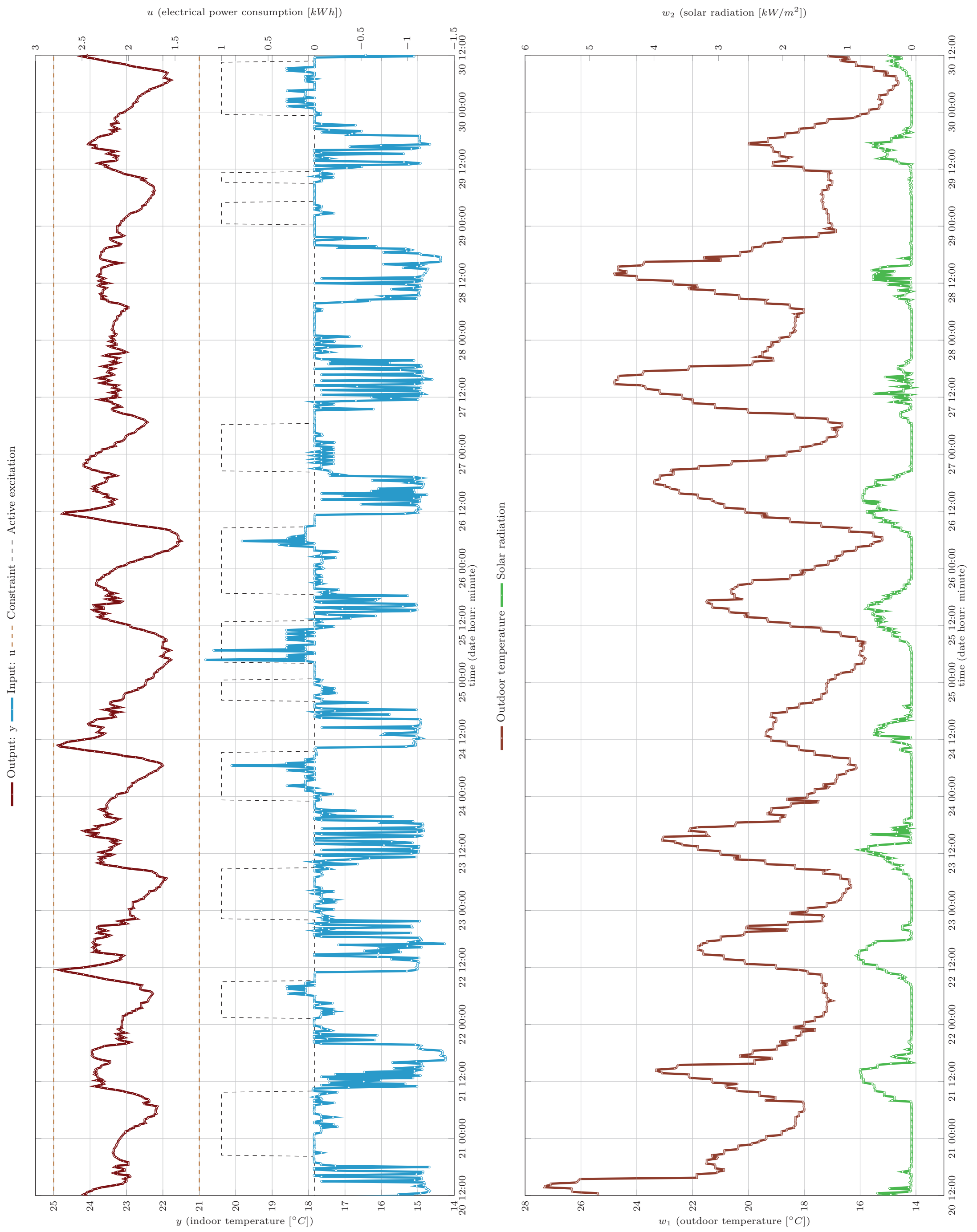


Figure 1.11: Experimental results from noon 20^{th} June 2021 to noon 30^{th} June 2021. Left:indoor temperature and electrical power consumption, and the black line indicates the time interval within which the active extication is active: high level: active, low level: inactive. Right: outdoor temperature and solar radiation

Experiment 4

Finally, we compare the proposed robust bi-level DeePC scheme with the default controller (Section. 1.3.4), which regulates the supply air temperature based on the return air temperature. A fixed setpoint at 22°C is given to the default controller in order to ensure the occupants' comfort throughout the day. It is worth noting that the default controller is a benchmark controller widely used in the building control community [92–94].

The experiment with the proposed controller was conducted from 23^{rd} July 2021 to 26^{th} July 2021, and the one with the default controller ran from 7^{th} July 2021 to 10^{th} July 2021. To ensure a fair comparison, the weather conditions for these two experiments were similar, as shown in Figure 1.12. The indoor temperature and electrical power consumption are plotted in Figure 1.12, alongside the statistics of these two experiments summarized in Table 1.4.

From Table 1.4 and Figure 1.12, more constraint violation is observed from the default controller than the proposed controller. One major underlying reason is that the default controller runs without the knowledge of a weather forecast and only cooled down the building when the return air temperature reached 23°C. We believe that is the reason accounting for the constraint violation in the day-1 experiment of the default controller. If the default controller could have predicted a high solar radiation and turned the cooling on constantly, the temperature constraint violation should have been avoided.

Besides the benefit of robust constraint satisfaction, the proposed controller is also more power efficient than the default controller under similar weather conditions, which in particular consumed 18.4% less electricity than the default controller (Table 1.4). Note that, to maintain the data quality, the proposed controller ran the active excitation scheme (Algorithm 1) regularly from 00:00 to 9:00. Thus, there is still the possibility to further improve the energy efficiency of the proposed controller.

	average indoor temperature	average outdoor temperature	average solar radiation	duration of constraint violation
Proposed controller	23.1°C	18.5°C	$0.17W/m^2$	$0h$
Default controller	23.2°C	18.8°C	$0.17W/m^2$	$5.75h$
	min/max indoor temperature	min/max outdoor temperature	min/max solar radiation	averaged hourly energy comsumption
Proposed controller	21.5°C 24.9°C	15.2°C 23.4°C	$0W/m^2$ $0.80W/m^2$	$1.15kWh$
Default controller	21.2°C 25.9°C	15.5°C 24.1°C	$0W/m^2$ $0.92W/m^2$	$1.41kWh$

Table 1.4: Statistics of the four-day comparison

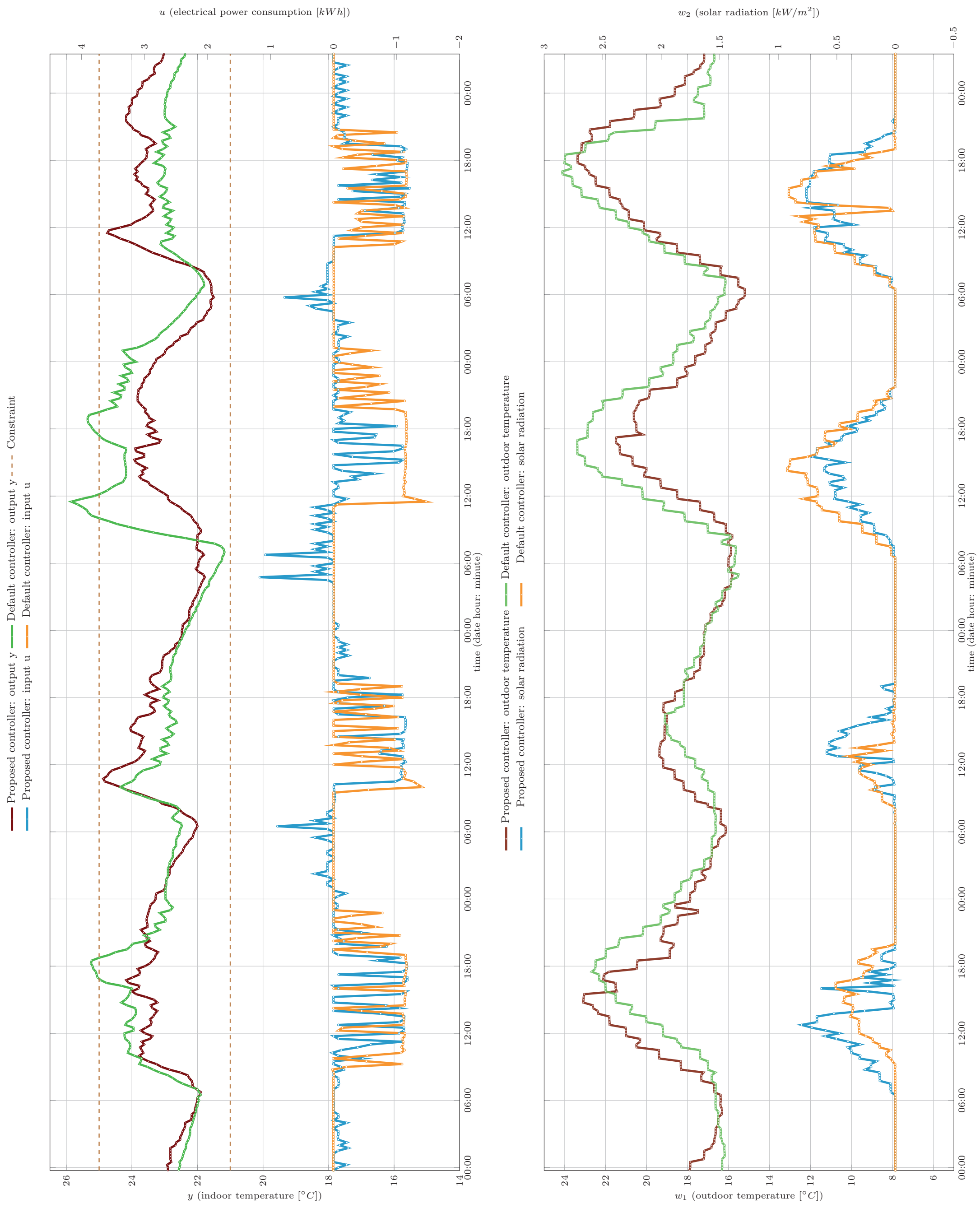


Figure 1.12: Comparison of the proposed robust controller and default controller. Left: indoor temperature and electrical power consumption. Right: outdoor temperature and solar radiation.

1.4 Final Remark

In this chapter, we explore the robustness aspect of direct data-driven controllers that are based on Willems' fundamental lemma. While data-driven control falls under the broader umbrella of control design, its key philosophy remains the characterization of system response. As a trajectory predictor, this characterization is integrated into the predictive control framework. Therefore, from my perspective, data-driven control is not entirely model-free, as there is still an implicit "model" used inside the controller. For the controllers studied in this chapter, this "model" is the KKT system that appears in bi-level scheme or the pseudo-inverse of the Hankel matrix in the single-level scheme.

The crucial difference between data-driven control and model-based control lies in whether it requires an explicit modelling/system identification procedure. In data-driven control, its "model" is fully characterized by data and is implicitly encapsulated as a functional module into the controller, thereby minimizing human effort and expert knowledge involved in understanding the targeted system. Thus, we can make two statements. First, since there is still a "model," data-driven control cannot outperform model-based control. Second, the primary goal of data-driven control is to minimize human involvement, mainly by automating the system identification procedure. Once the hypothesis model class is fixed, a data-driven control scheme should make minimal assumptions so that the user can tailor their controller based only on their preference without extra concern about the system responses.

It is worth mentioning that the feedback parametrization used in the single-level approach (Section 1.2) can also be used in the bi-level setup. However, since it depends on the QR decomposition of the numerically ill-conditioned Hankel matrices, it is numerically less robust than the feedback control investigated in Section 1.3. For completeness, the implicit feedback law and its numerical reformulation are included in the appendix.

2 Nonlinear Prediction and Control

Motivated by the success of linear direct data-driven control presented in the previous chapter, this chapter aims to extend those results to nonlinear systems. Rather than using the standard linearization approach, this chapter employs the idea of lifting. Lifting-based methods reframe a nonlinear problem as a linear one through nonlinear transformations. Therefore, the key focus of this chapter is to learn and characterize these nonlinear transforms. We will explore two different tracks: Koopman operator-based heuristics and reproducing kernel Hilbert space (RKHS) based lifting. These two approaches share a similar mathematical concept by viewing the evolution of a dynamical system as an evolution of the function evaluation. While the first approach directly models the dynamics of the function evaluation, the second approach views the states and inputs as a functional.

Outline

Due to the underlying difference between the approaches, we split this chapter into two main sections, whose background knowledge are accordingly allocated at the beginning of the corresponding sections (Section 2.1.1 and Section 2.2.1). In Section 2.1, a parametric algorithm that learns a finite dimensional representation of the Koopman operator is first summarized in Section 2.1.2. An optimization viewpoint of this algorithm is first introduced in Section 2.1.2 to give a more intuitive understanding. In the following Section 2.1.2, the mathematical reasoning behind this parametric scheme is studied via the scope of stochastic processes. After the introduction of a parametric algorithm, its semi-parametric version based on the Willems' fundamental lemma is investigated in Section 2.1.3. This is our first heuristic but natural extension of the Willems' fundamental lemma. Derived from a dual viewpoint, an RKHS extension of the Willems' fundamental lemma is investigated in Section 2.2.

2.1 Koopman Operator based Extensions

The Koopman operator enables the evolution of a nonlinear dynamic system to be represented via linear dynamics, albeit of an infinite dimension. It has been used extensively for the analysis of complex dynamic systems in fields such as fluid mechanics [95] and molecular physics [96] and in recent years it has been proposed as an approach that allows for some linear design tools to be used for nonlinear controller design [20, 97–100].

Central to this interest is the Dynamic Mode Decomposition (DMD) [101] algorithm and its extensions (e.g. [102, 103]), which are data-driven methods that allow the identification of the Koopman operator from data. Within DMD and its extensions, a *lifting* (or embedding), in which a dictionary of feature maps the original state to a higher dimensional feature space, uniquely characterizes the finite dimensional representation of a Koopman operator. This dictionary of feature maps is normally assumed as a-priori knowledge, however, as the dimension of the system increases, the number of feature maps that need to be considered tends to increase exponentially. Even though there are methods which learn the feature maps, e.g., [104–106], the learned maps are normally less interpretable. In this part, we will study an inverse parametric learning algorithm, whose learnt feature maps are the posterior mean of a Gaussian process.

2.1.1 Preliminary

Koopman Operator

We start by considering an autonomous, discrete-time nonlinear dynamical system

$$x^+ = F(x) \tag{2.1}$$

where $F : \mathcal{M} \to \mathcal{M}$ is the system update equation and $\mathcal{M} \subseteq \mathbb{R}^{n_x}$ is the state space. Our goal in this section is to demonstrate how the Koopman operator can be used to develop an infinite-dimensional linear dynamic system that can be used to compute the state evolution of this system.

Given a Hilbert space of smooth function $\mathscr{F} := \{g | g : \mathcal{M} \to \mathbb{R} \in C^\infty\}$ endowed with inner product $\langle \cdot, \cdot \rangle$, coined 'observables', the Koopman operator [18, 107] applied to the observable $g \in \mathscr{F}$ is defined as

$$\mathscr{K} g = g \circ F$$

The Koopman operator defines a new dynamical system in the function space $\mathscr{F}$ that governs the evolution of the observables. As the Koopman operator is an operator on a function space, $\mathscr{K}$ is in general infinite-dimensional, but critically it is *linear* even when the dynamics F are non-linear. By definition, the Koopman operator is a composite operator, and can be viewed as a shift operator as well. Regarding the fact of that the spectrum of a shift operator lie within a unit-circle and has been well-investigated [108], spectrum analysis is central to the research of Koopman operator in control engineering. In particular, we call an observable $\phi \in \mathscr{F}$ an eigenfunction associated with the eigenvalue $\lambda \in \mathbb{C}$ if $\mathscr{K}\phi = \lambda\phi$. From this, we can see that the eigenfunctions (or the linear combinations of the eigenfunctions) evolve linearly along the trajectories of our nonlinear system (2.1)

$$\phi(x^+) = \phi(F(x)) = \mathscr{K}\phi(x) = \lambda\phi(x) \ \ . \tag{2.2}$$

This leads to the key idea behind the Koopman operator: instead of tracking the state x of our system, we track the evolution of a set of observables f along the state trajectories. Specifically, given an observable $g \in span\left(\{\phi_k\}\right)$ with $g = \sum c_k(g)\phi_k$, where the weights $\{c_k(g)\}$ are called the Koopman modes of f, we notice that the evaluation of the observable at the current $g(x)$ is a *linear* function of the Koopman eigenfunctions evaluated at the current state

$$f(x) = \sum c_k(f)\phi_k(x) \ .$$

Its function evaluation at x^+ is therefore:

$$f(x^+) = \sum \lambda_k c_k(f)\phi_k(x) \ . \tag{2.3}$$

Such evaluation motivates the dynamic mode decomposition (DMD) algorithm as well as the extended DMD (EDMD) algorithm. We postpone the discussion about EDMD to Section 2.1.2. In the rest of this chapter, we do not distinguish eigenfunctions and their linear combination, as we only need the linear space spanned by them to determine the Koopman operator. With a bit of abuse of notation, a function within this space will be denoted by $\phi(\cdot)$.

Similar to the setup in discrete time dynamics, when a continuous time dynamical system:

$$\dot{x} = f(x)$$

is considered with $f(x) : \mathscr{M} \to T_x\mathscr{M}$, the Koopman operator is defined by:

$$\mathscr{K}^t g = g \circ \rho(x, t) \ ,$$

where $\rho(x, t)$ denotes the flow at time t starting from x at time 0. It is worth mentioning that the Koopman operator for continuous system is not necessarily densely defined [109], this is typically the case for systems that are not forward complete [110]. We refer the interested reader to [111] for more details, and we assume that the continuous Koopman operator exists and compact, especially in Section 2.1.2.

Subspace Identification

Subspace identification is a class of system identification methods that identify linear state space models of the form:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + w_k \\ y_k &= Cx_k + Du_k + v_k \ , \end{aligned} \tag{2.4}$$

Subspace identification is a class of system identification methods that identify linear state space models of the form: where $y_k \in \mathbb{R}^{n_y}$, $x_k \in \mathbb{R}^{n_x}$, $u_k \in \mathbb{R}^{n_u}$, $w_k \in \mathbb{R}^{n_x}$ and $v_k \in \mathbb{R}^{n_y}$ are the system outputs, states, control inputs, system noise and measurement noise. In order to stay close to the main topic of this thesis, the interested readers are referred to [112, 113] for an overview of subspace identification techniques. Most subspace identification methods fall in the unifying framework proposed by Van Overschee and De Moor [114]. In general, these algorithms obtain the Kalman filter states from input-output data, and then the state space model is computed by optimization with respect to the special structure in the filtering matrices. The key component of most of the subspace identification methods is the estimation of the extended observability matrix:

$$\Gamma_L = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{L-1} \end{bmatrix}. \tag{2.5}$$

In Section 2.1.2, the sequential PARSIM algorithm [115] is used.

Gaussian Processes

Gaussian process (GP) models are constructed from classical statistical models by replacing latent functions of parametric form (e.g. linear functions, truncated Fourier or Wavelet expansions, multi-layer perceptrons) by random processes with Gaussian prior. A Gaussian Process $\mathcal{GP}(\mu, k)$ is an infinite-dimensional distribution over the space of smooth real-valued functions $g : \mathbb{R}^N \to \mathbb{R}$, specialized by a priori mean $\mu : \mathbb{R}^N \to \mathbb{R}$ and covariance functions $k : \mathbb{R}^{N\times N} \to \mathbb{R}$ ([116]). $k(\cdot,\cdot)$ is called the kernel function, and it is stationary when $k(x, y)$ is a function of $x - y$. When a kernel is stationary, the underlying stochastic process is invariant to translation.

By definition, a Gaussian process is a second-order stochastic process whose function evaluation at a finite set of its index set $[x_1, x_2, x_3 \dots, x_n]$ follows a multi-variate Gaussian distribution $\mathcal{N}(\mu_X, K_{XX})$, where $\mu_X = [\mu(x_1), \dots, \mu(x_n)]^T$ and $K_{XX} = [k(x_i, x_j)]_{i,j=1}^n$. In general, $K_{A,B}$ denotes the cross-covariance between set A and B. If the measurement is contaminated by Gaussian observation noise, $p(y(x)|g(x)) \sim \mathcal{N}(g(x), \sigma^2)$ with σ^2 as measurement noise variance, then the predictive distribution at any point $x^* \in \mathbb{R}^N$ given data $\mathscr{D} = \{x_i, y_i\}_{i=1}^n$ is

$$\begin{aligned} p(g(x^*)|\mathscr{D}) &\sim \mathscr{GP}(\mu_{g|\mathscr{D}}(x^*), k_{g|\mathscr{D}}(x^*, x^*)) \\ \mu_{g|\mathscr{D}}(x^*) &= \mu(x^*) + K_{x^*X}\hat{K}_{XX}^{-1}y \\ k_{g|\mathscr{D}}(x^*, x^*) &= K_{x^*x^*} - K_{x^*X}\hat{K}_{XX}^{-1}K_{x^*X}^T \end{aligned} \tag{2.6}$$

where $\hat{K}_{XX} = K_{XX} + \sigma^2 I$ and $y = [y_1, y_2, \ldots, y_n]^T$. This defines a scalar-valued regression, for vector-valued regression, various methods have been proposed, such as in [117–119]. For simplicity, we will do vector-valued regression via a scalar-valued regression in each dimension in this thesis.

Gaussian process is a non-parametric statistical model, and it is widely used in controller design. Particularly, it is used to model any unknown ingredients within a dynamical system. This unknown ingredient can be the nonlinear part of a system model [120], or the response of a chemical process [121]. Interested readers are referred to [7] for GP based control applications.

Differential Parametric Quadratic Programming

Sensitivity analysis investigates the continuity of a parametric optimization problem. In most cases, it is only able to guarantee outer continuity of the optimal solution set and upper semi-continuity of the objective function [122]. Under a strong assumption of linear constraint qualification and second order sufficient condition, the implicit function theorem [123] can be applied to the KKT system to derive the partial derivative between the optimal value function and the parameter [124]. This idea has been applied to deep learning [125] and reinforcement learning [126]. It is noteworthy that the optimal solution set is not differentiable even for linear conic programming [127]. On the contrary, the optimal value function is smoother [128], and hence more frequently used in differential programming. In this thesis, we only apply the result in parametric convex programming, whose optimal value function is lower semi-continuous [129, Theorem 1.17]. When a general nonlinear parametric programming is considered, the directional derivative of its optimal value function exists [130, Thoerem 1]. Furthermore, this function is lower semicontinuous under the assumption of uniformly boundedness [131, Thoerem 5].

In this chapter, we consider the parametric quadratic programming(QP), the interested reader can refer to [132] for the case of linear conic programming. We use subscript $_q$ to denote the parameter used in QP and to avoid confusion. Consider a parametric QP, $\mathscr{Q}(e_q) := e_q \to z_q^*$ with parameters $\{Q_q \in \mathbb{S}^+, , q_q\ H_q,\ h_q,\ E_q\}$:

$$\begin{aligned} \min_{z_q}\ & \frac{1}{2} z_q^T Q_q z_q + q_q^T z \\ \text{s.t. } & H_q z_q \leqslant h_q, E_q z_q = e_q \end{aligned} \tag{2.7}$$

The KKT conditions for this QP are:

$$
\begin{aligned}
Q_q z^* + q_q + H_q^T \lambda^* + E_q^T \nu^* &= 0 \\
\operatorname{diag}\left(\lambda^*\right)(H_q z^* - h_q) &= 0 \\
E_q z_q^* - e_q &= 0
\end{aligned}
\tag{2.8}
$$

where z_q^*, ν^*, λ^* are the optimal primal and dual variables. Then the differentials of the KKT conditions can be computed by:

$$
\begin{aligned}
&\begin{bmatrix} Q_q & H_q^T & E_q^T \\ \operatorname{diag}(\lambda^*)H_q & \operatorname{diag}(H_q z^* - h_q) & 0 \\ E_q & 0 & 0 \end{bmatrix} \begin{bmatrix} dz_q \\ d\lambda \\ d\nu \end{bmatrix} \\
&= - \begin{bmatrix} dQ_q z_q^* + dq_q + dH_q \lambda^* + dE_q^T \nu^* \\ \operatorname{diag}(\lambda^*) dH_q z_q^* - \operatorname{diag}(\lambda^*) db \\ dE_q z^* - de_q \end{bmatrix}
\end{aligned}
\tag{2.9}
$$

The derivatives of z^* with respect to the parameters $(Q_q, q_q, H_q, h_q, E_q)$ and the function input f are given by the solution to the linear system defined in Equation (2.9). For example, the solution dz of (2.9) gives the result of $\frac{\partial z_q^*}{\partial Q_q}$ if we set $dQ_q = I$ and the differentials of other parameters to 0. The gradient of optimal value $L(z^*)$ with respect to Q is calculated accordingly by chain-rule as $\frac{\partial L(z_q^*)}{\partial z_q^*} \frac{\partial z_q^*}{\partial Q_q}$.

2.1.2 Parametric Approach

Regarding the dynamical equation defined by the eigenfunctions (2.3), choosing the function space spanned by $\{\phi_k(\cdot)\}_k$ is central to the spectrum analysis in Koopman operator theory. When finite dimensional approximation of the Koopman operator is considered, , we assume that there is a finite number of eigenfunctions, written in matrix form as $\Phi = \begin{bmatrix} \phi_1 & \phi_2 & \dots \end{bmatrix}^T$. As the problem is now finite, we can write the Koopman operator as a matrix $\mathcal{K} = K$ and the evolution of the eigenfunctions along the system trajectory as

$$\Phi(x^+) = K\Phi(x)$$

The observables, or performance outputs, are then a linear function of $\Phi(x)$

$$g(x) = C\Phi(x)$$

where $g := \begin{bmatrix} g_1 & g_2 & \dots \end{bmatrix}^T$ and C is a matrix of Koopman modes. If the eigenfunctions are assumed to lie within a linear space spanned by a dictionary of basis functions, the learning of K and C reduces to linear regression, which is termed the EDMD algorithm. However, when a-priori knowledge is not available, selection of the basis functions becomes non-trivial. In this part, without using the basis functions, we will instead learn the space spanned by eigenfunctions directly.

Setting the stage

We define a finite set of observables, called the ‘performance outputs’, $\{g_i\} \subset \mathscr{F}$, which define the outputs of interest for the targeted dynamical system. Meanwhile, a sequence of state/output measurement $\mathscr{D} = \{x_i, y_i := g(x_i)\}$ is available for this learning procedure. Note that there is no need to have the explicit form of functions g, a set of its function evaluation at $\{x_i\}_i$ is sufficient for this learning procedure. Therefore, they can be some sensor readouts or some user-defined nonlinear/linear mappings evaluated at $\{x_i\}$.

In the following two sections, we are going to derive the following algorithm:

Algorithm 2 Inverse Parametric Learning of Koopman Operator

```
Input:$\mathscr{D} = \{x_i, y_i := g(x_i)\}$, approximation order $j$
Output: $K, \{\phi_k\}$
Solve SVD of the Hankel matrix $\mathfrak{H}_L(y) = U\Lambda V := U\Phi$.
$K = U^{\dagger}_{1:end-n_y,1:j} U_{n_y+1:end,1:j}$, $C = u_{1:n_y,1:j}$
for k = 1:j do
    $\phi_k = \arg\min_{g_k(\cdot)\in\mathscr{F}} \sum_i \|\Phi_{k,i} - g_k(x_i)\|$
end for
```

Before delving into its development, the most interesting point of this algorithm is that the finite approximation K and its Koopman mode C are first learnt before the recovery of the lifting/feature maps. Therefore, we term this algorithm inverse parametric learning.

Optimization Viewpoint

The learning problem of the finite dimensional approximation of a Koopman operator can be posed nonlinear least square regression:

$$\begin{aligned} \min_{K,C,\Phi} \sum_i \|y_i - C\Phi(x_i)\|^2 + \|\Phi(x_i) - K\Phi(x_{i-1})\|^2 \\ \text{s.t. } \Phi \in \mathscr{F} \end{aligned} \tag{2.10}$$

If the eigenfunctions Φ are assumed to be in the span of a finite dictionary of known basis functions, then (2.10) reduces to a convex least square regression problem. This is the extended dynamic mode decomposition (EDMD) [102][1] approach, which has been shown to be effective if a good dictionary of functions Φ is available. However, once the dictionary of feature maps are fixed, the Koopman mode and Koopman operator are learned separately, the algorithm 2

[1] If the basis functions are identity functions of measurements sequence preceding the current time step, then a DMD approach is recovered.

shows that all three can be learned from data. We first state our assumption:

Assumption 3. *There is a class of universal approximator* $\mathscr{U} \subset \mathscr{F}$ *such that for* $\forall\ g \in \mathscr{F}$ *and* $\forall \epsilon > 0$, $\exists\, \Phi \in \mathscr{U}$ *such that* $||g - \Phi||_\infty < \epsilon$.

Assumption 3 implies that $\mathscr{U}$ is dense in $\mathscr{F}$. We note that Gaussian process regression and infinitely-wide neural networks are examples of such classes of functions. By this assumption, optimization problem (2.10) is equivalent to the following optimization problem almost everywhere

$$\begin{aligned} \min_{K,V,\{\hat{x}\},\Phi} \sum_i \|y_i - C\hat{x}_i\|^2 + \|\hat{x}_i - K\hat{x}_{i-1}\|^2 \\ \text{s.t. } \hat{x}_i = \Phi(x_i) \qquad \text{for all } x_i \\ \Phi \in \mathscr{U} \end{aligned} \tag{2.11}$$

This optimization problem admits a relaxation[2] as:

$$\begin{aligned} \min_{K,V,\{\hat{x}\},\Phi\in\mathscr{U}} \sum_i \overbrace{||y_i - C\hat{x}_i||^2 + ||\hat{x}_i - K\hat{x}_{i-1}||^2}^{\mathbf{1}} \\ + \overbrace{||\hat{x}_i - \Phi(x_i)||^2}^{\mathbf{2}} \end{aligned} \tag{2.12}$$

As indicated in Assumption 3, the second component of this optimization problem admits an arbitrarily small error and therefore this relaxation is tight. Hence, the optimization problems can now be separated into two problems, which can then be solved accordingly.

Problem 1 is solved first,

$$\min_{K,V,\{\hat{x}\}} \sum_i ||y_i - C\hat{x}_i||^2 + ||\hat{x}_i - K\hat{x}_{i-1}||^2 \tag{2.13}$$

which produces a lifted state sequence $\hat{x}_i$, as well as the matrices defining the linear dynamics K and C.

Problem 2 is then solved in order to find a mapping from the true system state x_i to the lifted system state $\hat{x}_i$ at the sampled points. The optimization is a standard regression problem.

$$\min_{\Phi\in\mathscr{U}} \sum_i \|\hat{x}_i - \Phi(x_i)\|^2 \tag{2.14}$$

In general, the key of this separation method is that regardless of the optimal state sequence generated in Problem (2.13), the optimal value of Problem (2.14) will be arbitrarily close to zero as a result of Assumption 3. As a result, the loss contributed by sub-optimization Prob-

[2] Relaxation means that every feasible solution in optimization (2.12) admits a feasible solution in optimization (2.11) by enforcing $\hat{x}_i = \Phi(x_i)$

lem (2.13) dominates the loss from sub-optimization Problem (2.14), such that Problem (2.14) is negligible in the original optimization Problem (2.12). Therefore, without loss of generality, we are able to apply any flexible regression algorithms that meet this requirement. In particular, the sub-optimization Problem (2.14) will be solved by Gaussian process regression in this thesis[3].

Remark 10. *It is worthwhile noting that, when the dictionary of feature maps coincide with $\mathscr{U}$, the optimal solution map from data to the $\{\Phi, K, C\}$ generated by EDMD is isometric to the solution map generated by the proposed method. The proposed algorithm is still advantageous over the EDMD method with less a-priori knowledge needed, since this algorithm only requires the existence of this function space without specifying its explicit structure.*

Problem (2.13) is equivalent to minimizing the mean squared error of the noise terms w_k and v_k in the following model

$$\begin{aligned} \hat{x}_{k+1} &= K\hat{x}_k + w_k \\ y_k &= C\hat{x}_k + v_k \end{aligned} , \tag{2.15}$$

which indicates that we can apply a subspace identification method to identify the corresponding matrices K, C and the states $\{\hat{x}\}$. Regarding an autonomous system, the subspace identification algorithm reduces to SVD, which recovers the first two steps in the inverse parametric learning algorithm 2. Even though the use of subspace identification naturally gives rise to an extension to system with control input, we postpone this discussion to Section 2.1.2 for the sake of clarity.

Remark 11. *In general, the learned feature maps are not necessarily the eigenfunctions, however, by diagonalization of the matrix K, we can recover the eigenfunctions, eigenvalues and the corresponding Koopman modes of the performance outputs [133]. Moreover, it is noteworthy that the state x is a very general concept, one could use previous control inputs or even the value function of an MPC controller to compose the states.*

Connection Between Koopman Operator and Observability

Before we derive the mathematical rationale behind the inverse parametric learning algorithm, we would take a detour first to develop a better understanding about the Koopman based learning problem. Firstly, the inverse parametric learning algorithm implies a strong connection between the Koopman operator and observability. To see this, we pose the following question:

Why do we learn these eigenfunctions but not the others?

[3] Even though exact interpolation is guaranteed when the measurement noise is assumed to be zero, overfit usually deteriorate the generalization performance of the learnt model. Therefore, a non-zero measurement noise is used in our implementation.

The eigenfunctions of the Koopman operator enjoy a special algebraic structure, in particular, the eigenfunctions that vanish nowhere form an Abelian group [133]. Specifically, if ϕ_1, ϕ_2 are eigenfunctions with eigenvalues λ_1, λ_2 respectively, then $\phi_1\phi_2$ is also an eigenfunction with eigenvalue $\lambda_1\lambda_2$. Based on this observation, the eigenfunctions related to an unstable mode or to a slow dynamic mode should dominate the learning problem, which is also the reason why Dynamic Mode Decomposition (DMD) tends to be numerically ill-conditioned. In order to learn the family of eigenfunctions, [134] proposed to construct a large family of possible eigenfunctions. However, we argue that instead of learning all the possible observable dynamics, we should select the eigenfunctions that are most informative with respect to 'performance outputs', because those output are of special interest. For example, if we control the temperature of a room, then we should learn the eigenfunctions that could properly reproduce the temperature dynamics but not the others. We thereby propose to use a well-developed concept in control science, observability, to quantify/regularize and to select the possible eigenfunctions.

For the sake of simplicity, we elaborate on this idea with a discrete-time, autonomous dynamical system. Assume that $\{\phi_k\}_{k=1}^{K}$ is the set of eigenfunctions with corresponding eigenvalues $\{\lambda_k\}$. Let f denote an observable that we focus on, and let its Koopman mode corresponding to ϕ_k be v_k, then

$$\mathcal{K}^i g = \hat{C}\Lambda^i \Phi(x), \tag{2.16}$$

where $\hat{C} = [c_1, c_2, \ldots, c_K]$ and $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_K)$. Given a sequence of N_i evaluations $y_i = f(x_i)$ of the observable f along the state sequence $\{x_i\}$, we have

$$\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{N_i} \end{bmatrix} = \hat{\Gamma}\Phi(x_0) = \begin{bmatrix} \hat{C}\Lambda^0 \\ \hat{C}\Lambda^1 \\ \vdots \\ \hat{C}\Lambda^{N_i-1} \end{bmatrix} \Phi(x_0). \tag{2.17}$$

Since the value of the eigenfunctions are fixed for a fixed x_0, the dynamics of the observable is determined by $\hat{\Gamma}$. This matrix leads to two observations:

1. The matrix $\hat{\Gamma}$ is exactly an extended observability matrix appearing in equation (2.5).
2. The dynamics of an observable depends both on the Koopman mode and the eigenvalues. Even though an eigenfunction may have a large eigenvalue, if it has a small Koopman mode with respect to the observable g (*i.e.* this eigenfunction has lower observability in g), then the dynamics corresponding to this eigenfunction is less relevant to the outputs that we would like to predict.

The observations above suggest an answer to the question we posed at the beginning of this section. Because the quality of the reconstruction of the performance outputs are considered in the original optimization problem (2.10), the eigenvalues learned from the proposed algorithms reflect a balance between state evolution and performance outputs reconstruction.

The key piece of a subspace identification method is exactly the estimation of the extended observability matrix, from which the most observable components are extracted by singular value decomposition (SVD).

Stochastic Process Viewpoint

Even though it is intuitive to develop the proposed algorithm 2 via the optimization scope, the mathematical object that guarantees the functionality of the proposed algorithm becomes unclear. In this part, we will justify the mathematical reasoning behind the proposed algorithm 2 via the scope of stochastic process. Meanwhile, we are able to quantify model uncertainty of the learnt model. Unlike the common concept of uncertainty/noise encountered in control science, the model uncertainty in this part is caused by our limited knowledge of an unknown system and by the use of a finite amount of data, even though the system can be described by deterministic dynamic equations. Problems in the same vein are broadly investigated in Bayesian learning [135], and find various applications in control science such as safe learning (see e.g. [136]).

In this part, we will establish the theory in which a Koopman operator evolves a distribution of observables instead of a specific observable. We first make the same assumption as in [137].

Assumption 4. *The state space $\mathscr{M} \subset \mathbb{R}^n$ is compact and the Koopman operator is therefore also compact.*

Similar to other GP based methods, we further assume that

Assumption 5. *The underlying* ***continuous time*** *dynamics f is deterministic.*

Following the Bayesian learning procedure, we assume an observable has an a priori distribution $g \sim \mathscr{GP}(\mu, k)$.

Corollary 2. *If an observable $g \sim \mathscr{GP}(\mu, k)$, then the Koopman operator applied to g is also a Gaussian process such that $\mathscr{K}^t g = \mathscr{GP}(\mathscr{K}^t \mu, \mathscr{K}^t k(\cdot, \cdot))$*

Proof. The Koopman operator is a linear operator. As a Gaussian process is closed under linear operators [116, 138], $\mathscr{K}^t g$ is a Gaussian process. Since a Gaussian process is fully characterized by its second order statistics [139], its mean and kernel functions can be calculated as follows:

$$\begin{aligned}\mathbb{E}(\mathscr{K}^t g(x)) &= \mathbb{E}_g(g(\rho(x,t))) \\ &= \int_{-\infty}^{+\infty} f(\rho(x,t)) p(g(\rho(x,t)) = \xi)) d\xi \\ &= \mu(\rho(x,t)) = \mathscr{K}^t \mu(x),\end{aligned} \tag{2.18}$$

$$\begin{aligned}
&\mathbb{E}((\mathcal{K}^t g(x) - \mathbb{E}(\mathcal{K}^t g(x)))(\mathcal{K}_g^t(y) - \mathbb{E}(\mathcal{K}^t g(y)))) \\
&= \mathbb{E}_g((g(\rho(x,t)) - \mu(\rho(x,t)))(g(\rho(y,t)) - \mu(\rho(y,t)))) \\
&= k(\rho(x,t), \rho(y,t)) \qquad (2.19) \\
&= \langle k(\rho(x,t),\cdot), k(\cdot, \rho(x,t)) \rangle \\
&= \langle \mathcal{K}^t k(x,\cdot), \mathcal{K}^t k(\cdot, x) \rangle =: \mathcal{K}^t k(x,y)
\end{aligned}$$

where $\mathbb{E}_g$ denotes expectation with respect to g. The first equality in Equation 2.18 and 2.19 follows Assumption 5. By linearity of the Koopman operator, $\mathcal{K}^t k(\cdot,\cdot)$ is positive definite and therefore a kernel function[4]. ■

Corollary 2 leads to the following corollary:

Corollary 3. *If an observable $g \sim \mathcal{GP}(\mu, k)$, then the trajectory of this observable $\{g(\rho(x,t))\}$ is a Gaussian process.*

The proof of Corollary 3 is simple because $\{f(\rho(x,t)\}$ fulfils the definition of a Gaussian process. More specifically, any finite snapshots of a $\mathcal{GP}$ observable trajectory forms a Gaussian random vector. In the rest of this paper, we denote $f(\rho(x,t))$ as $f(t)$ and $\rho(x,t)$ as $x(t)$ for the sake of compactness. We refer to $f \sim \mathcal{GP}(\mu, k)$ as a $\mathcal{GP}$ observable and $\{f(t)\}$ as the trajectory of the $\mathcal{GP}$ observable.

Up to this point, we have shown that, when an uncertain observable modelled by a Gaussian process, the trajectory of measurement/observation generated by this observable remains a Gaussian process not only in its coordinate (i.e. index set) defined by $\rho(\cdot,t)$ but also in the direction of time t. As the coordinate defined by $\rho(\cdot,t)$ is not necessarily Cartesian, the Koopman operator actually warps the state space $\mathcal{M}$ by diffeomorphism to a new manifold on which the observable remains Gaussian. In what follows, we will show that such warping (i.e. diffeomorphism) can be approximated by a finite dimensional linear transformation. To justify this learning process, we assume:

Assumption 6. *The trajectory of a $\mathcal{GP}$ observable is stationary.*

Even though we have established our theory based on a continuous time setup, the available data only have finite snapshots along this trajectory. In particular, we assume that the data $\mathcal{D} := \{x_, y_i := g(x_i)\}$ is sampled with a fixed sampling time T_s. The optimality of the inverse parametric learning algorithm 2 is summarized as:

Lemma 5. *Given the singular value decomposition (**SVD**) of $\mathfrak{H}_L(y) = U\Lambda V$ with columns of U and rows of V ordered by decreasing singular value, the optimal finite order approximation of*

[4] This notation is not rigour, as it may be confused with the commutation between the bilinear form and the expectation operator. However, as the idea is clear from the context, we keep this notation to avoid the introduction of the Hilbert-Schmidt operator [140, p.177].

the Koopman operator $\mathcal{K}$ of order j in the mean-square-error sense is

$$K = U^{\dagger}_{1:end-n_y,1:j} U_{n_y+1:end,1:j}$$

The a posteriori distribution of g is

$$g \sim U[1:n_y, 1:j] \begin{bmatrix} \phi_1 & \phi_2 & \dots & \phi_j \end{bmatrix}^T,$$

where $\phi_k \sim \mathcal{GP}(\mu, k|\{\phi_k(x_i) = \Phi_{k,i}\})$ and $\Phi = \Lambda V$

As the size of the data set is finite, the time interval of the data is accordingly finite. To enable the proof of Lemma 5, we first review the Karhunen–Loève theorem, which is later used to decompose the trajectory of a $\mathcal{GP}$ observable.

Theorem 2. *(**Karhunen–Loève theorem** ([141]))*
A centred[5] stochastic process $\{S_t\}_{t\in[0,T_e]}$ admits a decomposition

$$S_t = \sum_{k=0}^{+\infty} \phi_k e_k(t),$$

where ϕ_k are pairwise uncorrelated random variables and $e_k(\cdot)$ are continuous real-valued orthonormal basis functions in $L^2[0, T_e]$. In the case of a zero-mean Gaussian process, ϕ_k are independent centred Gaussian random variables.

Due to Assumption 4, the system evolves linearly in infinite dimensional feature space within a compact set, zeros and centred limit cycles are therefore the only possible equilibrium points. Hence, a $\mathcal{GP}$ observable is centred and Theorem 2 is applicable to the trajectory of a $\mathcal{GP}$ observable. In the following, we show the proof of Lemma 5.

Proof. **proof of Lemma 5**
By Corollary 3 and Assumption 6, we can apply Theorem 2 to a trajectory of a $\mathcal{GP}$ observable

$$Y(t) = \sum_{k=0}^{\infty} \phi_k e_k(t). \tag{2.20}$$

Since X are the states of the system, X is the full statistic of the system dynamics. Hence, ϕ is $\sigma(X)$-measurable, and there exists a Lebesgue measurable function mapping X to ϕ [142]. We can therefore rewrite Equation (2.20) as

$$Y(t) = \sum_{k=0}^{\infty} \phi_k(X(0)) e_k(t).$$

[5] Meaning that the process is zero-mean.

Following Assumption 6, we get

$$Y(t) = \sum_{k=0}^{\infty} \phi_k(X(0)) e_k(t) = g(X(t)) \sim \mathcal{GP}(\mathcal{K}^t \mu, \mathcal{K}^t k) \ .$$

$Y(t)$ is a realization of the stochastic process $\mathcal{K}^t g$. By Corollary 2, $\mathcal{K}^t g$ is a Gaussian process, and therefore its decomposition $\phi_k(X)$ is a Gaussian process[6]. Applying the Koopman operator to $e_k(\cdot)$, we get

$$\begin{aligned} Y(t) &= \sum_{k=0}^{\infty} \phi_k(X(0)) e_k(t) \\ &= \sum_{k=0}^{\infty} \phi_k(X(0)) \mathcal{K}^t e_k(0), \end{aligned} \tag{2.21}$$

Substitute sampling time to (2.21), we get

$$\begin{aligned} y_i &= \sum_{k=0}^{\infty} \phi_k(x_i) e_k(iT_s) \\ &= \sum_{k=0}^{\infty} \phi_k(x_i) \mathcal{K}^{iT_s} e_k(0). \end{aligned} \tag{2.22}$$

For the sake of clarity, we define the power of discrete-time Koopman operator as

$$\tilde{\mathcal{K}}^n g = g(\underbrace{F \circ F \cdots \circ F}_{n}(\cdot)),$$

where F denotes the discretization of the continuous dynamics f with sampling time T_s.The Hankel matrix of measurements can be rewritten as a discrete-time system regarding Assump-

[6] The proof that an uncorrelated decomposition of a Gaussian process only consists of Gaussian processes can be found in [143, 144].

tion 6,

$$
\begin{aligned}
&\mathfrak{H}_L(y) = \\
&\begin{bmatrix}
\sum_{k=0}^{+\infty} \phi_k(x(T_1))e_k(0) & \dots & \sum_{k=0}^{+\infty} \phi_k(x(T_{n-m+1}))e_k(0) \\
\sum_{k=0}^{+\infty} \phi_k(x(T_1))e_k(T_s) & \dots & \sum_{k=0}^{+\infty} \phi_k(x(T_{n-m+1}))e_k(T_s) \\
\vdots & \ddots & \vdots \\
\sum_{k=0}^{+\infty} \phi_k(x(T_1))e_k(LT_s - T_s) & \dots & \sum_{k=0}^{+\infty} \phi_k(x(T_{n-m+1}))e_k(LT_s - T_s)
\end{bmatrix} \\
&= \begin{bmatrix}
\sum_{k=0}^{+\infty} \phi_k(x(T_1))e_k(0) & \dots & \sum_{k=0}^{+\infty} \phi_k(x(T_{n-m+1}))e_k(0) \\
\sum_{k=0}^{+\infty} \phi_k(x(T_1))\tilde{\mathscr{K}} e_k(0) & \dots & \sum_{k=0}^{+\infty} \phi_k(x(T_{n-m+1}))\tilde{\mathscr{K}} e_k(0) \\
\sum_{k=0}^{+\infty} \phi_k(x(T_1))\tilde{\mathscr{K}}^2 e_k(0) & \dots & \sum_{k=1}^{+\infty} \phi_k(x(T_{n-m+1}))\tilde{\mathscr{K}}^2 e_k(0) \\
\vdots & \ddots & \vdots \\
\sum_{k=0}^{+\infty} \phi_k \tilde{\mathscr{K}}^{L-1} e_k(0) & \dots & \sum_{k=0}^{+\infty} \phi_k(x(T_{n-m+1}))\tilde{\mathscr{K}}^{L-1} e_k(0)
\end{bmatrix} \\
&= \underbrace{\begin{bmatrix}
e_1(0) & e_2(0) & \dots \\
\tilde{\mathscr{K}} e_1(0) & \tilde{\mathscr{K}} e_2(0) & \dots \\
\vdots & \vdots & \vdots \\
\tilde{\mathscr{K}}^{L-1} e_1(0) & \tilde{\mathscr{K}}^{L-1} e_2(0) & \dots
\end{bmatrix}}_{E}
\underbrace{\begin{bmatrix}
\phi_1(x(T_1)) & \dots & \phi_1(x(T_{n-m+1})) \\
\phi_2(x(T_1)) & \dots & \phi_2(x(T_{n-m+1})) \\
\vdots & \vdots & \vdots
\end{bmatrix}}_{\Phi} .
\end{aligned}
$$

[145] showed that the numerical finite realization of Karhunen–Loève decomposition is equivalent to the singular value decomposition (**SVD**), $\mathfrak{H}_L(y) = U\Lambda V$, with $E = U$ and $\Phi = \Lambda V$. Meanwhile, if U and V is ordered by decreasing singular value, then the optimal approximation in the mean-square-error sense up to order j is spanned by the first j dominant singular values and their columns/rows in U/V. Therefore, the corresponding finite order approximation of Koopman operator is $K = U^{\dagger}_{1:end-n_y,1:j} U_{n_y+1:end,1:j}$. This is sufficient to conclude the proof. ■

To interpret the inverse parametric learning algorithm 2 in a more intuitive way, we first denote the i^{th} column of U as U_i. U_1 captures the most dominant dynamics presented in the column space of $\mathfrak{H}_L(y)$. In control science language, U_1 is the most observable mode in the extended observability matrix. However, as we only have access to finitely many trajectories, the lifting ϕ_1 which maps state X to this mode U_1 cannot be captured completely. An a-priori Gaussian process is therefore used to model a distribution of this lifting. In particular, $\Phi_{i,j}$ is the realizations of all the i^{th} most dominant $\mathscr{GP}$ observable in the j^{th} $\mathscr{GP}$ observable trajectory. The a priori Gaussian process is then refined to its posterior distribution, which is still a Gaussian process.

Extensions to Systems with Inputs

As discussed in Section 2.1.2, the correspondence between SVD and subspace identification hints a natural extension of the Koopman operator to systems with control inputs. When a discrete time dynamics $x^+ = F(x,u)$ is considered, the corresponding model is assumed to be:

$$\mathscr{K}_u^t g := g(f(\cdot)) + Bu\,,$$

where $B : \mathbb{R}^{n_u} \to \mathscr{H}$ is a linear operator. This is a heuristic used in [20], and has been shown to provide good performance in some specific models. It is vital to remember that, this extension to system with control inputs is merely a heuristic. First, this models implies the eigenvalue of the dynamics is invariant to the amplitude of the control input, which holds only when the system is linear. Secondly, even for a linear system, analysis/modelling of system behaviours based on its open-loop eigenfunction is more or less redundant for control engineering, as they can be varied by the closed-loop control. If this extension works, it suggests that the underlying dynamics can be well approximated by linear models. Beyond this extension, there are other extension such as [99, 146], and to the best of this author's knowledge, all these extensions ends up with a nonlinear dynamics, which negates the benefit/motivation behind the use of the Koopman operator. Hence, we stick to the natural input-affine extension in this thesis. With the aforementioned remark in mind, we summarize a finite dimensional approximation of this extension by:

$$\begin{aligned} \Phi(x_k) &= K\Phi(x_k) + Bu_k \\ y_k &= C\Phi_k + Du_k \ . \end{aligned} \tag{2.23}$$

To learn this model, the steps 1 and 2 in Algorithm 2 is replaced by a subspace identification routine. Meanwhile, to determine the order of the system, dubbed n_ϕ, one might choose different methods, such as cross validation [147] or complexity criteria [148]. In subspace identification, we are able to choose different weightings in Problem (2.13) to achieve an optimal realization up to order n_ϕ [149], which is also the optimal finite approximation of observable dynamics up to order n_ϕ. Following Section 2.1.2, uncertainty of the lifted initial state can be quantified by the posterior distribution of a Gaussian process regressor, which gives:

$$\begin{aligned} \Phi_0 &= \mathscr{GP}(\mu_{g|\mathscr{D}}(x), k_{g|\mathscr{D}}(x_0)) && \text{state space to observables} \\ \Phi_{k+1} &= K\Phi_k + Bu_k && \text{observables dynamics} \\ y_k &= C\Phi_k + Du_k && \text{observables to measurements.} \end{aligned}$$

As Φ_0 follows a Gaussian distribution, following stochastic MPC is tractable via convex programming,

$$\begin{aligned}\min_{u}\max_{w}\sum_{i=1}^{n_h} & y_i^T Q y_i + u_{i-1}^T R u_{i-1}\\ \text{s.t. } & \Phi_0 \sim \mathcal{GP}(\mu_{g|\mathcal{D}}(x), k_{g|\mathcal{D}}(x_0)\\ & \Phi(x_k) = K\Phi(x_k) + Bu_k\\ & y_k = C\Phi_k + Du_k\\ & P(y_i \in \mathcal{Y}) \geq 1-\epsilon, P(u_i \in \mathcal{U}) \geq 1-\epsilon, i = 0, 1 \ldots n_h,\end{aligned}$$

where Q and R are penalty matrices for outputs and control inputs, n_h is the prediction horizon. $\mathcal{U}, \mathcal{Y}$ denotes the feasible sets for control inputs and measurements separately. Even though w lies in feature space and cannot be measured, the moving horizon scheme of stochastic MPC only uses the open-loop first step. Hence, following standard causal feedback law in predictive control can be applied:

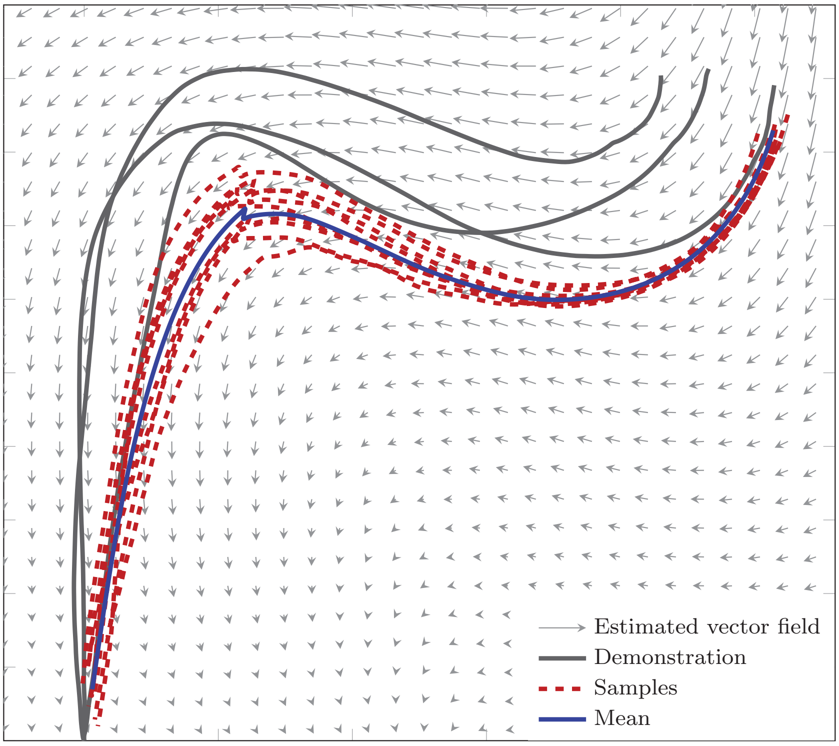
$$\begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{n_h-1} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \ldots & 0 \\ G_{11} & 0 & 0 & \ldots & 0 \\ G_{21} & G_{22} & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \\ G_{n_h 1} & G_{n_h 2} & G_{H3} & \ldots & 0 \end{bmatrix} (\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n_h-1} \end{bmatrix} - \begin{bmatrix} \bar{y}_1 \\ \bar{y}_2 \\ \vdots \\ \bar{y}_{n_h-1} \end{bmatrix}),$$

where $\bar{y}_i$ represents the nominal trajectory without uncertainty.

Numerical Results

Learning by demonstration enables robots to imitate human-level control by providing demonstrations ([150]). A popular method is proposed in [151], which is based on Gaussian mixture models (GMM). This method ensures global stability by enforcing a Lyapunov condition. However, this approach is not scalable with respect either to the dimension of the state space or to the amount of data.

In this validation test, the algorithm is applied to learn the dynamics of hand-written characters. We assume that drawing a character is governed by an autonomous, discrete-time dynamical system such that the location of the pen tip x evolves according to $x_{k+1} = F(x_k)$. We apply the proposed method to learn these dynamics. The effectiveness of the algorithm is shown in Figure.2.1 and Figure.2.2, whose data comes from [151] with 3 times demonstrations shown in the corresponding figures. All the sample curves start from the same initial point but with different initial states sampling in the feature space. We notice that in both cases, the uncertainty becomes larger when the curves turn, which is aligned with our intuition.



]

Figure 2.1: Learned 18th order dynamics of the character 'r'

In this section, we apply the proposed method to identify the Koopman operator of a bilinear model of a DC motor [152].

$$\begin{aligned}\dot{x}_1 &= -(R_a/L_a)x_1 + (k_m/L_a)x_2u + u_a/L_a\\ \dot{x}_2 &= -(B/J)x_2 + (k_m/J)x_1u - \tau_l/J\\ y &= x_2\end{aligned}$$

where x_1 is the rotor current, x_2 is the angular velocity and the control input u is the stator current. The parameters are $L_a = 0.314$, $R_a = 12.345$, $k_m = 0.253$, $J = 0.00441$, $B = 0.00732$, $\tau_l = 1.47$, and $u_a = 60$. This model is also used in [20], where 103 basis functions are used to model the feature maps based on EDMD algorithm. However, by using the inverse parametric learning algorithm 2, only 30 feature maps are needed to reproduce a good model, which demonstrates a much higher scalability than the EDMD algorithm. The validation of the model is shown in Figure 2.3, where also a local linearization of the model is used for comparison.

We run tests with 100 randomly sampled initial states and run a 1.5 second simulation with a random control input sequence. The root-mean-square error (RMSE) for the mean of the model learnt by the inverse parametric learning algorithm 2 is 0.021 while the RMSE for the

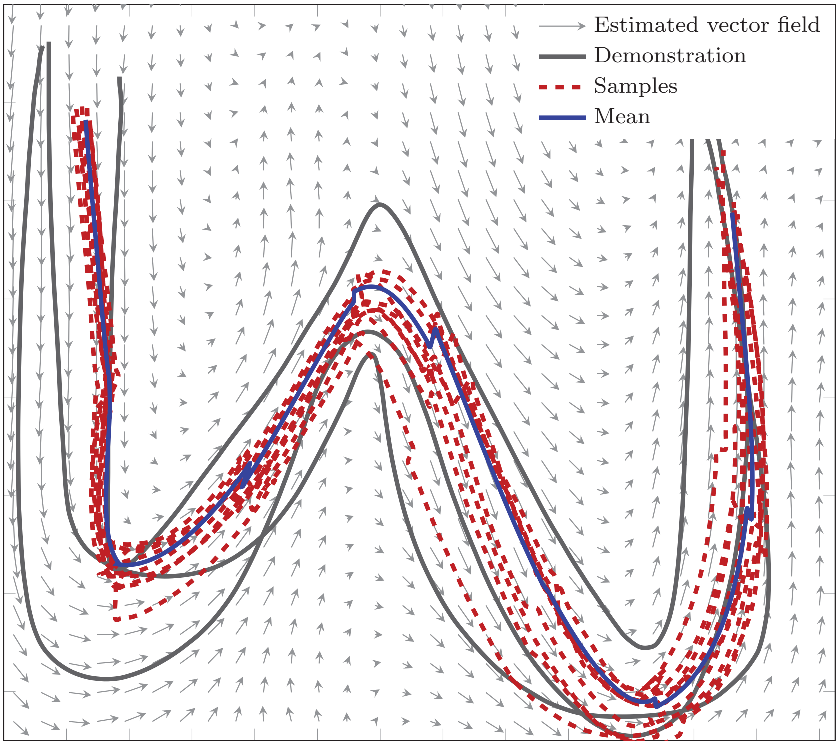


Figure 2.2: Learned 22th order dynamics of the character 'w'

model from [20] is 0.137 for the same initial states. In conclusion, our proposed method has much higher scalability and performance than current EDMD approaches. It is noted that the mean of the proposed method can properly track the real output, while its uncertainty evolves properly with real outputs always included. The closed-loop stochastic controller proposed in Section 2.1.2 is deployed to control the angular velocity x_2. The prediction horizon n_h is set to 10, Q and R are set to 1 and 0 respectively. The constraints are $u \in [-1,1]$ and $y \in [-1,1]$, where y is the angular velocity. The experiments are shown in Figure 2.4 and 2.5.

2.1.3 Semi-parametric Approach

Recall the lifted dynamics of the Koopman operator and its heuristic extension to systems with control inputs (2.23), it is natural to apply the Willems' fundamental lemma in the feature space to characterize the predictive trajectory without learning the explicit finite order representation of the Koopman operator. What is left to learn is the feature space within which the dynamics evolves linearly. This is the key idea behind this part.

Without loss of generality, we consider a system with control input and accordingly follow the extension studied in Section 2.1.2. By applying the Willems' fundamental lemma, a prediction

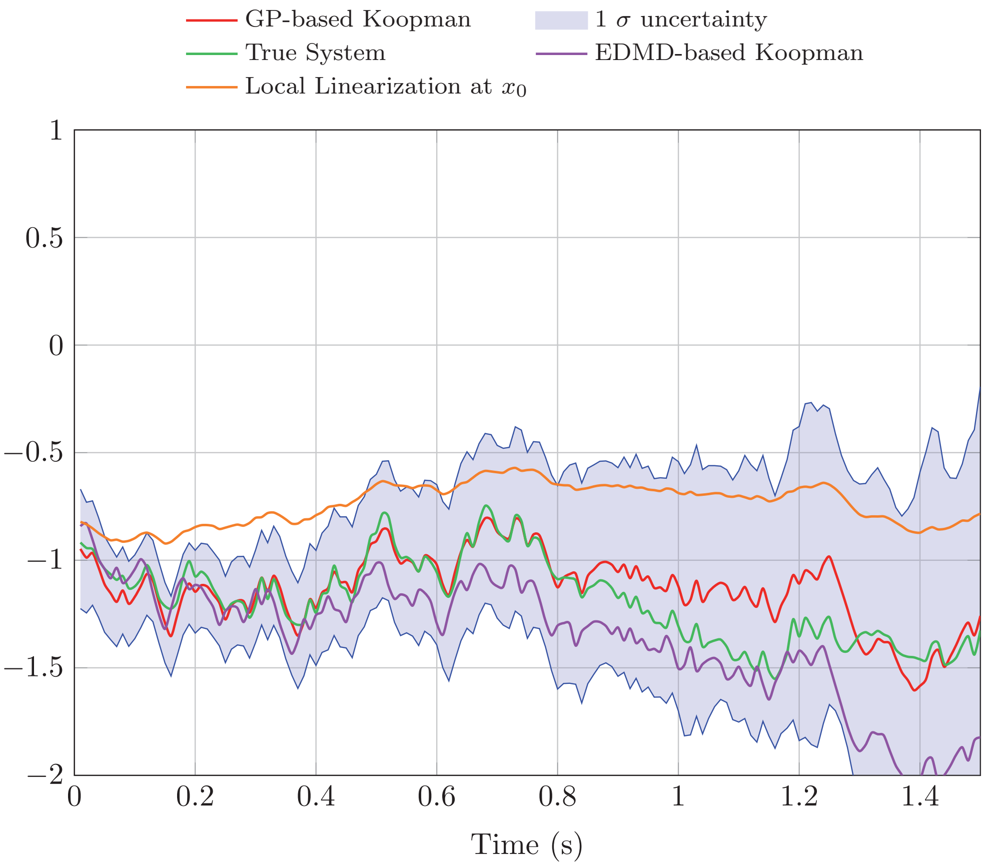


Figure 2.3: Comparison of the open-loop prediction given Koopman operator and the real state evolution

problem from Section 1.3.1 is reformulated to:

$$y_{pred} = \mathfrak{H}_{L,pred}(y_{\mathbf{d}})g(\tilde{w}_{pred}) \tag{2.24a}$$

$$g(\tilde{w}_{pred}) \in \underset{g_l,\sigma_l}{\arg\min} \frac{1}{2}\|\sigma_l\|^2 + \frac{\eta_g}{2}\|g_l\|^2$$
$$\text{s.t.} \begin{bmatrix} \Phi_\theta(x_{\mathbf{d}}) \\ \mathfrak{H}_{L,init}(u_{\mathbf{d}}) \\ \mathfrak{H}_{L,pred}(u_{\mathbf{d}}) \end{bmatrix} g_l = \begin{bmatrix} \phi_\theta(x_0) \\ u_{init} \\ u_{pred} \end{bmatrix}, \tag{2.24b}$$

where the lifting function $\phi_\theta(\cdot)$ is parametrized by θ. The finite order parametric representation of the Koopman operator (i.e. K, B, C, D in (2.23)) is not learned, hence we term this learning method a semi-parametric approach. By the minimization of the prediction error $\|y_{pred} - y_{\mathbf{d},pred}\|$ and by the use of differential parametric quadratic programming (Section 2.1.1), the parameter in $\Phi_\theta(\cdot)$ can be trained and optimized by gradient descent. In this case, the lifting/feature maps can be modelled by any differential functions, including neural networks. When a neural network is used, the learning of this model can make use of the power learning

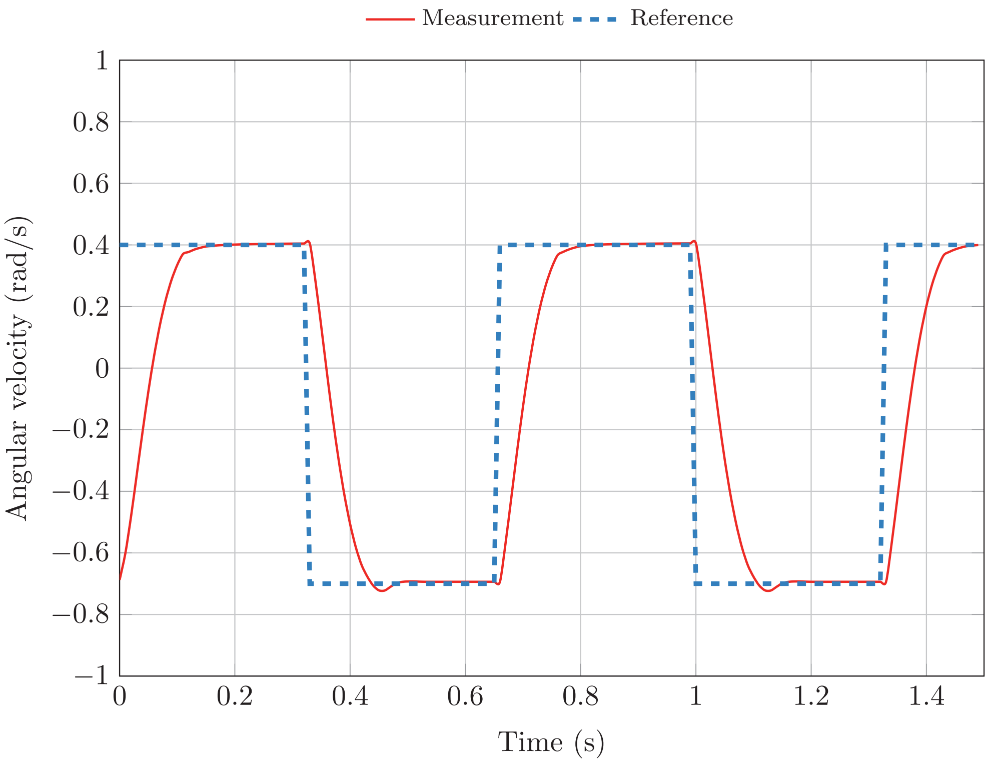


Figure 2.4: Angular velocity of the controlled motor

toolboxes from the machine learning community, such as `Pytorch` [153].

Numerical Reuslts

We firstly compare the control algorithm with the algorithm Koopman operator-based MPC controller (K-MPC) proposed in [20] by controlling a bilinear model of a DC motor. [154]

$$\begin{aligned}\dot{x}_1 &= -(R_a/L_a)x_1 - (k_m/L_a)x_2u + u_a/L_a\\ \dot{x}_2 &= -(B/J)x_2 + (k_m/J)x_1u - \tau_1/J\\ y &= x_2\end{aligned}$$

where x_1 is the rotor current, x_2 the angular velocity an the control input u is the stator current and the output y is the angular velocity. The parameters are $L_a = 0.314, R_a = 12.345, k_m = 0.253, J = 0.00441, B = 0.00732, \tau_1 = 1.47, u_a = 60$. The physical constraints on the control input are $u \in [-1,1]$.

We use 40 trajectories with time horizon $0.25s$ to construct a mosaic Hankel matrix. All trajectories are randomly initialized on the unit box $[-1,1]^2$. The control input that is used to

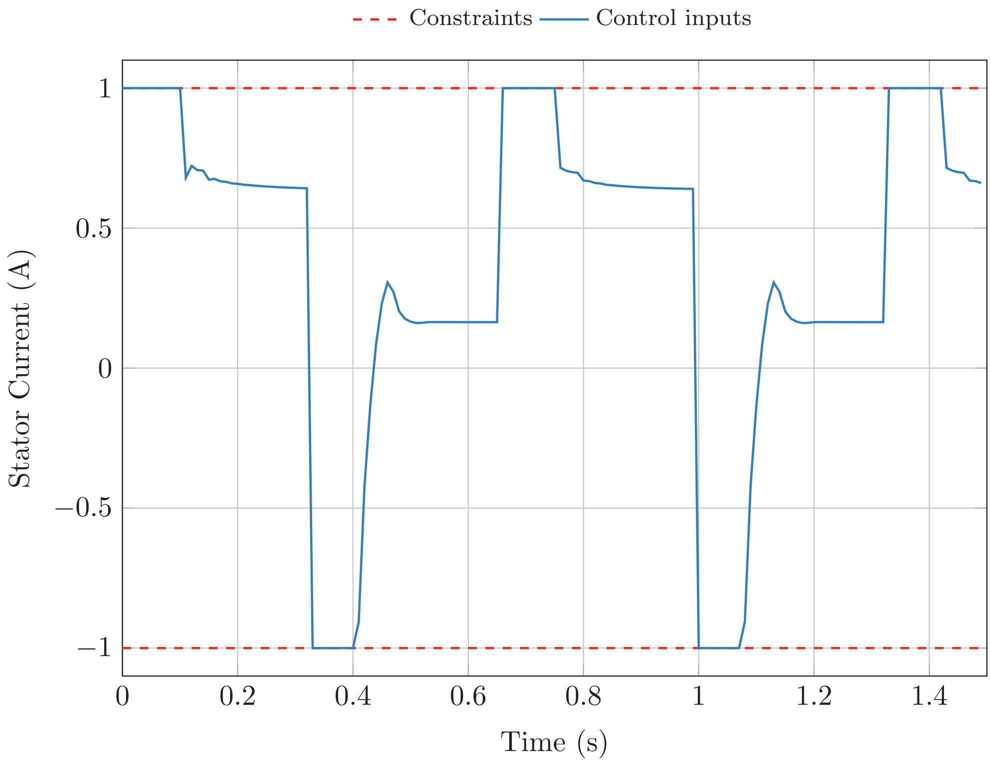


Figure 2.5: Control input from GP-based Koopman closed loop robust MPC

generate the dataset is randomly sampled from the uniform distribution over $[-1,1]$. Finally, we choose 40 thin plate spline radial basis function as lifting functions. The centers of these lifting functions are selected randomly within $[-1,1]^3$. Since the system states are not directly measurable, we construct the state by concatenating the current measurement with it one-step backward measurement. Hence, we have $C = [1,0,\dots,0]$, and we choose $Q = Q_{N_p} = 10$, $R = 0.01$. The prediction horizon is set to $N = 10$ with the input/output constraint $u \in [-1,1]$, $y \in [-0.4,0.4]$.

We benchmark the controller with a parametric Koopman based MPC (K-MPC) proposed in [20], where the controller runs in closed-loop for 3 seconds. The results are plotted in Figure 2.6 and 2.7, from which we can see that both controllers are capable of following the reference without constraint violation. The control input are not strictly the same between these two controllers, the input generated by the semi-parametric model vibrates a bit round the discontinuous points along the reference trajectory.

Our next simulation is to control the nonlinear Korteweg–de Vries(KdV) equation which models the propagation of acoustic waves in aplasma or shallow-water wave [155]. The

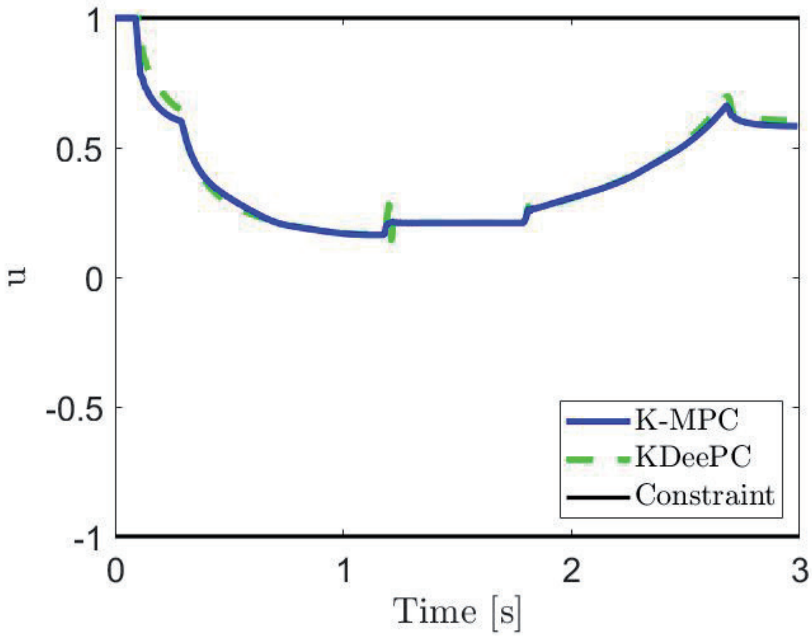


Figure 2.6: Feedback control input of a bilinear motor

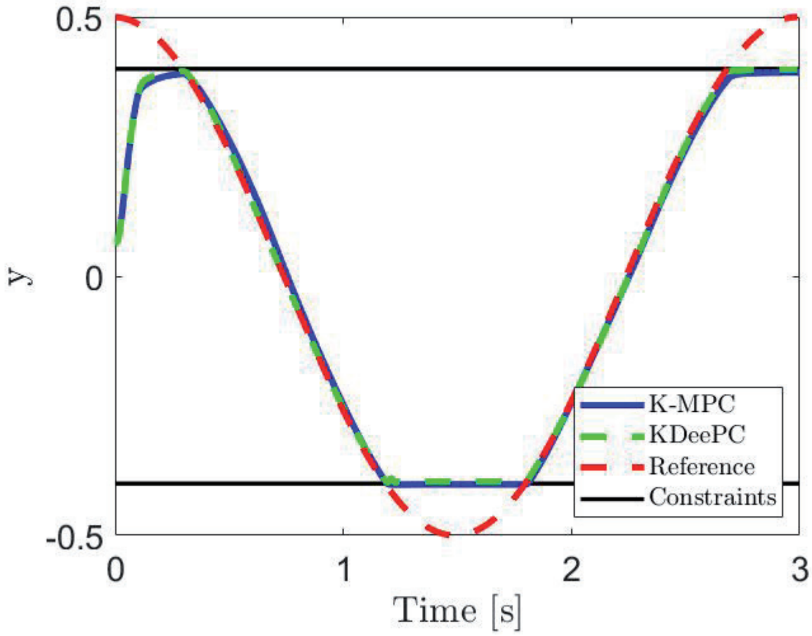


Figure 2.7: Angular velocity of a bilinear motor

equation is given as:

$$\frac{\partial y(t,x)}{\partial t} + y(t,x)\frac{\partial y(t,x)}{\partial x} + \frac{\partial^3 y(t,x)}{\partial x^3} = u(t,x)$$

where $y(t,x)$ is the unknown function and $u(t,x)$ the control input. The state is bounded by $x \in [-\pi,\pi]$, which is discretized into 128 points. The sampling time of the data is 0.02 seconds. The input is assumed to be of the form $u(t,x) = \sum_{i=1}^{3} u_i(t)v_i(x)$ where v_i consists of 3 spacial basis functions: $v_i(x) = e^{-25(x-\pi/2)^2}$ with $c_1 = -\pi/2$, $c_2 = 0$, $c_3 = \pi/2$. The input is constrained to $[-1,1]$. To generate the data and to run the closed-loop simulation, the states are initialized by a convex combination of 3 fixed spatial profiles $y_0^1 = e^{-(x-\pi/2)^2}$, $y_0^2 = -sin^2(x/2)$, $y_0^3 = e^{-(x+\pi/2)^2}$. We choose the states itself, the elementwise square of the state, the elementwise product of the states with its periodic shift as the lifting functions. The objective is set by $Q = I$, $R = \mathbf{O}$, with the prediction horizon $n_h = 5$. We collect 63 I/O trajectories to construct all the Hankel matrices.

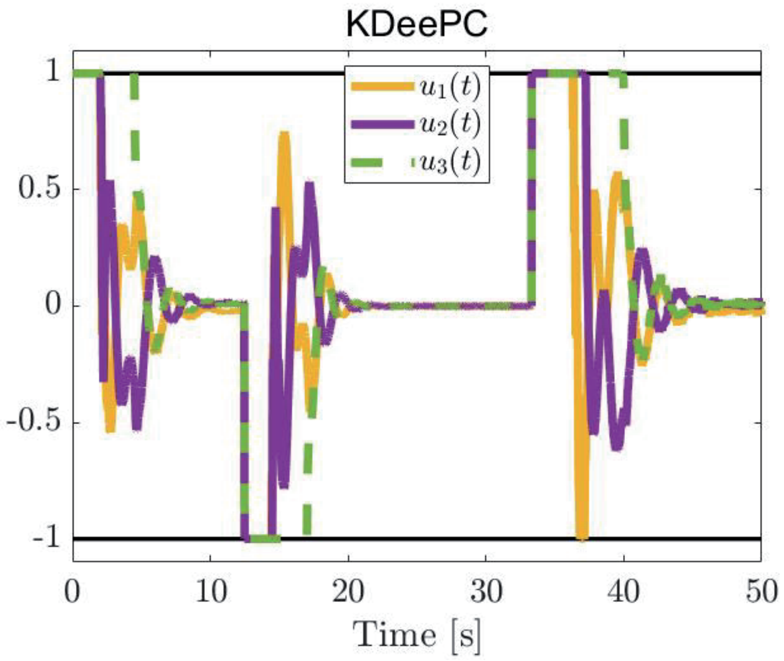


Figure 2.8: Feedback control input of KdV

2.2 RKHS based Extensions

From Section 2.1.2, we learn the fact that the extension of the Willems' fundamental lemma based on the Koopman operator theory is heuristic. At the same time, we have observed that the use of lifting has been a great success in autonomous system. Then we may ask:

Is there another lifting based viewpoint to extend nonlinear dynamics to linear dynamics?

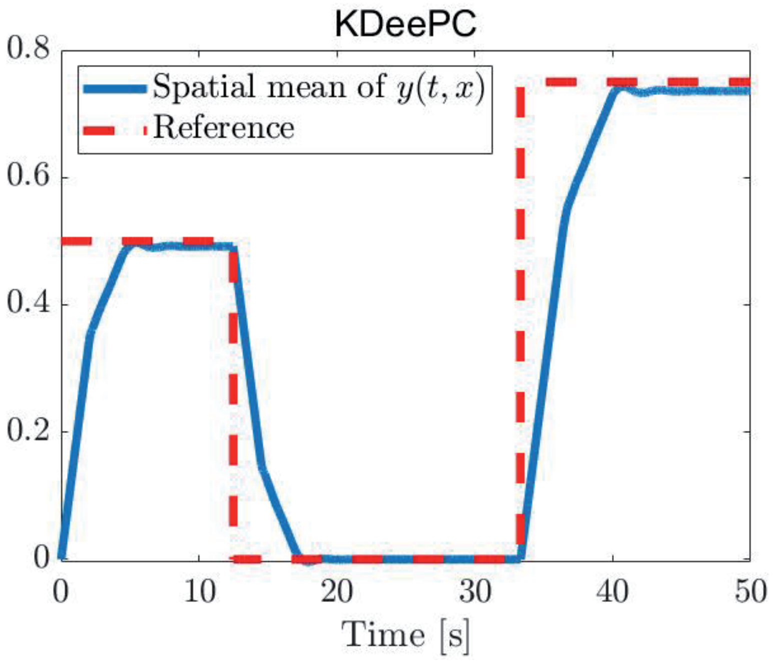


Figure 2.9: Tracking result

This part will give another possible mathematical framework without resorting to the Koopman operator theory.

2.2.1 Reproducing Kernel Hilbert Space

Definition 2. *[156] A reproducing kernel Hilbert space (**RKHS**) over a set X is a Hilbert space of functions from X to $\mathbb{R}$ such that for each $x \in X$, the evaluation functional $E_x g := g(x)$ is bounded.*

Given an RKHS H, the Riesz-representation theorem [157] guarantees that each $x \in X$ corresponds to a unique $k_x \in H$ such that $\langle g, k_x \rangle_H = g(x)$, where $\langle \cdot, \cdot \rangle_H$ denotes the inner product defined in H. It is self-dual, and its dual space H^* is the space of linear functional over H. Evaluational function is central to the analysis in the RKHS, as the spaced $\text{span}(\{k_x\}_{x \in X})$ is dense in H.

The kernel function defined on H is given by $K(x, y) = \langle k_x, k_y \rangle_H$, and it is positive-semidefinite. And we mention the definition of characteristic kernel and universal kernel [158] for the sake of completeness, where the definition of characteristic kernel [159] is adapted to the deterministic setup used in this part.

Definition 3. *Let $C_b(X)$ be the space of bounded continuous functions on a compact metric space X . A continuous positive definite kernel K on X is said to be universal if the corresponding RKHS H is dense in $C_b(X)$, i.e., for any $f \in C_b(X)$ and $\epsilon > 0$, there exists a function $h \in H$ such* *that* $\|f-h\|<\epsilon$.

Definition 4. *A kernel* k *is said to be characteristic if the map* $x \to k_x$ *is injective. The RKHS* H *is said to be characteristic if its reproducing kernel is characteristic*

Universal kernel on a compact set X is characteristic [160, Theorem 3], so as the kernel of form $k(x,y)=\psi(x-y)$ [161]. In [162], if the RKHS $H+\mathbb{R}$ is dense in $L^q(X,\mathbb{P})$ for any dirac measure $\mathbb{P}$ with $q>1$, then k corresponding to H is also characteristic.

Finally, as one might have noticed by its name, there is indeed unique correspondence between an RKHS and the covariance function of a second-order stochastic process (i.e. Gaussian process) [163, Chpater 2]. More background knowledge of RKHS theory is available in [156, 163, 164].

2.2.2 RKHS based Data-enabeled Method

We consider a linear dynamical system $\mathfrak{B}(\mathscr{A},\mathscr{B},\mathscr{C},\mathscr{D})$ evolving in the dual RKHS:

$$\begin{aligned} f_{i+1} &= \mathscr{A} f_i + \mathscr{B} E_{u_i} \\ E_{y_i} &= \mathscr{C} f_i + \mathscr{D} E_{u_i}\ , \end{aligned} \tag{2.25}$$

where f_i is a real-valued linear functional in RKHS H_x^*, similarly E_{u_i} and E_{y_i} are evaluational functionals in RKHS H_u^* and H_y^*. For the sake of consistency, we denote the dimension of H_x^* by n_x. The kernels of H_u and H_y are $k_u(\cdot,\cdot)$ and $k_y(\cdot,\cdot)$ respectively. Meanwhile, the dynamics are modelled by bounded linear operators $\mathscr{A}: H_x^* \to H_x^*$, $\mathscr{B}: H_u^* \to H_x^*$, $\mathscr{C}: H_x^* \to H_y^*$ and $\mathscr{D}: H_u^* \to H_y^*$. As one might have noticed, this approach shares the same philosophy as the Koopman operator: the behaviour of the observation is modelled instead of the state. Finally, it is noteworthy to point out that these dynamics are not necessarily infinite dimensional; more discussion about this point is allocated to Section 2.2.3.

Given a sequence of measurements $\{u_i\}_{i=1}^T$, $\{y_i\}_{i=1}^T$, we have two sequences of evaluation functionals as $\{E_{u_i}\}_{i=1}^T$, $\{E_{y_i}\}_{i=1}^T$. The corresponding n-column Hankel matrices are:

$$\mathfrak{H}_L(E_u) := \begin{bmatrix} E_{u_1} & E_{u_2} & \dots & E_{u_{T-L+1}} \\ E_{u_2} & E_{u_3} & \dots & E_{u_{T-L+2}} \\ \vdots & \vdots & & \vdots \\ E_{u_L} & E_{u_L+1} & \dots & E_{u_T} \end{bmatrix},\ \mathfrak{H}_L(E_y) := \begin{bmatrix} E_{y_1} & E_{y_2} & \dots & E_{y_{T-L+1}} \\ E_{y_2} & E_{y_3} & \dots & E_{y_{T-L+2}} \\ \vdots & \vdots & & \vdots \\ E_{y_L} & E_{y_{L+1}} & \dots & E_{y_T} \end{bmatrix}. \tag{2.26}$$

For simplicity, we further define $\nu(\{u_i\}_{i=1}^L,\{y_i\}_{i=1}^L) := [E_{u_1},\dots E_{u_L}, E_{y_1},\dots,E_{y_L}]^\top$. The Gram matrix of the stacked Hankel matrix $\mathfrak{H}_L(E_u, E_y) := [v_1, \dots, v_n]$ is then defined by

$$\begin{aligned} K_{i,j} :&= k(v(\{u_i\},\{y_i\}), v(\{u_j\},\{y_j\})) \\ &= \sum_{k=0}^{L-1} < E_{u_{i+k}}, E_{u_{j+k}} >_{H_u^*} + \sum_{k=0}^{L-1} < E_{y_{i+k}}, E_{y_{j+k}} >_{H_y^*} \\ &\overset{(a)}{=} \sum_{k=0}^{L-1} k_u(u_{i+k}, u_{j+k}) + \sum_{k=0}^{L-1} k_y(y_{i+k}, y_{j+k}) \,, \end{aligned} \tag{2.27}$$

where (a) holds by the fact that the Hilbert space is self-dual [157]. The corresponding RKHS generated by $k(v(\{u_i\},\{y_i\}), v(\{u_j\},\{y_j\}))$ is constructed by the following product topology [163, Chapter 1.4],

$$H^* := \underbrace{H_u^* \bigotimes \cdots \bigotimes H_u^*}_{L \text{ times}} \bigotimes \underbrace{H_y^* \bigotimes \cdots \bigotimes H_y^*}_{L \text{ times}} \,.$$

With the Fundamental Lemma 1, we can state the following theorem

Theorem 3. *Consider a controllable linear system $\mathfrak{B}(\mathscr{A},\mathscr{B},\mathscr{C},\mathscr{D})$ and assume E_u is persistently excited of order $N \geq n_x + L$. A trajectory of length L $\{\tilde{u}_i\}_{i=1}^L$ and $\{\tilde{y}_i\}_{i=1}^L$ is an element of $\mathfrak{B}_L(\mathscr{A},\mathscr{B},\mathscr{C},\mathscr{D})$, if and only if there exists $g \in \mathbb{R}^n$ such that*

$$g^T K g + k(\tilde{v}, \tilde{v}) - 2 \sum_{i=1}^n g_i k(\tilde{v}, v_i) = 0 \,, \tag{2.28}$$

where $\tilde{v} := v(\{\tilde{u}_i\}_{i=1}^L, \{\tilde{y}_i\}_{i=1}^L)$.

Proof. The inputs and outputs sequence containing the evaluation functional is an element of $\mathfrak{B}_L(\mathscr{A},\mathscr{B},\mathscr{C},\mathscr{D})$. Hence, by Fundamental Lemma 1, $\tilde{v} \in colspan(\mathfrak{H}_L(E_u, E_y))$ and there exists $g = [g_1, \dots, g_n]^\top \in \mathbb{R}^n$ such that

$$\begin{aligned} \sum_{i=1}^n g_i v_i = \tilde{v}_i &\overset{(a)}{\iff} \| \sum_{i=1}^n g_i v_i - \tilde{v} \| = 0 \\ &\overset{(b)}{\iff} \sum_{i,j=1}^n g_i < v_i, v_j > g_j - 2 \sum_{i=1}^i g_i < v_i, \tilde{v} > + < \tilde{v}, \tilde{v} >= 0 \\ &\iff g^T K g + k(\tilde{v}, \tilde{v}) - 2 \sum_{i=1}^n g_i k(\tilde{v}, v_i) = 0 \,. \end{aligned}$$

(a) holds by the uniqueness of the zero-element in a Hilbert space and (b) follows equation (2.27). ■

We are not able to state that span$(\mathfrak{H}_L(E_u, E_y)) = \mathfrak{B}_L(\mathscr{A},\mathscr{B},\mathscr{C},\mathscr{D})$. The most important reason is that the evaluation functional is not necessarily dense in the RKHS, while dynamics (2.25) is only defined on the set of evaluation functionals. More specifically, because the set of evaluational functional is a subset in RKHS, if $v(u_i, y_i)$ is a functional I/O trajectory, there may not exist any evaluation functional sequence that matches its scaling (i.e. $\alpha v(u_i, y_i)$ with $\alpha \neq 0$). This is what we expect to have in nonlinear systems, as closeness under scaling only holds true in linear systems. The structure of the evaluation functional in this model therefore implicitly enforce the structure of the nonlinear responses. With this idea in mind, we can also see that by enforcing inputs outputs to be evaluational functional also limits the possible realization of operator $\mathscr{A}$. More specifically, it is still unclear how restrictive this nonlinear structure is, and the proposed dynamics (2.25) still remain hypothetical.

Nonlinear Data-enabled Prediction

Similar to what we have done in Section 1.3.1, we are able to define a data-driven predictor based on the Theorem above. In particular, given an input-output sequence of length t_{init}, $\tilde{u}_m := \{\tilde{u}_1, \tilde{u}_2, \dots \tilde{u}_{t_{init}}\}$ and $\tilde{y}_m := \{\tilde{y}_1, \tilde{y}_2, \dots \tilde{y}_{t_{init}}\}$, an open-loop prediction of length n_h is to predict $\tilde{y}_p := \{\tilde{y}_{t_{init}+1}, \tilde{y}_{t_{init}+2}, \dots \tilde{y}_{t_{init}+n_h}\}$ if a sequence of inputs $\tilde{u}_p := \{\tilde{u}_{t_{init}+1}, \tilde{u}_{t_{init}+2}, \dots \tilde{u}_{t_{init}+n_h}\}$ is applied from $t_{init}+1$ to $t_{init}+n_h$. Theorem 3 indicates that the prediction problem is equivalent to the following optimization problem

$$\min_{y_p, g} g^T K g + k(\tilde{v}, \tilde{v}) - 2 \sum_{i=1}^{n} g_i k(\tilde{v}, v_i) \,, \tag{2.29}$$

where K is computed from the Hankel matrix $\mathfrak{H}_{t_{init}+n_h}(E_u, E_y)$ and $\tilde{v} := v(\{\tilde{u}_i\}_{i=1}^{t_{init}+n_h}, \{\tilde{y}_i\}_{i=1}^{t_{init}+n_h})$. Note that the prediction is achieved by an optimization problem instead of by solving the nonlinear equation (2.28) in order to better accommodate the numerical solvability of nonlinear equation (2.28), the presence of measurement noise, the model mismatch and the infinite dimensionality. At the same time, it is noteworthy to point out that a solution to equation (2.28) is a global minimizer of problem (2.29).

Remark 12. *A kernel heuristic is mentioned in [17, Section V], which relies heavily on their presumed Hammerstein and/or Wiener system structure. In particular, their algorithm decouples the reconstruction of the predicted trajectory and the selection of the weight by assuming an inverse map from RKHS to the states, which is not valid for most RKHS.*

Nonlinear Data-enabled Predictive Control

To convert the method in Section 2.2.2 into a predictive control scheme, the input sequence is optimized so that the corresponding output sequence is most desirable. Similar to what we have done in Section 1.3.1, it leads to an optimistic bi-level problem [74, Chapter 2] as follows:

$$\begin{aligned}\min_{u_p,y_p,g}\sum_{i=0}^{n_h} & l(u_{t_{init}+i}, y_{t_{init}+1+i})\\ \text{s.t.}\, & u_{t_{init}+i}\in\mathscr{U},\ y_{t_{init}+1+i}\in\mathscr{Y}\\ & y_p\in\operatorname{argmin}_{\tilde{y}_p} g^T K g + k(\tilde{v},\tilde{v}) - 2\sum_{i=1}^{n} g_i k(\tilde{v}, v_i)\,, \end{aligned}\tag{2.30a}$$

where $l(\cdot,\cdot)$ is the stage cost and $\mathscr{U}$, $\mathscr{Y}$ are constraints for control input and outputs.

2.2.3 Discussion and Practical Issues

In Section 2.2.2, the theory and the applications have been built, more theoretical details and practical issues are elaborated in this section.

On Existence of the Proposed Model

One obvious question is that whether the proposed model (2.25) makes any practical sense. Above all, the proposed model (2.25) includes standard linear systems (2.25) as a special instance.

Lemma 6. *If H_x, H_u, H_y are RKHS whose kernel are $k(x,y)=x^Ty$, then model* (2.25) *is equivalent to a linear dynamics.*

Proof. By Riesz representation theorem, for each $f_i\in H_x^*$ there exists a unique $x_i\in H_x$ such that $\forall\ \tilde{x}\in H_x$, $f_i(\tilde{x}) = < x_i^T, \tilde{x} >= x_i^T\tilde{x}$. The following proof finds the x_{i+1} such that $\langle x_{i+1},\cdot\rangle = f_{i+1}(\cdot)$. For any $\tilde{x}\in H_x$, equation (2.25) gives

$$\begin{aligned} f_{i+1}(\tilde{x}) =& \mathscr{A}\, f_i(\tilde{x}) + \mathscr{B}\, E_{u_i}(\tilde{x}) \overset{(1)}{=} \langle\mathscr{A}\, x_i, \tilde{x}\rangle + \langle\mathscr{B}\, u_i, \tilde{x}\rangle\\ =& \langle\mathscr{A}\, x_i + \mathscr{B}\, u_i, \tilde{x}\rangle \overset{(2)}{\Longrightarrow} f_{i+1}(\cdot) = \langle\mathscr{A}\, x_i + \mathscr{B}\, u_i, \cdot\rangle\,, \end{aligned}$$

where (1) follows the definition of the kernel function and (2) is weak-* convergence. In a similar way, $E_{y_i} = \mathscr{C}\, f_i + \mathscr{D}\, E_{u_i}$ can be reformulated. Hence, we conclude a standard linear system (2.25). ■

Remark 13. *Lemma 6 can be generalized to Hammerestein systems and Wiener systems. Without loss of generality, we consider a Hammerstein system, whose input nonlinearity enters the dynamics through an N_ϕ dimensional map $\phi(\cdot)$ as follows,*

$$\begin{aligned} x_{i+1} &= Ax_i + B\phi(u_i)\\ y_i &= Cx_i + D\phi(u_i)\,. \end{aligned}$$

If H_u is generated by the kernel $k(x,y) = <\phi(x), \phi(y)>_{\mathbb{R}^{N_\phi}}$ with N_ϕ the dimension of $\phi(\cdot)$ and

H_x , H_y are generated by a linear kernel, then the resulting dynamics of (2.25) *is a Hammerstein system following a similar proof of Lemma 6.*

On Persistent Excitation

The assumption of persistent excitation of E_u in Theorem 3 is defined according to the rank of its Hankel matrix $\mathfrak{H}_L(E_u)$. Checking rank of a matrix defined by functionals is not trivial, instead, the following procedure can simplify the rank calculation.

$$\operatorname{rank}(\mathfrak{H}_L(E_u)) = \operatorname{rank}(\mathfrak{H}_L(E_u)^T \mathfrak{H}_L(E_u)) =: K_u \,, \; (K_u)_{i,j} = \sum_{k=0}^{L-1} k_u(u_{i+k}, u_{j+k}) \,.$$

Hence, the condition of persistent excitation is determined by the rank of the corresponding Gram matrix. However, the condition of persistent excitation is only well-defined for finite dimensional dynamics, where H_x, H_u, H_y are finite dimensional, such as the RKHS corresponding to a linear kernel or a polynomial kernel. If, instead, the dynamics is infinite dimensional, persistent excitation is no longer guaranteed, and Theorem 3 only serves as a heuristic. Motivated by the persistent excitation condition, more informative data is more desirable in this case. This is particular the case when a universal kernel is used. This claim follows the fact that $\operatorname{rank}(K_u) \geq \operatorname{trace}(K_u)$ when E_{u_i} is within a unit ball of H_u^* [165][7]. Given the fact that a universal kernel $k_u(\cdot,\cdot)$ is strictly positive definite [164, Lemma 4.55] (i.e. $k_u(u_i, u_i) > 0 \; \forall \; i \in \mathbb{Z}$), the rank of K_u is non-decreasing regarding the amount of data. In conclusion, a relaxed condition of persistent excitation is required for an infinite dimensional system, and this author believes that this requires a more in-depth understanding about the geometric structure of the evaluational functional. Until the submission of this thesis, this author has not yet found out a solution.

On Choice of the Kernel

Even though the proposed method is non-parametric, the choice of kernel still determines the final performance. Due to the unique correspondence between the RKHS and the kernel function by Moore-Aronszajn theorem [163, Theorem 3], the choice of kernel function reflects our a-priori knowledge. Above all, to ensure a unique recovery of the predictive trajectory from its evaluational functional, the kernel needs to be characteristic. Beyond this basic requirement, side information can be included in the choice of kernel. For example, if $\overline{x}$ is the unique fixed point, one can choose a kernel k such that $k(\overline{x}, \overline{x}) = 0$. On top of this, linear and polynomial kernels imply symmetric dynamics around 0. An exponential kernel $k(x, y) = e^{x^T y}$ is used when H_x, H_u, H_y is spanned by polynomials, because polynomials are dense in the corresponding RKHS [164, Chapter 4]. Moreover, if the trajectories that are close to each other in the state space also show similar behaviour, the RBF kernel $k(x, y) = e^{\frac{\|x-y\|}{2}}$ can be used.

[7] By scaling the kernel function, this condition holds for any kernel function defined on a compact set.

Beyond the a-priori knowledge about the system, the choice of kernel function also affects the tractability of Problem (2.29) and Problem (2.30). We observe that the exponential kernel has relatively low numerical stability, as two distant trajectories result in a large gradient. We also observed that the RBF kernel leads to pervasive local minima, which causes poor performance when the Problem (2.30) and Problem (2.29) are solved by gradient based algorithms. Based on our experiments, the kernel function $k(x,y)=e^{\frac{-\|x-y\|}{2}}e^{x^Ty}$ in general has the best performance for examples tried so far.

Finally, even with all points discussed above, the choice of the kernel function is still non-trivial in general. If a dictionary of kernel functions $\{k(\cdot,\cdot)\}_{i=1}^{N_k}$ is available, the choice of the kernel can be optimized over the positive-weighted sum $k(\cdot,\cdot)=\sum_{i=1}^{N_k}\alpha_i k(\cdot,\cdot)$ via minimization of the prediction error in the training data set.

On Stochastic Model with Measurement Noise

When the data is contaminated by measurement noise, a kernel mean embedding [166] can be used, which evaluates the kernel function with respect to its distribution as

$$\mathbb{E}_{\tilde{X}}k(\cdot,x)\,, \tag{2.31}$$

where $x\sim\tilde{X}$ relating to the distribution of the measurement noise. If the distribution of measurement noise is known, such as Gaussian distribution, the Equation (2.31) has closed explicit form. If noise is unknown, then (2.31) can be evaluated by the empirical distribution.

Remark 14. *Notice that the measurement noise in each column of the Hankel matrix is not i.i.d. Hence, the empirical distribution has slower convergence rate than $O(\frac{1}{N})$.*

2.2.4 Numerical Results

Damped Pendulum

We consider a force acting on the tip of the damped pendulum. The dynamics are

$$\dot{x}_1=x_2\,,\qquad\qquad \dot{x}_2=-\frac{2g}{l}\sin(x_1)-\mu x_2^3+\frac{1}{l}|\cos(x_1)|u$$

with $g=9.8N/kg$, $l=0.5m$ and $\mu=0.1$ denoting a nonlinear friction factor. Only x_1 is observed as $y=x_1$. The output training data is generated by random input of ranging from -1 to 1, the sequence is measured with a sampling time of 0.04 seconds. 500 data points are used to defined the Problem (2.29) and Problem (2.30). Part of the output training data is shown in Figure 2.10, in which one can see the nonlinear modulation effect of the damped pendulum model. An open-loop prediction of $n_h=60$ step is carried out with $t_{init}=10$ previous step measured and the result is shown in Figure 2.11, where the kernel for inputs is $k_u(x,y)=0.2e^{\frac{-\|x-y\|^2}{6}}+e^{x^Ty}+0.01e^{\frac{-\|x-y\|^2}{6}}e^{x^Ty}$ and the one for outputs is $k_y(x,y)=0.2e^{\frac{-\|x-y\|^2}{6}}+$

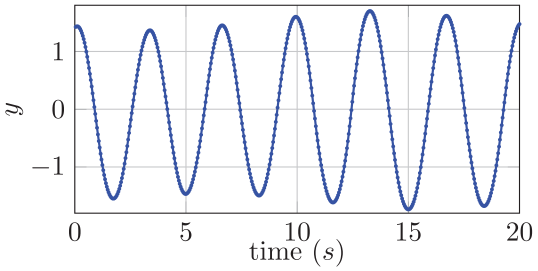


Figure 2.10: Snapshot of training data used in damped pendulum model

$e^{x^T y} + 0.01 e^{\frac{-\|x-y\|^2}{6}} e^{x^T y} + (1 + x^T y)^2$.

Bilinear Motor

We consider a bilinear motor [152] whose dynamics is

$$\dot{x}_1 = -\frac{R_a}{L_a} x_1 + \frac{k_m}{L_a} x_2 u + \frac{u_a}{L_a}, \quad \dot{x}_2 = -\frac{B}{J} x_2 + \frac{k_m}{J} x_1 u - \frac{\tau}{J}$$

where x_1 is the rotor current, x_2 is the angular velocity and the control input u is the stator current. Only the stator current is measured as $y = x_2$. The parameters are $L_a = 0.314$, $R_a = 12.345$, $k_m = 0.253$, $J = 0.00441$, $B = 0.00732$, $\tau = 1.47$, and $u_a = 60$. Due to the synthetic effect of the bilinear term and the bias term, the responses of the system at different operating points are wildly different, hence the data is generated by a sequence of $\mathcal{N}(\mu_n, \sigma_n^2)$, where the mean of the random signal μ_n is time varying in order to excite more modes around different operating points. In particular, the mean μ_n ranges from -0.5 to 0.5, and the variance $\sigma_n = 1$. 700 datapoints are used to defined the prediction and the optimal control problem. The sampling time of the generated sequence is 0.01 seconds. An open-loop prediction of 30 is carried out with 40 previous step measured, the corresponding result is shown in Figure 2.12, where the kernel for inputs is $k_u(x, y) = 0.1 e^{\frac{-\|x-y\|^2}{4}} + e^{\frac{-\|x-y\|^2}{4}} e^{x^T y}$ and the one for outputs is $k_y(x, y) = 0.1 e^{\frac{-\|x-y\|^2}{4}} + e^{\frac{-\|x-y\|^2}{4}} e^{x^T y}$.

Furthermore, a predictive control scheme is tested, where the 15 previous steps are used for a prediction horizon of 8 steps. The stage cost is $l(u, y) = (y - y_{\text{ref}})^T (y - y_{\text{ref}}) + 0.01 u^T u$. The proposed predictive control scheme is compared to the nonlinear model predictive control, which has explicit knowledge of the system dynamics and full access of the state measurement. A step-like reference signal is tracked with outcome shown in Figure 2.13 without considering output constraints[8], it is observed that the proposed method shows competitive performance against the model based control law, however, it fails to converge to the upper reference with a subtle bias and it has slight overshoot with respect to both set points.

[8] The consideration of output constraints makes the bi-level highly unsolvable.

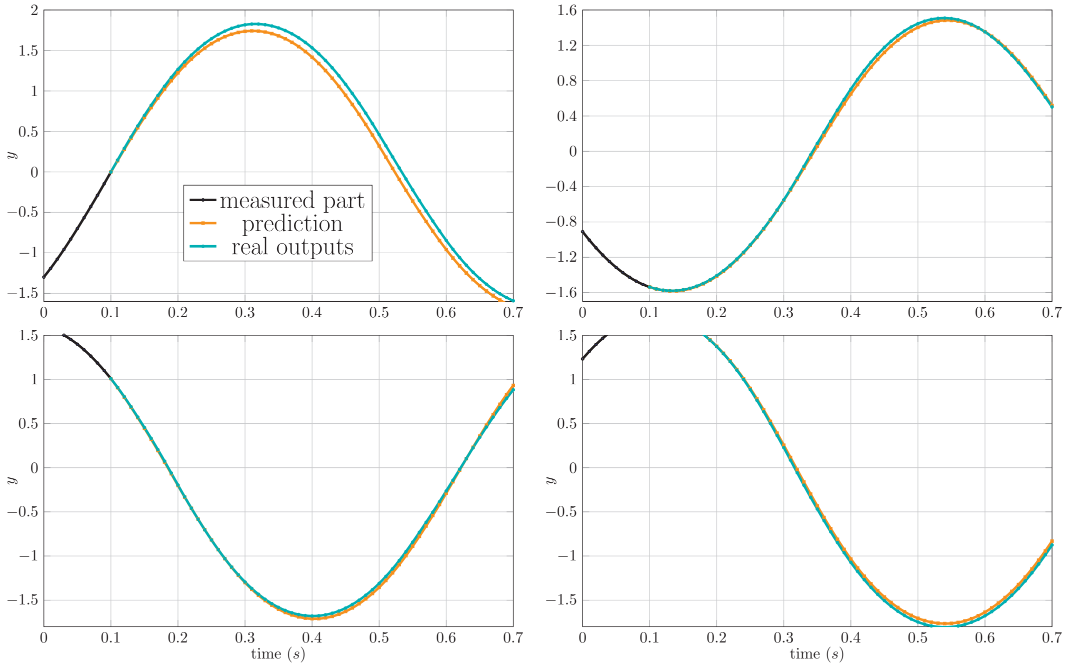


Figure 2.11: Open-loop prediction of damped pendulum. An asymmetric oscillation is observed and is learnt by the proposed method in subfigure (b). Each subplot is an open-loop prediction evaluated on different data. By referring to Section 2.2.2, the black curves are $\tilde{y}_m$ and orange curves are the open-loop prediction solved by problem (2.29)

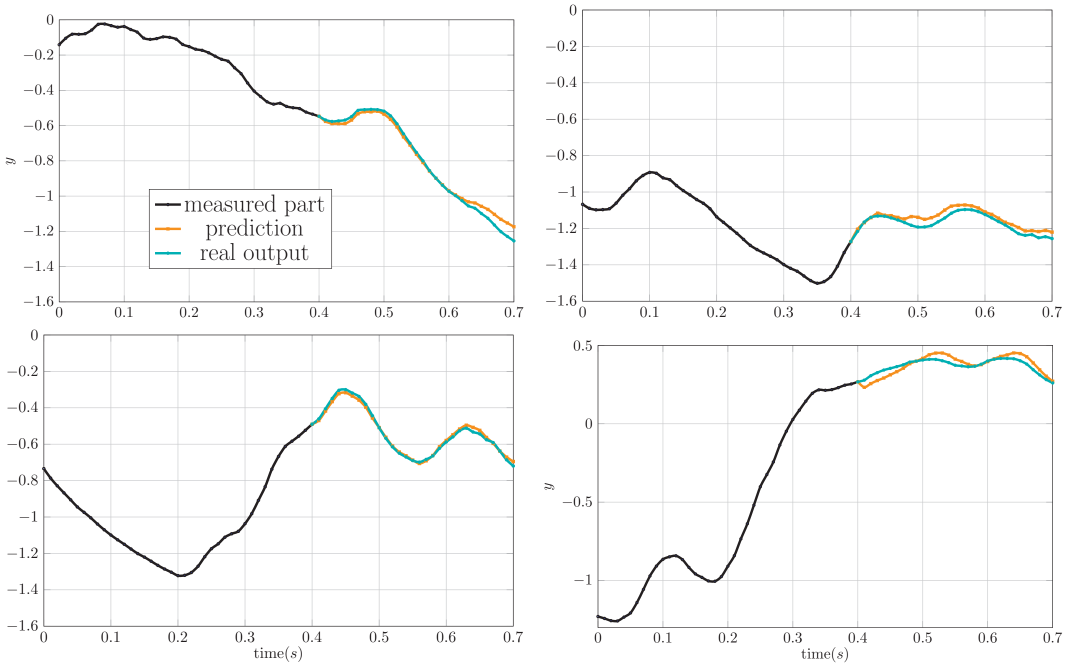


Figure 2.12: Open-loop output prediction with random biased input sequence of bilinear motor model. Interpretation of each subplot is similar to Figure 2.11

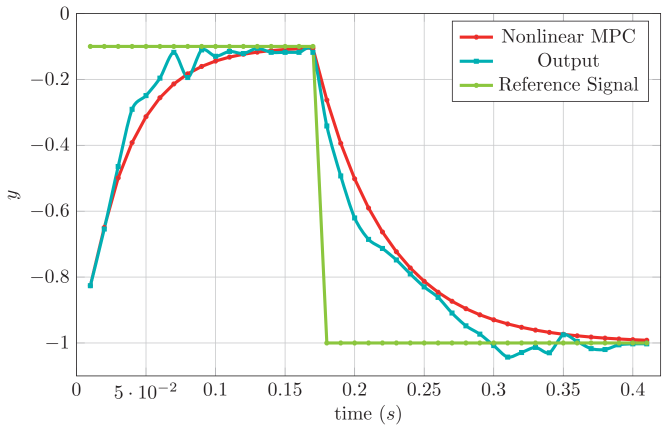


Figure 2.13: Closed-loop MPC control output

2.3 Final Remark

After studying the robustness aspect of direct data-driven control based on Willems' fundamental lemma for linear systems, we sought to extend these results to nonlinear systems in this chapter. Two approaches were considered: the Koopman operator-based heuristic and the reproducing kernel Hilbert space (RKHS)-based lifting. While these methods have shown promise in this chapter, it is important to discuss their limitations.

For the heuristic Koopman based extension, the major challenge is on the theoretical side. One argument is that, if we use the lifted states $\phi(x)$ as the underlying state, the resulting model is just a linear model, making the model linear. This view is agreed upon by the author. In fact, the inverse statement is also true: it is always possible to disguise a linear system as a nonlinear one. Consider a simple 1-d example $x^+ = 2x$, if we use the use e^x as the state instead, you will have the dynamics $e^{x^+} = (e^x)^2$. For most reported successful applications of the Koopman operator in nonlinear systems, the author conducted exhaustive tests on those reported models. With the exception of the KdV model and the bilinear motor model, using a linear controller can provide comparable or even higher performance than the Koopman-based method in all other models. Therefore, the author believes that there is some potential for Koopman operator-based methods, but the mathematical framework based on the Koopman operator theory may not be a proper choice for extending the application of linear control theory to nonlinear systems. This is also why all the proofs in Section 2.1 are given in autonomous systems.

In contrast, the RKHS-based extension addresses the limitation of linearity by imposing the nonlinear structure endowed by the evaluational functionals. However, the resulting method poses a significant computational challenge, making the rationale for not using system identification becomes unclear. Additionally, it is also unclear whether the proposed hypothetical dynamics (2.25) in dual RKHS can model nonlinear dynamics with multiple fixed point. For a stable linear system[9], the collection of its limiting behaviours will form a subspace, and thus its limit cycle and equilibrium points are not isolated. Therefore, the resulting topology induced by the collection of limiting behaviours require us to investigate the induced topology generated by the intersection of the set of evaluational functionals and a linear subspace. To the best of my knowledge, this topic is still an untouched area in the RKHS community.

In summary, the idea of lifting presents potential, but significant challenges remain. While both the Koopman-based and RKHS-based extensions have their limitations, they offer promising avenues for future research.

[9] This statement holds true in infinite dimensional linear system

Direct Data-driven Stability Analysis Part II

3 Robust Lyapunov Stability Analysis

Nonlinear Lyapunov stability analysis has been widely studied, where model-based approaches and sampling-based approaches form two main categories. In either approach, a Lyapunov candidate is optimized or synthesized by verifying the Lyapunov stability conditions. In the model-based approaches, the knowledge of the underlying model is explicitly used in the search of the Lyapunov function. On the contrary, the Lyapunov function is trained by penalizing the violation of the Lyapunov stability conditions on a dataset. Even though a standard Monte-Carlo sampling scheme can also give a probabilistic guarantee [167], it is always preferable to give a strict qualification in stability analysis. In this case, both model-based approaches and sample-based approaches require the explicit knowledge of the model. In particular, verification of the Lyapunov stability condition usually resort to nonlinear optimization or satisfiability modulo theory (SMT) solvers, such as dReal [168]. Note that when smooth dynamics are considered, one can write the Lyapunov stability condition with respect to any Lyapunov candidate into an explicit algebraic form (see e.g [169–171]). The SMT solver is accordingly used to checked whether these algebraic inequalities are satisfied up to some user-defined tolerance [172].

To the best of the authors' knowledge, the first numerical method that finds a Lyapunov function solves the Zubov equation [173]. The Zubov equation models the Lyapunov function as the solution to a linear partial differential equation (PDE). The approximation of this PDE is solved by series expansion [173], collocation method [174], etc. One main advantage of the model-based approach is the a-priori knowledge about the model can be used to reformulate the Lyapunov learning problem into a simpler problem. When polynomial dynamics are considered, a sum of square (SOS) programming relaxation can be used to search for polynomial Lyapunov functions [175, Chapter 4][176]. Due to the nice algebraic property of polynomials, the SOS framework has been further used to find the region of attraction [177, 178] and its sparsity structure has been used to improve its scalability [179, 180]. Parallel to the studies in polynomial dynamics, PWA dynamics are another genre attracting broad research interest [181, 182]. Such tremendous interest is also a result of the ubiquitous appearances of PWA functions in various controllers, such as ReLU-neural-networks-based controller and linear

MPC [183]. For the PWA setup, optimization based approaches play the central role, which mostly applies linear matrix inequality [178, 184, 185] and mixed integer programming [186].

Unlike the model-based approaches, sampling-based methods highly rely on an efficient strategy of generating informative samples. The counter-example guided inductive synthesis (GEGIS) [187, 188] is one major concept applied behind many sample-based approaches (see e.g. [185, 189, 190]). These approaches have direct access to the model or its simulator. During the learning process, they iteratively augment the sample dataset by adding counter examples to the Lyapunov candidate proposed in the current iteration. These algorithms train the Lyapunov function by penalizing the violation of the Lyapunov stability condition on the samples, and they converge when no further counter example can be generated [170, 189].

The search for a Lyapunov function is usually confined to a specific function class, such as generalized quadratic form [191] or positive definite kernel regressor [192]. In this part, we will focus on PWA Lyapunov functions defined on a compact set. Besides the advantages mentioned in the model-based approach paragraph, PWA Lyapunov candidates have shown nice interplay with Lipschitz dynamics. In particular, when the samples of system dynamics are assigned to the vertices of a grid, a robust Lyapunov stability condition on each simplex can be verified by only considering a tightened Lyapunov condition defined on its vertices. This family of methods is called the continuous piece-wise affine (CPA) method [193, 194]. The CPA method has been extend to more general problem setup: differential inclusion [195], switched system [196], etc. In this part, we also consider Lipschitz dynamics, but we do not assume that the data are located on the vertices. Therefore, we do not term our method a CPA method to avoid unnecessary confusion. More detailed comparison with the CPA method are postponed to Section 3.3.3.

The rest of this chapter is organized as follows: In Section 3.1, some necessary tools from convex analysis are reviewed alongside the statement of the problem setup. In Section 3.2, we will first generalize the Lyapunov theorem in Section 3.2.1, this generalization will later be used to develop a local Lyapunov condition with PWA Lyapunov candidate in Section 3.2.2. In the sequel, Section 3.3 applies this local condition to a set of uncertain function defined by data, whose robust satisfaction is summarized as in Theorem 7. This theorem is later used to define a convex inequality condition for the class of Lipschitz function in Section 3.3.2, where the learning problem will be summarized. A comparison between the proposed learning problem and other related works are given in Section 3.3.3. The learnability of the proposed scheme is studied in Section 3.4.1, after which the proposed learning problem is recast into an equivalent form to enable higher computational efficiency in Section 3.4.2. The general learning algorithm are summarized in 3.4.3 with a numerical validation in Section 3.5.

Due to ambiguity in the literature, we indicate our definition of polyhedra and polytopes:

- A polyhedron $P \subset \mathbb{R}^n$ is an intersection of finitely many half spaces: $\exists m \in \mathbb{N}, A \in$

$\mathbb{R}^{m\times n}, b \in \mathbb{R}^m$ such that

$$P = \{x \in \mathbb{R}^n \mid Ax - b \in (\mathbb{R}_+)^m\} .$$

When the polyhedron is bounded, with slightly abuse of notation, we denote the number of vertices by $|P|$.

- A polytope is a finite union of bounded polyhedra, which is not necessarily convex. Accordingly, a convex polytope is the convex hull of its vertices.

3.1 Preliminary

In this section, we will first review some results from convex analysis and then introduce the considered stability analysis problem.

3.1.1 Convex Analysis

Let $\mathbb{R}^n_C$ denote the set of compact convex subsets in $\mathbb{R}^n$, the support function of a compact set $C \in \mathbb{R}^n_C$ is defined by

$$V_C(g) := \max_{x\in C} g^\top x ,$$

for all $g \in \mathbb{R}^n$, and any convex set can be uniquely characterized by its support function. Meanwhile, the indicator function of a convex set $C \in \mathbb{R}^n_C$ is defined by

$$\iota_C(x) := \begin{cases} 0 , & x \in C \\ \infty , & x \notin C , \end{cases}$$

and this function is convex. With this definition, we also have

$$\iota_{C_1\cap C_2}(x) = \iota_{C_1}(x) + \iota_{C_2}(x) , \tag{3.1}$$

as $\iota_{C_1\cap C_2}(x) = 0$ if and only if $\iota_{C_1}(x) = 0$ (*i.e.* $x \in C_1$) and $\iota_{C_2}(x) = 0$ (*i.e.* $x \in C_2$), and thus $x \in C_1 \cap C_2$. A conjugate of a convex function $h : \mathbb{R}^n \to \mathbb{R}$ is defined, for $g \in \mathbb{R}^n$, by

$$h^*(g) := \max_{x\in\mathbb{R}^n} g^\top x - h(x) .$$

By this definition, the conjugate of a non-empty set indicator function is its support function [129, p. 11.4] as

$$V_C(\cdot) = \iota^*_C(\cdot) .$$

Given two proper, convex functions $h_1 : \mathbb{R}^n \longrightarrow \mathbb{R}$ and $h_2 : \mathbb{R}^n \longrightarrow \mathbb{R}$, the infimal convolution is defined by

$$h_1(x)\#h_2(x) := \inf_{y \in \mathbb{R}^n} h_1(y) + h_2(x-y)\,,$$

Geometrically, the epigraph of $h_1(\cdot)\#h_2(\cdot)$ is the Minkowski sum of the epigraph of $h_1(\cdot)$ and $h_2(\cdot)$. For the sake of simplicity, we denote

$$\#_i h_i(x) := \inf \sum_i h_i(y_i) \quad \text{s.t.} \; \sum_i y_i = x\,.$$

By Lagrangian multiplier, the following calculus of infimal convolution can be derived [197]

Proposition 4. *Let h_1, h_2 be two proper, convex functions, then*

$$\left(h_1 + h_2\right)^* (x) = h_1^*(x)\#h_2(x)^*\,,$$

3.1.2 Unknown dynamic, fixed data

In our problem setup, we consider an **unknown** continuous time dynamic system on a dimension n_x compact set $\mathcal{X} \subset \mathbb{R}^{n_x}$,

$$\frac{dx}{dt} = f(x)\,,$$

which we know has a locally asymptotically stable (LAS) equilibrium point (EP). And we have a size N_D **fixed** dataset sampled from this dynamic system as $D = \{(x_i, f_i = f(x_i))\}_{i=1}^{N_D}$ with $x_i \in \mathcal{X}$. The goal is to analyse the stability of this unknown dynamical system based on the fixed dataset D. Without loss of generality, we assume that

Assumption 7. *$\mathbf{0} \in \mathcal{X}$ is the LAS EP and we have access to a compact subset $\mathcal{X}_s$ of its region of attraction.*

Note that $\mathcal{X}_s$ models a conservative prior about the region of attraction (RoA) of the unknown dynamic system, e.g. deduced from engineering practice. Meanwhile, if no such information is available, one can also consider $\mathcal{X}_s = \{\mathbf{0}\}$. We denote the vector space of Lipschitz continuous functions on n_x dimensional domain $\mathcal{X}$ by $\mathrm{Lip}(\mathcal{X})^{n_x}$. To facilitate the analysis, we further assume that

Assumption 8. *Let $f \in \mathcal{F}$ with $\mathcal{F} \subset \mathrm{Lip}(\mathcal{X})^{n_x}$, such that*

$$\forall\; h \in \mathcal{F}, x \mapsto \|h(x)\| \text{ is bounded within } \mathcal{X}\,,$$

with upper bound denoted $\|h\|_\infty$ (clearly a norm on $\mathcal{F}$).

Remark 15. *Assumption 8 means that the underlying dynamic system is bounded within $\mathcal{X}$ (i.e. there will not exist infinite velocity). Meanwhile, $\mathcal{F}$ does not need to be a Hilbert space, a* *typical case being the whole* $\mathrm{Lip}(\mathscr{X})^{n_x}$. *If a Hilbert space* $\mathscr{F}$ *is used, it is usually assumed to be infinite dimensional to ensure a large modelling capability.*

Remark 16. *Assumption 8 implies that the evaluation operator defined by*

$$E_x(h) := h(x)\,,\ \forall\ x \in \mathscr{X},\ h \in \mathscr{F}$$

is bounded (i.e. continuous in h) with respect to the norm $\|\cdot\|_\infty$*, meaning that*

$$\forall\ x \in \mathscr{X}, \exists\ C_x \geq 0\,; \forall\ h \in \mathscr{F}, \|E_x(h)\| \leq C_x \|h\|_\infty.$$

Indeed, taking $C_x = 1$ *for all* $x \in \mathscr{X}$ *is always possible. However, as soon as* $\mathscr{F}$ *contains the space of polynomial vector fields, it cannot be complete with respect to this norm* $\|\cdot\|_\infty$*, as the Stone-Weierstrass theorem ensures that its uniform completion is the whole space of continuous functions (including non-Lipschitz ones). As a result, one might prefer the standard Lipschitz norm*

$$\|h\|_{\mathrm{Lip}} := \|h(\mathbf{0})\| + \underbrace{\sup_{x \neq y \in \mathscr{X}} \frac{\|h(x) - h(y)\|}{\|x - y\|}}_{\text{Lipschitz constant}}$$

for which $\mathrm{Lip}(\mathscr{X})^{n_x}$ *is complete and the* E_x *are also bounded:*

$$\begin{aligned}
\|E_x(h)\| &= \|h(x)\| \\
&= \|h(\mathbf{0}) + h(x) - h(\mathbf{0})\| \\
&\leq \|h(\mathbf{0})\| + \|h(x) - h(\mathbf{0})\| \\
&\leq \|h(\mathbf{0})\| + \|x\| \sup_{y \neq z \in \mathscr{X}} \frac{\|h(y) - h(z)\|}{\|y - z\|} \\
&\leq \underbrace{(1 + \|x\|)}_{C_x} \|h\|_{\mathrm{Lip}}
\end{aligned}$$

3.2 Piecewise affine functions for Lyapunov inference

In this section, we first try to augment the prior knowledge of stability in $\mathscr{X}_s$ to a bigger set $\mathscr{X}$ in Section 3.2.1. This result is refined to a Lyapunov candidate from the class of piece-wise affine (PWA) functions in in Section 3.2.2.

3.2.1 Lyapunov inference

Before proceeding to the stability condition, we introduce two additional concepts on functions $L : \mathbb{R}^{n_x} \to \mathbb{R}$. The sub-level set of L with level $\alpha \in \mathbb{R}$ is given by

$$L_{\leq \alpha} := \left\{x \in \mathbb{R}^{n_x} \mid L(x) \leq \alpha\right\}.$$

The Clarke generalized gradient of L at a point $x \in \mathbb{R}^{n_x}$ is the set given by

$$\partial_{\mathrm{Cl}} L(x) := \mathrm{co}\left\{ y \in \mathbb{R}^{n_x} : \begin{array}{l} \forall\, \epsilon > 0, \exists\, x_\epsilon \in \mathbb{R}^{n_x}; \\ \|x - x_\epsilon\| < \epsilon, \\ L \text{ is differentiable in } x_\epsilon, \\ \|y - \nabla L(x_\epsilon)\| < \epsilon \end{array} \right\}.$$

In [198, Theorem 2.5.1] it is proved that if L is Lipschitz continuous on a neighbourhood of x, then $\partial_{\mathrm{Cl}} L(x) \neq \emptyset$. The Clarke gradient is a generalized gradient in the sense that if L is continuously differentiable on a neighbourhood of x, then trivially $\partial_{\mathrm{Cl}} L(x) = \{\nabla L(x)\}$.

Now we summarize the Lyapunov stability condition on a compact set $\mathscr{X}$ in the following Lyapunov inference theorem.

Theorem 5. *Let Assumption 7 and 8 hold, and suppose that there exists a Lipschitz continuous function $L : \mathbb{R}^{n_x} \to \mathbb{R}$ such that:*

5.1) There exists $\alpha \in \mathbb{R}$ such that $\mathscr{X} = L_{\leq \alpha}$.

5.2) $f(x)^\top y < 0$ for all $x \in \mathscr{X} \setminus \mathscr{X}_s$, $y \in \partial_{\mathrm{Cl}} L(x)$.

Then for all $x_0 \in \mathrm{int}(\mathscr{X})$,

$$x(t|x_0) \xrightarrow[t\to\infty]{} \mathbf{0}$$

where $t \mapsto x(t|x_0)$ denotes the (unique by Lipschitz continuity of f – see Assumption 8) continuously differentiable solution to the Cauchy problem

$$\begin{cases} \dfrac{d}{dt} x(t|x_0) = f(x(t|x_0)) \\ x(0|x_0) = x_0. \end{cases}$$

Proof. The proof is similar to the proof of the Lasalle theorem [199]. Let $x_0 \in \mathrm{int}(\mathscr{X})$ and define the entering time

$$\tau(x_0) := \inf\{t \geq 0 : x(t|x_0) \in \mathscr{X}_s\}.$$

If $\tau(x_0) < \infty$, then $x(\tau(x_0)|x_0) \in \mathscr{X}_s$ and using the semigroup property and Assumption 7, one obtains $x(t + \tau(x_0)|x_0) = x(t|x(\tau(x_0)) \xrightarrow[t\to\infty]{} \mathbf{0}$ so that $x(t|x_0) \xrightarrow[t\to\infty]{} \mathbf{0}$.

We now proceed to the non-trivial case where $\tau(x_0) = \infty$, i.e. $\forall\ t \geq 0$, $x(t|x_0) \notin \mathscr{X}_s$, and consider the exit time $T := \inf\{t \geq 0 : x(t|x_0) \in \partial\mathscr{X}\}$. Notice that as $x_0 \in \mathrm{int}(\mathscr{X})$ and $t \mapsto x(t|x_0)$ is continuous, $T > 0$. Then, using Lipschitz continuity of L, condition 5.2) and [200, Lemma 2.15], for almost all $t \in [0, T)$, $\frac{d}{dt} L(x(t|x_0)) < 0$, so that $L(x(t|x_0))$ is decreasing on $[0, T)$ by the mean value inequality. This yields that $\forall\ t \in (0, T)$, one has $L(x(t|x_0)) < L(x_0) \leq \alpha$, and by condition 5.1) and continuity of $t \mapsto L(x(t|x_0))$, $T = \infty$ and thus $L(x(t|x_0))$ is decreasing on $[0,\infty)$ and for all $t \geq 0$, $x(t|x_0) \in \mathscr{X} \setminus \mathscr{X}_s$ regarding $\tau(x_0) = \infty$.

We now consider the compact set $\Omega := \overline{\mathscr{X} \setminus \mathscr{X}_s}$ from which the trajectory $x(t|x_0)$ never escapes. Since L is continuous and Ω is compact, by the Weierstrass extreme value theorem, one has that

$$-\infty < \ell := \inf_{\Omega} L,$$

so that the function $t \mapsto L(x(t|x_0))$ is decreasing and lower bounded, hence it has a limit $c := \lim_{t\to\infty} L(x(t|x_0)) \in \mathbb{R}$ that it does not attain in finite time.

We also consider the limit set

$$\Gamma(x_0) := \{p \in \Omega : \forall\, \epsilon, T > 0, \exists\, t \geq T;\, \|x(t|x_0) - p\| < \epsilon\},$$

that has the following property (see [199]): as $t \mapsto x(t|x_0)$ is bounded, $\Gamma(x_0) \subset \Omega$ is nonempty, compact and invariant (forward and backwards). Thus, $\exists\, p_0 \in \Gamma(x_0)$ and $\forall\, t \geq 0$, $x(t|p_0) \in \Gamma(x_0)$. Moreover, continuity of the function L and the definition of $\Gamma(x_0)$ ensure that for all $p \in \Gamma(x_0)$, $L(p) = c$, so that $\forall\, t \geq 0, L(x(t|p_0)) = c$. By Lipschitz continuity of L and $x(\cdot|p_0)$, we deduce that for almost all $t \geq 0$, $\frac{d}{dt}L(x(t|p_0)) = 0$. In addition to that, [200, Lemma 2.15] states that for almost all $t \geq 0$, $\exists\, y \in \partial_{\mathrm{Cl}} L(x(t|p_0))$ such that $0 = \frac{d}{dt}L(x(t|p_0)) = y^\top f(x(t|p_0))$. From these two points and condition 5.2) we deduce that for almost all $t \geq 0$, $x(t|p_0) \in \mathscr{X}_s$. Additionally, by definition of $\mathscr{X}_s$, $x(t|p_0) \underset{t\to\infty}{\longrightarrow} \mathbf{0}$, and by invariance and compactness of $\Gamma(x_0)$, we have $\mathbf{0} \in \Gamma(x_0)$.

Finally, we recall the definition of Assumption 7's local asymptotic stability condition:

$$\exists\, \epsilon > 0 \ s.t.\ \|x_1\| < \epsilon \Longrightarrow x(t|x_1) \underset{t\to\infty}{\longrightarrow} \mathbf{0}.$$

Moreover, by definition of $\Gamma(x_0) \ni \mathbf{0}$, for all $T > 0$ we are given a $t_0 \geq T$ such that $\|x(t_0|x_0)\| < \epsilon$. Those two observations, taking $x_1 = x(t_0|x_0)$, ensure that

$$x(t|x_0) \underset{t\to\infty}{\longrightarrow} \mathbf{0}.$$

. ■

Intuitively speaking, the proof of Theorem 5 considers two cases. The first part with finite entering time $\tau(x_0)$ concerns the case where $\mathbf{0} \in \mathrm{int}(\mathscr{X}_s)$. Meanwhile, the second part of the proof regarding $\tau(x_0) = \infty$ deals with the case where $\mathbf{0} \in \partial\mathscr{X}_s$. While so far numerical considerations lead us to limit to the former case, it is worth noticing that in theory Lyapunov inference can be performed even when the equilibrium point lies on the boundary of the prior region of attraction estimate (including the case $\mathscr{X}_s = \{\mathbf{0}\}$).

Remark 17. *The local asymptotic stability condition around* $\mathbf{0}$ *given in Assumption 7 is neces-*

sary. To see this, we construct a two-dimensional counter example:

$$\frac{dx}{dt} = \underbrace{\|x\|\,\|x-(1,0)\|^2}_{\text{normal speed}}(-x_2, x_1) + \underbrace{v_r(x)}_{\text{radial speed}}\ x \tag{3.2}$$

$$v_r(x) = \begin{cases} \frac{1}{\|x\|} - 1, & \|x\| \geq \frac{3}{4} \\ 1 - \frac{1}{2\|x\|}, & \frac{1}{4} \leq \|x\| < \frac{3}{4} \\ -1, & \|x\| < \frac{1}{4} \end{cases} \tag{3.3}$$

in the following, we would try to expand the ROA around $x_s = (1,0)$. Note that to shift this point to $\mathbf{0}$, one only has to do the change of variable $\tilde{x} = x - (1,0)$.

Now we start to analyse the behaviour of these dynamics in three cases (in addition to the trivial case of equilibrium point).

- *If $\|x_0\| = 1$, then $v_r(x_0) = 0$ so that $\|x(t)\|$ is constant (equal to 1) and the normal speed is positive unless / until $x(t) = (1,0)$.*
- *If $\|x_0\| = \frac{1}{2}$, again the radial speed is zero and the solution will stay on the circle centred at $\mathbf{0}$ with radius $\frac{1}{2}$, permanently rotating as the normal speed never vanishes. This is an unstable limit cycle.*
- *If $\|x_0\| > \frac{1}{2}$, then the radial speed will have same sign as $1 - \|x\|$ and $\|x(t)\|$ will converge to 1. In particular, if $\|x_0\| \geq \frac{3}{4}$, then $\|x(t|x_0)\| = 1 + (\|x_0\| - 1)e^{-t}$.*
- *If $\|x_0\| < \frac{1}{2}$, then the radial speed will stay negative and $x(t)$ will converge to $\mathbf{0}$. In particular, if $\|x_0\| < \frac{1}{4}$, then $\|x(t|x_0)\| = \|x_0\|e^{-t}$.*

The discussion above allows us to set $\mathscr{X}_s = \partial\mathscr{B}(0,1)$: it is true that $\forall x_0 \in \mathscr{X}_s$, $x(t|x_0) \xrightarrow[t\to\infty]{} (1,0)$. Assuming that $\mathscr{X} = \overline{\mathscr{B}}(0,\frac{5}{4}) \setminus \mathscr{B}(0,\frac{3}{4})$, with a Lyapunov candidate $L(x) = |\,\|x\| - 1\,|$, we have $L_{\leq\frac{1}{4}} = \mathscr{X}$, which fulfills sublevel set condition 1.1) in Theorem 5. Additionally, on $\mathscr{X} \setminus \mathscr{X}_s$, $\nabla L(x) = \operatorname{sgn}(\|x\| - 1)x$ and thus $f(x)^\top \nabla L(x) = -\|x\||1 - \|x\|| < 0$, satisfying condition 1.2) in Theorem 5. However, $\forall\ x_0 \in \mathscr{X} \setminus \mathscr{X}_s$, $x(t|x_0)$ never converges to $(1,0)$. Instead, it has a limit set $\Gamma(x_0) = \partial\mathscr{B}(0,1) \subset \partial\mathscr{X}_s$ (see proof of Theorem 5), and the trajectory will circulate forever around the $\partial\mathscr{B}(0,1)$ without convergence to any point (Fig. 3.1)

Remark 18. *Condition 5.1) is important. To see this, we can assume that $\mathscr{X} \subsetneq L_{\leq\alpha}$, by continuity of $L(x)$, there exists $x_{ext} \notin \mathscr{X}$ such that $L(x_{ext}) \leq \alpha$. For any $x_0 \in \mathscr{X} \setminus \mathscr{X}_s$, we can only ensure that the evaluation of $L(x)$ is decreasing, but it is possible to leave $\mathscr{X}$ heading towards x_{ext}. Note that our condition 2) in Theorem 5 only holds on $\mathscr{X}$, no stability guarantee can be given in this case.*

Remark 19. *Theorem 5 is a generalization of the classical Lyapunov-Massera local asymptotic stability theorem, included in it as the case where L is continuously differentiable and $\mathscr{X}_s = \{\mathbf{0}\}$. However, we do not assume that $L \geq 0$ with equality only satisfied in $\mathbf{0}$, as $\mathbf{0}$ is already assumed LAS.*

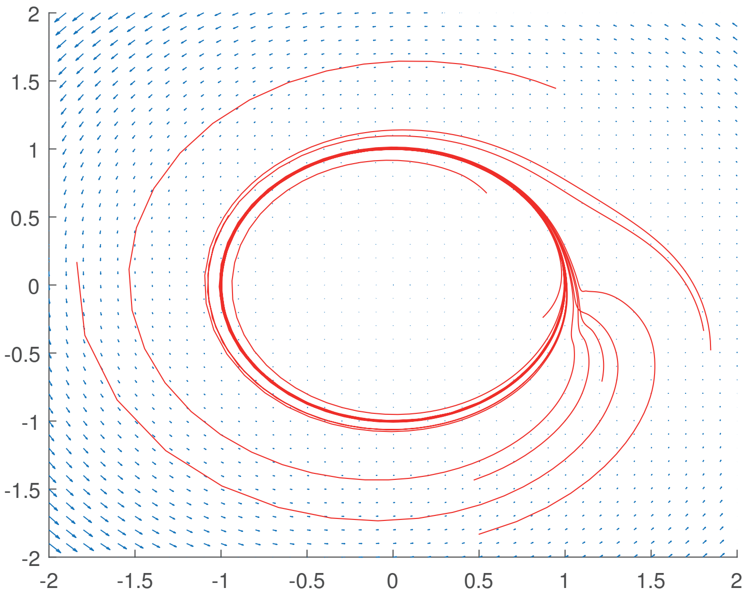


Figure 3.1: Vector field and trajectories with different initial points of dynamic system (3.2) that converges to $(1,0)$.

3.2.2 Piecewise Affine Lyapunov Function

PWA functions have strong modelling capability because they are dense in the space of continuous functions with a compact domain [201, Chapter 7.4]. This part will refine Theorem 5 to the Lipschitz continuous PWA Lyapunov function. For the sake of simplicity, we further assume

Assumption 9. *$\mathscr{X}$ is a polytope.*

When $\mathscr{X}$ is not a polytope, it can be inner-approximated by a polytope up to arbitrary accuracy, thus this assumption will not limit the application of the proposed analysis. Additionally, it is worth noting that the definition of polytope used in this thesis is not necessarily convex (see Section 3.1.2).

We now introduce our Lyapunov candidate under the form of a Lipschitz continuous PWA function. Let $\{C_k\}_{k=1}^{N_C}$ be an N_C-piece tessellation of $\mathscr{X}$ (i.e. $\cup_k C_k = \mathscr{X}$ and $\text{int}(C_k) \cap \text{int}(C_\ell) = \varnothing$ if $k \neq \ell$) where the C_k are convex polytopes (without loss of generality we take $C_1 \ni \mathbf{0}$). For $k \in \mathbb{N}_{N_C}$, we denote the vertices of C_k by $\{v_{j,k}\}_{j=1}^{|C_k|}$. Using this structure, a PWA Lyapunov candidate $L_{\mathscr{X}}$ is defined on $\mathscr{X}$ by

$$\forall k \in \mathbb{N}_{N_C}, x \in C_k,\ L_{\mathscr{X}}(x) = g_k^\top x + b_k\,. \tag{3.4}$$

With appropriate conditions on $g_k \in \mathbb{R}^{n_x}$, $b_k \in \mathbb{R}$, continuity of $L_{\mathscr{X}}$ on $\mathscr{X}$ is enforced: for any common vertex $v \in C_k \cap C_\ell$ (i.e. $\exists\, i \in \mathbb{N}_{|C_k|},\ j \in \mathbb{N}_{|C_\ell|}$ such that $v = v_{i,k} = v_{j,\ell}$), the condition

$$(g_k - g_\ell)^\top v = b_\ell - b_k \tag{3.5}$$

should hold. Then, $L_{\mathscr{X}}$ is Lipschitz continuous on $\mathscr{X}$ with

$$\|L_{\mathscr{X}}\|_{\text{Lip}} = |b_1| + \max_{1\le k\le N_C} \|g_k\| < \infty.$$

Remark 20. *Existence of a set $\{g_k, b_k\}_{k=1}^{N_C}$ such that* (3.4) *defines a continuous function is guaranteed for any tessellation of $\mathscr{X}$, as a constant function on $\mathscr{X}$ always satisfies the continuity condition (i.e. $g_k = \mathbf{0}$, $\forall\, k \in \mathbb{N}_{N_C}$ and $b_i = b_j$, $\forall\, i,\ j \in \mathbb{N}_{N_C}$). When the tessellation is defined by a power-diagram, existence of convex PWA functions is further guaranteed [202].*

Then, our PWA Lyapunov candidate $L_{\mathscr{X}}$ defined on $\mathscr{X}$ can be extended to the whole $\mathbb{R}^{n_x}$, such that Theorem 5 can be applied.:

$$L(x) := \begin{cases} \alpha + \eta\,\text{dist}(x, \mathscr{X}), & x \in \mathbb{R}^{n_x} \setminus \mathscr{X} \\ L_{\mathscr{X}}(x), & x \in \mathscr{X} \end{cases} \tag{3.6}$$

where $\eta := \max_{1\le k\le N_C} \|g_k\|$ is $L_{\mathscr{X}}$'s Lipschitz constant, and $\alpha \in \mathbb{R}$ is used to define the sublevel set (see Theorem 5) and can be tuned during the Lyapunov analysis.

Regarding our PWA LF candidate, the stability conditions in Theorem 5 can be restated as

Theorem 6. *Let Assumptions 7 to 9 hold, and consider the function $L_{\mathscr{X}}$ and L defined by* (3.4), (3.5) *and* (3.6). *If L moreover satisfies the following conditions*

6.1) α is such that the evaluation of L on the vertices satisfies

$$\forall\, j,k \text{ s.t. } v_{j,k} \in \mathscr{X} \setminus \partial\mathscr{X}, \qquad L(v_{j,k}) < \alpha \tag{3.7a}$$

$$\forall\, j,k \text{ s.t. } v_{j,k} \in \partial\mathscr{X}, \qquad L(v_{j,k}) = \alpha \tag{3.7b}$$

6.2) $\forall\, k \in \mathbb{N}_{N_C}$, $\forall\, x \in C_k \cap (\mathscr{X} \setminus \mathscr{X}_s)$,

$$f(x)^\top g_k < 0 \tag{3.8}$$

Then L is Lipschitz continuous and for all $x_0 \in \text{int}(\mathscr{X})$,

$$x(t|x_0) \underset{t\to\infty}{\longrightarrow} \mathbf{0}.$$

Proof. We first prove Lipschitz continuity of L. L is Lipschitz continuous on $\mathscr{X}$ by its definition (3.5). By (3.6), as the distance function is 1-Lipschitz, L is Lipschitz continuous on $\mathbb{R}^n \setminus \mathscr{X}$ with same Lipschitz constant as in $\mathscr{X}$. Then, to prove global Lipschitz continuity of L we only need to prove continuity at the boundary $\partial\mathscr{X}$, which is straightforward using condition (3.7b). Indeed, by construction of the C_k and Assumption 9, any $x \in \partial\mathscr{X}$ is a convex combination of some vertices $v_{j,k} \in \partial\mathscr{X}$: $\exists\, k \in \mathbb{N}_{N_C}, \{\lambda_j\}_{j=1}^{|C_k|} \subset [0,1]$ s.t.

$$v_{j,k} \in \operatorname{int}(\mathscr{X}) \Rightarrow \lambda_j = 0, \quad \sum_{j=1}^{|C_k|} \lambda_j = 1, \quad \sum_{j=1}^{|C_k|} \lambda_j v_{j,k} = x$$

so that $L(x) = g_k^\top x + b_k = \sum_j \lambda_j \underbrace{(g_k^\top v_{j,k} + b_k)}_{\alpha} = \alpha$, and $L = \alpha$ on $\partial\mathscr{X}$, which is consistent with the limit $\operatorname{dist}(x, \mathscr{X}) \to 0$ in (3.6).

We are now going to use Theorem 5 to complete our proof. We first check condition 5.1) : $\mathscr{X} = L_{\leq\alpha}$. As $\mathscr{X} \supset L_{\leq\alpha}$ is trivially deduced from (3.6), we focus on the converse inclusion. Let $x \in \mathscr{X}$ and let us prove that $x \in L_{\leq\alpha}$ i.e. $L(x) \leq \alpha$.

By construction, $\exists\, k \in \mathbb{N}_{N_C}$ s.t. $x \in C_k$, and thus $\exists\{\lambda_j\}_{j=1}^{|C_k|} \subset [0,1]$ s.t. $\sum_j \lambda_j = 1$ and $\sum_j \lambda_j v_{j,k} = x$. Again,

$$L(x) = \sum_{j=1}^{|C_k|} \lambda_j \underbrace{(g_k^\top v_{j,k} + b_k)}_{\leq\alpha} \leq \alpha$$

using condition 6.1), which is the announced result.

We then move on to condition 5.2) in Theorem 5. Let $x \in \mathscr{X} \setminus \mathscr{X}_s$ and define $K(x) := \{k \in \mathbb{N}_{N_C} \mid x \in C_k\}$.

If $x \in \operatorname{int}(\mathscr{X})$ and $K(x)$ has a single element k, then $x \in \operatorname{int}(C_k)$ and L is smooth on a neighbourhood of x with $\partial_{\mathrm{Cl}} L(x) = \{g_k\}$ and we conclude using condition 6.2).

Else, if $x \in \operatorname{int}(\mathscr{X})$ but $|K(x)| > 1$, we deduce from the previous point and the definition of the Clarke gradient that

$$\partial_{\mathrm{Cl}} L(x) = \operatorname{co}\{g_k \mid k \in K(x)\}$$

so that $\forall\; y \in \partial\, L(x)$, $\exists\{\lambda_k\}_{k\in K(x)} \subset [0,1]$ s.t.

$$\sum_{k\in K(x)} \lambda_k = 1 \text{ and } \sum_{k\in K(x)} \lambda_k g_k = y.$$

This yields that for such y

$$f(x)^\top y = \sum_{k\in K(x)} \lambda_k \underbrace{f(x)^\top g_k}_{<0} < 0$$

using condition 6.2).

Else, if $K(x)$ has a single element k but $x \in \partial\mathscr{X}$ (i.e. x is in the interior of a facet of $\mathscr{X}$), we notice that conditions (3.7a), (3.7b) enforce that L is constant equal to α on $C_k \cap \partial\mathscr{X}$ and less than α in $\operatorname{int}(C_k)$, which yields, using the definition (3.4), that g_k is orthogonal to $\partial\mathscr{X}$ in x and points outward of $\mathscr{X}$. Thus, the unit normal vector to $\partial\mathscr{X} \cap C_k$ at x pointing outward of $\mathscr{X}$ is given by

$$v(x) = \frac{g_k}{\|g_k\|}.$$

Let $\epsilon > 0$ and $x_\epsilon \in \mathbb{R}^{n_x} \setminus \mathscr{X}$ s.t. $\|x - x_\epsilon\| < \epsilon$. The definition (3.6) ensures that L is differentiable in x_ϵ and that

$$\nabla L(x_\epsilon) \underset{\epsilon \to 0}{\longrightarrow} \left(\max_{1 \le \ell \le N_C} \|g_\ell\| \right) v(x) = \left(\max_{1 \le \ell \le N_C} \|g_\ell\| \right) \frac{g_k}{\|g_k\|}.$$

denoting

$$\tilde{g}_k := \left(\max_{1 \le \ell \le N_C} \|g_\ell\| \right) \frac{g_k}{\|g_k\|} = \frac{\eta}{\|g_k\|}\, g_k,$$

we then deduce that $\partial_{\mathrm{Cl}} L(x) = \mathrm{co}\{g_k, \tilde{g}_k\} = [1, \eta/\|g_k\|]\, g_k$ and we conclude the proof using condition 6.2).

Eventually, in the last case where $|K(x)| > 2$ and $x \in \partial\mathscr{X}$ (i.e. x is on the boundary of a facet of $\mathscr{X}$), from the above we deduce that if $y \in \partial_{\mathrm{Cl}} L(x)$, then $\exists \{\lambda_k\}_{k \in K(x)}, \{\mu_k\}_{k \in K(x)} \subset [0,1]$ s.t. $\sum_{k \in K(x)} \lambda_k + \mu_k = 1$ and

$$y = \sum_{k \in K(x)} (\lambda_k\, g_k + \mu_k\, \tilde{g}_k) = \sum_{k \in K(x)} (\lambda_k + \frac{\eta}{\|g_k\|}\, \mu_k)\, g_k$$

and

$$f(x)^\top y = \sum_{k \in K(x)} (\lambda_k + \frac{\eta}{\|g_k\|}\, \mu_k) \underbrace{f(x)^\top g_k}_{<0} < 0$$

using condition 6.2). ■

Finally, we would wrap up this part by sorting out the logic flow in this theorectical Section 3.2 again. The ultimate goal is to extend some prior knowledge of RoA (i.e. $\mathscr{X}_s$) to a bigger set $\mathrm{int}(\mathscr{X})$ via PWA continuous function, which is not smooth. Theorem 5 gives this characterization with respect to a continuous Lyapunov candidate via its Clarke gradient evaluation within the set $\mathscr{X} \setminus \mathscr{X}_s$. A specific characterization based on a continuous PWA Lyapunov candidate is then summarized in Theorem 6. This theorem is useful as it allows us to define the Lyapunov candidate only on the region of interest (i.e. $\mathscr{X}$), while the general Thereom 5 requires the definition of the Lyapunov candidate on the whole state space. Moreover, Theorem 6 reformulates the stability analysis into the analysis on function evaluation on the vertices and the negativity test on each affine piece. Additionally, this negativity test is local with respect to each affine piece, which implies that a local refinement of the Lyapunov candidate $L(x)$ is possible, further discussion of this aspect will be given at the end of the following section.

3.3 Learning Robust PWA Lyapunov Function

Recall the ultimate goal of this part; we would like to learn a PWA Lyapunov function for an **unknown** Lipschitz dynamic $f(x)$ based on a given **fixed** dataset $D = \{x_i, f_i\}_{i=1}^{N_D}$. Regarding the underlying dynamic system, we further assume that an overestimate of the Lipschitz constant is given:

Assumption 10. *M is a known overestimate of the Lipschitz constant of f, i.e. for all $x, y \in \mathbb{R}^{n_x}$*

$$\|f(x) - f(y)\| \leq M\|x - y\|.$$

Remark 21. *In the case where $\mathscr{F}$ is a Hilbert space with inner product $\langle\cdot;\cdot\rangle_{\mathscr{F}}$, it is also possible to work with another assumption that might be more convenient in some settings, but introduces additional technicalities in the learning process, namely: M is a known overestimate of the Hilbert norm of f, i.e.*

$$\|f\|_{\mathscr{F}} = \sqrt{\langle f; f\rangle_{\mathscr{F}}} \leq M.$$

In order to formulate a tractable learning problem, we further assume:

Assumption 11. *The tessellation of $\overline{\mathscr{X} \setminus \mathscr{X}_s}$ is fixed by a given set $\{C_k\}_{k=1}^{N_C}$.*

The discussion about why we make this assumption is postponed to Remark 23. In this Section, we will show how to learn the PWA Lyapunov function on the tessellation $\{C_k\}_{k=1}^{N_C}$ based on Theorem 6. In the following subsections, we will gradually develop a Lyapunov function learning scheme under the form of an optimization problem.

Subsection 3.3.1 studies the negative condition for a general hypothesis space, and Subsection 3.3.2 studies the condition in the most basic hypothesis space, i.e. the Lipschitz function space.

3.3.1 Robust Lyapunov Condition

Although condition (3.8) is local to each affine piece, it still poses a numerically intractable infinite dimensional constraint, especially when the dynamic system is unknown and uncertain as $f \in \mathscr{F}$. In this subsection, we will show how this condition can be relaxed to a more tractable form via the calculus of the infimal convolution.

Based on dynamic system evaluation $f_i := f(x_i)$ on location x_i in the dataset D, the hypothesis space of the underlying dynamic system is tightened to

$$\mathscr{F}_D := \{h \in \mathscr{F} \mid E_{x_i} h = f_i, \forall\, i \in \mathbb{N}_{N_D}\}\,, \tag{3.9}$$

where N_D is the number of the collected data points. Accordingly, we further define

$$\mathscr{F}_{D,i} := \{h \in \mathscr{F} \mid E_{x_i} = f_i\}\,, \tag{3.10}$$

such that

$$\mathscr{F}_D = \cap_i \mathscr{F}_{D,i}\,. \tag{3.11}$$

The following theorem is the cornerstone representing the infinite dimensional Lyapunov constraint with a finite number of constraints.

Theorem 7. *The condition* $0 > g_k^\top f(x)$ *holds for any* $x \in C_k$ *if there exist a set* $\{\tilde{g}_{i,k}\}_{i=1}^{N_D} \subset \mathbb{R}^{n_x}$ *such that*

$$\sum_{i=1}^{N_D} \tilde{g}_{i,k} = g_k, \textit{ and } \sum_{i=1}^{N_D} \max_{x \in C_k} \iota^*_{\mathscr{F}_{D,i}}(E_x^* \tilde{g}_{i,k}) < 0. \tag{3.12}$$

Proof. The key idea in this proof is to view the negativity condition (3.8) as an evaluation of a conjugate function in direction g_k:

$$\begin{aligned}
0 > \max_{x \in C_k} \max_{f \in \mathscr{F}_D} g_k^T f(x) \quad & \overset{(a)}{\Longleftrightarrow} 0 > \max_{x \in C} \max_{f \in \mathscr{F}} g_k^T (E_x f) - \iota_{\mathscr{F}_D}(f) \\
& \overset{(b)}{\Longleftrightarrow} 0 > \max_{x \in C_k} \iota^*_{\mathscr{F}_D}(E_x^* g_k) \\
& \overset{(c)}{\Longleftrightarrow} 0 > \max_{x \in C_k} \iota^*_{\cap_i \mathscr{F}_{D,i}}(E_x^* g_k) \\
& \overset{(d)}{\Longleftrightarrow} 0 > \max_{x \in C_k} \left(\sum_{i=1}^{N_D} \iota_{\mathscr{F}_{D,i}}(E_x^* g_k) \right)^* \\
& \overset{(e)}{\Longleftrightarrow} 0 > \max_{x \in C_k} \#_i \iota^*_{\mathscr{F}_{D,i}}(E_x g_k) \\
& \overset{(f)}{\Longleftrightarrow} 0 > \max_{x \in C_k} \inf_{\sum \tilde{g}_{i,k} = g_k} \sum_i^{N_D} \iota^*_{\mathscr{F}_{D,i}}(E_x \tilde{g}_{i,k}) \\
& \overset{(g)}{\Longleftarrow} 0 > \inf_{\sum \tilde{g}_{i,k} = g_k} \max_{x \in C_k} \sum_i^{N_D} \iota^*_{\mathscr{F}_{D,i}}(E_x \tilde{g}_{i,k}) \\
& \overset{(h)}{\Longleftrightarrow} \exists \sum_{i=1}^{N_D} \tilde{g}_{i,k} = g_k \text{ s.t. } \max_{x \in C_k} \sum_i^{N_D} \iota^*_{\mathscr{F}_{D,i}}(E_x^* \tilde{g}_{i,k}) < 0.
\end{aligned}$$

(a) writes the feasible set $\mathscr{F}_D$ into the objective function by indicator function, and (b) follows the definition of the convex conjugate (see Section 3.1.1). (c) applies the decomposition given in (3.11), and (d) applies the calculus of indicator functions given in (3.1). (e) applies calculus of conjugate function in Proposition 4, whose explicit form is written in (f). (g) applies the min-max inequality [76, Chapter 3.14]. Finally, as often performed in robust optimization (see e.g. [203]), the infimum operator is replaced by an existence assertion in (h). ■

The key concept behind Theorem 7 is the decomposition of the hypothesis space in (3.11). In particular, $\iota^*_{\mathscr{F}_{D,i}}(E_x^* \tilde{g}_{i,k}) = \max_{f \in \mathscr{F}_{D,i}} g_{i,k}^\top f(x)$ is related to the uncertainty quantified from one data point, which usually has an easy-to-evaluate explicit closed solution. In comparison, the explicit solution is usually not available or difficult to evaluate when the whole dataset D is considered. For example, when $\mathscr{F}$ is the space of Lipschitz functions, then the uncertainty boundary quantified by one data point defines a shifted cone. However, the uncertainty upper and lower bounds are PWA and non-trivial to evaluate [204] when the whole dataset D

is used. Recall Assumption 8, it is also reasonable to consider a reproducing kernel Hilbert space (RKHS)[1], which underpins various uncertainty quantification methods such as Gaussian process regression [205] and deterministic error bound methods [206]. All these methods require computing the inverse of the Gram matrix or solving a second order cone program, which has an easy-to-evaluate explicit solution only when one data point is considered.

Remark 22. *In this part, we consider a decomposition of the hypothesis space $\mathscr{F}_D$ into the intersection of single-data models, i.e. $\mathscr{F}_D = \cap \mathscr{F}_{D,i}$. It is also possible to generalize the result to other decomposition, dubbed $\{\mathscr{F}_j\}$. If $\mathscr{F} = \cap_j \mathscr{F}_j$, where the conclusion in Theorem 7 needs slight modification accordingly:*

$$\sum_j \tilde{g}_{j,k} = g_k,\ 0 > \sum_j \max_{x \in C_k} \iota^*_{\mathscr{F}_j}(E^*_x \tilde{g}_{j,k}) \,.$$

Remark 23. *Even though function evaluation on a fixed location defines a linear operator from the hypothesis space $\mathscr{F}$ to $\mathbb{R}^{n_x}$, the mapping from the evaluation point to this operator is in general nonlinear. Assumption 11 is posed to develop a tractable formulation by avoiding this nonlinear mapping, and the next subsection will make use of this property. It is also noteworthy that, with a fixed tessellation, the parameters of the Lyapunov candidate on each affine piece (i.e. g_k, b_k on C_k) can be uniquely determined by the function evaluation on the vertices.*

Another main benefit of a fixed tessellation is that it allows a direct control over the model complexity of the Lyapunov candidate. In particular, consider two Lyapunov candidates $L_1(x)$ and $L_2(x)$ with their corresponding partitions $\{C_{1,k}\}$ and $\{C_{2,k}\}$. Then, we can state that $L_1(x)$ is a refinement of $L_2(x)$ (i.e. $L_1(x)$ has a higher degree of modelling capability than $L_2(x)$) if $\forall\ C_{2,k}$, $\exists \{C_{1,j}\}_{j \in \mathscr{I}_k}$ such that $\cup_j C_{1,j} = C_{2,k}$. As condition (3.8) *is local to each affine piece, if one affine piece C_k violates the assumptions of Theorem 7, then we can refine the model locally by further partition this piece.*

3.3.2 A Convex Tractable Case: Lipschitz Dynamics

Theorem 7 gives us a representation of condition (3.8), but such a representation remains abstract and hard to check numerically; thus, we will now recast this representation under a tractable form, in a specific case. Although we have discussed that it is possible to consider a more complex function space property on top of the Lipschitz property in Section 3.3.1, this section will show that a convex learning problem exists even when we consider the most basic hypothesis space:

$$\mathscr{F} = \mathrm{Lip}(\mathscr{X})^{n_x} \tag{3.13}$$

In such case, the following corollary holds:

[1] A reproducing kernel Hilbert space (**RKHS**) over a set X is a Hilbert space of functions from X to $\mathbb{R}$ such that for each $x \in X$, the evaluation functional $E_x g := g(x)$ is bounded [156]

Corollary 4. *The condition* $0 > g_k^\top f(x)$ *holds for any* $x \in C_k$ *if there exists a set* $\{\tilde{g}_{i,k}\}_{i=1}^{N_D} \subset \mathbb{R}^{n_x}$ *such that for all* $j \in \mathbb{N}_{|C_k|}$

$$\sum_{i=1}^{N_D} \tilde{g}_{i,k} = g_k, \text{ and } \sum_{i=1}^{N_D} \tilde{g}_{i,k}^\top f_i + \|\tilde{g}_{i,k}\| M \| v_{j,k} - x_i \| < 0. \tag{3.14}$$

Proof.

$$\begin{aligned}
\forall\, j, 0 > \sum_{i=1}^{N_D} \tilde{g}_{i,k}^\top f_i + \|\tilde{g}_{i,k}\| M \| v_{j,k} - x_i \| \quad & \overset{(a)}{\iff} 0 > \max_{x \in C_k} \sum_{i=1}^{N_D} \tilde{g}_{i,k}^\top f_i + \|\tilde{g}_{i,k}\| M \| x - x_i \| \\
& \overset{(b)}{\iff} 0 > \max_{x \in C_k} \sum_{i=1}^{N_D} \max_{y_i \in \mathscr{B}(f_i, M\|x - x_i\|)} \tilde{g}_{i,k}^\top y_i \\
& \overset{(c)}{\implies} 0 > \max_{x \in C_k} \sum_{i=1}^{N_D} \max_{f \in \mathscr{F}_{D,i}} \tilde{g}_{i,k}^\top f(x) \\
& \overset{(d)}{\iff} 0 > \max_{x \in C_k} \sum_{i=1}^{N_D} \iota_{\mathscr{F}_{D,i}}^* (E_x^* \tilde{g}_{i,k}) \, .
\end{aligned} \tag{3.15}$$

To show (a), we notice that the right-hand side of (3.15) is a convex maximization problem over a bounded convex polytope, its optimal solution is attained on its vertices, i.e. $\{v_{j,k}\}$. (b) uses the Cauchy-Schwarz inequality on the second term of the sum $\tilde{g}_{i,k}^\top y_i = \tilde{g}_{i,k}^\top f_i + \tilde{g}_{i,k}^\top (y_i - f_i)$. (c) follows the assumption of Lipschitz constant overestimate (Assumption 10). Finally, (d) applies the definition of the conjugate function. ■

Remark 24. *If instead of Assumption 10, we suppose that* $\|f\|_{\mathscr{F}} \le M$, *then one has a bound function* $B_M : \mathbb{R}^{n_x} \to \mathbb{R}_+$ *(depending on M) such that for any* $x \in \mathbb{R}^{n_x}$, $h^\star \in \operatorname{argmin}_{h \in \mathscr{F}_D} \|h\|_{\mathscr{F}}$,

$$\|f(x) - h^\star(x)\| \le B_M(x),$$

so that in the previous proof one has to replace $M\|x - x_i\|$ *with* $B_M(x)$. *However, the problem here is that* B_M *is not convex, so that relation* (b) *does not hold anymore, and we would need other arguments (out of the scope of this article) to obtain a finite dimensional constraint.*

Using Theorem 6 and Corollary 4, we get the following conditions for int($\mathscr{X}$) to be a positively invariant subset of the region of attraction of our unknown system:

$$
\begin{aligned}
&\exists\, \{\tilde{g}_{i,k} \mid i \in \mathbb{N}_{N_D}, k \in \mathbb{N}_{N_C}\} \subset \mathbb{R}^{n_x}, && \exists\, \{g_k \mid k \in \mathbb{N}_{N_C}\} \subset \mathbb{R}^{n_x}, \\
&\exists\, \{b_k \mid k \in \mathbb{N}_{N_C}\} \subset \mathbb{R}, && \exists\, \alpha \in \mathbb{R} \\
&\forall\, k, \ell \in \mathbb{N}_{N_C}, j \in \mathbb{N}_{|C_k|} \text{ s.t. } v_{j,k} \in C_\ell, && (g_k - g_\ell)^\top v_{j,k} = b_\ell - b_k \qquad (3.5) \\
&\forall\, k \in \mathbb{N}_{N_C}, j \in \mathbb{N}_{|C_k|}, \text{s.t. } v_{j,k} \in \mathscr{X} \setminus \partial\mathscr{X} && g_k^\top v_{j,k} + b_k < \alpha \qquad (3.7a) \\
&\forall\, k \in \mathbb{N}_{N_C}, j \in \mathbb{N}_{|C_k|}, \text{s.t. } v_{j,k} \in \partial\mathscr{X} && g_k^\top v_{j,k} + b_k = \alpha \qquad (3.7b) \\
&\forall\, k \in \mathbb{N}_{N_C}, && \sum_{i=1}^{N_D} \tilde{g}_{i,k} = g_k \qquad (3.12) \\
&\forall\, k \in \mathbb{N}_{N_C}, j \in \mathbb{N}_{|C_k|}, && \sum_{i=1}^{N_D} \tilde{g}_{i,k}^\top f_i + \|\tilde{g}_{i,k}\| M \|v_{j,k} - x_i\| < 0. \qquad (3.14)
\end{aligned}
$$

Hence, our Lyapunov inference now boils down to a finite number of equality and strict inequality tests. Regarding condition (3.14) for a fixed $k \in \mathbb{N}_{N_C}$, we introduce slack variables to transform this certification problem into an optimization problem, with optimality giving the best robust certificates possible. This results in the following optimization problem:

$$
\begin{aligned}
s^\star_{\alpha,\epsilon} := \min_{\{g_k, b_k\}, \{\tilde{g}_{i,k}\}\{s_{j,k}\}} & \textstyle\sum_{k=1}^{N_C} \sum_{j=1}^{|C_k|} s_{j,k} \\
\forall\, k, \ell \in \mathbb{N}_{N_C}, j \in \mathbb{N}_{|C_k|}, \quad & s_{j,k} \geq -\epsilon && (3.16a) \\
v_{j,k} \in C_\ell \implies \quad & (g_k - g_\ell)^\top v_{j,k} = b_\ell - b_k && (3.16b) \\
v_{j,k} \in \mathscr{X} \setminus \partial\mathscr{X} \implies \quad & g_k^\top v_{j,k} + b_k \leq \alpha - \epsilon && (3.16c) \\
v_{j,k} \in \partial\mathscr{X} \implies \quad & g_k^\top v_{j,k} + b_k = \alpha && (3.16d) \\
& \textstyle\sum_{i=1}^{N_D} \tilde{g}_{i,k} = g_k && (3.16e) \\
& \textstyle\sum_{i=1}^{N_D} \tilde{g}_{i,k}^\top f_{i_m} + \|\tilde{g}_{i,k}\| M \|v_{j,k} - x_{i_m}\| \leq s_{j,k} && (3.16f)
\end{aligned}
$$

where $\epsilon > 0$ is a user-defined negativity tolerance and $\alpha \in \mathbb{R}$ is the user-defined maximal value of the Lyapunov function. Constraint (3.16b) is the continuity condition, and constraint (3.16c) is the interior condition (3.7a) and constraint (3.16d) is the boundary condition (3.7b), both stated in Theorem 6. When the slack variables satisfy $s_{j,k} < 0$, constraints (3.16e) and (3.16f) correspond to the negative condition (3.8) stated in Theorem 6. Optimization problem (3.16) comes with the following result:

Theorem 8. *$\forall\, h \in \mathscr{F}$ whose function evaluations are consistent with the unknown underlying dynamic system f ($h(x_i) = f_i$, $\forall\, x_i, f_i \in D$), the solution to problem* (3.16) *defines a Lyapunov function for dynamic system h on $\mathscr{X}$ when its optimal value verifies $s^\star_{\alpha,\epsilon} = -\epsilon \sum_{k=1}^{N_C} |C_k|$.*

Proof. We need to show that an optimal solution $\{g_k^{\text{opt}}, b_k^{\text{opt}}, \tilde{g}_{i,k}^{\text{opt}}, s_{j,k}^{\text{opt}}\}$ satisfying

$$\sum_{j,k} s_{j,k}^{\text{opt}} = -\epsilon \sum_{k} |C_k|$$

will also satisfy the conditions given in Theorem 6.

First of all, by constraint (3.16b) the learnt PWA function is continuous (see Equation (3.4)). On top of this, by constraints (3.16d) and (3.16c), a solution to problem (3.16) recovers the condition 6.1) in Theorem 6. In the rest of this proof, we need to show that constraints (3.16e) and (3.16f) are equivalent to the negative condition 6.2) in Theorem 6 (i.e. $0 > g_k^\top f(x)$, $\forall\, x \in C_k$).

Note that, by $s_{j,k} \geq -\epsilon$ (constraint (3.16a)), the optimal value $\sum\limits_{j,k} s_{j,k}^{\text{opt}} = -\epsilon \sum_{k=1}^{N_C} |C_k|$ implies that $s_{j,k} = -\epsilon$, $\forall\, j \in \mathbb{N}_{|C_k|}$, $k \in \mathbb{N}_{N_C}$. Therefore, we have

$$\begin{aligned} & 0 > -\epsilon \geq \sum_{m=1}^{N_D} \tilde{g}_{i_m,k}^\top f_{i_m} + M \|\tilde{g}_{i_m,k}\| \|v_{j,k} - x_{i_m}\| \\ \overset{(a)}{\Longrightarrow}\ & 0 > \sum_{i=1}^{N_D} \tilde{g}_{i,k}^\top f_i + M \|\tilde{g}_{i,k}\| \|v_{j,k} - x_i\| \\ \overset{(b)}{\Longrightarrow}\ & 0 > g_k^\top h(x) \quad \forall\, x \in C_k,\ h \in \mathscr{F}_D\,, \end{aligned}$$

where (a) holds by taking $\tilde{g}_{i,k} = \mathbf{0}$, $\forall\, i$ such that $(x_i, f_i) \notin D$ and (b) is Corollary 4. Based on this, we can derive:

In summary, constraints (3.16f) and (3.16e) guarantees the satisfaction of condition 6.2) in Theorem 6 for any $h \in \mathscr{F}_D$. Hence, we conclude the proof. ■

The implication of this theorem is strong as it states that under the assumption of Lipschitz dynamics, we can learn/validate a LF by convex programming (3.16) even when the unknown underlying dynamic system f is nonlinear.

3.3.3 Comparison with related works

We would like to wrap up this subsection by comparing the proposed learning scheme with other existing methods. In comparison with other PWA Lyapunov function based methods (see e.g. [193, 195]), the proposed scheme shows two major differences. First, the location of the samples and the tessellation of the PWA Lyapunov candidate are decoupled in the proposed scheme. While in existing methods, the data are sampled on the vertices of the tessellation, therefore, the data locations are usually structural due to the choice of the tessellation. Secondly, the robust Lyapunov stability conditions considered in the existing methods only consider the model uncertainty quantified by one data point. On the contrary, the proposed scheme synthetically makes use of the uncertainty quantified by each data point while

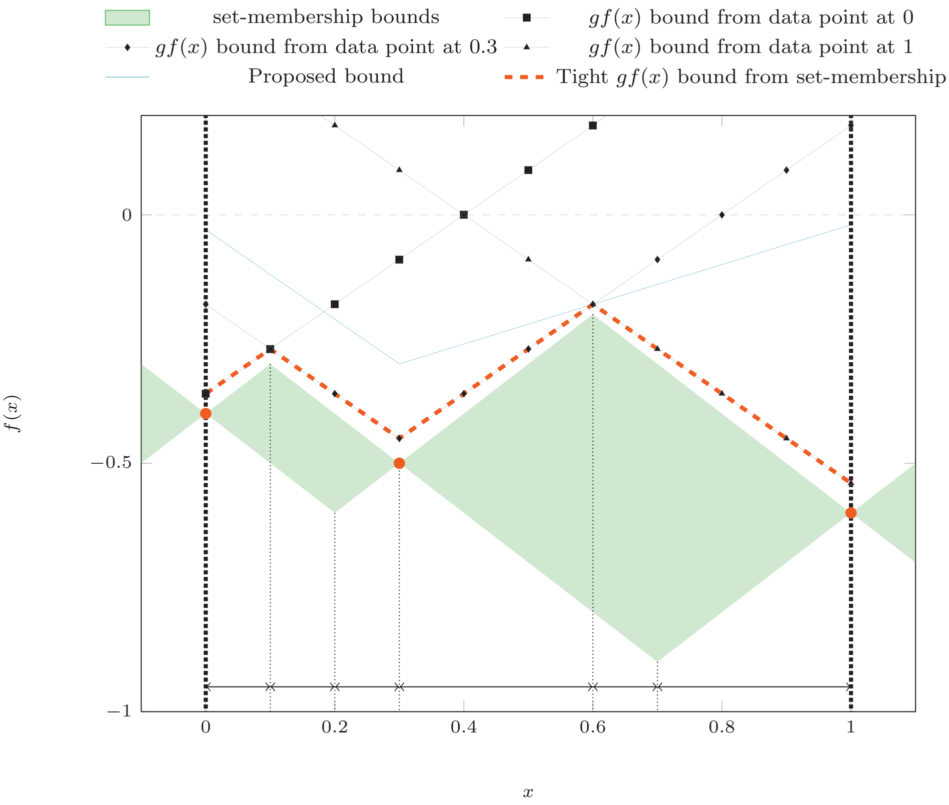


Figure 3.2: Comparison between set membership method and the proposed method. The proposed scheme is evaluated by $\tilde{g}_1 = 0, \tilde{g}_2 = 0.65, \tilde{g}_3 = 0.35$.

maintaining a convex tractable structure.

Another framework related to the proposed approach is the set-membership method [204]. In short, the set-membership method models the set of Lipschitz functions whose evaluation on the points $\{x_i\}$ are consistent with data $\{x_i, f(x_i)\}$. When $n_x = 1$, the evaluation upper/lower bounds given by this method are PWA. To better demonstrate the difference between the proposed scheme and the set-membership method, we consider a specific example in $\mathbb{R}$ (Figure 3.2), whose Lipschitz overestimate is set to $M = 1$ and the data points are:

$$\{\big(0, f(0) = -0.4\big), \big(0.3, f(0.3) = -0.5\big), \big(1, f(1) = -0.6\big)\}$$

Now, consider a Lyapunov function candidate $L(x) = gx$ with $g = 0.9$ within interval $[0,1]$. If the evaluation bounds of set-membership method are used, the Lyapunov decreasing condition needs to be examinated in all the sub-intervals generated by the PWA bounds (plotted as two-headed arrow in Figure 3.2). Determination of these sub-intervals is computationally heavy. Instead, if we hope to simplify the analysis by only taking one data point into consideration, none of the simple model generated by one data point can justify the Lyapunov decreasing condition (see black lines of different markers in Figure 3.2). These two aspects together imply the use of the whole dataset is necessary. The proposed scheme synthesizes the knowledge of simple models via a convex optimization. One optimal solution to the proposed scheme is plotted as a blue line in Figure 3.2, which only utilizes the last two points (i.e. $\tilde{g}_1 = 0$). It is worth noting that, in this example, if we only consider the left data and the right data point, the Lyapunov deceasing condition will fail even when the set-membership method is used. Hence, we can observe that the proposed method is able to search for the data points that are relevant to the Lyapunov decreasing condition. Additionally, this process is done by polynomial time convex optimization algorithms [207]. On the contrary, even though the set-membership method gives the tightest bound, checking the Lyapunov decreasing condition with these bounds is NP-hard, as it requires vertex elimination of the Voronoi cells.

Remark 25. *Note that in the proof of Theorem 4, we use the Cauchy-Schwarz inequality in* (3.15)*, that holds only with Euclidean 2-norm. Actually, other norms can be considered, and the resulting problem* (3.16) *will have different properties accordingly. In particular, if 1-norm or ∞-norm is used, the resulting problem is a linear program. For the sake of simplicity, the thesis uses the Euclidean norm only, and as a result the learning problem* (3.16) *is a second order cone programming (SOCP) (Details in Section. 3.4).*

3.4 Algorithm Development

After the introduction of the Lyapunov learning problem (3.16), we will discuss its learnability in Section 3.4.1. The original learning problem (3.16) will be recast to an equivalent but numerically more efficient form in Section 3.4.2. In the end, the main algorithms are summarized in Section 3.4.3.

3.4.1 Learnability

Above all, the set of Lyapunov functions is closed under positive scaling. More specifically, if a Lyapunov function $L(x)$ is learnt from problem (3.16), then its positive scaling $\lambda L(x)$ with $\lambda > 0$ is also a Lyapunov function. Thus, the user-defined value α in (3.16c) will not introduce conservativeness in the learning problem. Furthermore, due to the introduction of the slack variables $\{s_{j,k}\}$, the learning problem (3.16) is always feasible. It is natural to ask the following core question in the limiting case:

If $\mathscr{X}$ is the RoA of the underlying dynamic system f and we are allowed to evaluate the dynamic system f in any finite set of points $x \in \mathscr{X}$, can we always learn a Lyapunov function for any RoA subsets $\mathscr{X} \setminus \mathscr{X}_s$?

Unfortunately, the answer to this question is **no**, and a counterexample is summarized in the following Corollary:

Corollary 5. *Let Assumption 7 holds and let the hypothesis space $\mathscr{F}$ satisfy Equation* (3.13)*, if $\mathbf{0} \notin \operatorname{int}(\mathscr{X}_s)$, then the optimal value to problem* (3.16) *verifies:*

$$\sum_{j,k} s_{j,k}^{\text{opt}} > -\epsilon \sum_{k=1}^{N_C} |C_k|$$

Proof. Note that the user-defined tolerance is negative (i.e. $-\epsilon < 0$) and arbitrary, we need to show that there exist one $s_{j,k}^{\text{opt}} \geq 0$ in the optimal solution.

By Assumption 7, we have $f(\mathbf{0}) = \mathbf{0}$. Using the Lipschitz condition and Assumption 10 yields

$$\|f_i\| = \|f_i - \mathbf{0}\| \leq M\|x_i - \mathbf{0}\|$$

As $\mathbf{0}$ is the LAS EP, we have $\mathbf{0} \in \mathscr{X}_s$, and by assumption $\mathbf{0} \notin \operatorname{int}(\mathscr{X}_s)$, so $\mathbf{0} \in \partial\mathscr{X}_s$. Consider the affine piece C_k that contains $\mathbf{0}$ (i.e. $\mathbf{0} \in C_k$ and it is a vertex of C_k). By inspecting the constraint (3.14) and using Cauchy-Schwarz inequality, one has:

$$\begin{aligned} &\tilde{g}_{i,k}^{\top} f_i + M\|\tilde{g}_{i,k}\|\|\mathbf{0} - x_i\| \\ \geq &M\|\tilde{g}_i\|\|\mathbf{0} - x_i\| - \|\tilde{g}_i\|\|f_i\| \geq 0\,. \end{aligned}$$

Thus, the constraint (3.16f) will become:

$$s_{j,k} \geq \sum_{i=1}^{N_D} \tilde{g}_{i,k}^{\top} f_i + M\|\tilde{g}_{i,k}\|\|v_{j,k} - x_i\| \geq 0 > -\epsilon,$$

which is sufficient to conclude the proof. ■

Even though our theory of expanding a-priori knowledge from $\mathscr{X}_s$ to a bigger set $\mathscr{X}$ holds for any $\mathscr{X}_s$ (Section 3.2), Corollary 5 shows that, if the domain of a Lyapunov function contains

$\mathbf{0}$, then it is impossible to learn this function by only assuming Lipschitz continuity (Equation (3.13)). Similar observation of infeasibility was also made in the constructive proof of converse Lyapunov theorem [208]. Meanwhile, many numerical methods also have similar limitation around the invariant set [209, Chapter 2.11]; we refer the interested reader to [210, 211] for more details. In summary, if we do not further assume other function space structure on the hypothesis space $\mathscr{F}$, following assumption is required to ensure the learnability of problem (3.16):

Assumption 12. $\mathbf{0} \in \mathrm{int}(\mathscr{X}_s)$.

We would stress again that the assumption above is only necessary for the learning scheme based on problem (3.16), and a learning scheme without this assumption is left for future research.

After answering the aforementioned problem by Corollary 5, the follow-up core question is:

Given an RoA prior estimate $\mathscr{X}_s$, what condition on the dataset D should hold to enable learning the PWA Lyapunov function on $\mathscr{X} \setminus \mathscr{X}_s$?

Obviously, it is impossible to answer this question with a sufficient condition, regarding the arbitrariness of the unknown dynamic system f. However, if we only assume the function space to be Lipschitz (Equation (3.13)), we can still give an initial check of the learnability of problem (3.16). In order to discuss this necessary condition, we first define

$$r_i := \frac{\|f_i\|}{M} .$$

We state the necessary condition as follows:

Lemma 7. *Let Assumption 7 hold and let hypothesis space $\mathscr{F} = \mathrm{Lip}(\mathscr{X})^{n_x}$. If solutions to problem* (3.16) *define a Lyapunov function as in Theorem 6, then $\mathscr{X} \setminus \mathscr{X}_s \subset \cup_{i=1}^{N_D} \mathscr{B}(x_i, r_i)$.*

Proof. From the proof of Theorem 8, the solution to problem (3.16) is a Lyapunov function if for $\forall\ k \in \mathbb{N}_{N_C}$, $j \in \mathbb{N}_{|C_k|}$, $s_{j,k} = -\epsilon$. Let $x \in \mathscr{X} \setminus \mathscr{X}_s$, and without loss of generality, we suppose $x \in C_k$ for some $k \in \mathbb{N}_{N_C}$. By Corollary 4,

$$0 > -\epsilon = s_{j,k} \geq \sum_{m=1}^{N_D} \tilde{g}_{i_m,k}^{\top} f_{i_m} + M\|\tilde{g}_{i_m,k}\| \|x - x_{i_m}\| ,$$

which implies that there exist at least one $n \in \mathbb{N}_{N_d}$ such that

$$\begin{aligned} & 0 > \tilde{g}_{n,k}^{\top} f_n + M\|\tilde{g}_{n,k}\|\|x - x_n\| \\ \Longrightarrow & |\tilde{g}_{n,k}^{\top} f_n| > M\|\tilde{g}_{n,k}\|\|x - x_n\| \\ \overset{(a)}{\Longrightarrow} & \|f_n\|\|\tilde{g}_{n,k}\| > M\|\tilde{g}_{n,k}\|\|x - x_n\| \\ \Longleftrightarrow & x \in \mathscr{B}\left(x_n, \frac{\|f_n\|}{M}\right) \\ \Longleftrightarrow & x \in \mathscr{B}(x_n, r_n) \\ \Longrightarrow & x \in \cup_{i=1}^{N_D} \mathscr{B}(x_i, r_i) \end{aligned}$$

where (a) follows from the Cauchy-Schwarz inequality. Due to the inclusion condition is sastisfied for any point $x \in \mathscr{X} \setminus \mathscr{X}_s$, we conclude the proof with $\mathscr{X} \setminus \mathscr{X}_s \subset \cup_{i}^{N_D} \mathscr{B}(x_i, r_i)$. ■

This lemma shows the connection between learning a PWA Lyapunov function and the set covering problem, which was proved to be equivalent to a non-convex semi-infinite problem [212], and thus one should not try to check the condition in Lemma 7 numerically. Recall a key idea behind the problem (3.16): the global analysis on $\mathscr{X} \setminus \mathscr{X}_s$ is reduced to the analysis on the vertices. This inspires us to relax the continuous set covering problem to the covering problem of the vertices and we state this condition in the following Corrolary

Corollary 6. *Let Assumption 7 hold and let hypothesis space $\mathscr{F} = \mathrm{Lip}(\mathscr{X})^{n_x}$, if solutions to problem* (3.16) *define a Lyapunov function as in Theorem 6, then*

$$\{v_{i,k}\}_{\substack{1 \le k \le N_C \\ 1 \le i \le |C_k|}} \subset \cup_{i=1}^{N_D} \mathscr{B}(x_i, r_i)\,, \tag{3.17}$$

This necessary condition (3.17) can be checked in polynomial time. If this test fails, it means that there exist $v_{i,k} \notin \cup_{i=1}^{N_D} \mathscr{B}(x_i, r_i)$, and therefore, the data is not informative enough to learn a PWA Lyapunov function by only assuming the Lipschitz condition (Equation (3.13)). Accordingly, the learning process will be terminated. Intuitively, the points which $v_{i,k} \notin \cup_{i=1}^{N_D} \mathscr{B}(x_i, r_i)$ should suggest the location where additional samples are required.

3.4.2 Computationally efficient recasting

In this part, we will discuss how we recast the original problem (3.16) to an equivalent problem that can be handled numerically more efficiently.

Data Refinement

One main computational bottleneck for the original problem (3.16) comes from the number of decision variables. Without loss of generality, we consider an affine piece C_k. By inspecting the tessellation validation test (3.17), if a data point $(x_i, f(x_i))$ does not contatin any vertices of C_k in $\mathscr{B}(x_i, r_i)$ (i.e. $C_k \ni v_{j,k} \notin \mathscr{B}(x_i, r_i)$), then $\forall\ \tilde{g} \in \mathbb{R}^{n_x}$, following inequality holds

$$0 \leq \tilde{g}^\top f_i + M\|\tilde{g}\|\|v_{j,k} - x_i\|\,.$$

Hence, this data point x_i cannot help enforce the strict negative Lyapunov decreasing condition (3.16f), and can therefore be neglected in the constraints defined on affine piece C_k. Accordingly, the set of data points relevant to C_k are:

$$D_k := \big\{(x_i, f_i)\,\big|\, i \in I_k\big\} \quad \text{with} \quad I_k := \big\{i \in \mathbb{N}_{N_D}\,\big|\, \exists\ j \in \mathbb{N}_{|C_k|},\ v_{j,k} \in \mathscr{B}(x_i, r_i)\big\}$$

The condition defining this set essentially states the point should at least be possible to enforce strict negativity on constraint (3.16f) at one vertex. This technique can significantly reduce the computational cost. To see this, we consider a homogeneous tessellation within a unit hypercube centred at **0** within which data points scatter uniformly. We further assume that the affine pieces of the tessellation are hypercubes with edge width rel. Based on the Lipschitz condition (3.13), each data point will at most get involved in $O\big(\big(\frac{r_i}{rel}\big)^{n_x}\big) \ll N_C$. Notice that $r_i \leq .5$ (see Section 3.4.1), the number of decision variables are reduced to roughly $O\big(\frac{1}{2^{n_x}}\big)$ of the problem defined by the whole data set. In the numerical example we consider in Section 3.5, we observe on average an 81% reduction in the number of decision variables, which makes the problem tractable on a Laptop without memory overflow.

Explicit SOCP formualtions

In the numerical implementation, it is critical to convert the inequality constraint (3.16f) into a set of second-order cones, or Lorentz cone in particular [76]:

$$\begin{aligned} s_{j,k} &\geq \sum_{i \in I_k} \tilde{g}_{i,k}^\top f_i + t_{i,j,k} \\ t_{i,j,k} &\geq \|\tilde{g}_{i,k}\| M \|v_{j,k} - x_i\|\,, \end{aligned}$$

where $|I_k|$ auxiliary decision scalar variables $\{t_{i,j,k}\}$ are introduced per vertex $v_{j,k}$ in order to define the Lorentz cone. The resulting computational complexity per iteration in an interior point algorithm is $O(N_D N_C n_x^2)$ [213]. In comparison, without this reformulation, this inequality constraint will be directly handled by a block diagonal positive semi-definite matrix, whose computational complexity per iteration in an interior point algorithm is $O\big(|C_k|\,(N_D n_x)^3\big)$ [214, Chapter 1].

The Recast Problem

After introducing of the reformulation techniques, the learning problem we solved becomes:

$$
\begin{aligned}
s^{\star}_{\alpha,\epsilon} := \min_{\substack{\{g_k, b_k\}, \{\tilde{g}_{i,k}\} \\ \{s_{j,k}\}, \{t_{i,j,k}\}}} & \sum\nolimits_{k=1}^{N_C} \sum\nolimits_{j=1}^{|C_k|} s_{j,k} \\
\forall\, k, \ell \in \mathbb{N}_{N_C}, j \in \mathbb{N}_{|C_k|}, \quad & s_{j,k} \geq -\epsilon \\
v_{j,k} \in C_\ell \implies \quad & (g_k - g_\ell)^\top v_{j,k} = b_\ell - b_k \\
v_{j,k} \in \mathcal{X} \setminus \partial\mathcal{X} \implies \quad & g_k^\top v_{j,k} + b_k \leq \alpha - \epsilon \\
v_{j,k} \in \partial\mathcal{X} \implies \quad & g_k^\top v_{j,k} + b_k = \alpha \\
& \sum\nolimits_{i \in I_k|} \tilde{g}_{i,k} = g_k \\
& \sum\nolimits_{i \in I_k|} \tilde{g}_{i,k}^\top f_i + t_{i,j,k} \leq s_{j,k} \\
\forall\, i \in I_k, \quad & \|\tilde{g}_{i,k}\| M \| v_{j,k} - x_i \| \leq t_{i,j,k}
\end{aligned}
\tag{3.18}
$$

3.4.3 Algorithms

The final learning algorithm is summarized in Algorithm 3. Even though we use the standard tessellation algorithm in algorithm 3, generating a good tessellation is vital but non-trivial. Existing works mostly focus on the link between a convex liftable tessellation and the power diagram (see e.g. [215, 216]). However, as the Lyapunov function studied in this thesis is not necessarily convex, hence we leave the study of this topic in the future research. And we use the standard Delaunay triangularization in this part [217].

Algorithm 3

```
 Input: RoA prior $\mathcal{X}_s$, negativity tolerance $-\epsilon$, Lipschitz overestimate $M$, set level $\alpha$,
 Output: Lyapunov function $L(x)$
Refine a tessellation $\{C_k\}_{k=1}^{N_c}$ until it satisfies (3.17)
if tessellation is valid then
  Solve optimization (3.18)
  if Optimal value solution satisfies $s^{\star}_{\alpha,\epsilon} = -\epsilon \sum_{k=1}^{N_C} |C_k|$ then
    Return
  end if
else
  Return cannot learn $L(x)$.
end if
```

Sequential Space Partition

Based on the aforementioned strategies, the scalability of the learning problem (3.16) can still be improved by partitioning the region of interest $\mathcal{X}$ into a sequence of subset, such that

$$\mathcal{X}_s \subset \mathcal{X}_1 \subset \mathcal{X}_2 \cdots \subset \mathcal{X} .$$

Note that the logic behind the proposed algorithm is an augmentation of the a-priori knowledge in $\mathcal{X}_s$ to $\mathcal{X}$, this allows us to further improve the computational efficiency. The key idea is to gradually augment the volume of the RoA, and in each iteration, it is only necessary to learn the LF on the set $\mathscr{X}_i \setminus \mathscr{X}_{i-1}$. This sequential learning algorithm is summarized in Algorithm 4. It is noteworthy that, if one needs to recover the whole LF on $\mathscr{X} \setminus \mathscr{X}_s$, then it is necessary to impose the continuity condition on be boundary of $\partial\mathscr{X}_i$ between the i-th iteration and the $i+1$-th iteration. The corresponding algorithm is summarized in Algorithm 4.

Algorithm 4

```
  **Input**: Subset sequence $\{\mathscr{X}_i\}$ and the initial basin $\mathscr{X}_0 = \mathscr{X}_s$
  **Output**: Lyapunov function $L(x)$
$i = 1$
**while** $\{\mathscr{X}_s \neq \mathscr{X}\}$ **do**
  Run Algorithm. 3 with $\mathscr{X}_s$
  **if** Algorithm 3 failed **then**
    Return cannot learn $L(x)$.
  **end if**
  $i \leftarrow i + 1$
  $\mathscr{X}_s \leftarrow \mathscr{X}_i$
**end while**
```

3.4.4 Learning $\mathscr{X}$

Up to this point, we assume that a known $\mathscr{X}$ within which the stability analysis is conducted. However, such prior knowledge/assumption is not necessarily available. Instead, the users may only have access to a set of data and whole to find out a ROA based on this dataset. In this case, we will need to solve the Learning problem (3.16) without the boundary condition:

$$\begin{aligned} s_\epsilon^\star := \min_{\{g_k,b_k\},\{\tilde{g}_{i,k}\}\{s_{j,k}\}} & \textstyle\sum_{k=1}^{N_C}\sum_{j=1}^{|C_k|} s_{j,k} \\ \forall\, k,\ell \in \mathbb{N}_{N_C}, j \in \mathbb{N}_{|C_k|}, \quad & s_{j,k} \geq -\epsilon \\ v_{j,k} \in C_\ell \Longrightarrow \quad & (g_k - g_\ell)^\top v_{j,k} = b_\ell - b_k \\ v_{j,k} \in \mathscr{X} \setminus \partial\mathscr{X} \Longrightarrow \quad & g_k^\top v_{j,k} + b_k \leq \alpha - \epsilon \\ & \textstyle\sum_{i=1}^{N_D} \tilde{g}_{i,k} = g_k \\ & \textstyle\sum_{i=1}^{N_D} \tilde{g}_{i,k}^\top f_{i_m} + \|\tilde{g}_{i,k}\|M\| v_{j,k} - x_{i_m}\| \leq s_{j,k} \end{aligned} \tag{3.19}$$

Based on the solution to this problem, it is possible to determine a ROA by the following Corollary:

Corollary 7. *If the solution to problem* (3.19) *satisfies* $s_\epsilon^\star = -\epsilon\sum_{k=1}^{N_C}|C_k|$*, then any sublevel set* $L_{\leq\alpha} \subset \mathscr{X}$ *with* $\alpha \in \mathbb{R}$ *defines a ROA.*

Proof. If the solution to (3.19) satisfies $s_\epsilon^\star = -\epsilon\sum_{k=1}^{N_C}|C_k|$, by choosing $\alpha \in \mathbb{R}$ such that $L_{\leq\alpha} \subset \mathscr{X}$, then the optimal solution to (3.19) is also an optimal solution to:

$$\begin{aligned} & \min_{\{g_k, b_k\}, \{\tilde{g}_{i,k}\}\{s_{j,k}\}} \sum_{k=1}^{N_C} \sum_{j=1}^{|C_k|} s_{j,k} \\ \forall\, k, \ell \in \mathbb{N}_{N_C}, j \in \mathbb{N}_{|C_k|}, \quad & s_{j,k} \geq -\epsilon \\ v_{j,k} \in C_\ell \cap L_{\leq\alpha} \Longrightarrow \quad & (g_k - g_\ell)^\top v_{j,k} = b_\ell - b_k \\ v_{j,k} \in L_{\leq\alpha} \setminus \partial L_{\leq\alpha} \Longrightarrow \quad & g_k^\top v_{j,k} + b_k \leq \alpha - \epsilon \\ v_{j,k} \in \partial L_{\leq\alpha} \Longrightarrow \quad & g_k^\top v_{j,k} + b_k = \alpha \\ & \sum_{i=1}^{N_D} \tilde{g}_{i,k} = g_k \\ & \sum_{i=1}^{N_D} \tilde{g}_{i,k}^\top f_{i_m} + \|\tilde{g}_{i,k}\| M \| v_{j,k} - x_{i_m} \| \leq s_{j,k}\,, \end{aligned}$$

whose tessellation is given by $\{C_k \cap L_{\leq\alpha}\}_k^{N_C}$. Hence, Theorem 8 holds in $L_{\leq\alpha}$, which summarizes the proof.

■

Based on this corollary, we are able to learn the ROA within $\mathscr{X}$ by the following algorithm.

Algorithm 5

```
  Input: RoA prior $\mathscr{X}_s$, negativity tolerance $-\epsilon$, Lipschitz overestimate $M$,
  Output: Lyapunov function $L(x)$, ROA $L_{\leq\alpha}$
Refine a tessellation $\{C_k\}_{k=1}^{N_c}$ until it satisfies (3.17)
if tessellation is valid then
    Solve optimization (3.19)
    if Optimal value solution satisfies $s_\epsilon^\star = -\epsilon \sum_{k=1}^{N_C} |C_k|$ then
        find $\alpha \in$ such that $L_{\leq\alpha} \subset \mathscr{X}$
        Return
    end if
else
    Return cannot learn $L(x)$.
end if
```

Remark 26. *Besides the numerical improvements discussed in Section 3.4.2, it is also possible to parallelize the solution of problem* (3.16) *by distributed convex optimization. In particular, each affine piece* $\{C_k\}$ *has its local constraints* (3.16a), (3.16d), (3.16c), (3.16e) *and* (3.16f). *It is coupled with its adjacent pieces by the linear equality constraints* (3.16b). *This structure fits into the standard structure in distributed convex optimization and can effectively enable the solution of the large problem* (3.16) *into iterative solution of small scale SOCP defined only on each affine pieces* C_k.

3.5 Numerical Results

In this part, we are going to evaluate the proposed learning schemes in two different examples. In particular, we will make use of Algorithm 3, 4 and 5 in the first example and use Algorithm 5 in the second one. All the following results are implemented on a laptop with Intel i7-11800H and 32G memory, and the `Mosek` is used for solving the SOCP problem.

3.5.1 Non Polynomial Dynamics

We consider a two-dimensional nonlinear dynamic system:

$$\begin{aligned}\dot{x}_1(t) &= -0.9\sin(x_1(t))\cos(x_2(t)) + 0.2x_1(t)x_2(t) + 0.25x_2(t)^2 \\ \dot{x}_2(t) &= -1\sin(x_2(t))(|x_1(t)+0.2|) + 0.5\frac{x_1(t)x_2(t)}{\cos(x_2(t)) - 0.3x_1(t)}\end{aligned}$$

We assume that we know a ROA $\mathscr{X}_s = [-0.1, 0.1] \times [-0.1, 0.1]$. A dataset with only 200 samples within $[-1,1] \times [-1,1] \subset \mathbb{R}^2$ is used to learn the underlying Lyapunov function: the positions $\{x_i\}$ and speeds $\{f(x_i)\}$ of these samples are plotted in Figure 3.3, from which we can observe that this dataset is relatively sparse in $[-1,1] \times [-1,1]$. Judging by the speed sample, the dynamic system seems stable within the box of $[-0.4, 0.4] \times [-0.4, 0.4]$, while stability within the region $[-1,1] \times [-1,1] \setminus [-0.4, 0.4] \times [-0.4, 0.4]$ is unclear because of the speed samples in the lower right corner in Figure 3.3. Hence, we ran sequential space partition scheme (Algorithm 4). In particular, we first use Algorithm 3 in the region $[-0.4, 0.4] \times [-0.4, 0.4]$ with $\mathscr{X}_s = [-0.1, 0.1] \times [-0.1, 0.1]$. After we justify that $[-0.4, 0.4] \times [-0.4, 0.4]$ is a ROA, then we further apply Algorithm 5 to $[-1,1] \times [-1,1]$ with $\mathscr{X}_s = [-0.4, 0.4] \times [-0.4, 0.4]$. In both sub-problems, the negativity tolerances ϵ are set to 10^{-3} and the tessellation are both randomly generated by Delaunay triangularization [217].

The learnt Lyapunov function in $[-0.4, 0.4] \times [-0.4, 0.4]$ is shown in Figure 3.4, while the ROA we finally end up is shown in Figure 3.6. Moreover, the Lyapunov function learnt from Algorithm 5 in $\mathscr{X}_s = [-0.1, 0.1] \times [-0.1, 0.1] \setminus [-0.4, 0.4] \times [-0.4, 0.4]$ is shown in Figure. Figure 3.6 also shows the evaluation of $\frac{dL}{dx} f(x)$ with respect to the underlying dynamic system, whose maximal evaluation is -1.525×10^{-2}. In accordance with our guess, the learnt ROA in Figure 3.6 cuts off the lower right corner, because this region does not seem to be stable. To see that, we simulate the underlying dynamic system by setting the initial states to points in our dataset. The simulated trajectories are plotted in Figure 3.7, and please note that these trajectories are not used in the learning scheme at all.

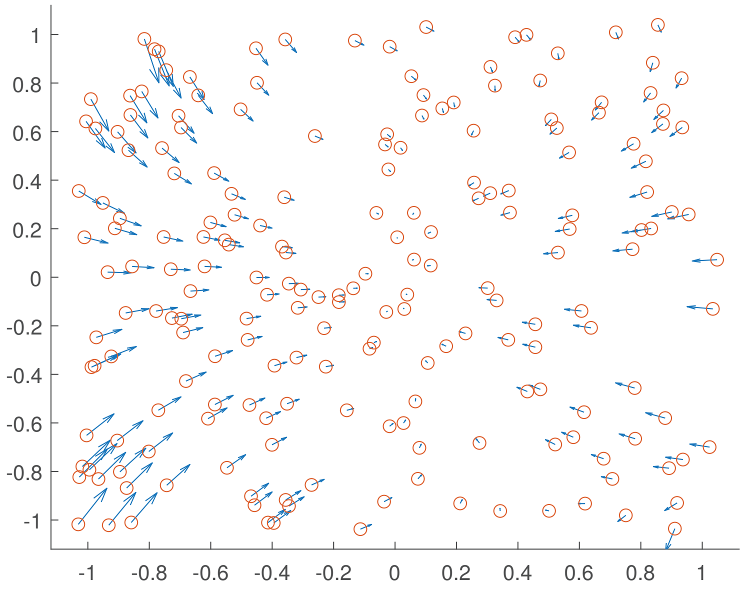


Figure 3.3: Visualization of the data used for the learning scheme

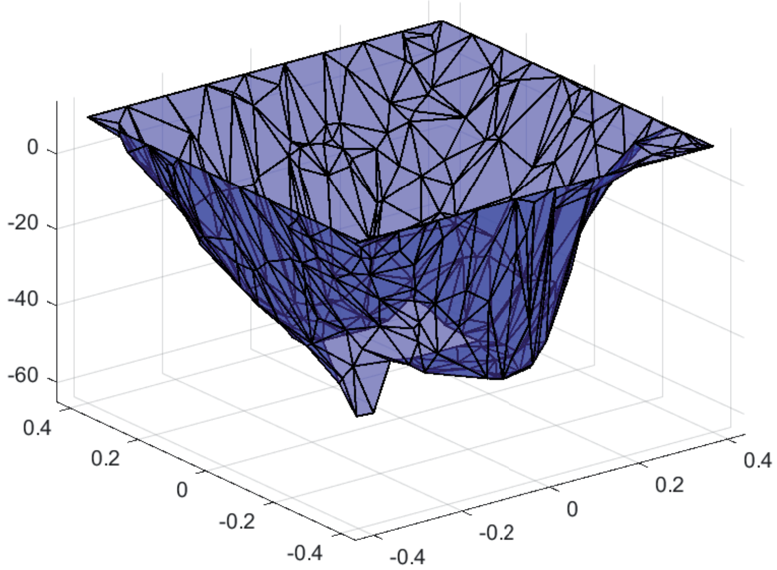


Figure 3.4: Visualization of the Lyapunov function on $[-0.4, 0.4] \times [-0.4, 0.4]$

3.5.2 Reverse Time Van Del Pol Oscillator

In this part, we consider the reverse time Van Del Pol oscillator:

$$\dot{x}_1(t) = -2x_2(t)$$
$$\dot{x}_2(t) = -0.8 * x_1(t) - 10(x_1(t)^2 - 0.21)x_2(t)\,.$$

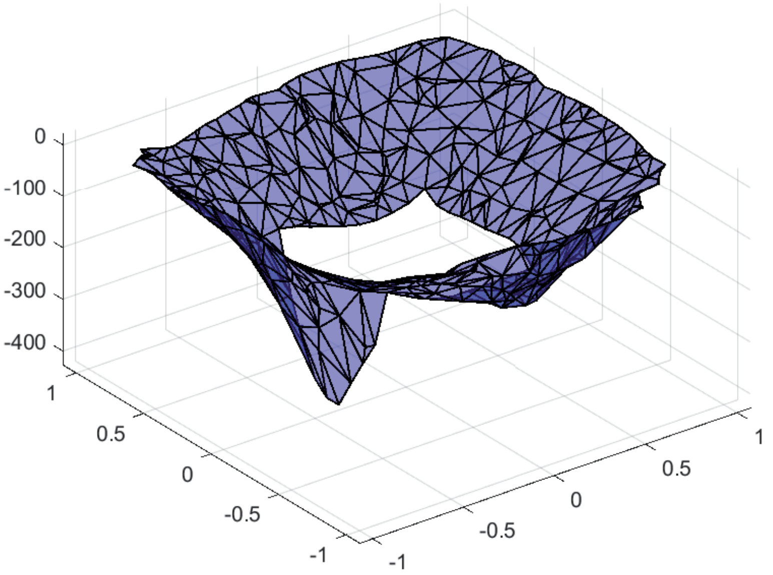


Figure 3.5: Visualization of the Lyapunov function on the learnt ROA within $[-1,1] \times [-1,1]$

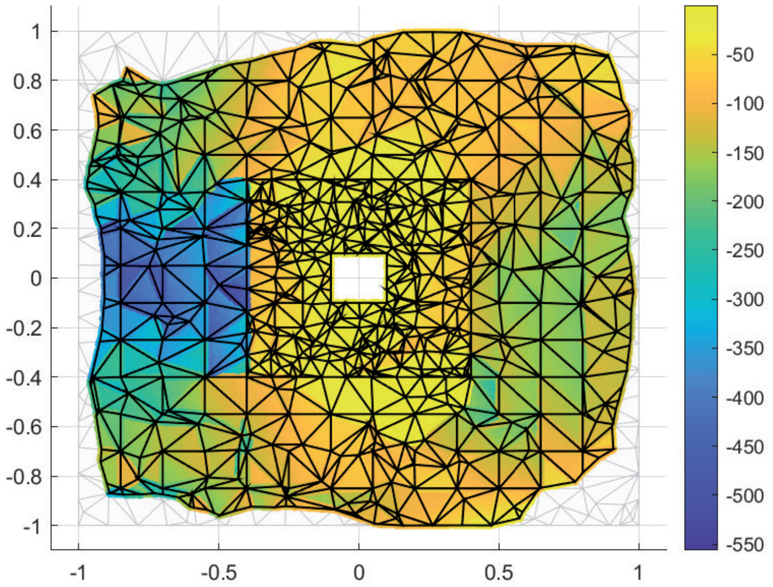


Figure 3.6: Evaluation of $\frac{dL}{dx} f(x)$ on the learnt ROA, the gray triangularization in the background is the tessellation used to solve Problem (3.19), while the coloured region in the front is ROA.

We know an a-priori polytopic ROA $\mathscr{X}_s$, which is plotted in the center of Figure 3.11. A dataset with only 400 samples within $[-0.5,0.5] \times [-0.5,0.5] \subset \mathbb{R}^2$ is used to learn the underlying

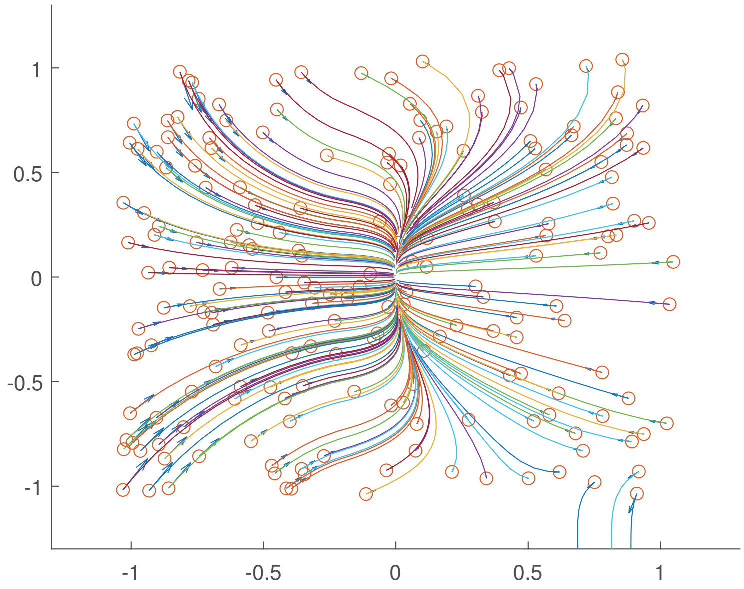


Figure 3.7: Simulation of the underlying dynamic system

Lyapunov function: the positions $\{x_i\}$ and speeds $\{f(x_i)\}$ of these samples are plotted in Figure 3.8. Similar to what we did in the last example, we simulate these data forward in Figure 3.9, while these trajectories not used in the learning scheme. We can observe that both the lower right corner and the upper left corner in Figure 3.9 correspond to regions of unstable states. Even with only the access to the data in Figure 3.8, we can not give a clear idea about which region is safe, hence we apply Algorithm 5 to $[-0.5, 0.5] \times [-0.5, 0.5]$. In particular, the negativity tolerance ϵ is set to 10^{-3} and the tessellation is randomly generated by Delaunay triangularization [217]. The learnt Lyapunov function and its evaluation of $\frac{dL}{dx} f(x)$ on the learnt ROA are respectively plotted in Figure 3.10 and Figure 3.11. In particular, the maximal evaluation of the $\frac{dL}{dx} f(x)$ on the learnt ROA is -1.947×10^{-2}.

3.6 Conclusion

Our results

In this work, we went all the way from proving a variant of stability theorem with non-smooth Lyapunov functions (LF), to actually implementing an algorithm for data-based region of attraction (RoA) estimation with unknown dynamic system. In the process, we went through proving a theorem for piecewise affine (PWA) LF computation and deriving a convex optimization program for computing such LF.

The originality of the method we propose is that it only requires a fixed dataset to compute an

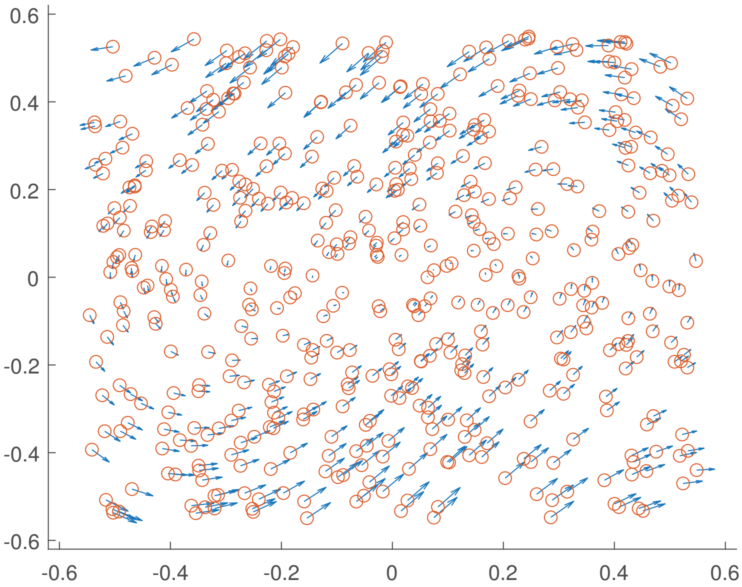


Figure 3.8: Visualization of the data used for the learning scheme

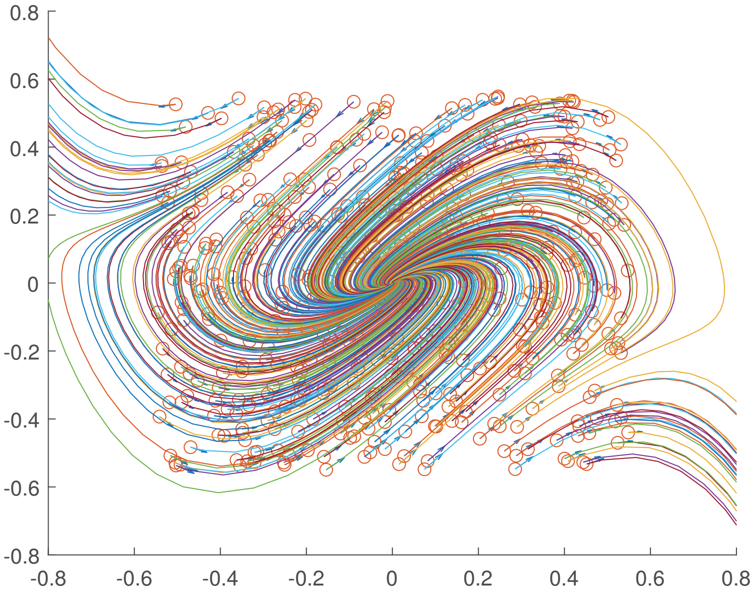


Figure 3.9: Simulation of the reverse time Van Del Pol oscillator

estimate of the RoA, from which it allows the user to deduce global information from local data and knowledge on the Lipschitz constant of the dynamic system. Hence, it can be used to study systems whose dynamics cannot be easily sampled at will, through a relatively simple

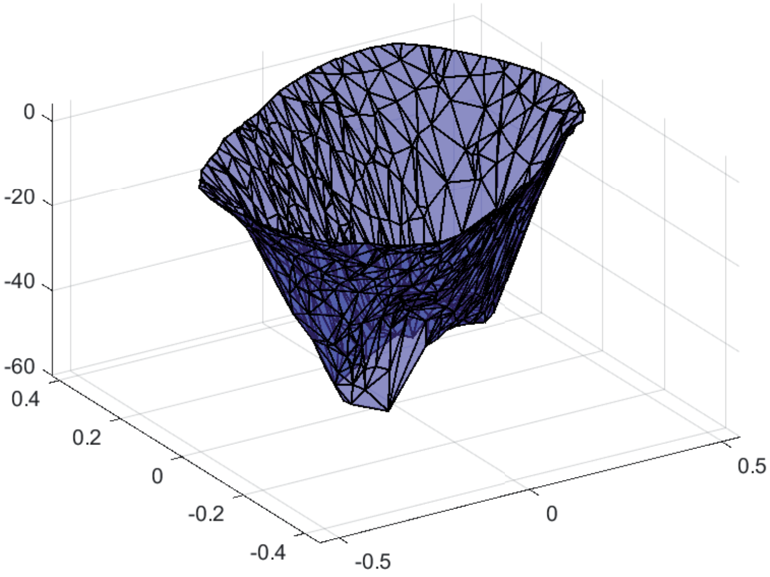


Figure 3.10: Visualization of the Lyapunov function on the learnt ROA

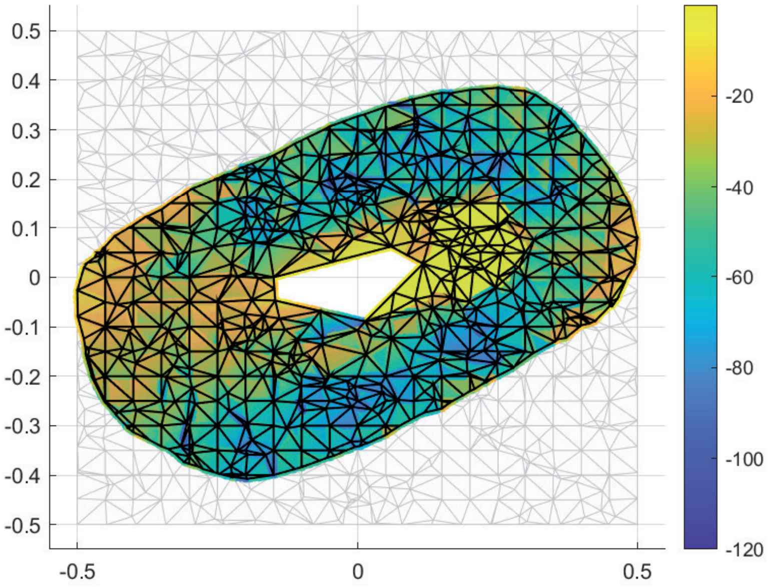


Figure 3.11: Evaluation of $\frac{dL}{dx} f(x)$ on the learnt ROA, the gray triangularization in the background is the tessellation used to solve Problem (3.19), while the coloured region in the front is ROA. The polytopic hole in the middle is $\mathscr{X}_s$.

optimization problem that can be handled with interior point methods.

Future works

In this work, we propose a learning scheme that can learn a robust Lyapunov function with a fixed dataset. However, with such minimal knowledge of the underlying dynamic system results a big convex optimization even for a low-dimensional dynamic system. In the future works, we should investigate how the side information or other a-priori knowledge of the underlying dynamic system can be incorporated into the learning scheme so that the proposed learning scheme can handle a higher-dimensional dynamic system.

Other Contributions Part III

In this part, we will take a detour to another two major topics that I have been explored during my Ph.D, which includes self-triggered control and real-time nonlinear MPC solver design. Each topic is assigned a specific chapter, and one selected result is presented in the corresponding chapter. Each chapter is written to be self-contained such that they do not affect the integrity of the first two parts. For the readers that mainly hope to have a rough idea of the result presented in these two chapters before reading them into details, I briefly summarize them as follows:

Lian, Y., Jiang, Y., Stricker, N., Thiele, L. and Jones, C.N., 2021. *Resource-aware stochastic self-triggered model predictive control. IEEE Control Systems Letters, 6, pp.1262-1267.*
In this section, we investigate a continuous-time stochastic linear system where an aperiodic zero-order-hold predictive controller is applied. The main challenge addressed is the predictive uncertainty propagation of this aperiodic closed-loop control law. In particular, we consider two aspects. Firstly, in self-triggered control, the triggering time instance of an actuator is a control input and therefore a decision variable in the MPC problem. Another problem that needs to be resolved in the MPC scheme is that the control input is constant between two adjacent triggers. When a closed-loop prediction scheme is used, the predictive controller needs to react to uncertainty by some parametric feedback control law. However, such reaction only occurs at the triggered time instances. Therefore, a new characterization of uncertainty propagation needs to be considered. To provide such a characterization, this paper offers two fascinating results. Firstly, we provide the first result, to the best of our knowledge, that decomposes information by product sigma field. Without resorting to the inner product space structure, such decomposition is achieved solely based on the knowledge of the topology of the stochastic process. Additionally, this decomposition enables us to quantify which part of the information is available to update the feedback control law, which leads to the second result in this paper. We characterize the uncertainty propagation by two ordinary differential equations (ODEs), while the considered dynamics are governed by stochastic differential equations. In these ODEs, we observe an interesting correlation in the direction of time.

Lian, Y., Jiang, Y., Opila, D.F. and Jones, C.N., 2023.A Proximal-Point Lagrangian Based Parallelizable Nonconvex Solver for Bilinear Model Predictive Control. *Conditionally accepted by Transactions on Control Systems Technology*
This part presents a real-time solver for the bilinear model predictive control (MPC) problem, in which we leverage parametric explicit quadratic programming to improve the performance of the sequential quadratic programming (SQP) algorithm in detecting the active set. The key contributions of this work lie in two aspects. Firstly, we bridge the gap between explicit MPC and nonlinear MPC by incorporating a parametric QP step in the SQP algorithm. Secondly, we recover a proximal alternating linearized method [218] in the dual space, which suggests potential applications in broader nonlinear MPC problems. The proposed algorithm not only offers interesting theoretical insights, but also achieves comparable performance to the fine-

tuned SQP solver, acados [219]. In addition, we deploy the proposed solver on a real-world bilinear motor with a low-cost embedded system and successfully achieve a reference tracking task at a frequency of 500 Hz. Overall, our work demonstrates the practicality and effectiveness of the proposed solver in real-time control applications.

4 Self-triggered MPC

In this chapter, we will briefly review the work that I have done in the direction of self-triggered control. Particularly, we will mainly elaborate on the result of stochastic self-triggered MPC and then summarize the contributions in other papers at the end.

4.1 Introduction

Most devices in Internet of Things (IoT) networks and wireless sensing systems are operated with some limited resource factors, such as battery life or hardware longevity. In order to maintain desirable performance, a minimal number of triggers are required to best exploit the limited resource. Event-triggered control and self-triggered control are two main control schemes [220] accommodating this issue. In particular, control under an event-triggered scheme is updated *reactively* by determining a trigger condition, for which a sensor has to continuously monitor the trigger condition. Contrarily, a self-triggered scheme updates *proactively* by planning the next trigger in advance, leaving the sensor and controller in idle mode. Due to the limitation of the resource factors, especially battery life, a self-triggered scheme is often preferable and is, therefore, the research object of this work.

The key ingredient of a self-triggered controller is the decision of the triggering time sequence. The triggering time can be chosen as long as possible to minimize resource consumption as in [221, 222]. However, to balance performance and resource consumption more effectively, the response of the resource is explicitly considered in the model predictive control (MPC) problem in [223, 224]. The former work solves a mixed-integer problem and is designed for discrete-time systems, while the latter work solves a non-convex continuous-time optimal control problem and has been later generalized to a distributed control scheme in one of our previous result [225].

Running a triggered system within an uncertain environment while maintaining system performance is challenging. Especially for the self-triggered controllers, the lack of sensor measurement between consecutive triggers requires extra consideration of the uncertainty propagation. In [226], a nominal control law is determined based on a nominal system, while the discrepancy between the nominal and measured trajectories serves as the triggering condition. In [227–229], the idea of tube-MPC enables the design of robust self-triggered controllers for both discrete-time and continuous-time linear systems. [230] used a min-max optimization to optimize the worst-case performance. Although it is capable of handling general uncertainties in nonlinear systems, the resulting non-convex robust optimization problem is NP-hard [68]. Except for [230], other previous works mainly decouple the effects of uncertainty from the nominal system, and the feedback control laws are updated with a fixed frequency.

In this part, a resource-aware stochastic predictive control scheme is designed for a stochastic linear system where the process noise is explicitly considered in the predictive control problem. In particular, a discrete-time zero-order-hold linear feedback control law is integrated into the closed-loop predictive control problem. The update time instances of this feedback control law distribute non-uniformly on the time axis, which we term *time-inhomogeneous*, and are optimized within the predictive control problem. The contributions of this work are summarized into two aspects:

- A sigma field decomposition strategy is proposed to enable the analysis of a time-inhomogeneous control.

- A discrete-time closed-loop feedback control law for stochastic self-triggered MPC is proposed.

4.2 Deterministic Self-Triggered MPC

This section recaps the main idea of deterministic resource-aware self-triggered control. We consider a linear time invariant (LTI) system in continuous time:

$$\frac{dx(t)}{dt} = Ax(t) + Bu(t) \tag{4.1}$$

with state $x(\cdot) : [0,\infty) \to \mathbb{R}^{n_x}$ and control input $u(\cdot) : [0,\infty) \in \mathbb{R}^{n_u}$. A self-triggered controller determines both the value of the control inputs and the time instances at which the control input is changed. In the framework of direct optimal control [231], a self-triggered controller parameterizes its control inputs over the time horizon $[0, t_N]$ by

$$u(t) = \sum_{k=0}^{N-1} v_k \cdot \zeta_k(t, t_k, t_{k+1}), \tag{4.2}$$

where the orthogonal functions $\zeta_k \in \mathscr{L}^2[t_0, t_N],\ k \in \mathbb{Z}_0^{N-1}$ model the triggering property with a piece-wise constant function

$$\zeta_k(t, t_k, t_{k+1}) = \begin{cases} 1 & t \in (t_k, t_{k+1}] \\ 0 & \text{otherwise.} \end{cases} \tag{4.3}$$

For the sake of compactness, we use notation $v \in \mathbb{R}^{Nn_u} := [v_0^\top, v_1^\top \dots, v_{N-1}^\top]^\top$ to stack the control coefficients, and define the triggering time interval $\Delta_k := t_{k+1} - t_k$ and use the notation $\Delta = [\Delta_0, ..., \Delta_{N-1}]^\top$.

A self-triggered agent updates its control inputs at triggering time instances $\{t_k\}_{k=0}^{N-1}$. When the control law is fixed within $(t_k, t_{k+1}]$, the resource r is recharged at a constant rate ρ until saturation. More specifically, $\dot{r}(t) = h(\overline{r} - r(t))\rho$ for all $t \in [t_k, t_{k+1})$, where $\overline{r}$ is the saturated value and $h(\cdot)$ is a heaviside function with $h(s) = 1$ if $s > 0$ and 0 elsewhere. When the agent is triggered to update the control input, the resource is discharged by an amount $\eta(\Delta_k)$ to pay the update cost. Hence, the resource at triggering time instants $\{t_k\}_{k=0}^{N-1}$ is

$$r(t) = \begin{cases} r_0 & t = t_0 \\ \lim\limits_{t \to t_k^-} r(t) - \eta(\Delta_k) & t \in \{t_k\}_{k=1}^{N-1} \end{cases} \tag{4.4}$$

with an initially available resource r_0 at t_0. Here, $t \to t_k^-$ represents the left limits, i.e., $t \to t_k$ and $t < t_k$. Moreover, the resource r is required to be lower bounded by $\underline{r}$. In conclusion, a resource-aware self-triggered agent can update its control input when its resource is sufficiently high to stay above the lower bound $\underline{r}$. Otherwise, it must wait until enough resource is available. Once the controller is triggered at time t_0, the resource-aware self-triggered control solves the following optimization problem to plan the next trigger time t_1 and the control input within $[t_0, t_1]$,

$$\begin{aligned} \min_{x(\cdot), v, \Delta} \quad & \sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} l(x(\tau), v_k) d\tau + M(x(t_N)) && \text{(4.5a)} \\ \text{s.t.} \quad & x(t_0) = x_0\ ,\ r(t_0) = r_0 && \text{(4.5b)} \\ & \forall\, t \in [t_0, t_N],\ \frac{dx(t)}{dt} = Ax(t) + Bu(t), && \text{(4.5c)} \\ & \forall\, t \in [t_0, t_N],\ x(t) \in \mathscr{X}\ ,\ u(t) \in \mathscr{U}\ , && \text{(4.5d)} \\ & \forall\, k \in \{0, 1, ..., N-1\} && \\ & r(t_{k+1}) = \min\{\rho\Delta_k + r(t_k) - \eta(\Delta_k), \overline{r}\} && \text{(4.5e)} \\ & r(t_{k+1}) \in [\underline{r}, \overline{r}]\ , && \text{(4.5f)} \\ & \Delta_k \in [\underline{\Delta}, \overline{\Delta}] && \text{(4.5g)} \end{aligned}$$

where $l(\cdot,\cdot)$ and $M(\cdot)$ in (4.5a) are stage cost and terminal cost, respectively. (4.5e) is a simplified yet equivalent formulation of the resource dynamics (4.4) [224] and the resource is bounded by (4.5f). The constraints of the triggering time interval in (4.5g) protects the system from being Zeno[1] or frozen. $\mathscr{X} \subseteq \mathbb{R}^{n_x}$ and $\mathscr{U} \subseteq \mathbb{R}^{n_u}$ in (4.5d) model the state and input constraints. The initial state and resource are given by (4.5b).

4.3 Stochastic Self-Triggered MPC

In this section, we consider the LTI system (4.1) contaminated by a Wiener process noise. This is described by the stochastic differential equation (SDE)

$$dx(t) = (Ax(t) + Bu(t))dt + dW \,, \tag{4.6}$$

where W denotes a multi-dimensional Wiener process with statistics $\mathbb{E}\{W(t)W(s)\} = Q\min(s,t)$ and $\mathbb{E}\{W(t)\} = 0$.

The open-loop evolution of the system's state distribution (4.6) is widely studied in filter theory [232] and the state evolution remains Gaussian $\mathscr{N}(\mu(t),\, P(t))$, where

$$\frac{d\mu(t)}{dt} = A\mu(t) + Bu(t)\,, \tag{4.7a}$$

$$\frac{dP(t)}{dt} = AP(t) + P(t)A^\top + Q\,. \tag{4.7b}$$

Above all, given the dynamics in (4.7), it is straightforward to adapt the deterministic formulation in (4.5) to generate an open-loop resource-aware stochastic MPC. The main focus and contribution of this work are to develop a **closed-loop** scheme with respect to the dynamics (4.7). In particular, a feedback control law is explicitly considered in the predictive control problem and this feedback control law should satisfy the following requirements:

1. The feedback control law can only change its value when the controller is triggered, otherwise, the control inputs remain constant.

2. The feedback control law is not updated at a fixed frequency, and its update time instances are decision variables of the self-triggered problem.

In the following, the dynamics of the state distribution driven by a discrete time feedback control law are developed by using the technique of sigma field decomposition. This results in a resource-aware stochastic self-triggered MPC and its numerical implementation is discussed at the end of this section. In order to convey the main idea of the proposed scheme, we state the main results intuitively and provide the mathematical details in the Appendix.

[1] Zeno means that the triggering time Δ can be zero.

4.3.1 Stochastic Process Decomposition

Considering an ordered triggering time sequence $\{t_k\}_{k=0}^{N}$, a sigma field $\mathscr{F}_k$ collects all the stochastic events occurring between $[t_0, t_k]$, whereas $\mathscr{F}_0$ includes all the deterministic events. Because the controller can update only when it is triggered, we propose to partition the stochastic events by time intervals. In particular, the collection of stochastic events between two consecutive triggers is defined by $\mathscr{F}_{k,k+1} := \sigma(\mathscr{F}_{k+1} \backslash \mathscr{F}_k)$, where $\sigma(\cdot)$ denotes the minimal sigma field. The following lemma indicates that there is no information loss with the partitioning $\{\mathscr{F}_{k,k+1}\}_{k=0}^{N-1}$.

Lemma 8. *For a given Wiener process W with a stopping time sequence $\{t_k\}_{k=0}^{N-1}$, if $t_j > t_i$ holds almost surely for all $j > i$, the sigma field at time t_N can be decomposed as $\mathscr{F}_N = \sigma(\cup_{i=0}^{N-1} \mathscr{F}_{i,i+1})$, where $\mathscr{F}_{i,i+1}$ is independent from $\mathscr{F}_{j,j+1}$ for all $i \neq j$.*

Proof. We first recall the definition $\mathscr{F}_{k,k+1} := \sigma(\mathscr{F}_{k+1} \backslash \mathscr{F}_k)$. According to the independence property of a Wiener process, $\mathscr{F}_{i,i+1} \perp \mathscr{F}_{j,j+1}$ holds for all $i \neq j$, therefore we have $\sigma(\cup_{i=0}^{N-1} \mathscr{F}_{i,i+1}) \subset \mathscr{F}_N$. Then, we show the equality holds by contradiction. If $\mathscr{F}_N \neq \sigma(\cup_{i=0}^{N-1} \mathscr{F}_{i,i+1})$, by definition of $\mathscr{F}_{i,i+1}$, there exists $i \in \mathbb{Z}_0^{N-1}$ such that $\mathscr{F}_{t_i} \neq \mathscr{F}_{t_i^+} := \sigma(\cap_{t > t_i} \mathscr{F}_t)$, which violates the continuity of a Wiener process [233]. Hence, the proof concludes ■

Remark 27. *This lemma holds for any càdlàg Lévy process [232], which is practical for real-world applications as all the analysis is established from the current time step or, in particular, the sigma fields accumulated up to the current time instant.*

Remark 28. *Equation* (4.8) *holds due to product topology given by the Lemma 8, which reflects the fact that the conditional covariance matrix $P_k(t) := \mathbb{E}(P(t)|\mathscr{F}_{k,k+1})$ is a projection onto the $\mathscr{L}^2$ space of the progressively measurable process on $\mathscr{F}_{k,k+1}$.*

Lemma 8 is the key component of the feedback control law analysis, which enables us to decompose the statistics of the state evolution into non-overlapping time intervals. Here, we focus on the decomposition of the covariance matrix $P(t)$ because of its close link with the feedback control law. The projection[2] of the covariance matrix $P(t)$ onto the stochastic events within $(t_k, t_{k+1}]$ is defined by $P_k(t) := \mathbb{E}(P(t)|\mathscr{F}_{k,k+1})$, and Lemma 8 implies that

$$\forall\, t \in [t_0, t_N]\,,\ P(t) = \sum_{i=0}^{N-1} P_i(t)\,. \tag{4.8}$$

Based on this decomposition, the open-loop evolution of the conditional dynamics of $P_k(t)$

[2] This is a geometric interpretation of conditional expectation [142]. Given a squared-integrable random variable X in Sigma field $\mathscr{F}$ and a sub-Sigma field $\mathscr{G} \subset \mathscr{F}$, then $\mathbb{E}(X|\mathscr{G}) = \arg\min_{Y \in \mathscr{G}} \mathbb{E}((X-Y)^2)$.

are given by

$$\frac{dP_k(t)}{dt} = \begin{cases} 0 & t \in [t_0, t_k] \\ AP_k(t) + P_k(t)A^\top + Q & t \in (t_k, t_{k+1}] \\ AP_k(t) + P_k(t)A^\top & t > t_k\,. \end{cases} \tag{4.9}$$

Note that substituting (4.9) into (4.8) yields the dynamics in (4.7b).

4.3.2 Discrete-Time Feedback Covariance Dynamics

To alleviate the perturbation caused by the process noise in (4.6), a feedback control law is introduced to regulate the state deviation around the expected trajectory $\mu(t) := \mathbb{E}\{x(t)\}$. Based on the standard self-triggered scheme in (4.2) and (4.3), the feedback control law is defined by

$$u(t) = \sum_{k=0}^{N-1} (v_k + K(x(t_k) - \mu(t_k)))\zeta(t, t_k, t_{k+1})\,, \tag{4.10}$$

where v_k is the nominal control input determined by the expected dynamics of $\mu(t)$. Note that this is a **discrete-time** linear control law written in continuous time, and it respects the self-triggered control scheme such that the control input remains constant within time interval $(t_k, t_{k+1}]$ given by

$$u(t) = v_k + K(x(t_k) - \mu(t_k))\,,\ t \in (t_k, t_{k+1}]\,.$$

Meanwhile, due to the fact that the state is accurately measured at time instance t_0, there is no feedback at t_0. Recall (4.7), the evolution of the state distribution under the control law (4.10) is characterized by its mean and covariance, where the nominal input v_k governs the mean dynamics by

$$\frac{d\mu(t)}{dt} = A\mu(t) + Bv_k,\ \ t \in (t_k, t_{k+1}],\ \ k \in \mathbb{Z}_0^{N-1}. \tag{4.11}$$

As the feedback part in (4.10) reacts to the deviation from the nominal dynamics of $\mu(t)$, the covariance dynamics is therefore governed by the feedback control gain $K \in \mathbb{R}^{n_u \times n_x}$. The following theorem gives the covariance dynamics.

Theorem 9. *Let the feedback control law be defined by* (4.10). *The dynamics of the covariance is given by*

$$\frac{dP(t)}{dt} = AP(t) + P(t)A^\top \tag{4.12a}$$

$$+ BKP_{k,t}(t) + P_{t,k}(t)(BK)^\top + Q\,,$$

$$\frac{dP_{t,k}(t)}{dt} = AP_{t,k}(t) + BKP(t_k) \tag{4.12b}$$

with correlation matrix $\lim_{t\to t_k, t>t_k} P_{t,k}(t) = P(t_k)$ *and*

$$P_{t,k}(t) := \mathbb{E}\{\mathbb{E}\{(x(t)-\mathbb{E}\{x(t)\})(x(t_k)-\mathbb{E}\{x(t_k)\})^\top|\mathscr{F}_k\}\} \tag{4.13}$$

for all $t \in (t_k, t_{k+1}]$ *and* $k \in \mathbb{Z}_0^{N-1}$.

To prove this theorem, we first state the Itô lemma [233]:

Lemma 9 (Itô's Lemma)**.** *For a given drift-diffusion process* $dx = a\,dt + b\,dW$*, if function* $f(\cdot)$ *is twice-differentiable, Iťo's formula holds as*

$$df = \left(\frac{\partial f}{\partial t} + a\frac{\partial f}{\partial x} + \frac{b^2}{2}\frac{\partial^2 f}{\partial x^2}\right)dt + b\frac{\partial f}{\partial x}dW\,.$$

Proof. This proof consists of two steps:

1. Derive the decomposed covariance dynamics $\{P_k(\cdot)\}_{k=0}^{N-1}$.
2. Reconstruct the ensemble covariance dynamics $P(t)$.

Decomposed covariance dynamics
Conditioning on the sigma field $\mathscr{F}_{k,k+1}$, the control law is

$$\begin{aligned}&\mathbb{E}(u(t)\,|\,\mathscr{F}_{k,k+1}) = \\ &\begin{cases} v_i & t\in(t_i,t_{i+1}],\ i\in\mathbb{Z}_0^k \\ v_i + K\mathbb{E}\{x(t)-\mu(t)|\mathscr{F}_{k,k+1}\} & t\in(t_i,t_{i+1}],\ i\in\mathbb{Z}_{k+1}^{N-1}. \end{cases}\end{aligned} \tag{4.14}$$

Notice that under a predictive control scheme, $\{v_i\}_{i=0}^{N-1}$ are determined at t_0, hence $\{v_i\}_{i=0}^{N-1}$ are $\mathscr{F}_0$ measurable and furthermore $\mathscr{F}_{k,k+1}$ is measurable. Before t_{k+1}, none of the triggers can generate feedback with respect to the events in $\mathscr{F}_{k,k+1}$ because $\mathscr{F}_{k,k+1}$ happens later than t_k. These facts conclude the conditional control law in (**??**). Based on the system dynamics (4.6), the mean dynamics (4.11) and the conditional control inputs (**??**), the SDE of the conditional deviation dynamics of $x(t)-\mu(t)$ is

$$\begin{aligned}&\mathbb{E}\{d(x(t)-\mu(t))|\mathscr{F}_{k,k+1}\} = \\ &\begin{cases} 0 & t\in[t_0,t_k] \\ A(\mathbb{E}\{x(t)-\mu(t)|\mathscr{F}_{k,k+1}\})dt + dW & t\in(t_k,t_{k+1}] \\ \big[A(\mathbb{E}\{x(t)-\mu(t)|\mathscr{F}_{k,k+1}\}) + B & t\in(t_i,t_{i+1}] \\ \quad\cdot K(\mathbb{E}\{d(x(t_i)-\mu(t_i))|\mathscr{F}_{k,k+1}\})\big]\,dt & i\in\mathbb{Z}_{k+1}^{N-1}. \end{cases}\end{aligned} \tag{4.15}$$

More specifically, this dynamic means that when $t \le t_k$, there is no uncertainty generated by $\mathscr{F}_{k,k+1}$, and $\mathbb{E}(x(t)-\mu(t)|\mathscr{F}_{k,k+1}) = 0$. After that, the stochastic events within interval $(t_k, t_{k+1}]$

do not generate any feedback before t_{k+1} and the deviation evolves in an open-loop form. After t_{k+1}, no new $\mathscr{F}_{k,k+1}$-measurable events can happen anymore and the feedback control law comes into effect.

As $P_k(t) = \mathbb{E}\{\mathbb{E}\{(x(t)-\mu(t))(x(t)-\mu(t))^\top | \mathscr{F}_{k,k+1}\}\}$, we can apply Itô's Lemma (Lemma 9) to the deviation dynamics. As a result, we have $\frac{dP_k(t)}{dt} = 0$ for all $t \in [t_0, t_k]$. And we have

$$\begin{aligned} dP_k(t) &= (AP_k(t) + P_k(t)A^\top + Q)dt \\ &\quad + \underbrace{\mathbb{E}\{\mathbb{E}\{x(t) - \mu(t)|\mathscr{F}_{k,k+1}\}dW^\top\}}_{(a)} \\ &\quad + \underbrace{\mathbb{E}[dW\,\mathbb{E}(x(t) - \mu(t)|\mathscr{F}_{k,k+1})^\top]}_{(b)} , \end{aligned}$$

holds for all $t \in (t_k, t_{k+1}]$, where $(a) = 0$ and $(b) = 0$ as $\mathbb{E}\{\mathbb{E}\{x(t) - \mu(t)|\mathscr{F}_{k,k+1}\}\} = 0$. We thus, conclude

$$\frac{dP_k(t)}{dt} = AP_k(t) + P_k(t)A^\top + Q \,, \forall\, t \in (t_k, t_{k+1}] \,,$$

which shares a form similar to (4.7b).

The last piece is the intervals in which the feedback control law takes effect. Without loss of generality, we consider one interval $(t_i, t_{i+1}]$ with $i \in \mathbb{Z}_{k+1}^{N-1}$, where we have

$$\frac{dP_k(t)}{dt} = AP_k(t) + P_k(t)A^\top + BKP_{i,t,k}(t) + P_{t,i,k}(t)(BK)^\top$$

with $P_{t,i,k}(t) = P_{i,t,k}(t)^\top :=$

$$\mathbb{E}\{\mathbb{E}\{(x(t) - \mu(t))(x(t_i) - \mu(t_i)^\top\}|\mathscr{F}_{k,k+1}\} \,.$$

Applying Itô's Lemma again, we have

$$\frac{dP_{t,i,k}}{dt} = AP_{t,i,k} + BKP_{t,i,k}(t_i) = AP_{t,i,k} + BKP_k(t_i) \,,$$

where the second equality holds by definition.

Reconstruct general dynamics $P(t)$
Considering interval $(t_i, t_{i+1}]$, we have following facts:

1. $\forall\, k \geq i+1$, we have $P_k(t) = 0$.

2. Because the feedback is not active for the sigma-fields $\mathscr{F}_{k,k+1}$, $\forall\, k \geq i$, we have $P_{t,i,k} = 0$, $\forall\, k \geq i$.

Based on the previous derivation, we have

$$\frac{dP(t)}{dt} \overset{(a)}{=} \sum_{k=0}^{N-1} \frac{dP_k(t)}{dt} = \underbrace{\sum_{k=0}^{i-1} \frac{dP_k(t)}{dt}}_{①} + \underbrace{\frac{dP_i(t)}{dt}}_{②} + \underbrace{\sum_{k=i+1}^{N-1} \frac{dP_k(t)}{dt}}_{③}$$

$$= \underbrace{\sum_{k=0}^{i} (AP_i(t) + P_i(t)A^\top)}_{(b)} + Q + \underbrace{\sum_{k=0}^{i-1} BKP_{k,t,i}(t) + P_{t,k,i}(BK)^\top}_{(c)} ,$$

where (a) holds by Lemma 8. ① corresponds to the components whose feedback is active, ② incorporates the stochastic event happening in the current interval $(t_i, t_{i+1}]$, while ③ are the future stochastic events which have no effect yet. The first aforementioned fact allows the reformulation of (b) as

$$(b) = \sum_{k=0}^{N-1} (AP_k(t) + P_k(t)A^\top) .$$

Similarly, the second aforementioned fact reformulates (c) as

$$(c) = \sum_{k=0}^{N-1} BKP_{i,t,k}(t) + P_{t,i,k}(t)(BK)^\top .$$

Hence, by equation (4.8), we conclude

$$\frac{dP(t)}{dt} = AP(t) + P(t)A^\top + BKP_{k,t}(t) + P_{t,k}(t)(BK)^\top + Q .$$

In a similar approach, we have

$$\frac{dP_{t,k}(t)}{dt} = AP_{t,k}(t) + BKP(t_k) ,$$

which concludes the proof. ■

Before proceeding to the predictive control problem, we discuss the physical meaning behind Theorem 9. Equation (4.13) is the definition of $P_{t,k}(t)$, $\forall\ k \in \mathbb{Z}_0^{N-1}$ and the rest of Theorem 9 summarizes its dynamics. In particular, Matrix Q in (4.12a) models the uncontrolled uncertainty happening during interval $(t_k, t_{k+1}]$ and $P_{t,k}$ models the stabilization effect of the feedback control law, $P_{t,k}$ in (4.12b) is the correlation between current time instance t and the previous trigger moment t_k, which reflects the fact that the feedback control law within $(t_k, t_{k+1}]$ only uses information up to t_k to generate a constant feedback. The final piece of Theorem 9, $\lim_{t \to t_k, t > t_k} P_{t,k}(t) = P(t_k)$, links the dynamics between $(t_{k-1}, t_k]$ and $(t_k, t_{k+1}]$. In particular, as the feedback control law updates at t_k, $P_{t,k}$ gets reset at t_k and drops the information $P_{t,k-1}(t_k)$ from the last interval.

Remark 29. *We have $P_{t,0}(t_0) = 0$ and $P(t_0) = 0$ in the first time interval $[t_0, t_1]$, . Hence, by*

Theorem 9, the covariance dynamics in $t \in [t_0, t_1]$ is

$$\frac{dP(t)}{dt} = AP(t) + P(t)A^\top + Q\,, \quad \frac{dP_{t,0}(t)}{dt} = 0, \tag{4.16}$$

which is consistent with the fact that there is no effective feedback within the first interval $[t_0, t_1]$.

4.3.3 Model Predictive Control Scheme

In this section, we summarize a stochastic MPC controller, that incorporates the dynamics derived in Theorem 9 into the self-triggered MPC scheme. In particular, the controller optimizes the expected performance while ensuring input/state constraint satisfaction up to some user defined probability. For the sake of compactness, the saturated resource dynamics are denoted by $g(r(t_k), \Delta_k) := \min\{\rho\Delta_k + r(t_k) - \eta(\Delta_k), \overline{r}\}$. In general, the nominal inputs $\{v_k\}_{k=0}^{N-1}$, the feedback control K, the triggering time instances $\{t_k\}_{k=1}^{N}$ are determined by following problem.

$$\min_{\substack{K,v,\Delta \\ \mu(\cdot),P(\cdot)}} \sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} \mathbb{E}\left(l(x(\tau), u(\tau))\right) d\tau + \mathbb{E}(M(x(t_N))) \tag{4.17a}$$

$$\text{s.t. } \forall\, t \in (t_k, t_{k+1}],\ \forall\, k \in \{0, 1, \ldots, N-1\},$$

$$\begin{cases} \dfrac{d\mu(t)}{dt} = A\mu(t) + Bv_k\,, \\ \dfrac{dP(t)}{dt} = AP(t) + P(t)A^\top + Q \\ \qquad\qquad + BKP_{k,t}(t) + P_{t,k}(t)(BK)^\top, \\ \dfrac{dP_{t,k}(t)}{dt} = AP_{t,k}(t) + BKP(t_k), \end{cases} \tag{4.17b}$$

$$\forall\, t \in [t_0, t_N]\,, \begin{cases} x(t) \sim \mathcal{N}(\mu(t), P(t)), \\ \mathbb{P}(x(t) \in \mathcal{X}) \geq 1 - \epsilon_x, \end{cases} \tag{4.17c}$$

$$\forall\, t \in \{t_k\}_{k=0}^{N-1}, \begin{cases} u(t_k) \sim \mathcal{N}(K\mu(t_k), KP(t_k)K^\top), \\ \mathbb{P}(u(t_k) \in \mathcal{U}) \geq 1 - \epsilon_u, \end{cases} \tag{4.17d}$$

$$\forall\, k \in \{0, 1, \ldots, N-1\}\,,$$

$$\begin{cases} r(t_{k+1}) = g(r(t_k), \Delta_k),\ \Delta_k \in [\underline{\Delta}, \overline{\Delta}], \\ P_{t,k}(t_k) = P(t_k),\ r(t_{k+1}) \in [\underline{r}, \overline{r}], \end{cases} \tag{4.17e}$$

where ϵ_x and ϵ_u are the threshold that the chance constraints (4.17c) and (4.17d) are required to stay above. Notice that due to the feedback with respect to a random event, the actual input value $u(t)$ is uncertain as well. On the practical side, if the feasible $\mathcal{X}$ and $\mathcal{U}$ are polytopic, the chance constraints can be conservatively approximated by an explicit reformulation [234, Chapter 3]. Without loss of generality, we consider $\mathbb{P}(H_{x,i}^\top x(t) \leq h_{x,i})$, $i \in \mathbb{Z}_{i=1}^{n_h}$, where n_h is the number of inequality constraints with respect to x and $H_{x,i} \in \mathbb{R}^{n_x}$ and $h_{x,i} \in \mathbb{R}$. As $x(t)$ follows a Gaussian distribution, any of these constraints can be reformulated as

$$H_{x,i}^{\top}\mu(t) \leq h_{x,i} - \sqrt{H_{x,i}^{\top}P(t)H_{x,i}}\,\mathcal{N}^{-1}(1-\epsilon_x)\,,$$

where $\mathcal{N}^{-1}(\cdot)$ is the inverse cumulative probability distribution function, i.e., $\mathbb{P}(x \leq \mathcal{N}^{-1}(1-\epsilon_x)) = 1-\epsilon_x$.

4.3.4 Implementation Discussion

When Pproblem (4.17) is solved within a direct optimal control scheme, the integration of the ordinary differential equations can be achieved by numerical integration methods such as the Runge-Kutta method or the collocation method [235]. We recommend to use the collocation method, because the triggering time instances are decision variables. If Runge-Kutta is used, the integration depends on high order terms of $\{\Delta_k\}_{k=0}^{N-1}$, which results in low numerical stability. Instead, a collocation method depends linearly on $\{\Delta_k\}_{k=0}^{N-1}$ and hence is numerically more stable.

4.4 Numerical Result

The proposed algorithm is tested on a double integrator with state $x(t) = (x_1(t), x_2(t))$, whose SDE is

$$dx(t) = \left(\begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix} x(t) + \begin{bmatrix}0\\ 1\end{bmatrix} u(t)\right) dt + dW.$$

The controller is designed to track a reference signal oscillating between 1 and −0.4. Only the stage cost is considered with $l(x(t), u(t)) = 10(x_1(t) - \text{ref}(t))^2 + 0.1u(t)^2$ and ref(t) is the tracking reference. The parameters for the chance constraints in (4.17c) and (4.17d) are $\epsilon_x = 0.01$, $\epsilon_u = 0.01$. The recharging rate is 1 with a trigger cost of 0.4. To show the effectiveness of the proposed algorithm, we consider two different cases, $Q = 0.01I$ and $Q = 10^{-4}I$. The former case can be considered as more dangerous than the second case as it has larger process noise. In both cases, we consider a input chance constraint in $[-10, 10]$ with a prediction horizon $N = 10$.

In the first case, the covariance of the process noise is set to be $Q = 0.01I$ and the output is bounded by $y \in [-2, 1]$. In this case, the reference overlaps with the output's upper bound, and the standard deviation of the process noise is around 10% the scale of the output.

A Monte-Carlo simulation of the output responses is shown in Figure **??**, where the controller tries to stay close to the reference, however, as the output is upper bounded by 1, it stays below the upper reference to ensure safety. Regarding another reference signal at −0.4, because it is far away from both constraints, the fluctuations of all the sampled experiments centre around the reference −0.4.

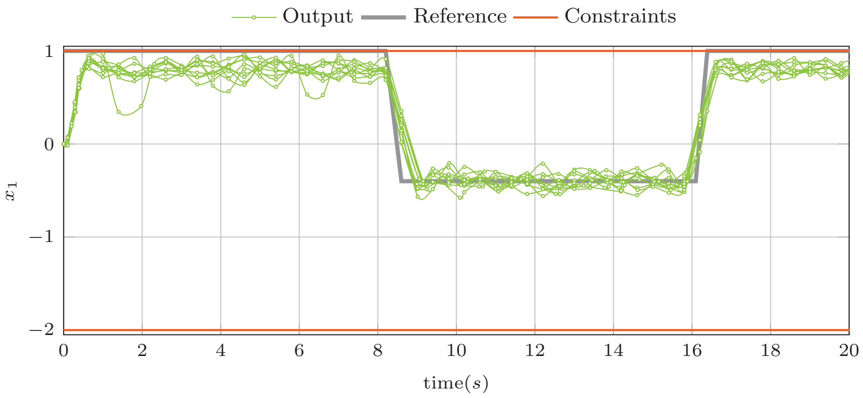


Figure 4.1: Resource response of the stochastic self-triggered MPC ($Q = 10^{-2}I$)

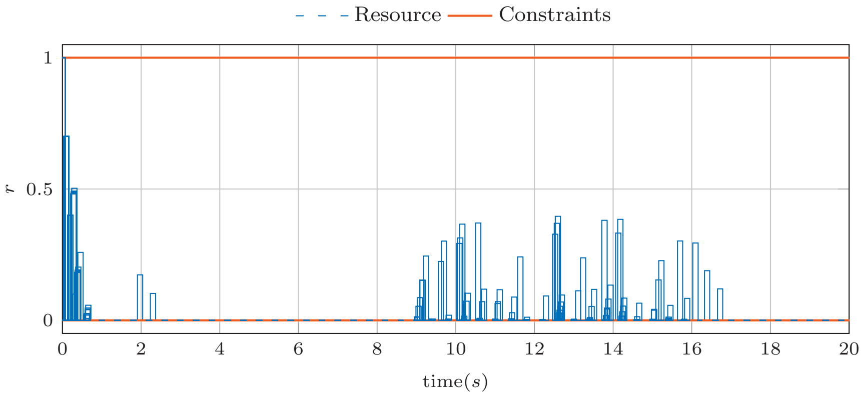


Figure 4.2: Resource response of the stochastic self-triggered MPC ($Q = 10^{-2}I$)

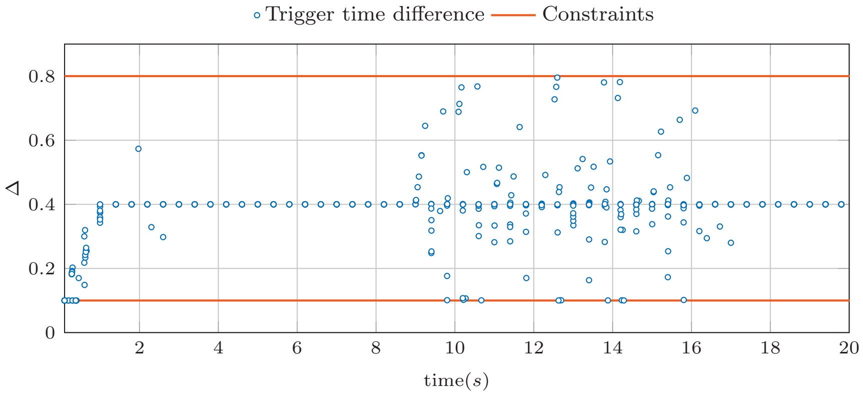


Figure 4.3: Triggering time response of the stochastic self-triggered MPC ($Q = 10^{-2}I$)

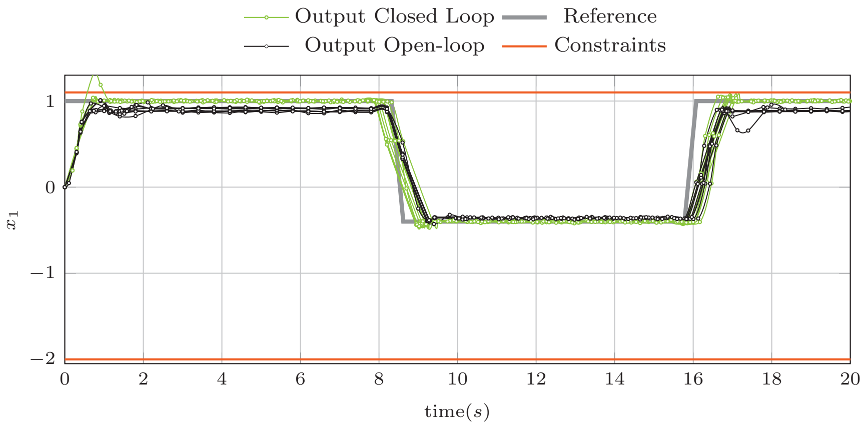


Figure 4.4: Output of the stochastic self-triggered MPC ($Q = 10^{-4} I$)

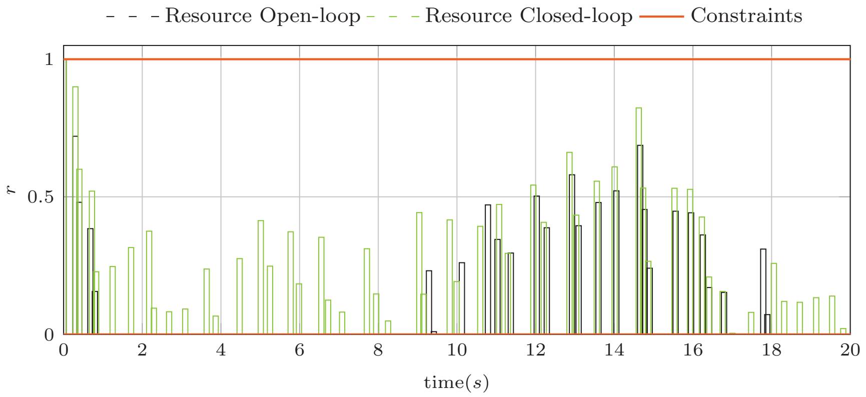


Figure 4.5: Resource response of the stochastic self-triggered MPC ($Q = 10^{-4} I$)

Figure 4.2 and Figure 4.3 show the responses of the resource and the triggering time difference Δ of all Monte-Carlo runs. When the reference is close to the bound, the controller uses the shortest triggering time confined by the resource dynamics (*i.e.* 0.4s) and consumes all the resource. When the reference is farther away from the bound, the resource starts to recharge, which is also reflected as the time between triggering times is larger than 0.4 in Figure 4.3. However, the resource level is lower in comparison with another case because the process noise is large and a more frequent triggers are required to guarantee the controller performance.

In the second case, a smaller process noise $Q = 10^{-4} I$ is considered and the output is bounded by $y \in [-2, 1.1]$. In this case, we compare our proposed scheme against the open-loop scheme to show the necessity of the closed-loop control. Monte-Carlo samples of the output responses are shown in Figure 4.4, where the output of the proposed scheme tightly tracks the reference. Meanwhile, as a stochastic control scheme, one can see that there is a sampled trajectory that violates the upper bound at around 1s. The open-loop scheme also performs the task properly, however, we can see that it stays farther away from the reference 1. To make a cleaner and more informative plot, the resource of one sampled trajectory is shown in Figure 4.5, where we can see that the resource of the closed-loop scheme tends to ramp up when the output is already around the reference and tends to decrease when the reference signal changes. Meanwhile, we can see that the resource consumption in the open-loop scheme is much higher than that of the closed-loop scheme, which means that the open-loop is less desirable as it requires more frequent triggers to maintain the system performance.

4.5 Final Remarks and Other Contributions

Having introduced the resource-aware self-triggered MPC and its stochastic version in this chapter, we can see that the major challenge for self-triggered approach concerns its computational aspect. As the triggering time is a decision variable and the actuator cannot be further updated between two consecutive triggers, the computational aspect becomes more challenging, including the coordination among self-triggered devices, uncertainty propagation and even its numerical implementation. Other work that I carried out in this research direction mainly resolve the aforementioned challenges, which allows me to make use of the self-triggered concept to resolve other applied problems:

- Stricker, N., **Lian, Y.**, Jiang, Y., Jones, C.N. and Thiele, L., 2021, November. Joint energy management for distributed energy harvesting systems. *In Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems (pp. 575-577).*: use self-triggered control to minimize energy use of the actuator, whose energy supply comes from a PV-based enregy harvest system. As the energy availability depends on the illumination condition, which is uncertain depending on the weather condition or the occupant behaviours. Hence, limiting the triggers and accordingly its power consumption, while maintaining the performance, is vital to the reliable operation of this category of devices.

- **Lian, Y.**, Jiang, Y., Stricker, N., Thiele, L. and Jones, C.N., 2021. Robust Resource-Aware Self-Triggered Model Predictive Control. *IEEE Control Systems Letters, 6, pp.1724-1729.*: the robust version of resource-aware self-triggered MPC. The setup is similar to what we did in the stochastic version, while the ellipsoidal calculus is used. The resulting uncertainty propagation dynamics is slightly different from its stochastic version, where the correlation in the direction of time still presents.

- **Lian, Y.**, Wildhagen, S., Jiang, Y., Houska, B., Allgöwer, F. and Jones, C.N., 2020, December. Resource-aware asynchronous multi-agent coordination via self-triggered MPC. *In 2020 59th IEEE Conference on Decision and Control (CDC) (pp. 685-690). IEEE.*: Coordination of multiple self-triggered control object, while allowing each of them to operate/trigger asynchronously.

- **Lian, Y.**, Jiang, Y., Jones, C.N. and Opila, D.F., 2022. Scheduling Delays and Curtailment for Household Appliances With Deterministic Load Profiles Using MPC. *IEEE Control Systems Letters, 6, pp.3301-3306.*: Use the self-triggered control idea to schedule the home appliances with a continuous time formulation. This formulation can reduce the amount of decision variables in comparison with its discrete time formulation, and hence reduce the computational time.

- Stricker, N., **Lian, Y**., Jiang, Y., Jones, C.N. and Thiele, L., 2021, November. Self-triggered Control with Energy Harvesting Sensor Nodes. *To appear in Transactions on Cyber-Physical Systems. IEEE.*: Experimental validation of the self-triggered scheme on enery harvesting sensor nodes for long-term operation.

5 Proximal Bilinear MPC solver

5.1 Introduction

Bilinear systems were originally introduced in [236, 237] to model systems where the dynamics involve products of the inputs and the states. These dynamics may result from the linearization of a nonlinear input affine system and are most commonly used to model convection, and spinning in chemical processes and mechanical systems [238, 239]. Additionally, by means of the concept of Carleman linearization [240], it has been shown that bilinear systems are capable of modeling general nonlinear systems [241]. Meanwhile, with the help of various sophisticated tools such as Lie algebra [242, Chapter 2] and Volterra series [243], bilinear control theory has been explored in depth and has found various successful applications [244–248].

Nonlinear model predictive control (NMPC) is one of the most successful approaches used to control bilinear systems [249–252]. The main idea of NMPC is to achieve the desired performance by optimizing the input in a receding horizon scheme while enforcing state and input constraints [253]. This requires the solution of a nonlinear optimal control problem (OCP) online within a limited update time. Therefore, an efficient solver is a must to run NMPC in real time[1].

Among various real-time NMPC methods, designing and executing an online solver that can run in parallel via distributed algorithms has been a trend over the past decade [254]. Compared to centralized solution approaches, parallelizable methods split the problem into multiple smaller problems such that the computational resources can be utilized more efficiently by exploiting the structure of the OCP being solved. A classical approach used in distributed optimization is based on dual decomposition, where, for example, a gradient-based method [255, 256] or a semi-smooth Newton method [257] have been used to solve the concave dual problem. Another famous approach is the Alternating Direction Method of

[1] In this chapter, real-time means that the MPC solver should return the solution fast enough to enable a desirable operation of the targeted system. Based on our experience, for a mechatronic/mechanical system, the MPC solver should be at least five times faster than the sampling frequency.

Multipliers, which parallelizes the computation by introducing auxiliary variables [254, 258]. These two methods lack convergence guarantees for nonconvex problems and hence are only formally applicable to linear systems. In [259], an augmented Lagrangian based distributed optimization algorithm is proposed, which has been applied to parallelize the computation of MPC problems in [260, 261]. However, despite being parallelized, these algorithms require a solution to multiple non-convex optimization problems in each iteration, which are still numerically intense.

Decomposing an NMPC problem into a set of small-scale problems mainly leverages the linear equality constraints that appear in NMPC problems, which can reflect the topology of a network system or that naturally emerge in the temporal direction via the introduction of auxiliary variables. The latter approach is the horizon splitting method [262, 263], or sometimes termed Schwarz decomposition [264]. It splits the predictive trajectories into short sequential sequences, where linear couplings naturally enforce the equality between the initial states and the terminal state of two adjacent short sequences, hence the name. Within the scope of horizon splitting, tools beyond distributed algorithms have been leveraged to improve efficiency further. The banded structure of the KKT system is the most investigated object in this setup. In [265], a general parallel solver is summarized by a binary-tree-structured algorithm, and in [263], an approximation scheme is introduced to develop a parallel Ricatti solver. However, these algorithms still handle the nonconvex problem directly and, as such, are still numerically challenging.

Another category of methods that are widely used in real-time NMPC leverages the super-linear local convergence property of Newton-type methods to accelerate the online convergence given a good initialization of the decision variables. This category of methods roughly defines the “warm-start" strategy, whose initialization usually derives from the solution information gathered from the preceding time step. A basic approach directly shifts the solution from the last iteration [266], and then a Newton iteration ensures efficient local convergence. Under the umbrella of sequential quadratic programming (SQP), the sensitivity information of the local solution is further used to initialize the KKT system, where an initial guess of active constraints is the most challenging object. In [267], the piece-wise affine property of linear model predictive control (MPC) is used to estimate the change of the active constraint. This idea is generalized in [268] under the name of real-time iteration (RTI), where a sensitivity analysis of the local solution is used to give a piece-wise affine update of the control law.

Instead of solving the NMPC directly online, explicit MPC shifts the online computational burden offline. It treats the MPC control law as a nonlinear mapping from the initial state to control input, and this control law is precomputed offline to enable efficient online calls. In a linear MPC setup, the optimal control law is locally affine [269, 270], and this piece-wise affine parametric solution is first used to pre-compute the MPC control law offline in [270]. However, this algebraic property only holds for linear systems, and its application to nonlinear MPC is limited without approximation [271].

this chapter studies a new proximal-point Lagrangian based algorithm (PPL), which combines the ideas of horizon splitting, explicit MPC, and real-time SQP. In contrast to a standard horizon-splitting approach, a novel interlacing horizon-splitting scheme is introduced. The advantages of the proposed controller are summarized as follows:

1. The PPL algorithm runs computationally efficient iterations, which only require an evaluation of a multi-parametric QP (mpQP) solution and to solve a sparse linear equation system.
2. The detection of the active set is shifted to the mpQP solution, whose problem size is independent of the prediction horizon.
3. A novel interlacing horizon splitting scheme is introduced. The resulting problem has the same number of decision variables as the original NMPC problem without introducing auxiliary variables.
4. The PPL algorithm will not abort even when an infeasible initial state is given. It will output a solution that at least satisfies the input constraint.

5.1.1 Preliminaries

We first recap some existing results from the field of multi-parametric quadratic programming (mpQP) used later in this chapter. A generic convex mpQP can be written in the form of

$$\begin{aligned} \min_{x} \ & \frac{1}{2} x^\top Q x + \theta^\top S x && (5.1a)\\ \text{s.t.} \ & Ax \le b + C\theta \ , && (5.1b) \end{aligned}$$

with decision variables $x \in \mathbb{R}^{n_x}$ and parameters $\theta \in \mathbb{R}^{n_p}$. Here, matrices $Q \in \mathbb{S}_{+}^{n_x}$, $S \in \mathbb{R}^{n_p \times n_x}$, $A \in \mathbb{R}^{m \times n_x}$, $C \in \mathbb{R}^{m \times n_p}$ and vector $b \in \mathbb{R}^m$ are given data. Moreover, we denote by Ω the set of all parameters θ for which (5.1) is feasible. For a mpQP (5.1) with a strongly convex value function, it has been shown (see, e.g., [272]) that Ω is a polyhedron while the solution map $x^\star(\theta) : \mathbb{R}^{n_p} \to \mathbb{R}^{n_x}$ is a continuous piecewise affine (PWA) function of the parameters. Each affine piece is called a critical region [273, Chapter 7.1.2]. Meanwhile, the Lipschitz-continuity holds at $x^\star(\cdot)$, i.e., there exists a positive constant $\eta > 0$ such that for any $\theta_1, \theta_2 \in \Omega$, we have

$$\left\| x^\star(\theta_1) - x^\star(\theta_2) \right\| \le \eta \left\| \theta_1 - \theta_2 \right\| . \tag{5.2}$$

We now recall some definitions from the field of nonlinear programming (NLP). Let us consider NLPs in a generic form

$$\min_{x} \ f(x) \quad \text{subject to} \begin{cases} g(x) = 0 & \mid \lambda, \\ h(x) \le 0 & \mid \kappa. \end{cases} \tag{5.3}$$

Throughout the rest of this chapter, we write Lagrangian multipliers right after the constraints such that $\lambda \in \mathbb{R}^{n_g}$ and $\mathbb{R}^{n_h} \ni \kappa \geq 0$ denote, respectively, the Lagrangian multipliers of the equality constraints and inequality constraints. Functions $f : \mathbb{R}^{n_x} \to \mathbb{R}$, $g : \mathbb{R}^{n_x} \to \mathbb{R}^{n_g}$ and $h : \mathbb{R}^{n_x} \to \mathbb{R}^{n_h}$ are assumed twice continuously differentiable. A primal-dual solution $(x^*, \lambda^*, \kappa^*)$ is called a Karush–Kuhn–Tucker (KKT) point of (5.3) if the following conditions are satisfied [274, Chapter 12.3]

$$\begin{aligned} \nabla f(x^*) + \nabla g(x^*)\lambda^* + \nabla h(x^*)\kappa^* = 0, \\ g(x^*) = 0,\ h(x^*) \leq 0, \\ \forall\, i \in 1, ..., n_h,\ \ [\kappa^*]_i \cdot [h(x^*)]_i = 0, [\kappa^*]_i \geq 0, \end{aligned} \tag{5.4}$$

where $[\kappa^*]_i$ and $[h(x^*)]_i$ define the i-th element of κ^* and $h(x^*)$, respectively. For a given feasible x, we denote by $\mathscr{A}(x)$ the active set at x, i.e., the index set that includes the equality constraints and the inequality constraints that holds equality at x. When the set of active constraint gradients (i.e., $\left[\nabla g(x), \nabla h_{i \in \mathscr{A}(x)}(x)\right]$) is linearly independent at point x, the linear constraint qualification (LICQ) holds [274, Chapter 12.2]. Furthermore, we say the second-order sufficient condition (SOSC) holds at point x, if its hessian $\nabla^2 h(x)$ is positive definite semidefinite on the null space spanned by active constraint gradients [275]. Finally, we say the strict complementary condition (SCC) holds if a dual variable equals zero only when the corresponding constraint is active. Then, we state the definition of regular KKT point for NLP (5.3).

Definition 5. *[275] A given KKT point $(x^*, \lambda^*, \kappa^*)$, is called a regular KKT point if the LICQ, SOSC, and the SCC hold.*

For a given KKT point $(x^*, \lambda^*, \kappa^*)$, if it is regular, then there exists an open neighborhood $\mathscr{B}(x^*)$ around x^* such that the active set is fixed for any $x \in \mathscr{B}(x^*)$, (i.e., $\mathscr{A}(x) = \mathscr{A}(x^*)$) [275]). Regularity at KKT points guarantees the local convergence property when a Newton-type method is applied to solve (5.3) [274].

When the inequality constraint $h(x) \leq 0$ defines a convex set, the first-order optimality condition (5.4) can be further simplified for the sake of compactness:

$$0 \in \nabla f(x^*) + \nabla g(x^*)\lambda^* + \mathscr{N}_{\mathscr{X}}(x^*),$$

with the convex set $\mathscr{X} := \{x \in \mathbb{R}^{n_x} | h(x) \leq 0\}$ and $\mathscr{N}_{\mathscr{X}}(x^*) := \left\{y \middle| (y - x^*)^\top (x - x^*) \leq 0,\ \forall\ x \in \mathscr{X}\right\}$ denotes the normal cone of convex set $\mathscr{X}$ at x^*.

In contrast to the standard Hestenes-Powell augmented Lagrangian method [276, 277], a variant of an augmented Lagrangian method, termed proximal-point Lagrangian [278], is used in this chapter. Given an equality constraint optimization problem

$$\min_x \ f(x) \quad \text{s.t.} \quad g(x) = 0\,,$$

its linearized proximal-point Lagrangian around $\overline{x}$ is defined by

$$\mathscr{L}^{\rho}(x,\lambda,\overline{x}) := f(x) + (\nabla g(\overline{x})\lambda)^{\top} x + \frac{\rho}{2}\|x-\overline{x}\|^2 . \tag{5.5}$$

Note that when $\rho = 0$, it recovers the linearized Lagrangian. For the sake of completeness, the Hestenes-Powell augmented Lagrangian is defined by $f(x) + \lambda^{\top} g(x) + \frac{\rho}{2}\|g(x)\|^2$.

5.2 Problem Formulation

this chapter considers discrete-time, time-invariant bilinear dynamic systems:

$$x_{k+1} = Ax_k + Bu_k + \sum_{i=1}^{n_u} C_i x_k [u_k]_i + B_w w_k \tag{5.6}$$

with state $x_k \in \mathbb{R}^{n_x}$, control inputs $u_k \in \mathbb{R}^{n_u}$ and disturbance w_k at time instant k. For the sake of simplicity, we group the bilinear coefficient matrices $C = [C_1^{\top}, \ldots, C_{n_u}^{\top}]^{\top} \in \mathbb{R}^{n_x \cdot n_u \times n_x}$ and assume that the states and control inputs are subject to the polyhedral constraints

$$\begin{aligned} x_k \in \mathscr{X} &:= \left\{ x \in \mathbb{R}^{n_x} \,\middle|\, P_x x \le p_x \right\} , \\ \text{and } \; u_k \in \mathscr{U} &:= \left\{ u \in \mathbb{R}^{n_u} \,\middle|\, P_u u \le p_u \right\} . \end{aligned}$$

The dynamics (5.6) also includes the case without process noise. For the case with process noise, to make the problem tractable, we consider solving a certainty equivalent problem, where the prediction of the nominal process noise w_k is available. This assumption holds in many energy-related applications: solar radiation in photovoltaic power systems, the outdoor climate in building control, and power generation in airborne wind energy systems, to name a few. An MPC controller can be designed by recursively solving the following optimal control problem in a receding horizon fashion,

$$\min_{x_0,x,u} \sum_{k=0}^{N-1} \ell_k(x_k,u_k) + \ell_N(x_N) \tag{5.7a}$$

subject to:

$$x_0 = x(t), \tag{5.7b}$$

$$\forall\, k \in \{0,1,\ldots,N-1\},$$

$$x_{k+1} = Ax_k + Bu_k + \sum_{i=1}^{n_u} C_i x_k [u_k]_i + B_w w_k \qquad \mid \lambda_k, \tag{5.7c}$$

$$x_{k+1} \in \mathscr{X},\; u_k \in \mathscr{U} \qquad \mid \kappa_k, \tag{5.7d}$$

with $x = [x_1^{\top}, \ldots, x_N^{\top}]^{\top}$, $u = [u_0^{\top}, \ldots, u_{N-1}^{\top}]^{\top}$ and prediction horizon $N \in \mathbb{Z}_{>0}$. For the sake of consistency, variables indexed by bracketed time, such as $x(t)$, denote the actual measurements read out from sensors. Meanwhile, variables indexed by a subscript, such as x_k, denote the predictive “virtual" variables used in the MPC problem. In problem (5.7), the stage cost $\ell_k(\cdot,\cdot)$, $k \in \mathbb{Z}_0^{N-1}$ and terminal cost $\ell_N(\cdot)$ are quadratic and strongly convex, i.e.,

$$\begin{aligned}\ell_k(x,u) =& \frac{1}{2}x^\top Q_k x + q_k^\top x + \frac{1}{2}u^\top R_k u + r_k^\top u,\\ \ell_N(x) =& \frac{1}{2}x^\top Q_N x + q_N^\top x\end{aligned}$$

with user-defined parameters $Q, Q_N \in \mathbb{S}_{++}^{n_x}$, $R \in \mathbb{S}_{++}^{n_u}$ $q_k \in \mathbb{R}^{n_x}$, $r_k \in \mathbb{R}^{n_u}$. Notice that although its objective is strongly convex, solving nonconvex Problem (5.7) is challenging due to the bilinear dynamics (5.7c).

5.3 Algorithm Development

In this part, we will study an algorithm to solve the bilinear MPC problem (5.7) efficiently. Before delving into the algorithmic details, we would first state the logic behind the design of the PPL algorithm. As reviewed in Section 5.1, real-time application of MPC mainly applies and SQP solver with warm-start strategies, or uses explicit MPC. When a good initialization is not available, the detection of active inequality constraints becomes the major performance bottleneck for the SQP algorithm. This requires the design of sophisticated active set detection strategies or the use of merit functions (see e.g. [274, Chapter 18.2] [279, Chapter 2.3] [280, Chapter 2.3]). In the worst case, a poor estimate of the active set will lead to an infeasible QP subproblem, which in the end, aborts the progress of the SQP algorithm. On the contrary, the information of the active set is implicitly saved as critical regions in the explicit solution of explicit MPC. However, its application is limited to linear systems (Section 5.1).

this chapter aims at bringing the benefits of explicit MPC to the SQP method in the application of bilinear MPC. In particular, instead of the direct use of an explicit MPC, the explicit solution will play the role of an active set detector in this chapter. In the rest of this section, we will first introduce a novel interlacing horizon splitting scheme, after which the parallelizable parametric nonconvex solver is elaborated. The convergence properties of the proposed solver are studied in Section 5.3.3. An interpretation of the proposed scheme in the dual space is given in Section 5.3.4, followed by a comparison with related works in Section 5.3.5.

5.3.1 Interlacing horizon splitting reformulation

This section presents the interlacing horizon splitting scheme used later to develop a parallelizable parametric solver to deal with (5.7). As depicted in Fig. 5.1, its main idea is to bind the k-th input u_k with state x_{k+1}. To this end, we introduce the shorthand $\xi_0 = x_0$ and $\xi_k = [u_{k-1}^\top, x_k^\top]^\top$ for all $k \in \mathbb{Z}_1^N$ with associated constraint sets $\Xi_0 = \{\xi \in \mathbb{R}^{n_x} : \xi_0 = x_0 = x(t)\}$ and

$$\begin{aligned}\Xi_k =& \{\xi \in \mathbb{R}^{n_u+n_x} : \xi \in \mathcal{U} \times \mathcal{X}\},\ k \in \mathbb{Z}_1^N\\ =& \{\xi \in \mathbb{R}^{n_u+n_x} : P_\xi \xi \leq p_\xi\}\end{aligned} \tag{5.8}$$

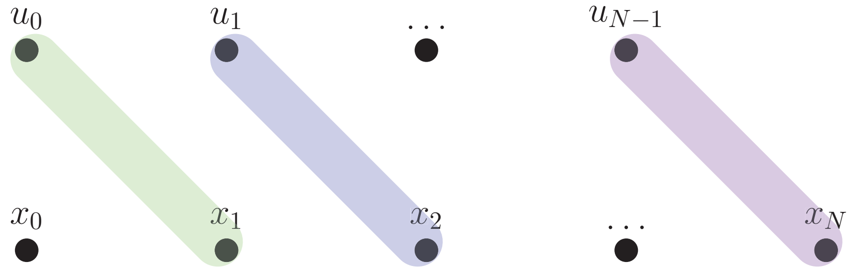


Figure 5.1: Visualization of the interlacing horizon splitting.

with $P_\xi = \operatorname{diag}(P_u, P_x)$ and $p_\xi = [p_u^\top, p_x^\top]^\top$. Moreover, we denote the decoupled objective by

$$\begin{aligned} F_0(\xi_0) &= \frac{1}{2}\xi_0^\top Q_0 \xi_0 + q_0^\top \xi_0, \\ F_k(\xi_k) &= \frac{1}{2}\,\|\xi_k\|^2_{\operatorname{diag}(R_{k-1},Q_k)} + [r_{k-1}^\top, q_k^\top]^\top \xi_k,\ k \in \mathbb{Z}_1^N \end{aligned}$$

and summarize the bilinear dynamics (5.6) by

$$D_k \xi_k + E_k \xi_{k+1} + (S_{k+1}\xi_{k+1} \otimes \mathbf{I}_{n_x})^\top G_k \xi_k = d_k$$

with coefficients $d_k = -B_w w_k,\ k \in \mathbb{Z}_0^{N-1}$,

$$\begin{aligned} D_0 = A,\ D_k &= [\mathbf{0}_{n_x \times n_u},\ A],\ k \in \mathbb{Z}_1^{N-1} \\ E_k &= [B, -\mathbf{I}_{n_x}],\ S_k = [\mathbf{I}_{n_u},\ \mathbf{0}_{n_u \times n_x}],\ k \in \mathbb{Z}_0^{N-1} \\ \text{and}\ \ G_0 = C,\ G_k &= \left[[\mathbf{0}_{n_x \times n_u},\ C_1]^\top, \dots, [\mathbf{0}_{n_x \times n_u},\ C_{n_u}]^\top\right]^\top \end{aligned}$$

for all $k \in \mathbb{Z}_1^{N-1}$. Accordingly, Problem (5.7) can be rewritten as

$$\min_{\xi}\ \sum_{k=0}^{N} F_k(\xi_k) \tag{5.9a}$$

$$\begin{aligned} \text{s.t.}\ \ &(S_{k+1}\xi_{k+1} \otimes \mathbf{I}_{n_x})^\top G_k \xi_k \\ &\quad + D_k \xi_k + E_k \xi_{k+1} = d_k \qquad\qquad \mid \lambda_k,\ k \in \mathbb{Z}_0^{N-1} \end{aligned} \tag{5.9b}$$

$$\xi_k \in \Xi_k, \qquad\qquad \mid \kappa_k,\ k \in \mathbb{Z}_0^{N}\ . \tag{5.9c}$$

5.3.2 Proximal-point Lagrangian Based Parallelizable Solver

Based on the interlacing splitting, the proximal-point Lagrangian introduced in (5.5) are used to design the algorithm. On the one hand, it gives zero local duality gap even under the nonlinear/non-convex dynamics [129, Chapter 11.K]. On the other hand, it enables a **convex** QP, accordingly an mpQP, formulation of parallelizable local problems. In particular, for a given primal trajectory $\overline{\xi}$ (i.e inputs and states) and a dual trajectory $\{\lambda_k\}_0^{N-1}$, the linearized proximal-point Lagrangian of (5.9) w.r.t the equality constraint (5.9b) is given by

$$\begin{aligned}\mathscr{L}^{\rho}(\xi,\lambda,\overline{\xi}) = \mathscr{L}_0^{\rho}(\xi_0,\lambda_0,\overline{\xi}_0,\overline{\xi}_1) + \mathscr{L}_N^{\rho}(\xi_N,\lambda_{N-1},\overline{\xi}_{N-1},\overline{\xi}_N)&\\ +\sum_{k=1}^{N-1}\mathscr{L}_k^{\rho}(\xi_k,\lambda_{k-1},\lambda_k,\overline{\xi}_{k-1},\overline{\xi}_k,\overline{\xi}_{k+1})&\end{aligned} \tag{5.10}$$

with

$$\begin{aligned}&\mathscr{L}_0^{\rho}(\xi_0,\lambda_0,\overline{\xi}_0,\overline{\xi}_1) := F_0(\xi_0) &(5.11a)\\ &\quad +\lambda_0^{\top}\left[D_0 + (S_1\overline{\xi}_1 \otimes \mathbf{I}_{n_x})^{\top}G_0\right]\xi_0 + \frac{\rho}{2}\left\|\xi_0 - \overline{\xi}_0\right\|^2,\\ &\mathscr{L}_k^{\rho}(\xi_k,\lambda_{k-1},\lambda_k,\overline{\xi}_{k-1},\overline{\xi}_k,\overline{\xi}_{k+1}) := &(5.11b)\\ &\quad F_k(\xi_k) + \lambda_{k-1}^{\top}\left(E_{k-1} + \mathrm{mat}(G_{k-1}\overline{\xi}_{k-1})\cdot S_k\right)\xi_k\\ &\quad +\lambda_k^{\top}\left[D_k + (S_{k+1}\overline{\xi}_{k+1}\otimes \mathbf{I}_{n_x})^{\top}G_k\right]\xi_k + \frac{\rho}{2}\left\|\xi_k - \overline{\xi}_k\right\|^2,\\ &\mathscr{L}_N^{\rho}(\xi_k,\lambda_{N-1},\overline{\xi}_{N-1},\overline{\xi}_N) := F_N(\xi_k) + \frac{\rho}{2}\left\|\xi_N - \overline{\xi}_N\right\|^2\\ &\quad +\lambda_{N-1}^{\top}\left(E_{N-1} + \mathrm{mat}(G_{N-1}\overline{\xi}_{N-1})\cdot S_N\right)\xi_N &(5.11c)\end{aligned}$$

with $\rho > 0$ and $\lambda := [\lambda_0^{\top},\lambda_1^{\top},\ldots,\lambda_{N-1}^{\top}]^{\top}$. From the proximal-point Lagrangian (5.10), the benefits of the interlacing horizon splitting become clear. Firstly, the problem is decomposed into $N+1$ independent subproblems in ξ. Secondly, each subproblem is a convex mpQP. Furthermore, the use of the proximal-point Lagrangian allows a simplification of $\mathscr{L}_k^{\rho}(\cdot)$ to a modified proximal form (Section. 5.4.1) [281].

If the primal-dual solution (ξ^*,λ^*) of (5.9) is a regular KKT point, we have solving (5.9) equivalent to solving

$$\begin{aligned}\max_{\lambda}\left(-\sum_{k=0}^{N-1}\lambda_k^{\top}d_k + \min_{\xi}\ \mathscr{L}^{\rho}(\xi,\lambda,\overline{\xi}=\xi^*)\right)&\\ \text{subject to } \xi\in\Xi = \Xi_0\times\ldots\times\Xi_N.&\end{aligned} \tag{5.12}$$

As $\mathscr{L}^{\rho}$ is decoupled in ξ, our main idea to develop a parallelizable algorithm solving (5.9) is to design a primal-dual algorithm to solve the dual problem (5.12) with an iterative update in $\overline{\xi}$.

Algorithm 6 outlines the main steps of the PPL algorithm for solving (5.9). Step 1) deals with decoupled problem (5.13) in parallel, which has explicit solutions as **convex** mpQPs. Particularly, their solution maps are piece-wise affine functions and can be pre-computed offline (See Section 5.4.1). Based on the local solutions ξ, Step 2) evaluates the sensitivities, including the Hessian approximation of the Lagrangian $\mathscr{L}^0$, the gradients of the decoupled objective F_k and the bilinear dynamics residual c_k. The active Jacobian $\hat{P}_{\xi}^{k}$ are constructed based on the active set at local solutions ξ_k. The terminal condition is given in Step 3). It is clear that if these termination conditions hold, we have the iterate (ξ,λ) satisfying the

Algorithm 6 Proximal-point Lagrangian Based Online Solver for Bilinear MPC

Input: an initial guess of $(\overline{\xi}, \lambda)$, a stop tolerance $\epsilon > 0$, a proximal weight ρ and a slack penalty μ

Repeat:

1. Solve decoupled mpQP problems **sequentially or in parallel**,

$$\begin{aligned}
&\min_{\xi_0 \in \Xi_0} \ \mathscr{L}_0^{\rho}(\xi_0, \lambda_0, x(t), \overline{\xi}_1), && \text{(5.13a)}\\
&\min_{\xi_k \in \Xi_k} \ \mathscr{L}_k^{\rho}(\xi_k, \lambda_{k-1}, \lambda_k, \overline{\xi}_{k-1}, \overline{\xi}_k, \overline{\xi}_{k+1}), \ k \in \mathbb{Z}_1^{N-1}, && \text{(5.13b)}\\
&\min_{\xi_N \in \Xi_N} \ \mathscr{L}_N^{\rho}(\xi_N, \lambda_{N-1}, \overline{\xi}_{N-1}, \overline{\xi}_N). && \text{(5.13c)}
\end{aligned}$$

   In all the following steps, ξ_k, $k \in \mathbb{Z}_0^N$ denote optimal solutions of the above QPs.

2. Evaluate sensitivities

$$\begin{aligned}
&H \approx \nabla_{\xi\xi} \mathscr{L}^0(\xi, \lambda, \xi), && \text{(5.14a)}\\
&g_k = \nabla F_k(\xi_k) - \nabla \mathscr{L}_k^{\rho}(\xi, \lambda, \xi) \ , \ k \in \mathbb{Z}_0^N, && \text{(5.14b)}\\
&c_k = D_k \xi_k + E_k \xi_{k+1} \\
&\qquad\qquad + (S_{k+1}\xi_{k+1} \otimes \mathbf{I}_{n_x})^\top G_k \xi_k - d_k, && \text{(5.14c)}
\end{aligned}$$

   and the active Jacobian $\hat{P}_{\xi}^k$ at local solution ξ_k. Here, we use simplified notation $\mathscr{L}_k^{\rho}(\xi, \lambda, \xi)$, $k \in \mathbb{Z}_0^N$. for (5.11).

3. Terminate if $\max_k \|c_k\| \le \epsilon$ and $\max_k \rho \|\xi_k - \overline{\xi}_k\| \le \epsilon$ hold.

4. Solve equality-constrained QP

$$\begin{aligned}
&\min_{\Delta\xi, s} \ \frac{1}{2} \Delta\xi^\top H \Delta\xi + \sum_{k=1}^{N} \left\{ g_k^\top \Delta\xi_k + \mu \|s_k\|^2 \right\} && \text{(5.15a)}\\
&\text{s.t.} \ \Delta\xi_0 = 0 && \text{(5.15b)}\\
&\qquad E_k \Delta\xi_{k+1} + (S_{k+1}\xi_{k+1} \otimes \mathbf{I}_{n_x})^\top G_k \Delta\xi_k \\
&\qquad\quad + \mathrm{mat}(G_k \xi_k) \cdot S_{k+1} \Delta\xi_{k+1} \\
&\qquad\quad + c_k + D_k \Delta\xi_k = 0 \ | \ \lambda_k^{\mathrm{QP}}, \ k \in \mathbb{Z}_0^{N-1} && \text{(5.15c)}\\
&\qquad \hat{P}_{\xi}^k \Delta\xi_k = s_k, \ k \in \mathbb{Z}_1^N. && \text{(5.15d)}
\end{aligned}$$

5. Update $\overline{\xi}^+ = \xi + \Delta\xi$ and $\lambda^+ = \lambda^{\mathrm{QP}}$.

first-order optimality condition

$$\mathbf{O}(\epsilon) \in \nabla_{\xi}\mathscr{L}^0(\xi,\lambda,\bar{\xi}) + \mathcal{N}_{\Xi}(\xi)$$

with $\Xi = \Xi_0 \times \Xi_1 \times \cdots \times \Xi_N$ and the primal feasibility condition

$$\left\| D_k\xi_k + E_k\xi_{k+1} + (S_{k+1}\xi_{k+1} \otimes \mathbf{I}_{n_x})^\top G_k\xi_k \right\| = \mathbf{O}(\epsilon)$$

for all $k \in \mathbb{Z}_0^{N-1}$ up to a small error of order $\mathbf{O}(\epsilon)$. Step 4) deals with a structured equality-constrained QP (5.15). In order to overcome the potential infeasibility caused by the linearization of nonlinear dynamic (5.6) in constraint (5.15c), we introduce a decoupled slackness s_k for each active constraint (5.15d). This makes QP (5.15) always feasible regardless of the feasibility of the original problem (5.9). Note that step 4) is similar to an SQP step, while its active sets are directly generated by the mpQP solutions. On top of this, the mpQPs in step 1) are also always feasible. Therefore, if one applies Algorithm 6 as an online solver for MPC, the resulting MPC controller is always feasible, i.e., the iteration of Algorithm 6 will never fail before termination, and it is independent of the initial condition $x(t)$. This property is desirable in real-world applications because the handling of infeasibility requires a careful design/tuning of a relaxed problem. , even for a feasible problem, the standard SQP algorithm may abort due to an incorrect estimate of the active sets. More specifically, if the estimated active set leads to an infeasible QP, the iterations of the SQP algorithm will stop. In summary, the interlacing horizon splitting scheme enables the mpQPs formulation. The PPL algorithm thereby iteratively calls the mpQP solutions, and the inputs to the mpQPs are iteratively updated in the SQP-style step 4).

Remark 30. *As discussed above, the proposed solver is always feasible even when the initial state makes the NMPC problem infeasible. This property is also observed in the compositions of the projection operator [282, 283], whose convergence to a point pair that are closest to all the sets is proved. However, in a nonconvex setup, the property of the convergent results is unclear and remains open for future research.*

5.3.3 Local Convergence Property

This section shows that the PPL Algorithm 6 asymptotically converges to the local optimal solution of (5.7) at a quadratic rate. The logic behind the constructive proof follows two facts: local mpQPs (5.13) have a Lipschitz-continuous solution map; and the coupled QP (5.15) is equivalent to a Newton-type method. To this end, we introduce the following lemma first to establish the quadratic contraction of the solution of (5.15).

Lemma 10. *Let the KKT point* (ξ^*,λ^*) *of Problem* (5.9) *be regular such that the solution* ξ^* *is a local minimizer. Moreover, let the Hessian evaluation* H *satisfy*

$$H = \nabla_{\xi\xi}\mathscr{L}^0(\xi,\lambda,\xi) + \mathbf{O}(\|\xi - \bar{\xi}\|). \tag{5.16}$$

and let μ *in* (5.15) *satisfy* $\frac{1}{\mu} \le \mathbf{O}(\|\xi - \bar{\xi}\|)$. *Then, there exists* $\alpha > 0$*, the solution to* (5.15) *locally satisfies,*

$$\left\|\bar{\xi}^{+} - \xi^{*}\right\| \le \alpha \left\|\xi - \xi^{*}\right\|^{2}, \ \|\lambda^{+} - \lambda^{*}\| \le \alpha\|\xi - \xi^{*}\|^{2}. \tag{5.17}$$

The term "local" in the statement means that the initial guess of primal-dual iterates locates within a small neighborhood around the local solution (ξ^*, λ^*).

Proof. Based on the definition 5 of regular KKT point, we have that the active sets are not changed locally [284]. Then, the standard analysis of Newton's method gives

$$\begin{aligned}\left\|\begin{bmatrix}\bar{\xi}^{+} - \xi^{*}\\ \lambda^{+} - \lambda^{*}\end{bmatrix}\right\| &\le \|H - \nabla_{\xi\xi}\mathscr{L}^{0}(\xi,\lambda,\xi)\| \cdot \mathbf{O}\left(\|\xi - \xi^{*}\|\right)\\ &+ \mathbf{O}\left(\|\xi - \xi^{*}\|^{2}\right) \overset{(5.16)}{\le} \mathbf{O}\left(\|\xi - \xi^{*}\|^{2}\right)\end{aligned}$$

as discussed in [274, Chapter 3.3], which concludes the proof. ■

Intuitively speaking, this lemma states that if the iterates given by (5.13) are close to the optimal solution to (5.7), then the distance between iterate given by (5.15) and the optimal solution contract quadratically. The following theorem intends to show that this quadratic contraction also holds even when we consider the iterates given by (5.13). Based on condition (5.16), we have that there exists a constant $\alpha > 0$ such that the local quadratic contraction (5.17) holds. Then, we define by ξ^+ the solution of (5.13) based on the updated primal-dual iterate $(\bar{\xi}^{+}, \lambda^{+})$ such that we can summarize the local convergence result as follows.

Theorem 10. *Let all assumptions in Lemma 10 be satisfied. The iterates* ξ *of Algorithm 6 locally converges to the local minimizer* ξ^* *of Problem* (5.9) *with quadratic rate.*

Proof. As discussed in Section 5.1.1, we have the local solution map $\xi^{\star}(\bar{\xi}, \lambda)$ of convex mpQPs (5.13) are Lipschitz continuous such that we have

$$\left\|\xi^{\star}(\bar{\xi}^{+}, \lambda^{+}) - \xi^{\star}(\xi^{*}, \lambda^{*})\right\| = \|\xi^{+} - \xi^{*}\| \le \eta \left\|\begin{bmatrix}\bar{\xi}^{+} - \xi^{*}\\ \lambda^{+} - \lambda^{*}\end{bmatrix}\right\|$$

with a constant $\eta > 0$. Here, we use the fact $\xi^* = \xi^{\star}(\xi^*, \lambda^*)$, i.e., if we initialize Algorithm 6 with the optimal solution (ξ^*, λ^*), the solution of convex mpQPs (5.13) is equal to the local minimizer ξ^*. Moreover, iterate ξ^+ is the solution of (5.13) if one starts the Algorithm 6 with $(\bar{\xi}^{+}, \lambda^{+})$ as the initial guess. Substituting (5.17) into the inequality above yields

$$\|\xi^{+} - \xi^{*}\| \le \alpha \cdot \eta \left\|\xi - \xi^{*}\right\|^{2},$$

which is sufficient to establish the local quadratic convergence of iterates ξ to the local minimizer ξ^* [274, Chapter 3.3]. ■

It is worth mentioning that the same order of local convergence speed holds in a wide range of second-order algorithms, such as the SQP Algorithm [274, Chapter 18.7] and the augmented Lagrangian based alternating direction inexact Newton (ALADIN) method [259]. The theoretical importance of Theorem 1 shows that such convergence rate is still preserved even when another layer of mpQPs is added (i.e., step 1) in Algorithm 6). Therefore, regarding the motivation discussed at the beginning of this Section 5.3, the PPL algorithm not only achieves efficient convergence as the Newton-type algorithm but also achieves an efficient active set detection mechanism via the concept of explicit MPC (i.e., mpQPs). Hence, in comparison with the standard SQP, the detection of active sets via mpQPs makes the PPL algorithm advantageous. Additionally, such a benefit does not increase the iteration cost significantly, which in practice, retains a low absolute computational time. This is not the case in other SQP-style extensions, such as the ALADIN method, and we postpone the detailed comparison with ALADIN to Section 5.3.5.

5.3.4 Dual Interpretation

In this part, we would show a different but more intuitive view of the PPL algorithm. The expansion of the first-order information g_k used in (5.15) gives

$$g_k = \left(\mathrm{diag}(R_{k-1}, Q_k) + \rho \mathbf{I}_{n_x+n_u}\right)\xi_k + [r_{k-1}^\top; q_k^\top]^\top + P_\xi^k \kappa_k.$$

Moreover, recall that P_ξ is the parameter of the inequality constraints (5.8), and κ_k are the corresponding dual variables (5.9c), which are generated by the mpQPs solutions directly. By inspecting the objective function in (5.15), the quadratic penalty term $\mu\|s_k\|^2$ and g_k together recover an augmented Lagrangian defined at ξ, where the active inequality constraints are dualized. This observation leads to a dual interpretation of the PPL algorithm.In the local problems (5.13) (i.e., step 1) in Algorithm 6), the system dynamics are dualized with fixed dual variables (i.e., $\{\lambda_k\}$) based on the proximal-point Lagrangian. The dual variables to the inequality constraints (i.e., $\{\kappa_k\}$) are thereby updated. Accordingly, the coupled QP problem (5.15) (i.e., step 4) in Algorithm 6) updates the dynamics dual with the dual variables to the inequality constraints fixed. Hence, the PPL algorithm can be viewed as an alternating direction method in the dual space. Via the scope of this dual interpretation, the coupled QP (5.15) is not a relaxed problem, as the augmented Lagrangian is an exact penalty function [274, Chapter 17].

With this dual interpretation at hand, we are ready to elaborate on the reasoning behind the use of proximal-point Lagrangian. Firstly, the dual variables model first-order local properties [285], and an aggressive update should therefore be avoided due to such locality. The proximal term (i.e., $\frac{\rho}{2}\|\xi_k - \overline{\xi}_k\|^2$ in the local steps (5.13)) realize this goal. This is important, especially when a good estimate of dual variables is not yet available. Secondly, as dual variables encompass first-order information, linearization is therefore needed. More specifically, the dual to the dynamics is updated to $\overline{\xi}$ in the coupled QP step. Hence the proximal-point Lagrangian is linearized around $\overline{\xi}$ in (5.13). It is worth noting that the design logic similar to the aforementioned one also appears in other nonconvex primal alternating direction methods, such as [218]. To the best of our knowledge, the PPL algorithm is the first algorithm to bring this idea to the dual space.

5.3.5 Comparison with Related Work

In this part, we will compare the proposed scheme with other related results, particularly the augmented Lagrangian based alternating direction inexact Newton (ALADIN) method. The ALADIN method is also an extension of the SQP algorithm, but it can only handle linear coupling. Hence, auxiliary variables that duplicate the states are introduced in order to handle the bilinear dynamics. More specifically, ALADIN reformulates the bilinear MPC problem (5.7) to the following equivalent problem:

$$\min_{x_0,x,u} \sum_{k=0}^{N-1} \ell_k(x_k,u_k) + \ell_N(x_N)$$

subject to:

$$\begin{aligned}
& x_0 = x(t), \\
& \forall\, k \in \{0,1,...,N-1\}, \\
& z_{k+1} = Ax_k + Bu_k + \sum_{i=1}^{n_u} C_i x_k [u_k]_i + B_w w_k \\
& x_{k+1} = z_{k+1} \ \mid \tilde{\lambda}_k, \qquad (5.18a) \\
& z_{k+1} \in \mathscr{X},\ x_k \in \mathscr{X},\ u_k \in \mathscr{U}
\end{aligned}$$

where auxilary variables z_k duplicate the states x_k, and a linear coupling constraint is thereby introduced in (5.18a). This is the standard horizon splitting scheme used in other nonlinear MPC algorithms [261–264], where inputs and states of the same time step are grouped together. This leads to the first advantage of this chapter's proposed splitting scheme: no auxiliary variables are introduced, so the problem size remains unchanged. This benefit also leads to a limitation of the PPL algorithm: the proposed splitting scheme requires that the stage cost and the constraints are decoupled between states and inputs. How to overcome this limitation is left for future research.

In the ALADIN algorithm, following N nonconvex subproblems $\{\mathscr{P}_k\}_{k=0}^{N-1}$ can be solved in parallel:

$$\forall k \in \{0,1,\ldots,N-2\}$$

$$\mathscr{P}_k := \min_{\substack{x_k, u_k \\ z_{k+1}}} \ell_k(x_k, u_k) + \begin{bmatrix} \tilde{\lambda}_{k-1} \\ -\tilde{\lambda}_k \end{bmatrix}^\top \begin{bmatrix} x_k \\ z_{k+1} \end{bmatrix} \\ + \frac{\rho}{2} \left\| \begin{bmatrix} x_k \\ z_{k+1} \end{bmatrix} - \begin{bmatrix} \overline{x}_k \\ \overline{z}_{k+1} \end{bmatrix} \right\|^2$$

subject to:

$$\begin{aligned} & z_{k+1} = Ax_k + Bu_k + \sum_{i=1}^{n_u} C_i x_k [u_k]_i + B_w w_k \\ & z_{k+1} \in \mathscr{X},\ x_k \in \mathscr{X},\ u_k \in \mathscr{U} \end{aligned}$$

$$\mathscr{P}_{N-1} := \min_{\substack{x_{N-1}, u_{N-1} \\ z_N}} \ell_k(x_{N-1}, u_{N-1}) + \ell(z_N) + \tilde{\lambda}_{N-1}^\top x_{N-1} \\ + \frac{\rho}{2} \| x_k - \overline{x}_k \|^2$$

subject to:

$$\begin{aligned} & z_N = Ax_{N-1} + Bu_{N-1} + \sum_{i=1}^{n_u} C_i x_k [u_k]_{N-1} + B_w w_{N-1} \\ & z_N \in \mathscr{X},\ x_{N-1} \in \mathscr{X},\ u_{N-1} \in \mathscr{U} \end{aligned}$$

After the parallel iteration, the ALADIN algorithm applies a relaxed SQP-style step to the reformulated problem (5.18) in order to update the coupling dual variables $\{\tilde{\lambda}_k\}$. The differences between the ALADIN and the proposed scheme now become clear:

- The proposed scheme only needs to solve a convex QP, whose solutions can be precomputed offline via mpQPs. Instead, the ALADIN algorithm has to solve a nonconvex problem online. Even though these nonconvex subproblems can be computed in parallel, the resulting computational cost per iteration is still significantly higher than the proposed scheme.
- The proposed scheme directly handles the nonlinear coupling (i.e. the bilinear dynamics) and therefore does not need to introduce auxiliary variables to duplicate the states. As a result, the SQP step used in the proposed scheme solves a smaller problem than the one solved in the ALADIN.

Due to the use of a SQP-style update and use of augmented Lagrangian methods [259, Section 4], the PPL algorithm and the ALADIN algorithm have a certain similarity. However, their focus during the algorithm design are different. ALADIN focuses more on the allocation of the computational complexity, while the PPL algorithm aims at efficient iteration with a good active detection scheme. That is why the ALADIN tends to handle the non-convexity directly, as this can be handled by different computational nodes. On the contrary, the PPL algorithm is customized to bilinear MPC to have a lower computational cost per iteration and fewer decision variables. In summary, even though both the ALADIN and the proposed scheme can be viewed as extensions to the SQP algorithm, the ALADIN is more tailored for distributed computation. The proposed scheme is instead tailored for efficient online computation. Finally, we wrap up this section by summarizing the benefits of the proposed scheme as follows:

- It brings the efficiency of explicit MPC into an NMPC setup. The integration of the explicit mpQP solution provide two benefits: it returns an accurate primal solution when good estimates of the dual variables $\{\lambda_k\}$ are given, and it significantly improves the real-time efficiency by providing the active set estimation.
- It retains the SQP structure. This not only preserves the convergence rate of the SQP algorithm but also makes the PPL algorithm compatible with any existing acceleration strategy developed for real-time SQP, such as warm-start.
- It enjoys high computational efficiency even without parallelization. With proper implementation, this efficiency may be further improved by parallelization for some processors.

Remark 31. *It is worth mentioning that the proposed scheme reduces to an ALADIN algorithm when the dynamics are linear (i.e., $C_k = 0$). In this case, the resulting algorithm is similar to the one studied in [261], where global convergence is also guaranteed [286].*

Remark 32. *The alternating direction method of multipliers (ADMM) can be used to solve a biconvex optimization problem [254, Chapter 9.2]. With the standard formulation given in [254, Chapter 9.2], the ADMM algorithm needs to solve a nonconvex QP problem in each iteration, which is proved to be NP-hard even for the calculation of a local minimizer [287, 288]. The resulting computational cost per iteration is significantly higher, and such ADMM formulation is therefore not suitable for our comparison. If the proposed interlacing horizon splitting scheme is applied instead, the resulting ADMM algorithm gets rid of the solution of an nonconvex QP, which is also one contribution of this chapter. However, the ADMM algorithm is still not suitable for comparison. On the one hand, a convergence guarantee exists only when there is no state constraint, which is not desirable in MPC applications. Based on our test on the numerical given in the following Section 5.5.1, we did not observe the convergence of the ADMM after 3000 iterations (equivalently 1 minute in absolute time). On the other hand, the bilinear dynamics are squared in the augmented Lagrangian. The resulting problem is no longer an mpQP, and cannot be precomputed offline. As a result, multiple inequality constraint QPs are required to be solved in each iteration, leading to a much higher computational cost. Finally, even though we did not observe convergence in our numerical study, if it happens to converge for some specific cases, the convergence rate of a nonconvex ADMM algorithm is at most sublinear [289]. Therefore, it requires more iterations and accordingly, more computational time to converge.*

5.4 Implementation Details

This section elaborates on the implementation details of Algorithm 6 with a particular emphasis on run-time aspects and a limited memory requirement. Here, the implementation of Steps 3) and 5) turns out to be straightforward such that we focus on the implementation of Steps 1), 2) and 4).

5.4.1 mpQP Subproblems

We summarize the local mpQPs (5.13) into a uniform form

$$\mathbb{P}(\theta_k): \min_{\xi_k \in \Xi_k} \frac{1}{2}\xi_k^\top \mathcal{Q}_k \xi_k + \theta_k^\top \xi_k \tag{5.20}$$

with parametric inputs $\theta_k \in \mathbb{R}^{n_x+n_u}$ and coefficient matrices $\mathcal{Q}_k = \text{diag}(R_{k-1}, Q_k) + \rho \mathbf{I}_{n_x+n_u}$ for all $k \in \mathbb{Z}_1^N$. Here, the first problem is omitted as its solution is fixed by $\xi_0 = x(t)$ due to the initial state constraint enforced by Ξ_0. Based on the formulation of $\mathscr{L}_k^\rho$, we can work out the explicit form of θ_k as follows,

$$\begin{aligned} \theta_k =& [r_{k-1}^\top; q_k^\top]^\top + \left(E_{k-1} + \text{mat}(G_{k-1}\overline{\xi}_{k-1}) \cdot S_k\right)^\top \lambda_{k-1} \\ &+ \left[D_k + (S_{k+1}\overline{\xi}_{k+1} \otimes \mathbf{I}_{n_x})^\top G_k\right]^\top \lambda_k - \rho \overline{\xi}_k \end{aligned} \tag{5.21a}$$

$$\begin{aligned} \theta_N =& [r_{N-1}^\top; q_N^\top]^\top - \rho\overline{\xi}_N \\ &+ \left(E_{N-1} + \text{mat}(G_{N-1}\overline{\xi}_{N-1}) \cdot S_N\right)^\top \lambda_{N-1}. \end{aligned} \tag{5.21b}$$

Evaluating these parameters only requires matrix-vector multiplications such that the complexity is $\mathcal{O}(N \cdot (n_x + n_u)^2)$. In this chapter, we use the enumeration-based multi-parametric QP algorithm from [290] for generating solution maps $\xi_k^\star : \mathbb{R}^{n_x+n_u} \to \mathbb{R}^{n_x+n_u}$ of (5.20). The complexity of pre-processing the small-scale QPs (5.20) depends on the number of critical regions $N_{\text{R},k}$ over which the PWA optimizers $\xi_k^\star(\cdot)$ are defined [272]. Here, we assume that each parametric QP is post-processed, off-line, to obtain binary search trees [291] in $\mathcal{O}(N_{\text{R},k}^2)$ time. Once the trees are constructed, they provide for a fast evaluation of the solution maps in (5.20) in time that is logarithmic in the number of regions, thus establishing the $\mathcal{O}(\sum_k \log_2(N_{R,k}))$ on-line computational bound. The memory requirements are directly proportional to the number of critical regions, with each region represented by a finite number of affine half-spaces. Finally, it is worth mentioning that if $Q_i = Q_j,\ q_i = q_j,\ R_i = R_j,\ r_i = r_j,\ \forall\ i \neq j$, then the i-th mpQP subproblems (5.20) is identical to the j-th one, and one mpQP solution can therefore serve for two subproblems. Identical subproblems happen in many MPC applications as the stage cost are usually fixed throughout the prediction horizon.

5.4.2 Sensitivities Evaluation

Step 2) of Algorithm 6 evaluates the sensitivities g_k, c_k, $\hat{P}_\xi^k$ and H. As we consider the quadratic cost, the gradients g_k can be easily evaluated with analytical form. Moreover, the primal feasibility residual c_k and active Jacobian $\hat{P}_\xi^k$ are also straightforward. Therefore, we focus on the computation of the Hessian matrix H in this subsection.

As we used an interlacing horizon splitting scheme, the exact Hessian $\nabla_{\xi\xi}L^0(\xi,\lambda,\xi)$ is not block diagonal with respect to each y_k but the banded block diagonal. However, as the off-diagonal blocks only involve the bilinear dynamics, we can work out each block analytically as follows

$$\nabla_{\xi\xi}L^0(y,\lambda,y) = \begin{bmatrix} \mathcal{Q}_0 & S_{0,1} & & \\ S_{1,0} & \mathcal{Q}_1 & S_{1,2} & \\ & \ddots & \ddots & \ddots \\ & & S_{N-1,N} & \mathcal{Q}_N \end{bmatrix}$$

with blocks

$$\begin{aligned} & S_{0,1} = S_{1,0}^\top = [C_1^\top \lambda_0, \dots, C_{n_u}^\top \lambda_0, \mathbf{0}_{n_x\times n_x}] \in \mathbb{R}^{n_x\times(n_u+n_x)} \\ & S_{k,k+1} = S_{k,k+1}^\top \\ = & \begin{bmatrix} C_1^\top \lambda_k & \dots & C_{n_u}^\top \lambda_k & \mathbf{0}_{n_x\times n_x} \\ \mathbf{0}_{n_u} & \dots, & \mathbf{0}_{n_u} & \mathbf{0}_{n_u\times n_x} \end{bmatrix} \in \mathbb{R}^{(n_x+n_u)\times(n_x+n_u)} \end{aligned}$$

for all $k \in \mathbb{Z}_1^{N-1}$. It is clear that evaluating the exact Hessian is equivalent to evaluate $C_i^\top \lambda_k$ for all $i \in \mathbb{Z}_1^{n_u}$ and $k \in \mathbb{Z}_0^{N-1}$. Therefore, its computational complexity is only $\mathcal{O}(N n_u n_x^2)$. In practice, some heuristics can be adopted to achieve a better numerical robustness on the convergence performance of Algorithm 6 such as enforcing $H \approx \nabla_{\xi\xi}\mathcal{L}^0 \succ 0$ by adding a regularization term, i.e., $H = \nabla_{\xi\xi}\mathcal{L}^0 + \sigma\mathbf{I}$ with $\sigma \geq 0$ [292].

5.4.3 Coupled QP

The coupled QP (5.15) has no inequality constraints such that solving (5.15) is equivalent to solving linear equations defined by the KKT system:

$$\underbrace{\begin{bmatrix} H + \rho \hat{P}_\xi^\top \hat{P}_\xi & J^\top \\ J & \end{bmatrix}}_{\mathcal{H}} \underbrace{\begin{bmatrix} \Delta\xi \\ \lambda^{\mathrm{QP}} \end{bmatrix}}_{w} = \begin{bmatrix} -g \\ -c \end{bmatrix} \tag{5.22}$$

with

$$J = \begin{bmatrix} \tilde{D}_0 & \tilde{E}_0 & & & \\ & \tilde{D}_1 & \tilde{E}_1 & & \\ & & \ddots & \ddots & \\ & & & \tilde{D}_N & \tilde{E}_N \end{bmatrix}, \quad \begin{aligned} \hat{P}_\xi &= \mathrm{diag}(\hat{P}_\xi^1, \dots, \hat{P}_\xi^N) \\ g &= [g_0^\top, g_1^\top, \dots, g_N^\top]^\top \\ c &= [c_0^\top, c_1^\top, \dots, c_N^\top]^\top \end{aligned}$$

and for all $k \in \mathcal{I}_0^N$,

$$\begin{aligned}\widetilde{D}_k &= D_k + (S_{k+1}\overline{\xi}_{k+1} \otimes \mathbf{I}_{n_x})^\top G_k,\\ \widetilde{E}_k &= E_k + \operatorname{mat}(G_k z_k) \cdot S_{k+1}.\end{aligned}$$

If we rearrange the KKT matrix $\mathscr{H}$ by resorting w as

$$(\Delta\xi_0,\, \lambda_0^{\mathrm{QP}},\, \Delta\xi_1,\, \lambda_1^{\mathrm{QP}}, \ldots, \Delta\xi_{N-1},\, \lambda_{N-1}^{\mathrm{QP}},\, \Delta\xi_N),$$

a tri-blocked-diagonal sparsity pattern appears in the KKT matrix $\mathscr{H}$, such that the Schur-complement based back-forward sweeps can be used to solve the linear equation efficiently. In order to better illustrate this idea, we consider $N = 2$ such that the resulting rearranged KKT system is

$$\left[\begin{array}{cccc|c} \mathscr{Q}_0 & \widetilde{D}_0^\top & S_{0,1} & & \\ \widetilde{D}_0 & & \widetilde{E}_0 & & \\ S_{1,0} & \widetilde{E}_0^\top & \widetilde{\mathscr{Q}}_1 & \widetilde{D}_1^\top & S_{1,2} \\ & & \widetilde{D}_1 & & \widetilde{E}_1 \\ \hline & & S_{2,1} & \widetilde{E}_1^\top & \widetilde{\mathscr{Q}}_2 \end{array}\right] \begin{bmatrix} \Delta\xi_0 \\ \lambda_0^{\mathrm{QP}} \\ \Delta\xi_1 \\ \lambda_1^{\mathrm{QP}} \\ \Delta\xi_2 \end{bmatrix} = \begin{bmatrix} -g_0 \\ -c_0 \\ -g_1 \\ -c_1 \\ -g_2 \end{bmatrix}$$

with $\widetilde{\mathscr{Q}}_k = \operatorname{diag}(R_{k-1}, Q_k) + \mu(\hat{P}_\xi^k)^\top \hat{P}_\xi^k$. We start the backward sweep by considering the whole KKT matrix as a 2x2 block matrix. Then, applying the Schur complement with respect to the lower left block $\widetilde{\mathscr{Q}}_2$ yields a reduced KKT matrix

$$\left[\begin{array}{cc|cc} \mathscr{Q}_0 & \widetilde{D}_0^\top & S_{0,1} & \\ \widetilde{D}_0 & & \widetilde{E}_0 & \\ \hline S_{1,0} & \widetilde{E}_0^\top & \widetilde{\mathscr{Q}}_1 + S_{1,2}\widetilde{\mathscr{Q}}_2^{-1}S_{2,1} & \widetilde{D}_1^\top + S_{1,2}\widetilde{\mathscr{Q}}_2^{-1}\widetilde{E}_1^\top \\ & & \widetilde{D}_1 + \widetilde{E}_1\widetilde{\mathscr{Q}}_2^{-1}S_{2,1} & \widetilde{E}_1\widetilde{\mathscr{Q}}_2^{-1}\widetilde{E}_1^\top \end{array}\right].$$

Applying the Schur complement once more results in a reduced KKT system with respect to only $(\Delta\xi_0, \lambda_0)$ such that the substitution of the initial condition $\Delta\xi_0 = 0$ can enable a forward sweep to recover the primal-dual solution $(\Delta\xi, \lambda)$. This method has been shown that it is equivalent to the Riccati recursion in dealing with LQR problems [293]. As the update of the right-hand side of the KKT system only requires matrix-vector multiplication, we observe that the computational complexity of this linear solver is dominated by the matrix update (i.e., computation of the Schur complement), which is $\mathscr{O}(N(n_x + n_u)^3)$.

5.5 Numerical Results

This section studies the PPL algorithm on two bilinear system examples. The PPL algorithm is first compared against other state-of-art solvers on a building control problem running on a laptop computer. The algorithm is then implemented in an embedded microcontroller for speed control of a DC motor. The binary search tree of the mpQP solutions used in the PPL algorithm is generated by the multi-parametric toolbox (`MPT 3.0`) [294].

5.5.1 Bilinear Building Control

In this part, the PPL algorithm is compared with an efficient optimal control solver `acados` [279] and the ALADIN algorithm, which is implemented by `ALADIN-`α toolbox [295]. The code generation in `acados` is based on the SQP method with exact Hessians and without/with condensing. All the algorithms use the mirror method to regularize the indefinite QP problem [292]. It is worth mentioning that `acados` is highly optimized for MPC, whose linear algebra subroutine `BLASFEO` [296] and QP solver [297] exploit the structure in MPC. On top of that, a sophisticated active set detection scheme by exact penalty function is implemented in `acados` [279, Chapter 2.3]. Hence, this comparison can demonstrate the performance of the PPL algorithm.

We considered a multi-zone building model reported in [86] with room indices shown in Figure. 5.2. Due to the space limit, the parameters of the model (i.e., A, B, B_w, C, matrices) are included in the supplementary material on Github. In this multi-zone building (Figure 5.2), room 2 is the corridor linking a large warehouse (room 1) and two offices (rooms 3 and 4). The indoor temperature of room 1 is controlled by an independent HVAC system, while another HVAC controls the temperature of all other rooms. The corresponding control inputs ($u \in \mathbb{R}^2$) are the valve positions in the air handling unit, where the heat transfer between the air and the hot water flowing in the heating coil results in the bilinear term in the system dynamics. As a result, the control inputs can manipulate the supply air temperature in a nonlinear way, which accordingly controls the indoor temperatures. In summary, this is a 15 dimensional model (i.e. $x \in \mathbb{R}^{15}$) with two dimensional control inputs, the states include the indoor temperature, wall temperature between two different rooms, wall temperature that stands between a specific room, and outdoor, and supply air temperature control. Process noises are outdoor temperature and solar radiation (i.e. $w \in \mathbb{R}^2$). In building control, a common practice is to apply certainty equivalence control [298], which uses weather forecast as the nominal disturbance in the MPC formulation. Meanwhile, the building evolves under the actual weather condition that is similar but not identical to the weather forecast. Real-world weather data is used for this numerical study.

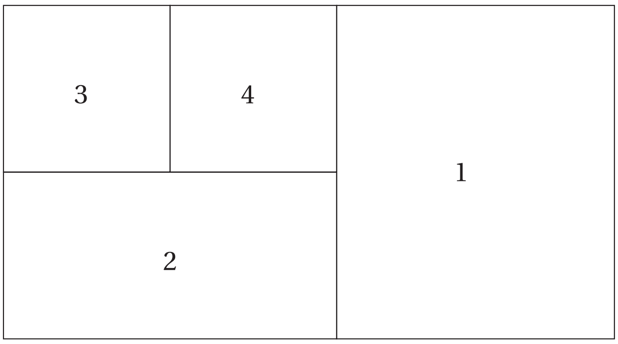


Figure 5.2: Schematic diagram of the multi-zone building

Using this approach, 100 Monte-Carlo tests were conducted with recorded weather from tomorrow.io [89] in winter (Fig.5.3(a)-(b) plot one sampled weather condition). The weather forecast used in the MPC problem is the recorded weather perturbed by zero-mean random noise, while the simulation uses the recorded weather (i.e., the weather forecast curves in Figure 5.3 (a)-(b)). The prediction horizon is set to 8 with an objective of minimizing energy consumption, whose loss function is

$$\ell_k(x_k, u_k) = u_k^2 \,.$$

The indoor temperature is bounded within $[22, 24]°C$ to ensure occupant comfort. The control input (i.e., fractional valve position) is bounded within $[0, 1]$.

Remark 33. *It is possible to define an objective as $\ell_k(x_k, u_k) = |u_k|$. The resulting local problem can be reformulated as a linear program and, thus, also an mpQP. We use a quadratic loss function here to avoid unnecessary confusion.*

For this numerical test, all solvers use the same initialization in the first iteration and apply the same warm-start strategy to generate the initialization for the following iterations. In particular, the shifted solution from the last iteration is used to warm-start. The computational time is the sum of the CPU time returned by the solver. The results in this subsection are generated by a laptop with an Intel i7-11800H 16-core processor and 32 GB memory. Meanwhile, as Step 1) in the PPL Algorithm 6 is a convex QP, the solution time without using mpQPs is also investigated. In particular, the mpQP solution in this example has 729 critical regions, and the resulting binary search tree is of depth 13 (Section 5.4.1). The parallelization of the PPL algorithm is done by using `OpenMP`, while the parallelization in ALADIN relies on the parallel computing toolbox from `Matlab`. The statistics of the solution time are summarized in Table 5.1, where the maximal solution time indicates the solution time when good initialization is not available (i.e., cold-start). Additionally, the mean solution time reflects the averaged performance when a good initialization is available.

Above all, Table 5.1 shows that the ALADIN method is not desirable for fast MPC applications, the need to solve multiple nonconvex problems significantly slows down its speed (Section 5.3.5). We only report the parallelized solution time for ALADIN, and the non-parallelized solution time is at least three times slower. Regarding the PPL algorithm, the overhead caused by parallelization pays off only when mpQP solutions are not used. In this case, step 1) in the PPL algorithm requires solving a QP whose computational cost is significantly higher than calling the mpQP solution. Thus, it is easier to gain performance improvement by parallelization in this case. On the contrary, as calling mpQP solution is already computationally highly efficient, improving performance by parallelization may require more involved code design, such as caching. We believe that is the reason why the use of `OpenMP` does not accelerate the mpQP based implementation in this case. Note that this observation does not negate the benefit of parallelization. On the one hand, the efficiency of parallelization depends on the computing unit and the compiler. The use of `OpenMP` and a general purpose `Intel` CPU in this numerical example may not be the most efficient implementation. On the other hand, the construction of the mpQP solutions may not be computationally affordable for large-scale systems. If solving convex QPs online is needed, then performance improvement is easy to achieve by parallelization, which is also justified by this numerical test.

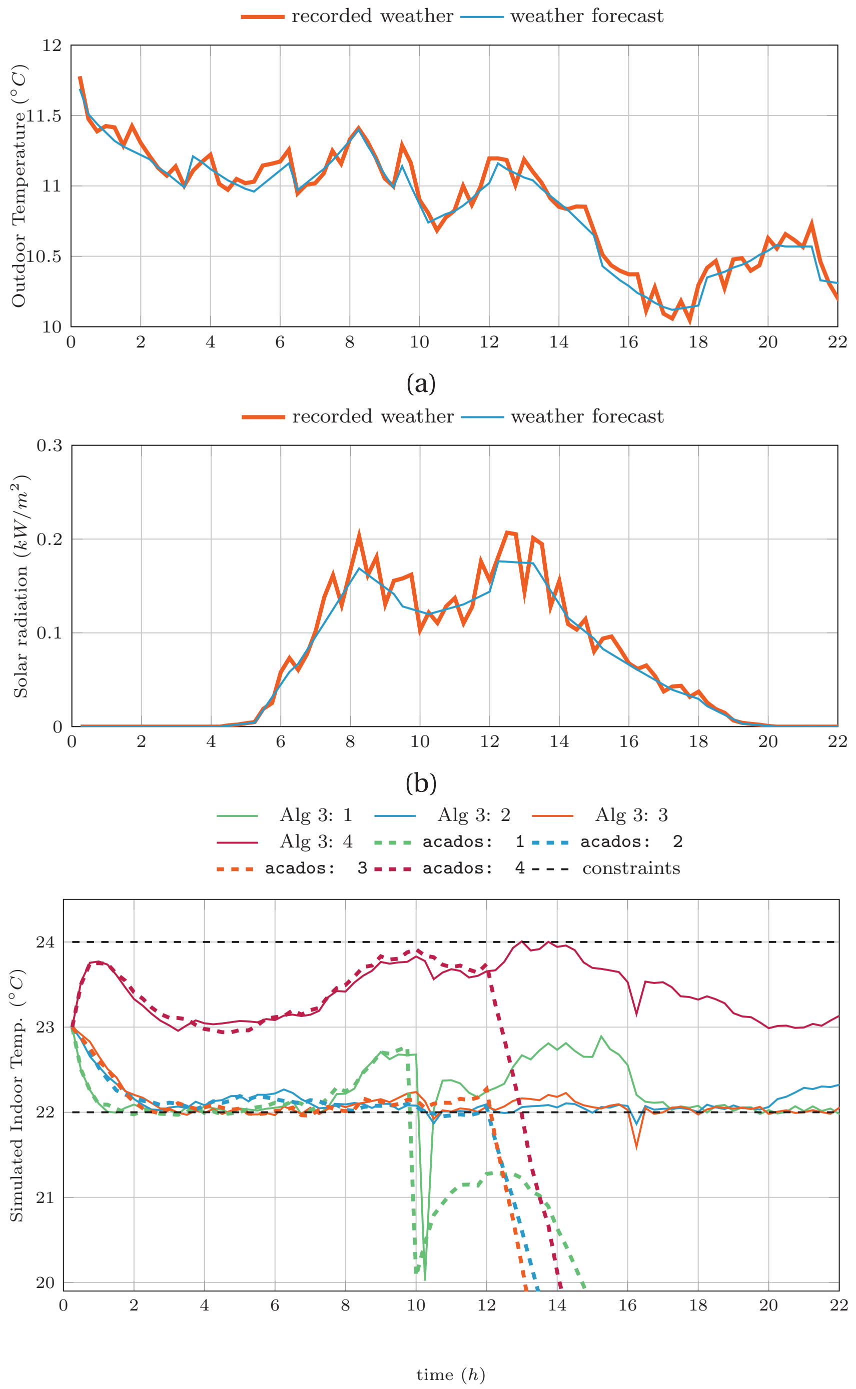
In comparison with `acodos`, when the proposed scheme uses mpQPs solution without parallelization, the maximal solution is on average 71% faster than those of `acados` without condensing. Regarding the mean solution time, even though the proposed scheme is faster than `acados` without condensing by 17%, but it is 48% slower than the mean solution of `acados` with condensing. Note that both the QP solver and the linear algebra routine in `acados` is highly optimized for NMPC; the results in Table 5.1 show that a tailored SQP solver is highly efficient when a good initialization is available. As Step 4) in the proposed scheme is similar to an SQP iteration, the PPL algorithm also shows comparable performance to a tailored SQP solver for warm-started iterations. This computational efficiency is aligned with our discussion given in Section 5.3. On top of this benefit, the proposed scheme shows much better performance when a good initialization is not available. This justifies the use of mpQP solutions, which improves the detection of the active sets. In summary, this numerical study proves the efficiency and efficacy of the PPL algorithm, and it also suggests a further possibility of performance improvement by a more sophisticated combination of the proposed scheme and a tailored SQP solver; we leave this for a future study.

Besides the observation given in Table 5.1, we also observe that the PPL algorithm is more robust to the choice of initialization strategy. More specifically, if the initialization is only partially warm-started by setting the predictive input sequence to **0** (i.e., cold-start inputs but warm start all the other variables), both `acados` and ALADIN will return NaN during the simulation for all the Monte-Carlo tests. On the contrary, the proposed scheme will always converge even when all the decision variables are initialized by **0**. This observation aligns with the motivation of the PPL algorithm and justifies the benefit of Step 1) in the PPL algorithm. Meanwhile, this robustness might be beneficial in some applications. For example, set-point change in tracking control makes initialization more challenging.

Last but not least, the property that the PPL algorithm is feasible even with an infeasible initial state is useful in practice, which is typically the case in building control. Due to the uncertain occupant behavior, such as opening the window, the indoor climate can be significantly perturbed, resulting in an infeasible initial state for the MPC problem. Consider a case where the occupant opens the window to bring in the fresh air when he first arrives in room 1 at 10:00 A.M, this move causes a sudden drop in indoor temperature as shown in Fig. 5.3 (c). Such sudden temperature drop causes infeasibility, which leads to the failures of the `acados` solver. However, the PPL algorithm can still give reasonable control inputs and quickly recovers the indoor temperature to a comfortable level.

5.5.2 Bilinear DC Motor Control with a C2000 Microcontroller

Next, the PPL algorithm is deployed on an embedded system, a Texas Instruments C2000 LaunchPad XL F28379D, to control the speed of a field-controlled DC motor. The dynamics of



(a)

(b)

(c)

Figure 5.3: Case study of sudden indoor temperature drop: (a) - (b): a sample of the recorded and forecast weather condition, (a) outdoor temperature, (b) solar radiation. The forecast is used as the nominal weather in the NMPC problem, while recorded weather is used for the simulation of building dynamics. (c): simulation of indoor temperature of different rooms (room index depicted in Figure 5.2).

Table 5.1: Statistics of the solution time at different tolerance (the entries of top two performance in each row are stressed by boldface black and boldface blue respectively)

method		Algorithm 6				`acados`		`ALADIN`
tol	sol time (ms)	parallel (mpQPs)				condensing		parallel
		yes (yes)	no (yes)	yes (no)	no (no)	yes	no	
10^{-4}	max	**8.404**	**6.304**	8.926	12.272	19.858	10.711	2553
	mean	0.5280	**0.4835**	0.6583	0.8751	**0.3671**	0.7931	897.4
10^{-5}	max	**8.838**	**6.629**	9.371	12.905	21.755	11.804	2859
	mean	0.6864	**0.6286**	0.8558	1.137	**0.4020**	0.8687	953.2
10^{-6}	max	**8.968**	**6.753**	9.593	13.510	22.532	12.251	3369
	mean	0.7814	**0.7156**	0.9743	1.295	**0.4163**	0.8995	1053

the field-controlled DC motor are bilinear,

$$
\begin{aligned}
\frac{dx_1}{dt} &= -\frac{R_a}{L_a}x_1 - \frac{K_m}{L_a}x_2 u + \frac{V_s}{L_a},\\
\frac{dx_2}{dt} &= -\frac{B}{J}x_2 + \frac{K_m}{J}x_1 u - \frac{T_e}{J},
\end{aligned}
$$

where states x_1, x_2 are respectively armature current and angular velocity, and the control input u is field current. V_s and T_e define the external torque respectively, which are chosen as 60 V and 0 Nm for this experiment. The remaining parameters are identified on a real field-controlled DC motor (Fig. 5.5b) as shown in Table 5.2.

Table 5.2: Parameters of the Field Controlled DC Motor

Parameter	Variable	Value
Armature Resistance	R_a	10 [ohm]
Armature Inductance	L_a	60 [mH]
Motor Constant	K_m	0.2297 [V (A rad/s)$^{-1}$]
Damping Ratio	B	0.0024 [Nm(rad/s)$^{-1}$]
Inertia	J	0.008949 [kg m^2]

We first provide some background on the motor behaviour to gain insight into the NMPC solutions. Typically, the armature current dynamics x_1 are much faster than the mechanical dynamics, so it is useful to consider the motor behaviour after the current dynamics have decayed. The torque-speed curves of the motor are shown in Fig. 5.4 (a) for various field currents. This is the electrical torque produced by the motor for the given speed and field current. If the mechanical torques (drag+external) match this torque at a given speed, it is an equilibrium point. For example, we may observe that the no-load speed for this motor (without drag) is 87 rad/s at the full 3 A field current. The typical operating region of this type of motor is at speeds higher than the full-field line, roughly $[80, 180]$rad/s for low torques. Operation below this speed is undesirable because the armature currents exceed the 3A continuous thermal limits regardless of the field current selected. This is shown in current-speed curve (Fig. 5.4 (b)), where armature current are plotted as a function of speed for different field currents (i.e control input). Hence, the curves below the red dashed line in Figure 5.4 (b) also show the set of desired operating points that allows long-term operation without overheating.

The continuous dynamics are discretized by the Euler method with a sampling time of 10ms. The prediction horizon is set to 3^2 with a convergence tolerance at 10^{-4}. The MPC controller conducts speed control, which tracks a reference speed ω_{ref}. This motor operates around [80,180] rad/s, and for most reference torque/speed combinations within this range, there are two possible field current solutions as shown by overlapping lines in Fig. 5.4 (a). The low field current (i.e., control input) solution results in a higher armature current, usually above the 3 A limit (Fig. 5.4 (b)). Long-term operation on this equilibrium point will result in armature overheating even though it tracks the reference speed. However, in order to have an agile motor response, the armature current should be able to operate above 3 A for short intervals. Therefore, we do not enforce a constraint on the armature current, while the field current (i.e., control input) is bounded within [1,3] A.

In this control setup, the desired operating point has an armature current lower than 3 A, which corresponds to the higher of two viable field currents (Fig 5.4). A proper choice of the loss function can help the solver to converge to the desired operating point. First of all, it is not desirable to use the speed regulating stage cost $\ell(x, u) = ([x_k]_2 - \omega_{\text{ref}})^2$, as the solver tends to select the lower field current command which will overheat the armature. This is particularly the case in the presence of noise based on our observation of different hardware in the loop simulations. We suspect that this can be explained by the torque-speed curve (Fig. 5.4 (a)). When operating with a low field current, the curves are relatively flat and a slight change in the field current can lead to a rapid change in speed. This implies that the solver can give better local convergence behavior in this region, so the solver tends to converge to this undesired operating point. To avoid this issue and push the solution to the preferred operating point, we offer a reference armature current and field current, whose steady state solution has an explicit form by substituting ω_{ref} into the system dynamics. The stage cost is designed to

$$\ell(x, u) = 20([x_k]_1 - I_{\text{ref}})^2 + ([x_k]_2 - \omega_{\text{ref}})^2 + 10(u - u_{\text{ref}})^2,$$

[2] The prediction horizon is set based on some recent results with **commercial** solver from ODYS [299, 300], where they use the same C2000-series hardware to deploy MPC on a synchronous machine. In their setup, the input constraints are neglected, the prediction horizon is two, and the linearized model is used instead of the nonlinear model.

where I_{ref} and u_{ref} are reference armature current and reference field current.

The experimental setup is shown in Fig. 5.5 with the explanation given in its caption. Two experiments are carried out on velocity tracking control, which both track a triangular reference speed that varies between 100 rad/s and 140 rad/s. In the first case, we only have a speed constraint within $[80, 180]$ rad/s, while this constraint is tightened to $[110, 180]$ rad/s in the second case. Thus, the speed lower bound is inactive in the first case, but active and satisfied in the second case.

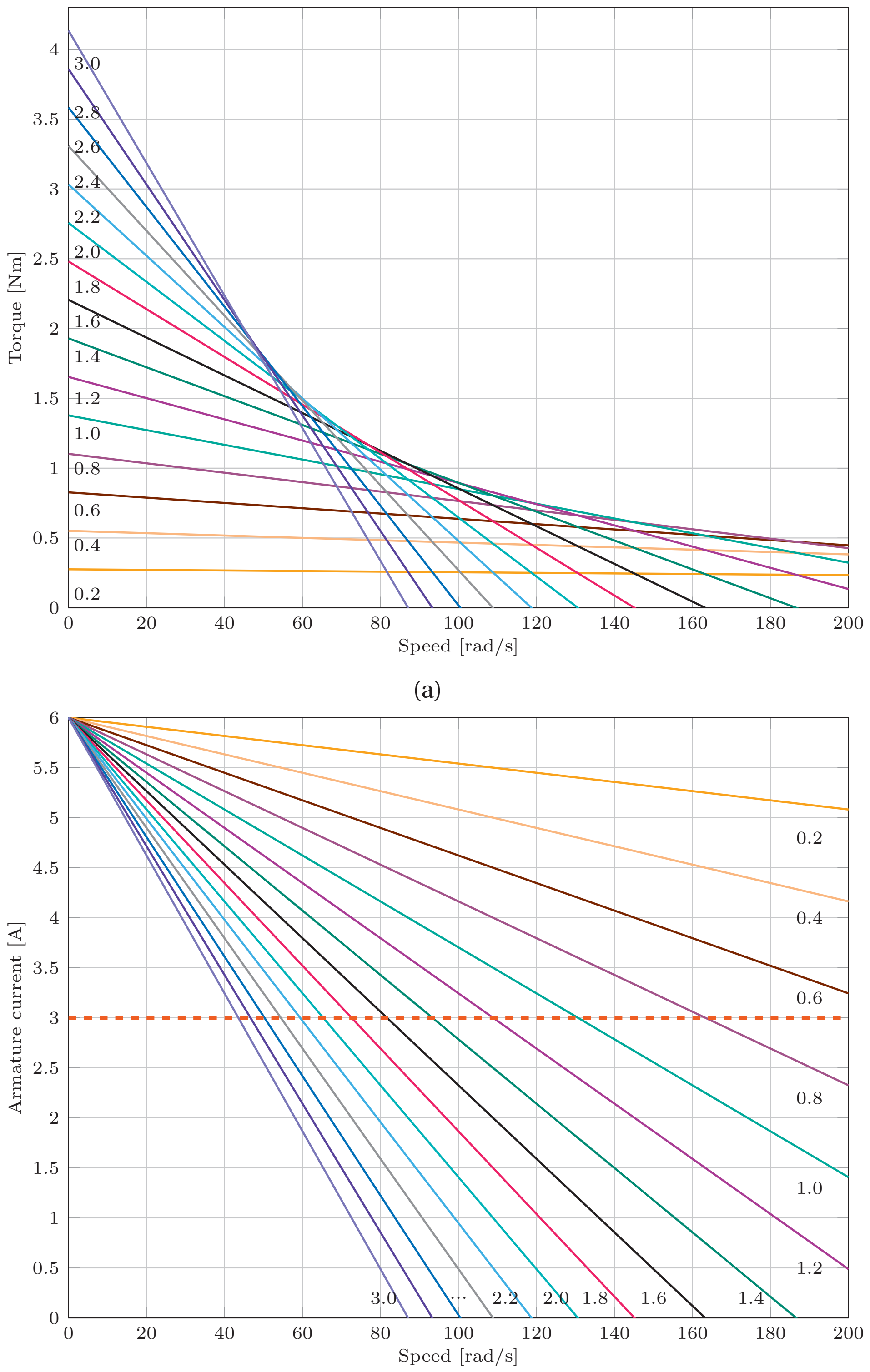
To verify the real-time controller performance, we first simulate these experiments using control hardware in the loop methods. Both the controller and a simulated motor run on the same C2000 microcontroller (Fig. 5.5c). The results are shown in Fig. 5.6, with the armature current disturbed by white noise to reproduce the switching noise encountered in real-world experiments. In simulations, the NMPC properly executes the speed control, and the speed constraint is satisfied in the second case as expected.

The hardware in-the-loop result above can already justify the efficiency of the proposed solver, a similar setup for performance proof is also used in the **commerical** product in [299, 300]. But for the sake of completeness, we carry out the experiment on a real motor. The hardware experimental results are shown in Fig. 5.7. The measured signals are post-processed with a low-pass zero-phase filter. In this experiment, the PPL algorithm successfully executes the control in real time with a 10 ms MPC update rate. In particular, the maximum and average execution time of the PPL algorithm in this embedded system is 2.088 ms and 1.764 ms respectively. Thus, the solver can run up to 500 Hz, which is sufficiently fast regarding the 10 ms sampling time of the targeted system.

However, in our real-world experiments, the tracking performance is somewhat lacking. From Fig. 5.7, we can observe that the NMPC tries to track the signal and periodic triangular speed trajectories are recorded with noticeable tracking errors. It is noteworthy that the NMPC satisfies the lower speed limit in the second test, which justifies the constraints enforced by the NMPC. The reasons will be investigated in future work but are possibly due to poor estimates of the unmeasured drag torque or other parameter errors, or inaccurate delivery of the current command. In summary, based on the hardware in the loop experiments and the real-world experiments, the efficiency and real-time capability of the PPL algorithm has been proven. Even though it is not the main focus of this chapter, we believe that there are still a few imrovments to be carried out on our experiment platform to fully expolit the capability of the NMPC control, and we leave this for future investigation.

5.6 final Remark

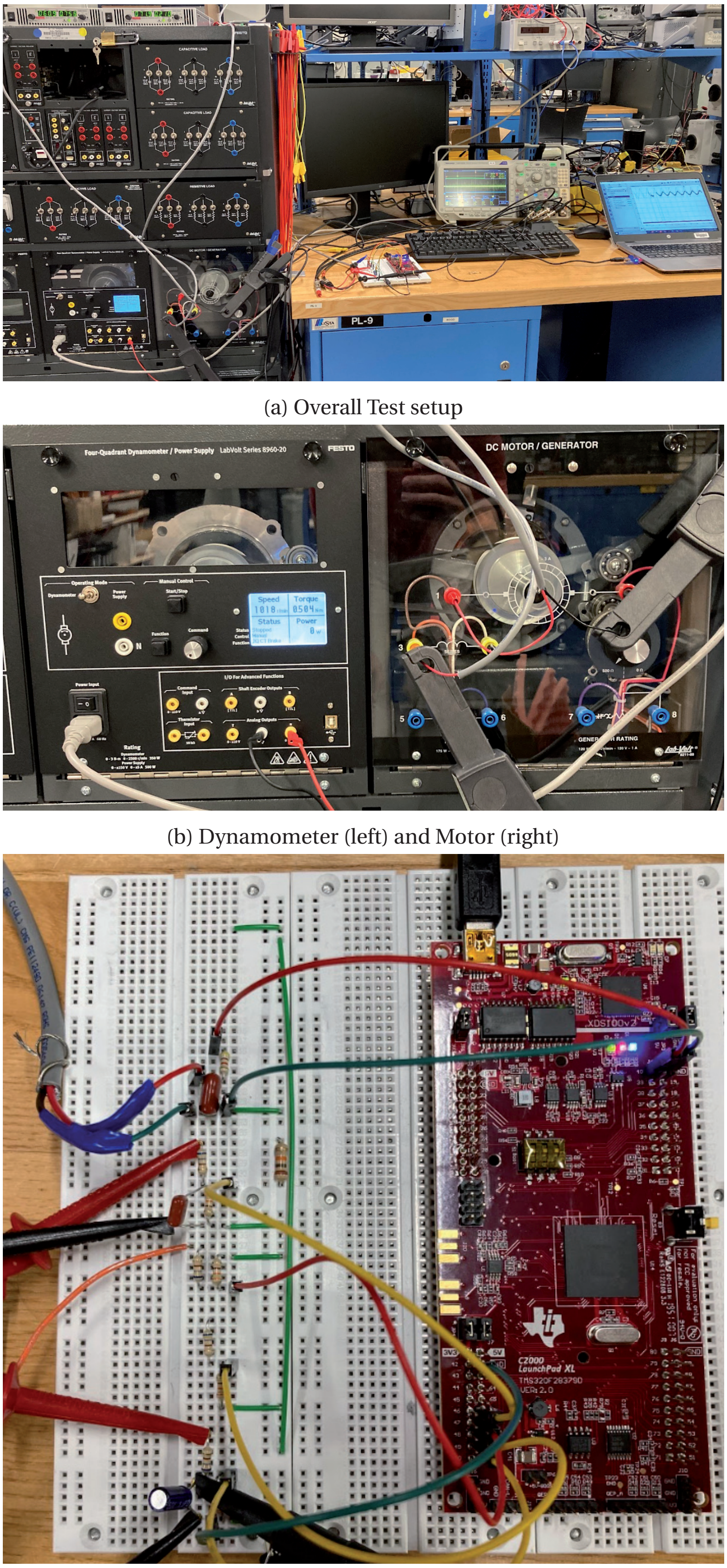
This chapter investigates a novel proximal-point Lagrangian based nonconvex solver for bilinear model predictive control. The PPL algorithm combines the ideas of explicit MPC, horizon splitting, and real-time SQP algorithms, and a novel horizon splitting scheme is



(a)

(b)

Figure 5.4: Toruqe current curves of the DC motor. Top: Torque-Speed curve at different field currents (indicated in Amps on the left end of each straight line). Bottom: Armature current as a function of speed for different field currents (indicated in Amps on the right end of each straight line). To avoid armature overheating, the armature current should stay below the 3 A thick red dashed line in long term operation.



(a) Overall Test setup

(b) Dynamometer (left) and Motor (right)

(c) TI LaunchPad XL and analog I/O filtering

Figure 5.5: Hardware testing of the proposed MPC algorithm with a real motor controlled by a

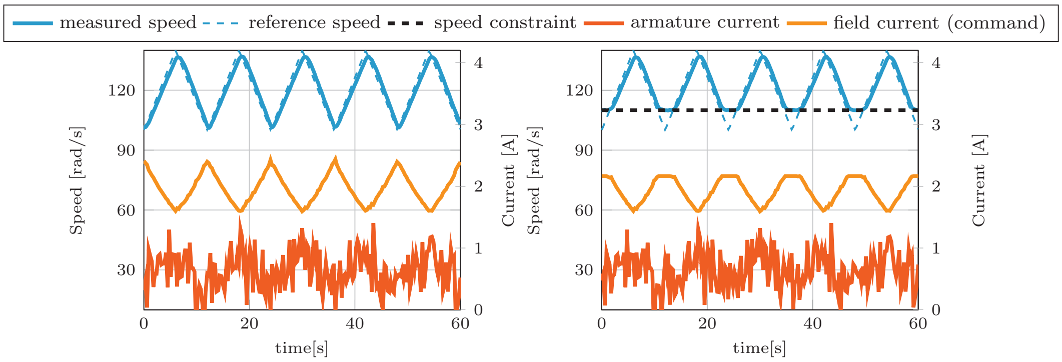


Figure 5.6: Hardware in the loop simulation of the motor control with the C2000 microcontroller. Left: speed constraint not active. Right: speed constraint $[110, 180]$ rad/s active (black dashed line). Raw signals are plotted.

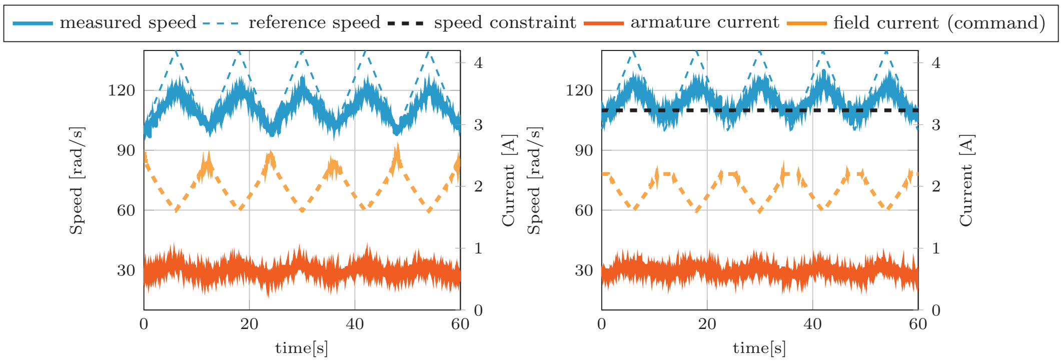


Figure 5.7: Hardware testing of the proposed MPC algorithm with a real motor controlled by a C2000 microcontroller. Left: speed constraint not active. Right: speed constraint $[110, 180]$ rad/s active (black dashed line). The measured signals are post-processed with a low-pass zero-phase forward-reverse filter.

proposed to enable this integration. The numerical efficiency of the PPL algorithm is validated by a building control simulation and an experiment on a real field-controlled DC motor with TI C2000 LaunchPad XL F28379D microcontroller. Particularly, the latter experiment proves the real-time capability of the PPL algorithm, which successfully solves the NMPC problem in 1.764 ms on average.

Concluding Remarks Part IV

6 Conclusion and Outlook

This thesis presents a systematic approach for integrating data into the pipeline of controller design and stability analysis. Except for the result presented in Section 2.1, WE demonstrate that data can be directly used to characterize system behaviour, while explicit identification of the underlying model is not necessary

The first part of this thesis focuses on controller design, with separate chapters dedicated to linear and nonlinear systems. The major contribution in the linear controller design is a systematic bi-level approach to handle process noise with noisy I/O measurements. This approach can incorporate heuristic techniques to adapt the predictor to slowly time-varying linear dynamics. As reported in this chapter, this framework has been tested on real-world building and provides long-term reliable service for three weeks. In our follow-up experiment, which is not included in this thesis, this bi-level framework can be hierarchically integrated into the demand response service. We have run another five-week experiment to showcase the capability of the proposed framework. While some technical details are currently tailored for building control, its general framework and most numerical techniques are applicable to general LTI systems. The next step for this framework is definitely to deploy the proposed framework in faster mechatronic systems. Autonomous driving is one promising application, where we already have some initial results in collaboration with Jiawei Wang. Another aspect that would be interesting is the update behaviour of the underlying optimization problem. In particular, once the I/O dataset is updated online iteratively, the data linearly enters the equality constraints on the left-hand side, this is no longer a convex parametric QP, but it can definitely quantify sensitivity of the proposed framework particularly with respect to the measurement noise.

In the design of nonlinear controllers, the primary contribution is to demonstrate the effective use of data to characterize system responses through lifting. However, the author believes that the key message of this chapter is not focused on numerical techniques or mathematical models behind the numerical algorithms. Rather, two function space viewpoints are essential takeaways. To develop these viewpoints, we must consider the dynamical behaviours as the evolution of observables. This perspective may seem unusual to most control engineers, but it is intuitive to non-control engineers. The evolution of a system is perceived through the variation of its measurements. When viewed in this way, we arrive at two different function space viewpoints: for a function, we can model either the function itself or our interactions with it. The former begins with parametric forms or hypotheses about the function space, while the latter starts with the set of functionals we will use. These two aspects together indicate that modelling the users, including their interactions or measurement behaviours, is sufficient for them to manipulate the dynamical systems. While we may need additional information to fully capture the underlying dynamics, it is mostly redundant for controller design, at least from the users' viewpoint. Extending this viewpoint is theoretically intriguing, but the crucial question is why we should use this viewpoint to understand a dynamical system. Furthermore, this perspective results in new hypothetical model structures. Thus, the long-standing question remains: which models/methods are best suited to specific classes of dynamics? The answer to this question should consider practicality and numerical performance as well. Therefore, before delving too deep, it is reasonable to step back and reconsider whether and when a direct data-driven approach is desirable for nonlinear control.

In the second part of this thesis, we focus on stability analysis that relies solely on data, which represents a significant contribution to the field of control theory. Specifically, we present two major contributions in this area. First, we extend the Lyapunov stability theory from analysing a single equilibrium point to a forward invariant set, which is a non-trivial extension that has important implications for the practical implementation of control systems. Secondly, we demonstrate that the use of a complex and informative model can be effectively relaxed and simplified by synthetically assembling multiple simple submodels that are part of the larger complex model. There are several avenues for future work in this area. First, we need to explore the applicability of this method to more complex function spaces, while maintaining its computational merit. This will involve developing new techniques to deal with the increased complexity of the models and data. Secondly, while our current result is based on Fenchel duality, it will be important to explore other duality structures to determine their effectiveness for this type of stability analysis. Overall, this work represents an important step forward in the development of data-driven techniques for control system analysis and design.

In the preceding paragraphs, a summary of the main results and their implications for future research has been provided. However, this brief summary fails to convey the full scope of the challenges and opportunities facing the field in the coming decades. As a young researcher in this field, the author would like to offer a more comprehensive perspective on the matter.

First and foremost, one of the primary concerns that needs to be systematically addressed is robustness to uncertainty. This includes both aleatoric and epistemic uncertainty, and the thesis has explored various approaches to addressing these issues using different mathematical tools. However, underpinning all of these mathematical tools is the topology of the function space being studied. For example, the Wasserstein distance considers the topology of the probability distribution, while the Reproducing Kernel Hilbert Space (RKHS) investigates the topology induced by the bounded evaluation functional. The relevance of study topology is proved in its numerical meaning by providing different bound used in the algorithm. In summary, a significant challenge for future research is to develop one or more systematic approaches to handle control problems while considering the corresponding topological structure. Based on that, we are able to endow the underlying topological space with a specific geometric structure. For example, when we use the RKHS, we have the inner product space and the corresponding geometric structure in the function space. This offers us a friendly interface to interact with those infinite dimensional objects. However, it may also prohibit us from considering the geometric information contained in the state space, which presents strong nonlinearity/nonconvexity due to the nonlinear evaluation map between the x in state space and the E_x in RKHS.

Building on top of topology, the geometric aspect gives a promising and highly unexplored direction. Actually, I once planned to write the thesis under the title of “geometric aspects of learning-based control". Even though I choose not to use this title in the end, the concept behind this title remains a central concept that can lead to significant advances in the field. Mathematical tools from symplectic geometry, variational analysis and nonlinear operator theory will be the key enabler for this direction. This is already hinted by the Chapter 4 and the bilinear MPC solver result in Chapter 6. In particular, the elliptic and symplectic structure that presents in most locally monotone object will be a fascinating direction to explore. The author has already obtained some initial results in this direction, and believes that further exploration in this area will be essential for future progress in learning-based control.

In conclusion, while data-driven approaches offer a holistic view of control science, they also present significant challenges. Classic control methods based on divide-and-conquer principles have helped simplify and sort out the structure of control science over decades. The direct use of data pushes us to deal with all the problems related to the controller design, especially the nonlinearity, simultaneously. This requires every control researcher to rethink the mathematical structure we have been used to for decades. Without question, the existing mathematical framework used in control science may not be suitable for this new control challenge without modifications, and a bottom-up refresh of control science may be needed to develop new tools that are better suited for data-driven control. This thesis is my initial attempt and I believe that many results, including the discussion in the preceding paragraphs, should suggest promising directions for future research.

Appendix Part V

In this part, we will develop the close-loop prediction that acts on the solution to the equation system. Meanwhile, all the assumption made in Chapter 1 remain valid. We specify the robust bi-level optimization problem to be solved as follows

$$
\begin{aligned}
&\min_{\substack{\overline{g},K,\\ \overline{u}_{pred},\overline{y}_{pred}}} \max_{w_{pred}\in\mathcal{W}} J(y_{pred},u_{pred}) \\
&\qquad \text{s.t. } K\in\mathcal{K}_c && \text{(6.1a)}\\
&\qquad\quad \forall\ w_{pred}\in\mathcal{W} \\
&\qquad\quad \overline{u}_{pred}+u_{fb}(w_{pred})\in\mathcal{U} && \text{(6.1b)}\\
&\qquad\quad \overline{y}_{pred}+y_{fb}(w_{pred})\in\mathcal{Y} && \text{(6.1c)}\\
&\overline{g}+Kw_{pred}\in\operatorname*{arg\,min}_{g_l,\sigma_l}\frac{1}{2}\|\sigma_l\|^2+\frac{1}{2}\eta_g\|g_l\|^2 && \text{(6.1d)}\\
&\qquad\quad \text{s.t } Hg_l=\begin{bmatrix} y_{init}+\sigma_l\\ u_{init}\\ w_{init}\\ \overline{u}_{pred}+u_{fb}(w_{pred})\\ w_{pred}\end{bmatrix},
\end{aligned}
$$

where $\mathcal{K}_c$ is the set of causal feedback control laws. By Assumption 2, we need to model a set-valued solution map of the lower level problem, which maps $\mathcal{W}$ to $\overline{g}+K\mathcal{W}$. To better elaborate this challenge, we write down the KKT system of the lower level problem (6.1d),

$$
\forall\ w_{pred}\in\mathcal{W}\begin{cases}
\overbrace{H(\overline{g}+Kw_{pred})=\begin{bmatrix} y_{init}+\sigma\\ u_{init}\\ w_{init}\\ u_{pred}\\ w_{pred}\end{bmatrix}}^{(a)}\\
\begin{bmatrix} I_{(t_{init}n_y)} & \mathbf{O}\end{bmatrix}\kappa(w_{pred})=\sigma(w_{pred})\\
H^{\top}\kappa(w_{pred})=-\eta_g(\overline{g}+Kw_{pred}),
\end{cases} \qquad \text{(6.2)}
$$

where $\kappa(w_{pred})$ and $\sigma(w_{pred})$ indicates that σ and κ depend on the value of w_{pred} and this dependence is not necessarily linear [301]. To intuitively explain the reason why this is nonlinear, one can consider that w_{pred} not only perturbs the right-hand-side of the term (a) in (6.2), it also perturbs the matrix H by HKw_{pred}. A further complexity is that K is confined to be causal.

We conclude the main result in the following theorem

Theorem 11. *The following single level robust optimization problem is equivalent to the bi-level* *robust optimization problem* (6.1).

$$\min_{\substack{\overline{g},\overline{\sigma},\overline{\kappa}\\ K_p,\overline{u}_{pred},\overline{y}_{pred}}} \max_{w_{pred}\in\mathscr{W}} J(y_{pred},u_{pred}) \tag{6.3a}$$

$$s.t.\ K = \begin{bmatrix} Q_{a,:,n_{init}+1:n_r} & Q_b \end{bmatrix} K_p \tag{6.3b}$$

$$\begin{bmatrix} H \\ \mathfrak{H}_{L,pred}(y_{\boldsymbol{d}}) \end{bmatrix} \overline{g} = \begin{bmatrix} y_{init}+\overline{\sigma} \\ u_{init} \\ w_{init} \\ \overline{u}_{pred} \\ \mathbf{0} \\ \overline{y}_{pred} \end{bmatrix} \tag{6.3c}$$

$$\mathfrak{H}_{L,pred}(w_{\boldsymbol{d}})K = I_{(n_h n_w)} \tag{6.3d}$$

$$\mathscr{T}\overline{\kappa} := \begin{bmatrix} -H^\top \\ [I_{(t_{init}n_y)}\ \boldsymbol{O}] \end{bmatrix} \overline{\kappa} = \begin{bmatrix} \eta_g \overline{g} \\ \overline{\sigma} \end{bmatrix} \tag{6.3e}$$

$$(I-P)\begin{bmatrix} \eta_g K \\ \mathfrak{H}_{L,init}(y_{\boldsymbol{d}})K \end{bmatrix} = \boldsymbol{O} \tag{6.3f}$$

$$\begin{aligned} &\forall\ w_{pred}\in\mathscr{W}\ ,\ \begin{bmatrix} \mathfrak{H}_{L,pred}(u_{\boldsymbol{d}}) \\ \mathfrak{H}_{L,pred}(y_{\boldsymbol{d}}) \end{bmatrix} K w_{pred} = \begin{bmatrix} u_{fb} \\ y_{fb} \end{bmatrix} \\ &u_{fb}+\overline{u}_{pred}\in\mathscr{U}\ ,\ y_{fb}+\overline{y}_{pred}\in\mathscr{Y}\ , \end{aligned} \tag{6.3g}$$

where $n_{init} := t_{init}\times(n_u+n_w)+n_u$, Q_a *and* Q_b *are defined by the following QR decomposition of* $H_{fb}\in\mathbb{R}^{n_r\times n_c}$

$$H_{fb}^\top := \begin{bmatrix} \mathfrak{H}_{L,init}(u_{\boldsymbol{d}}) \\ \mathfrak{H}_{L,init}(w_{\boldsymbol{d}}) \\ \mathfrak{H}_{L,pred}(u_{\boldsymbol{d}}) \end{bmatrix}^\top = \begin{bmatrix} Q_a & Q_b \end{bmatrix} \begin{bmatrix} R \\ \boldsymbol{O} \end{bmatrix} .$$

Meanwhile, K_p *is a blockwise lower triangular matrix as*

$$K_p = \begin{bmatrix} K_{p,1,1} & \boldsymbol{O} & \boldsymbol{O} & \dots & \boldsymbol{O} \\ K_{p,2,1} & K_{p,2,2} & \boldsymbol{O} & \dots & \boldsymbol{O} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ K_{p,n_h,1} & K_{p,n_h,2} & K_{p,n_h,3} & \dots & K_{p,n_h,n_h} \end{bmatrix} .$$

In particular, $K_{p,i,j}$ *are dense matrix blocks, whose sizes are* $n_u\times n_w$ *for* $\forall\ i<n_h$ *and are* $(n_c-n_r)\times n_w$ *when* $i=n_h$*. Finally, we enforce*

$$P = \mathscr{T}(\mathscr{T}^\top\mathscr{T})^{-1}\mathscr{T}^\top ,$$

with $\mathscr{T}$ *defined in* (6.3e).

Proof. This proof consists of two parts

- Reformulation of the KKT system such that $g = \overline{g} + K\mathscr{W}$ models the set of optimal solutions to the lower level problem.

- Formulation of the feedback control law K to respect causality.

Reformulating KKT Systems:
The key idea is to decompose the KKT system (6.2) of the lower level problem (6.1d) into two sub systems, one corresponds to the nominal decision variables and the other corresponds to the perturbation caused by the feedback law. Based on this idea, we will reformulate the terms in (6.2) accordingly. The first component we consider is the primal feasibility condition

$$\forall\ w_{pred} \in \mathscr{W}, \ \overbrace{\begin{bmatrix} \mathfrak{H}_{L,init}(y_{\mathbf{d}}) \\ \mathfrak{H}_{L,init}(u_{\mathbf{d}}) \\ \mathfrak{H}_{L,init}(w_{\mathbf{d}}) \\ \mathfrak{H}_{L,pred}(u_{\mathbf{d}}) \\ \mathfrak{H}_{L,pred}(w_{\mathbf{d}}) \end{bmatrix}}^{H} (\overline{g} + Kw_{pred}) = \begin{bmatrix} y_{init} + \sigma \\ u_{init} \\ w_{init} \\ u_{pred} \\ w_{pred} \end{bmatrix} \tag{6.4}$$

Regarding the $y_{init} + \sigma$ term, we enforce the following decomposition

$$\sigma = \underbrace{(\mathfrak{H}_{L,init}(y_{\mathbf{d}})\overline{g} - y_{init})}_{\overline{\sigma}} + \underbrace{\mathfrak{H}_{L,init}(y_{\mathbf{d}})Kw_{pred}}_{\sigma_{fb}(w_{pred})} \tag{6.5}$$

Regarding the u_{init} and w_{init} terms in (6.4), the causality constraint on K is relevant. In particular, the feedback caused by w_{pred} cannot change u_{init} and w_{init}, therefore, we enforce

$$\begin{bmatrix} \mathfrak{H}_{L,init}(u_{\mathbf{d}}) \\ \mathfrak{H}_{L,init}(w_{\mathbf{d}}) \end{bmatrix} \overline{g} = \begin{bmatrix} u_{init} \\ w_{init} \end{bmatrix} \Longrightarrow \underbrace{\begin{bmatrix} \mathfrak{H}_{L,init}(u_{\mathbf{d}}) \\ \mathfrak{H}_{L,init}(w_{\mathbf{d}}) \end{bmatrix} K = \mathbf{O}}_{(a)} . \tag{6.6}$$

Because u_{pred} is also a decision variable, we skip this term by simply enforcing

$$\mathfrak{H}_{L,pred}(u_{\mathbf{d}})(\overline{g} + Kw_{pred}) = \overline{u}_{pred} + u_{fb}(w_{pred}) .$$

The last ingredient in (6.4) is w_{pred}, where we enforce

$$\begin{aligned} \mathfrak{H}_{L,pred}(w_{\mathbf{d}})g = &\mathfrak{H}_{L,pred}(\overline{g} + Kw_{pred}) \\ &\overset{(b)}{=} \mathfrak{H}_{L,pred}(w_{\mathbf{d}})Kw_{pred} = w_{pred} \ , \forall\ w_{pred} \in \mathscr{W} \\ &\Longrightarrow \mathfrak{H}_{L,pred}(w_{\mathbf{d}})K = I_{(n_h n_w)} \ , \end{aligned}$$

where (b) holds by enforcing

$$\mathfrak{H}_{L,pred}(w_{\mathbf{d}})\overline{g} = \mathbf{0}\,. \tag{6.7}$$

Thus, we can now conclude the constraints (6.3c) and (6.3d).

We recall the stationary condition for the KKT system (6.2),

$$\begin{aligned}
\forall\ w_{pred} &\in \mathscr{W}\,,\\
H^{\top}\kappa(w_{pred}) &= -\eta_g(\overline{g} + Kw_{pred})\,,\\
\begin{bmatrix} I_{(t_{init}n_y)} & \mathbf{O}\end{bmatrix}\kappa(w_{pred}) &= \overline{\sigma} + \sigma_{fb}(w_{pred})\,.
\end{aligned}$$

Similarly to the decomposition applied to σ in (6.5), the dual variables κ are decomposed as

$$\kappa = \overline{\kappa} + \kappa_{fb}(w_{pred})\,.$$

Hence, we rewrite the stationary conditions as

$$\begin{bmatrix} -H^{\top} \\ [I_{(t_{init}n_y)}\ \mathbf{O}]\end{bmatrix}(\overline{\kappa} + \kappa_{fb}(w_{pred})) = \begin{bmatrix} \eta_g(\overline{g} + Kw_{pred}) \\ \overline{\sigma} + \mathfrak{H}_{L,init}(y_{\mathbf{d}})Kw_{pred}\end{bmatrix}\,,$$

which is decomposed into two systems

$$\begin{bmatrix} -H^{\top} \\ [I_{(t_{init}n_y)}\ \mathbf{O}]\end{bmatrix}\overline{\kappa} = \begin{bmatrix} \eta_g\overline{g} \\ \overline{\sigma}\end{bmatrix}\,, \tag{6.8a}$$

$$\begin{bmatrix} -H^{\top} \\ [I_{(t_{init}n_y)}\ \mathbf{O}]\end{bmatrix}\kappa_{fb} = \begin{bmatrix} \eta_g K \\ \mathfrak{H}_{L,init}(y_{\mathbf{d}})K\end{bmatrix} w_{pred}\,, \tag{6.8b}$$

where (6.8a) corresponds to the constraint (6.3e). For sake of compactness, we denote

$$\mathscr{T} := \begin{bmatrix} -H^{\top} \\ [I_{(t_{init}n_y)}\ \mathbf{O}]\end{bmatrix}\,.$$

As the dual variables did not enter the objective function, we only need to ensure that the linear equations (6.8b) is solvable for arbitrary $w_{pred} \in \mathscr{W}$, which means that

$$\text{colspan}(\begin{bmatrix} \eta_g K \\ \mathfrak{H}_{L,init}(y_{\mathbf{d}})K\end{bmatrix}) \subset \text{colspan}(\mathscr{T})$$

Note that the projection operator [302, Chapter 1.12] onto colspan($\mathscr{T}$) is defined by

$$P = \mathscr{T}(\mathscr{T}^{\top}\mathscr{T})^{-1}\mathscr{T}^{\top}\,.$$

Then the residual of the projection is always null for arbitrary $w_{pred} \in \mathscr{W}$, which is

$$\begin{aligned}(I-P)\begin{bmatrix}\eta_g K\\ \mathfrak{H}_{L,init}(y_\mathbf{d})K\end{bmatrix} w_{pred} = \mathbf{0}\,,\ \forall\ w_{pred} \in \mathscr{W}\\ \Longrightarrow (I-P)\begin{bmatrix}\eta_g K\\ \mathfrak{H}_{L,init}(y_\mathbf{d})K\end{bmatrix} = \mathbf{O}\,,\end{aligned}$$

corresponding to constraint (6.3f).

In conclusion, constraints (6.3c), (6.3d), (6.3e) and (6.3f) models the set-valued solution map of the lower level problem (6.1d).

Formulation of Causal Feedback:
In this part, we first show the structure that K must have to ensure causal feedback and then use the QR decomposition to enforce causality.

In terms of causality, it includes two parts. First, the feedback law cannot changes what has already happened, which has been enforced by the term (a) in constraint (6.6). Therefore, each column block of matrix K should satisfy the following subspace condition,

$$K_{:,j} \subset \mathrm{null}\left(\begin{bmatrix}\mathfrak{H}_{L,init}(u_\mathbf{d})\\ \mathfrak{H}_{L,init}(w_\mathbf{d})\end{bmatrix}\right)\ \forall\ j = 1,\ldots,n_h\,.$$

Second, the feedback from $w_{pred,j}$ should not be able to change the inputs that occur no later than $w_{pred,j}$, which includes $u_{pred,1:j}$. As the j-th column block in matrix K, $K_{:,j}$, defines the feedback ingredient with respect to $w_{pred,j}$, we therefore further have the following subspace condition

$$K_{:,j} \subset \mathrm{null}\left(\mathfrak{H}_{L,pred,1:j}(u_\mathbf{d})\right)\,.$$

In conclusion, we have

$$K_{:,j} \subset \mathrm{null}\left(\begin{bmatrix}\mathfrak{H}_{L,init}(u_\mathbf{d})\\ \mathfrak{H}_{L,init}(w_\mathbf{d})\\ \mathfrak{H}_{L,pred,1:j}(u_\mathbf{d})\end{bmatrix}\right)\ \forall\ j = 1,\ldots,n_h\,, \tag{6.9}$$

where $\mathfrak{H}_{L,pred,1:j}(u_\mathbf{d})$ denotes the first j row blocks in $\mathfrak{H}_{L,pred}(u_\mathbf{d})$.

To implement this null space condition (6.9) in a numerically efficient manner, we define $H_{fb} \in \mathbb{R}^{n_r \times n_c}$ and the QR decomposition of its transpose as follows

$$H_{fb}^\top := \begin{bmatrix}\mathfrak{H}_{L,init}(u_\mathbf{d})\\ \mathfrak{H}_{L,init}(w_\mathbf{d})\\ \mathfrak{H}_{L,pred}(u_\mathbf{d})\end{bmatrix}^\top = \begin{bmatrix}Q_a & Q_b\end{bmatrix}\begin{bmatrix}R\\ \mathbf{O}\end{bmatrix}\,.$$

Recall a useful property of QR decomposition [69]: the range of the first n rows of $H_{fb} \in \mathbb{R}^{n_r \times n_c}$

is spanned by the first n columns of Q_a. Because $[Q_a, Q_b]$ is an unitary matrix, the remaining columns in Q_a and the matrix Q_b spans the null space of the first n rows in matrix H_{fb}. The null space in (6.9) is represented by

$$\begin{aligned} &\text{null}\left(\begin{bmatrix} \mathfrak{H}_{L,init}(u_{\mathbf{d}}) \\ \mathfrak{H}_{L,init}(w_{\mathbf{d}}) \\ \mathfrak{H}_{L,pred,1:j}(u_{\mathbf{d}}) \end{bmatrix}\right) = \text{colspan}([Q_{a,:,n_{init}+(j-1)n_u+1:n_r}, Q_b]) \\ &\Longrightarrow K_{:,j} \subset \text{colspan}([Q_{a,:,n_{init}+(j-1)n_u+1:n_r}, Q_b]) \,, \end{aligned}$$

where $n_{init} := t_{init} \times (n_u + n_w) + n_u$ and . All the constraints on the null spaces can be summarized as

$$K = \begin{bmatrix} Q_{a,:,n_{init}+1:n_r} & Q_b \end{bmatrix} K_p \,,$$

where K_p is a blocked lower triangular matrix as

$$K_p = \begin{bmatrix} K_{p,1,1} & \mathbf{O} & \mathbf{O} & \dots & \mathbf{O} \\ K_{p,2,1} & K_{p,2,2} & \mathbf{O} & \dots & \mathbf{O} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ K_{p,n_h,1} & K_{p,n_h,2} & K_{p,n_h,3} & \dots & K_{p,n_h,n_h} \end{bmatrix} .$$

In particular, $K_{p,i,j}$ are dense matrix blocks, whose sizes are $n_u \times n_w$ for $\forall\ i < n_h$ and are $(n_c - n_r) \times n_w$ when $i = n_h$, where n_r is the number of rows in H_{fb}. In conclusion, this parametrization enforces the causality constraint (6.3b).

Note that, in theory, causality should be considered in $y_{pred} = \mathfrak{H}_{L,pred}(y_{\mathbf{d}})g$ as well, but actually $\mathfrak{H}_{L,pred}(y_{\mathbf{d}})$ should not be considered in the QR decomposition. The reason is twofold. Firstly, by the fundamental lemma 1, the trajectory prediction y_{pred} should be uniquely determined once y_{init}, u_{init}, w_{init}, u_{pred} and w_{pred} are given. Hence, consideration of $\mathfrak{H}_{L,pred}(y_{\mathbf{d}})$ is redundant. Secondly, our prediction y_{pred} is calculated from a noisy Hankel matrix as $y_{pred} = \mathfrak{H}_{L,pred}(y_{\mathbf{d}})g$, which breaks the null-space formed by the fundamental lemma **??**. Thus, $\mathfrak{H}_{L,pred}(y_{\mathbf{d}})$ is not considered in the matrix H_{fb}

Thus, we summarize the equivalence between problem (6.1) and problem (6.3b). ■

Bibliography

[1] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction.* MIT press, 2018.

[2] D. Jones et al. “Characterising the Digital Twin: A systematic literature review”. In: *CIRP Journal of Manufacturing Science and Technology* 29 (2020), pp. 36–52.

[3] L. Ljung. “System identification”. In: *Wiley encyclopedia of electrical and electronics engineering* (1999), pp. 1–19.

[4] N. Lanzetti et al. “Recurrent neural network based MPC for process industries”. In: *2019 18th European Control Conference (ECC)*. IEEE. 2019, pp. 1005–1010.

[5] A. Chiuso and G. Pillonetto. “System identification: A machine learning perspective”. In: *Annual Review of Control, Robotics, and Autonomous Systems* 2 (2019), pp. 281–304.

[6] S. Limanond and J. Si. “Neural network-based control design: an LMI approach”. In: *IEEE Transactions on Neural Networks* 9.6 (1998), pp. 1422–1429.

[7] J. Kocijan. *Modelling and control of dynamic systems using Gaussian process models.* Springer, 2016.

[8] Z. Hou. “The parameter identification, adaptive control and model free learning adaptive control for nonlinear systems”. In: *Shenyang: North-eastern University* (1994).

[9] G. O. Guardabassi and S. M. Savaresi. “Virtual reference direct design method: an off-line approach to data-based control system design”. In: *IEEE Transactions on Automatic Control* 45.5 (2000), pp. 954–959.

[10] S. Levine. “Reinforcement learning and control as probabilistic inference: Tutorial and review”. In: *arXiv preprint arXiv:1805.00909* (2018).

[11] J. C. Willems and J. W. Polderman. *Introduction to mathematical systems theory: a behavioral approach.* Vol. 26. Springer Science & Business Media, 1997.

[12] J. C. Willems et al. “A note on persistency of excitation”. In: *Systems & Control Letters* 54.4 (2005), pp. 325–329.

[13] J. Coulson, J. Lygeros, and F. Dörfler. “Data-enabled predictive control: In the shallows of the DeePC”. In: *2019 18th European Control Conference (ECC)*. IEEE. 2019, pp. 307–312.

[14] J. G. Rueda-Escobedo and J. Schiffer. “Data-driven Internal Model Control of Second-order Discrete Volterra Systems”. In: *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE. 2020, pp. 4572–4579.

[15] M. Guo, C. De Persis, and P. Tesi. “Data-driven stabilization of nonlinear polynomial systems with noisy data”. In: *arXiv preprint arXiv:2011.07833* (2020).

[16] A. Bisoffi, C. De Persis, and P. Tesi. “Data-based stabilization of unknown bilinear systems with guaranteed basin of attraction”. In: *Systems & Control Letters* 145 (2020), p. 104788.

[17] J. Berberich and F. Allgöwer. “A trajectory-based framework for data-driven system analysis and control”. In: *2020 European Control Conference (ECC)*. IEEE. 2020, pp. 1365–1370.

[18] B. Koopman and J. v. Neumann. “Dynamical systems of continuous spectra”. In: *Proceedings of the National Academy of Sciences* 18.3 (1932), pp. 255–263.

[19] A. Bittracher, P. Koltai, and O. Junge. “Pseudogenerators of spatial transfer operators”. In: *SIAM Journal on Applied Dynamical Systems* 14.3 (2015), pp. 1478–1517.

[20] M. Korda and I. Mezić. “Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control”. In: *Automatica* 93 (2018), pp. 149–160.

[21] A. M. Lyapunov. “The general problem of the stability of motion”. In: *International journal of control* 55.3 (1992), pp. 531–534.

[22] A. Der Kiureghian and O. Ditlevsen. “Aleatory or epistemic? Does it matter?” In: *Structural safety* 31.2 (2009), pp. 105–112.

[23] A. Kendall and Y. Gal. “What uncertainties do we need in bayesian deep learning for computer vision?” In: *Advances in neural information processing systems* 30 (2017).

[24] Y. Gal and Z. Ghahramani. “Dropout as a bayesian approximation: Representing model uncertainty in deep learning”. In: *international conference on machine learning*. PMLR. 2016, pp. 1050–1059.

[25] J. Anderson et al. “System level synthesis”. In: *Annual Reviews in Control* 47 (2019), pp. 364–393.

[26] A. Boodi et al. “Intelligent systems for building energy and occupant comfort optimization: A state of the art review and recommendations”. In: *Energies* 11.10 (2018), p. 2604.

[27] Z. Afroz et al. “Modeling techniques used in building HVAC control systems: A review”. In: *Renewable and sustainable energy reviews* 83 (2018), pp. 64–84.

[28] J. Drgoňa et al. “All you need to know about model predictive control for buildings”. In: *Annual Reviews in Control* (2020).

[29] M. Tesfay et al. “Adaptive-model predictive control of electronic expansion valves with adjustable setpoint for evaporator superheat minimization”. In: *Building and Environment* 133 (2018), pp. 151–160.

[30] M. Maasoumy et al. “Online simultaneous state estimation and parameter adaptation for building predictive control”. In: *Dynamic Systems and Control Conference*. Vol. 56130. American Society of Mechanical Engineers. 2013, V002T23A006.

[31] D. A. Bristow, M. Tharayil, and A. G. Alleyne. “A survey of iterative learning control”. In: *IEEE control systems magazine* 26.3 (2006), pp. 96–114.

[32] M. Minakais, S. Mishra, and J. T. Wen. “Groundhog Day: Iterative learning for building temperature control”. In: *2014 IEEE International Conference on Automation Science and Engineering (CASE)*. IEEE. 2014, pp. 948–953.

[33] X. Yan, Q. Ren, and Q. Meng. “Iterative learning control in large scale HVAC system”. In: *2010 8th World Congress on Intelligent Control and Automation*. IEEE. 2010, pp. 5063–5066.

[34] G. Ceusters et al. “Model-predictive control and reinforcement learning in multi-energy system case studies”. In: *Applied Energy* 303 (2021), p. 117634.

[35] E. Mocanu et al. “On-line building energy optimization using deep reinforcement learning”. In: *IEEE transactions on smart grid* 10.4 (2018), pp. 3698–3708.

[36] G. T. Costanzo et al. “Experimental analysis of data-driven control for a building heating system”. In: *Sustainable Energy, Grids and Networks* 6 (2016), pp. 81–90.

[37] Z. Zhang and K. P. Lam. “Practical implementation and evaluation of deep reinforcement learning control for a radiant heating system”. In: *Proceedings of the 5th Conference on Systems for Built Environments*. 2018, pp. 148–157.

[38] H. Kazmi et al. “Generalizable occupant-driven optimization model for domestic hot water production in NZEB”. In: *Applied Energy* 175 (2016), pp. 1–15.

[39] F. Ruelens et al. “Reinforcement learning applied to an electric water heater: From theory to practice”. In: *IEEE Transactions on Smart Grid* 9.4 (2016), pp. 3792–3800.

[40] I. Markovsky and P. Rapisarda. “On the linear quadratic data-driven control”. In: *2007 European Control Conference (ECC)*. IEEE. 2007, pp. 5313–5318.

[41] I. Markovsky and P. Rapisarda. “Data-driven simulation and control”. In: *International Journal of Control* 81.12 (2008), pp. 1946–1959.

[42] C. De Persis and P. Tesi. “Formulas for data-driven control: Stabilization, optimality, and robustness”. In: *IEEE Transactions on Automatic Control* 65.3 (2019), pp. 909–924.

[43] Y. Lian and C. N. Jones. “Nonlinear Data-Enabled Prediction and Control”. In: *arXiv preprint arXiv:2101.03187* (2021).

[44] Y. Yu et al. “On controllability and persistency of excitation in data-driven control: Extensions of willems’ fundamental lemma”. In: *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE. 2021, pp. 6485–6490.

[45] P. Schmitz, T. Faulwasser, and K. Worthmann. “Willems’ fundamental lemma for linear descriptor systems and its use for data-driven output-feedback MPC”. In: *IEEE Control Systems Letters* 6 (2022), pp. 2443–2448.

[46] J.-W. van Wingerden et al. “Data-enabled predictive control with instrumental variables: the direct equivalence with subspace predictive control”. In: *arXiv preprint arXiv:2209.05210* (2022).

[47] J. Berberich et al. “Robust constraint satisfaction in data-driven MPC”. In: *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE. 2020, pp. 1260–1267.

[48] H. J. van Waarde and M. K. Camlibel. “A matrix Finsler’s lemma with applications to data-driven control”. In: *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE. 2021, pp. 5777–5782.

[49] J. Berberich et al. “Robust data-driven state-feedback design”. In: *2020 American Control Conference (ACC)*. IEEE. 2020, pp. 1532–1538.

[50] J. Coulson, J. Lygeros, and F. Dorfler. “Distributionally robust chance constrained data-enabled predictive control”. In: IEEE, 2021.

[51] F. Dorfler, J. Coulson, and I. Markovsky. “Bridging direct & indirect data-driven control formulations via regularizations and relaxations”. In: *IEEE Transactions on Automatic Control* (2022).

[52] L. Xu et al. “A data-driven convex programming approach to worst-case robust tracking controller design”. In: *arXiv preprint arXiv:2102.11918* (2021).

[53] I. Markovsky and F. Dörfler. “Behavioral systems theory in data-driven analysis, signal processing, and control”. In: *Annual Reviews in Control* 52 (2021), pp. 42–64.

[54] E. Elokda et al. “Data-enabled predictive control for quadcopters”. In: *International Journal of Robust and Nonlinear Control* 31.18 (2021), pp. 8916–8936.

[55] J. Berberich et al. “Data-driven model predictive control: closed-loop guarantees and experimental results”. In: *at-Automatisierungstechnik* 69.7 (2021), pp. 608–618.

[56] G. Monge. “Mémoire sur la théorie des déblais et des remblais”. In: *Mem. Math. Phys. Acad. Royale Sci.* (1781), pp. 666–704.

[57] Y. Brenier. “Polar factorization and monotone rearrangement of vector-valued functions”. In: *Communications on pure and applied mathematics* 44.4 (1991), pp. 375–417.

[58] L. V. Kantorovich. “On the translocation of masses”. In: *Dokl. Akad. Nauk. USSR (NS)*. Vol. 37. 1942, pp. 199–201.

[59] A. Tolstoi. “Methods of finding the minimal total kilometrage in cargo transportation planning in space”. In: *TransPress of the National Commissariat of Transportation* 1 (1930), pp. 23–55.

[60] F. L. Hitchcock. “The distribution of a product from several sources to numerous localities”. In: *Journal of mathematics and physics* 20.1-4 (1941), pp. 224–230.

[61] P. M. Esfahani and D. Kuhn. “Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations”. In: *Mathematical Programming* 171.1 (2018), pp. 115–166.

[62] M. Arjovsky, S. Chintala, and L. Bottou. “Wasserstein generative adversarial networks”. In: *International conference on machine learning*. PMLR. 2017, pp. 214–223.

[63] C. Villani. *Optimal transport: old and new*. Vol. 338. Springer, 2009.

[64] G. Peyré, M. Cuturi, et al. “Computational optimal transport: With applications to data science”. In: *Foundations and Trends® in Machine Learning* 11.5-6 (2019), pp. 355–607.

[65] D. Dowson and B. Landau. “The Fréchet distance between multivariate normal distributions”. In: *Journal of multivariate analysis* 12.3 (1982), pp. 450–455.

[66] L. Huang et al. “Decentralized data-enabled predictive control for power system oscillation damping”. In: *IEEE Transactions on Control Systems Technology* 30.3 (2021), pp. 1065–1077.

[67] Y. Lian and C. N. Jones. “From System Level Synthesis to Robust Closed-loop Data-enabled Predictive Control”. In: *arXiv preprint arXiv:2102.06553* (2021).

[68] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust optimization*. Princeton university press, 2009.

[69] G. W. Stewart. *Matrix Algorithms: Volume 1: Basic Decompositions*. SIAM, 1998.

[70] G. H. Golub and C. F. Van Loan. *Matrix computations*. Vol. 3. JHU press, 2013.

[71] J. Löfberg. “YALMIP : A Toolbox for Modeling and Optimization in MATLAB”. In: *In Proceedings of the CACSD Conference*. Taipei, Taiwan, 2004.

[72] L. Gurobi Optimization. *Gurobi Optimizer Reference Manual*. 2021.

[73] F. Oldewurtel, C. N. Jones, and M. Morari. “A tractable approximation of chance constrained stochastic MPC based on affine disturbance feedback”. In: *2008 47th IEEE conference on decision and control*. IEEE. 2008, pp. 4731–4736.

[74] S. Dempe. *Foundations of bilevel programming*. Springer Science & Business Media, 2002.

[75] H. Neudecker. “A matrix trace inequality”. In: *Journal of mathematical analysis and applications* 166.1 (1992), pp. 302–303.

[76] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[77] J. Nocedal and S. Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[78] J. Löfberg. *Minimax approaches to robust model predictive control*. Vol. 812. Linköping University Electronic Press, 2003.

[79] J. Coulson, J. Lygeros, and F. Dörfler. “Regularized and distributionally robust data-enabled predictive control”. In: *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE. 2019, pp. 2696–2701.

[80] F. Dörfler. *Data-Enabled Predictive Control of Autonomous Energy Systems*. http://people.ee.ethz.ch/~floriand/docs/Slides/Dorfler_INI_2019.pdf. Accessed:2021-05-11. 2019.

[81] W. Favoreel, B. De Moor, and M. Gevers. "SPC: Subspace predictive control". In: *IFAC Proceedings Volumes* 32.2 (1999), pp. 4004–4009.

[82] J. Berberich et al. "Linear tracking MPC for nonlinear systems Part II: The data-driven case". In: *IEEE Transactions on Automatic Control* (2022).

[83] G. Marafioti, R. R. Bitmead, and M. Hovd. "Persistently exciting model predictive control". In: *International Journal of Adaptive Control and Signal Processing* 28.6 (2014), pp. 536–552.

[84] M. K. S. Faradonbeh, A. Tewari, and G. Michailidis. "Input perturbations for adaptive control and learning". In: *Automatica* 117 (2020), p. 108950.

[85] D. Sturzenegger. "Model predictive building climate control: Steps towards practice". PhD thesis. ETH Zurich, 2014.

[86] F. Belić, D. Slišković, and Ž. Hocenski. "Detailed Thermodynamic Modeling of Multi-Zone Buildings with Resistive-Capacitive Method". In: *Energies* 14.21 (2021), p. 7051.

[87] *EMU Electronics AG*. https://www.emuag.ch/. Accessed: 2021-06-01.

[88] *InfluxDB*. https://www.influxdata.com. Accessed: 2021-06-01.

[89] *Tomorrow.io*. https://www.tomorrow.io. Accessed: 2021-06-01.

[90] J. Lofberg. "YALMIP: A toolbox for modeling and optimization in MATLAB". In: *2004 IEEE international conference on robotics and automation (IEEE Cat. No. 04CH37508)*. IEEE. 2004, pp. 284–289.

[91] A. Swales et al. "Open Modbus/TCP Specification". In: *Schneider Electric* 29 (1999), pp. 3–19.

[92] R. Godina et al. "Optimal residential model predictive control energy management performance with PV microgeneration". In: *Computers & Operations Research* 96 (2018), pp. 143–156.

[93] A. Aswani et al. "Reducing transient and steady state electricity consumption in HVAC using learning-based model-predictive control". In: *Proceedings of the IEEE* 100.1 (2011), pp. 240–253.

[94] S. Yang et al. "Model predictive control with adaptive machine-learning-based model for building energy efficiency and comfort optimization". In: *Applied Energy* 271 (2020), p. 115147.

[95] C. W. Rowley et al. "Spectral analysis of nonlinear flows". In: *Journal of fluid mechanics* 641 (2009), pp. 115–127.

[96] H. Wu et al. "Variational Koopman models: slow collective variables and molecular kinetics from short off-equilibrium simulations". In: *The Journal of Chemical Physics* 146.15 (2017), p. 154104.

[97] A. Mauroy and J. Goncalves. "Linear identification of nonlinear systems: A lifting technique based on the Koopman operator". In: *arXiv preprint arXiv:1605.04457* (2016).

[98] A. Surana and A. Banaszuk. “Linear observer synthesis for nonlinear systems using Koopman operator framework”. In: *IFAC-PapersOnLine* 49.18 (2016), pp. 716–723.

[99] M. O. Williams et al. “Extending data-driven Koopman analysis to actuated systems”. In: *IFAC Symposium on Nonlinear Control Systems (NOLCOS)*. 2016.

[100] M. E. Villanueva, C. N. Jones, and B. Houska. “Towards global optimal control via koopman lifts”. In: *Automatica* 132 (2021), p. 109610.

[101] P. J. Schmid. “Dynamic mode decomposition of numerical and experimental data”. In: *Journal of fluid mechanics* 656 (2010), pp. 5–28.

[102] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley. “A data–driven approximation of the koopman operator: Extending dynamic mode decomposition”. In: *Journal of Nonlinear Science* 25.6 (2015), pp. 1307–1346.

[103] H. Arbabi and I. Mezic. “Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the Koopman operator”. In: *SIAM Journal on Applied Dynamical Systems* 16.4 (2017), pp. 2096–2126.

[104] S. E. Otto and C. W. Rowley. “Linearly-recurrent autoencoder networks for learning dynamics”. In: *arXiv preprint arXiv:1712.01378* (2017).

[105] N. Takeishi, Y. Kawahara, and T. Yairi. “Learning Koopman invariant subspaces for dynamic mode decomposition”. In: *Advances in Neural Information Processing Systems*. 2017, pp. 1130–1140.

[106] R. Vinuesa and S. L. Brunton. “Enhancing computational fluid dynamics with machine learning”. In: *Nature Computational Science* 2.6 (2022), pp. 358–366.

[107] B. O. Koopman. “Hamiltonian systems and transformation in Hilbert space”. In: *Proceedings of the National Academy of Sciences* 17.5 (1931), pp. 315–318.

[108] I. Gohberg, S. Goldberg, and M. Kaashoek. *Basic classes of linear operators*. Birkhäuser, 2012.

[109] J. A. Rosenfeld et al. “Occupation kernels and densely defined Liouville operators for system identification”. In: *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE. 2019, pp. 6455–6460.

[110] D. Angeli and E. D. Sontag. “Forward completeness, unboundedness observability, and their Lyapunov characterizations”. In: *Systems & Control Letters* 38.4-5 (1999), pp. 209–217.

[111] B. P. Russo and J. A. Rosenfeld. “Liouville operators over the Hardy space”. In: *Journal of Mathematical Analysis and Applications* 508.2 (2022), p. 125854.

[112] P. Van Overschee and B. De Moor. *Subspace identification for linear systems: Theory—Implementation—Applications*. Springer Science & Business Media, 2012.

[113] S. J. Qin. “An overview of subspace identification”. In: *Computers & chemical engineering* 30.10-12 (2006), pp. 1502–1513.

[114] P. Van Overschee and B. De Moor. “A unifying theorem for three subspace system identification algorithms”. In: *Automatica* 31.12 (1995), pp. 1853–1864.

[115] S. J. Qin, W. Lin, and L. Ljung. “A novel subspace identification approach with enforced causal models”. In: *Automatica* 41.12 (2005), pp. 2043–2053.

[116] C. E. Rasmussen. “Gaussian processes in machine learning”. In: *Advanced lectures on machine learning*. Springer, 2004, pp. 63–71.

[117] M. A. Álvarez and N. D. Lawrence. “Computationally efficient convolved multiple output Gaussian processes”. In: *Journal of Machine Learning Research* 12.May (2011), pp. 1459–1500.

[118] E. V. Bonilla, K. M. Chai, and C. Williams. “Multi-task Gaussian process prediction”. In: *Advances in neural information processing systems*. 2008, pp. 153–160.

[119] C. A. Micchelli and M. Pontil. “On learning vector-valued functions”. In: *Neural computation* 17.1 (2005), pp. 177–204.

[120] J. Kabzan et al. “Learning-based model predictive control for autonomous racing”. In: *IEEE Robotics and Automation Letters* 4.4 (2019), pp. 3363–3370.

[121] H. A. Shukla. *Optimization Methods for Control: From Embedded Programmable Hardware to Data-Driven Process Optimization*. Tech. rep. EPFL, 2021.

[122] A. L. Dontchev and R. T. Rockafellar. *Implicit functions and solution mappings*. Vol. 543. Springer, 2009.

[123] S. G. Krantz and H. R. Parks. “Introduction to the Implicit Function Theorem”. In: *The Implicit Function Theorem*. Springer, 2003, pp. 1–12.

[124] S. M. Robinson. “Generalized equations and their solutions, part II: applications to nonlinear programming”. In: *Optimality and stability in mathematical programming*. Springer, 1982, pp. 200–221.

[125] L. El Ghaoui et al. “Implicit deep learning”. In: *arXiv preprint arXiv:1908.06315* 2 (2019).

[126] M. Zanon and S. Gros. “Safe reinforcement learning using robust MPC”. In: *IEEE Transactions on Automatic Control* (2020).

[127] A. Mohammad-Nezhad. “Conic Optimization: Optimal Partition, Parametric, and Stability Analysis”. PhD thesis. Lehigh University, 2019.

[128] A. V. Fiacco. *Mathematical programming with data perturbations*. CRC Press, 2020.

[129] R. T. Rockafellar and R. J.-B. Wets. *Variational analysis*. Vol. 317. Springer Science & Business Media, 2009.

[130] J. M. Danskin. “The theory of max-min, with applications”. In: *SIAM Journal on Applied Mathematics* 14.4 (1966), pp. 641–664.

[131] W. W. Hogan. “Point-to-set maps in mathematical programming”. In: *SIAM review* 15.3 (1973), pp. 591–603.

[132] A. Agrawal et al. “Differentiating through a cone program”. In: *arXiv preprint arXiv:1904.09043* (2019).

[133] M. Budišić, R. Mohr, and I. Mezić. “Applied koopmanism”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 22.4 (2012), p. 047510.

[134] M. Korda and I. Mezić. “Learning Koopman eigenfunctions for prediction and control: the transient case”. In: *arXiv preprint arXiv:1810.08733* (2018).

[135] C. Robert. *The Bayesian choice: from decision-theoretic foundations to computational implementation*. Springer Science & Business Media, 2007.

[136] A. K. Akametalu et al. “Reachability-based safe learning with Gaussian processes”. In: *53rd IEEE Conference on Decision and Control*. IEEE. 2014, pp. 1424–1431.

[137] M. Korda and I. Mezić. “On convergence of extended dynamic mode decomposition to the Koopman operator”. In: *Journal of Nonlinear Science* 28.2 (2018), pp. 687–710.

[138] A. Papoulis and S. U. Pillai. *Probability, random variables, and stochastic processes*. Tata McGraw-Hill Education, 2002.

[139] C. M. Bishop. *Pattern recognition and machine learning*. springer, 2006.

[140] J. B. Conway. *A course in functional analysis*. Vol. 96. Springer, 2019.

[141] H. Stark and J. W. Woods. *Probability, random processes, and estimation theory for engineers*. Prentice-Hall, Inc., 1986.

[142] J. Jacod and P. Protter. *Probability essentials*. Springer Science & Business Media, 2012.

[143] G. Feldman and P Graczyk. “On the Skitovich–Darmois theorem for compact Abelian groups”. In: *Journal of Theoretical Probability* 13.3 (2000), pp. 859–869.

[144] V. Skitovitch. “On a property of the normal distribution”. In: *DAN SSSR* 89 (1953), pp. 217–219.

[145] J. J. Gerbrands. “On the relationships between SVD, KLT and PCA”. In: *Pattern recognition* 14.1-6 (1981), pp. 375–381.

[146] J. L. Proctor, S. L. Brunton, and J. N. Kutz. “Generalizing Koopman theory to allow for inputs and control”. In: *SIAM Journal on Applied Dynamical Systems* 17.1 (2018), pp. 909–930.

[147] B. Efron. “The estimation of prediction error: covariance penalties and cross-validation”. In: *Journal of the American Statistical Association* 99.467 (2004), pp. 619–632.

[148] H. Akaike. “A new look at the statistical model identification”. In: *Selected Papers of Hirotugu Akaike*. Springer, 1974, pp. 215–222.

[149] T. Gustafsson. “Subspace-based system identification: weighting and pre-filtering of instruments”. In: *Automatica* 38.3 (2002), pp. 433–443.

[150] A. Billard et al. “Robot programming by demonstration”. In: *Springer handbook of robotics* (2008), pp. 1371–1394.

[151] S. M. Khansari-Zadeh and A. Billard. “Learning stable nonlinear dynamical systems with gaussian mixture models”. In: *IEEE Transactions on Robotics* 27.5 (2011), pp. 943–957.

[152] S Daniel-Berhe and H Unbehauen. “Experimental physical parameter estimation of a thyristor driven DC-motor using the HMF-method”. In: *Control Engineering Practice* 6.5 (1998), pp. 615–626.

[153] A. Paszke et al. “Pytorch: An imperative style, high-performance deep learning library”. In: *Advances in neural information processing systems* 32 (2019).

[154] S. Daniel-berhe and H. Unbehauen. “Parameter Estimation of the Nonlinear Dynamics of a Thyristor Driven DC-Motor Experimental Set-Up Using HMF-Method”. In: *IFAC Proceedings Volumes* 30.11 (1997). IFAC Symposium on System Identification (SYSID’97), Kitakyushu, Fukuoka, Japan, 8-11 July 1997, pp. 203 –208. DOI: https://doi.org/10.1016/S1474-6670(17)42847-3.

[155] R. M. Miura. “The Korteweg–deVries equation: A survey of results”. In: *SIAM review* 18.3 (1976), pp. 412–459.

[156] S. Saitoh and Y. Sawano. *Theory of reproducing kernels and applications*. Springer, 2016.

[157] G. K. Pedersen. *Analysis now*. Vol. 118. Springer Science & Business Media, 2012.

[158] I. Steinwart. “On the influence of the kernel on the consistency of support vector machines”. In: *Journal of machine learning research* 2.Nov (2001), pp. 67–93.

[159] K. Fukumizu, F. R. Bach, and M. I. Jordan. “Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces”. In: *Journal of Machine Learning Research* 5.Jan (2004), pp. 73–99.

[160] A. Gretton et al. “A kernel method for the two-sample-problem”. In: *Advances in neural information processing systems* 19 (2006).

[161] K. Fukumizu et al. “Kernel measures of conditional dependence”. In: *Advances in neural information processing systems* 20 (2007).

[162] K. Fukumizu, F. R. Bach, and M. I. Jordan. “Kernel dimension reduction in regression”. In: *The Annals of Statistics* 37.4 (2009), pp. 1871–1905.

[163] A. Berlinet and C. Thomas-Agnan. *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media, 2011.

[164] I. Steinwart and A. Christmann. *Support vector machines*. Springer Science & Business Media, 2008.

[165] M. Fazel, H. Hindi, and S. P. Boyd. “A rank minimization heuristic with application to minimum order system approximation”. In: *Proceedings of the 2001 American Control Conference.(Cat. No. 01CH37148)*. Vol. 6. IEEE. 2001, pp. 4734–4739.

[166] K. Muandet et al. “Kernel mean embedding of distributions: A review and beyond”. In: *arXiv preprint arXiv:1605.09522* (2016).

[167] M. Hertneck et al. “Learning an approximate model predictive controller with guarantees”. In: *IEEE Control Systems Letters* 2.3 (2018), pp. 543–548.

[168] S. Gao, S. Kong, and E. M. Clarke. “dReal: An SMT solver for nonlinear theories over the reals”. In: *International conference on automated deduction*. Springer. 2013, pp. 208–214.

[169] Y.-C. Chang, N. Roohi, and S. Gao. “Neural lyapunov control”. In: *Advances in neural information processing systems* 32 (2019).

[170] H. Dai et al. “Counter-example guided synthesis of neural network lyapunov functions for piecewise linear systems”. In: *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE. 2020, pp. 1274–1281.

[171] J. Kapinski et al. “Simulation-guided Lyapunov analysis for hybrid dynamical systems”. In: *Proceedings of the 17th international conference on Hybrid systems: computation and control*. 2014, pp. 133–142.

[172] S. Gao, J. Avigad, and E. M. Clarke. “δ-complete decision procedures for satisfiability over the reals”. In: *International Joint Conference on Automated Reasoning*. Springer. 2012, pp. 286–300.

[173] V. I. Zubov. *Methods of AM Lyapunov and their application*. P. Noordhoff, 1964.

[174] P. Giesl. *Construction of global Lyapunov functions using radial basis functions*. Vol. 1904. Springer, 2007.

[175] P. A. Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. California Institute of Technology, 2000.

[176] P. A. Parrilo and S. Lall. “Semidefinite programming relaxations and algebraic optimization in control”. In: *European Journal of Control* 9.2-3 (2003), pp. 307–321.

[177] A. Oustry, M. Tacchi, and D. Henrion. “Inner approximations of the maximal positively invariant set for polynomial dynamical systems”. In: *IEEE Control Systems Letters* 3.3 (2019), pp. 733–738.

[178] D. Henrion and M. Korda. “Convex computation of the region of attraction of polynomial control systems”. In: *IEEE Transactions on Automatic Control* 59.2 (2013), pp. 297–312.

[179] M. Tacchi et al. “Approximating regions of attraction of a sparse polynomial differential system”. In: *IFAC-PapersOnLine* 53.2 (2020), pp. 3266–3271.

[180] A. A. Ahmadi and A. Majumdar. “DSOS and SDSOS optimization: more tractable alternatives to sum of squares and semidefinite optimization”. In: *SIAM Journal on Applied Algebra and Geometry* 3.2 (2019), pp. 193–230.

[181] Z. Sun. “Stability of piecewise linear systems revisited”. In: *Annual Reviews in Control* 34.2 (2010), pp. 221–231.

[182] H. Lin and P. J. Antsaklis. “Stability and stabilizability of switched linear systems: a survey of recent results”. In: *IEEE Transactions on Automatic control* 54.2 (2009), pp. 308–322.

[183] A. Alessio and A. Bemporad. “A survey on explicit model predictive control”. In: *Nonlinear model predictive control*. Springer, 2009, pp. 345–369.

[184] M. Johansson and A. Rantzer. “Computation of piecewise quadratic Lyapunov functions for hybrid systems”. In: *1997 European Control Conference (ECC)*. IEEE. 1997, pp. 2005–2010.

[185] H. Ravanbakhsh and S. Sankaranarayanan. “Learning control lyapunov functions from counterexamples and demonstrations”. In: *Autonomous Robots* 43.2 (2019), pp. 275–307.

[186] R. Schwan, C. N. Jones, and D. Kuhn. “Stability Verification of Neural Network Controllers using Mixed-Integer Programming”. In: *arXiv preprint arXiv:2206.13374* (2022).

[187] A. Solar-Lezama et al. “Combinatorial sketching for finite programs”. In: *Proceedings of the 12th international conference on Architectural support for programming languages and operating systems*. 2006, pp. 404–415.

[188] A. Solar-Lezama. *Program synthesis by sketching*. University of California, Berkeley, 2008.

[189] S. Chen et al. “Learning lyapunov functions for hybrid systems”. In: *Proceedings of the 24th International Conference on Hybrid Systems: Computation and Control*. 2021, pp. 1–11.

[190] A. Abate et al. “Formal synthesis of Lyapunov neural networks”. In: *IEEE Control Systems Letters* 5.3 (2020), pp. 773–778.

[191] T. A. Johansen. “Computation of Lyapunov functions for smooth nonlinear systems using convex optimization”. In: *Automatica* 36.11 (2000), pp. 1617–1626.

[192] P. Giesl et al. “Approximation of Lyapunov functions from noisy data”. In: *arXiv preprint arXiv:1601.01568* (2016).

[193] S. F. Marinósson. “Lyapunov function construction for ordinary differential equations with linear programming”. In: *Dynamical Systems: An International Journal* 17.2 (2002), pp. 137–150.

[194] P. M. Julian. “A high-level canonical piecewise linear representation: Theory and applications”. PhD thesis. Universidad Nacional del Sur (Argentina), 1999.

[195] R. Baier, L. Grüne, and S. F. Hafstein. “Linear programming based Lyapunov function computation for differential inclusions”. In: *Discrete & Continuous Dynamical Systems-B* 17.1 (2012), p. 33.

[196] S. F. Hafstein. “AN ALGORITHM FOR CONSTRUCTING LYAPUNOV FUNCTIONS.” In: *Electronic Journal of Differential Equations* 2007 (2007).

[197] J.-B. Hiriart-Urruty and C. Lemaréchal. *Fundamentals of convex analysis.* Springer Science & Business Media, 2004.

[198] F. H. Clarke. *Optimization and Nonsmooth Analysis.* Classics in Applied Mathematics. SIAM, 1990.

[199] J. P. La Salle. *The stability of dynamical systems.* SIAM, 1976.

[200] M. Della Rossa. “Non-Smooth Lyapunov Functions for Stability Analysis of Hybrid Systems”. PhD thesis. Institut National des Sciences Appliquées de Toulouse, 2020.

[201] H. L. Royden and P. Fitzpatrick. *Real analysis.* Vol. 32. Macmillan New York, 1988.

[202] F. Aurenhammer. “A criterion for the affine equivalence of cell complexes in R d and convex polyhedra in R d+ 1”. In: *Discrete & Computational Geometry* 2.1 (1987), pp. 49–64.

[203] A. Ben-Tal and A. Nemirovski. “Robust optimization–methodology and applications”. In: *Mathematical programming* 92.3 (2002), pp. 453–480.

[204] J.-P. Calliess et al. “Lazily adapted constant kinky inference for nonparametric regression and model-reference adaptive control”. In: *Automatica* 122 (2020), p. 109216.

[205] C. K. Williams and C. E. Rasmussen. *Gaussian processes for machine learning.* Vol. 2. 3. MIT press Cambridge, MA, 2006.

[206] P. Scharnhorst et al. “Robust Uncertainty Bounds in Reproducing Kernel Hilbert Spaces: A Convex Optimization Approach”. In: *arXiv preprint arXiv:2104.09582* (2021).

[207] Y. Nesterov and A. Nemirovskii. *Interior-point polynomial algorithms in convex programming.* SIAM, 1994.

[208] S. F. Hafstein*. “A constructive converse Lyapunov theorem on asymptotic stability for nonlinear autonomous ordinary differential equations”. In: *Dynamical Systems* 20.3 (2005), pp. 281–299.

[209] P. Giesl and S. Hafstein. “Review on computational methods for Lyapunov functions”. In: *Discrete & Continuous Dynamical Systems-B* 20.8 (2015), p. 2291.

[210] A. Subbaraman and A. R. Teel. “A Matrosov theorem for strong global recurrence”. In: *Automatica* 49.11 (2013), pp. 3390–3395.

[211] S. P. Meyn and R. L. Tweedie. *Markov chains and stochastic stability.* Springer Science & Business Media, 2012.

[212] H. Krieg. *Modeling and solution of continuous set covering problems by means of semi-infinite optimization.* Fraunhofer Verlag, 2019.

[213] E. D. Andersen, C. Roos, and T. Terlaky. “On implementing a primal-dual interior-point method for conic quadratic optimization”. In: *Mathematical Programming* 95.2 (2003), pp. 249–277.

[214] L. Vandenberghe and M. S. Andersen. “Chordal graphs and semidefinite optimization”. In: *Foundations and Trends in Optimization* 1.4 (2015), pp. 241–433.

[215] K. Rybnikov. *Polyhedral partitions and stresses.* Queen's University at Kingston, 2000.

[216] F. Aurenhammer. "Power diagrams: properties, algorithms and applications". In: *SIAM Journal on Computing* 16.1 (1987), pp. 78–96.

[217] B. Delaunay et al. "Sur la sphere vide". In: *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk* 7.793-800 (1934), pp. 1–2.

[218] J. Bolte, S. Sabach, and M. Teboulle. "Proximal alternating linearized minimization for nonconvex and nonsmooth problems". In: *Mathematical Programming* 146.1 (2014), pp. 459–494.

[219] R. Verschueren et al. "acados—a modular open-source framework for fast embedded optimal control". In: *Mathematical Programming Computation* 14.1 (2022), pp. 147–183.

[220] W. Heemels, K. H. Johansson, and P. Tabuada. "An introduction to event-triggered and self-triggered control". In: *Proc. 51th Conf. Decis. Control.* 2012, pp. 3270–3285.

[221] J. B. Berglind, T. Gommans, and W. Heemels. "Self-triggered MPC for constrained linear systems and quadratic costs". In: *IFAC Proceedings Volumes* 45.17 (2012), pp. 342–348.

[222] D. Bernardini and A. Bemporad. "Energy-aware robust model predictive control based on noisy wireless sensors". In: *Automatica* 48.1 (2012), pp. 36–44.

[223] E. Henriksson et al. "Self-triggered model predictive control for network scheduling and control". In: *IFAC-PapersOnLine* 45.15 (2012), pp. 432–438.

[224] S. Wildhagen, C. N. Jones, and F. Allgöwer. "A resource-aware approach to self-triggered model predictive control". In: *IFAC-PapersOnLine* 53.2 (2020), pp. 2733–2738.

[225] Y. Lian et al. "Resource-Aware Asynchronous Multi-Agent Coordination via Self-Triggered MPC". In: *Proc. 59th Conf. Decis. Control.* 2020, pp. 685–690.

[226] H. Li and Y. Shi. "Event-triggered robust model predictive control of continuous-time nonlinear systems". In: *Automatica* 50.5 (2014), pp. 1507–1513.

[227] E. Aydiner, F. D. Brunner, W. Heemels, et al. "Robust self-triggered model predictive control for constrained discrete-time LTI systems based on homothetic tubes". In: *Proc. Eur. Control Conf.* 2015, pp. 1587–1593.

[228] F. D. Brunner, M. Heemels, and F. Allgöwer. "Robust self-triggered MPC for constrained linear systems: A tube-based approach". In: *Automatica* 72 (2016), pp. 73–83.

[229] M. Farina and R. Scattolini. "Tube-based robust sampled-data MPC for linear continuous-time systems". In: *Automatica* 48.7 (2012), pp. 1473–1476.

[230] C. Liu et al. "Robust self-triggered min–max model predictive control for discrete-time nonlinear systems". In: *Automatica* 89 (2018), pp. 333–339.

[231] H. G. Bock and K.-J. Plitt. "A multiple shooting algorithm for direct solution of optimal control problems". In: *IFAC-PapersOnLine* 17.2 (1984), pp. 1603–1608.

[232] V. Krishnan. *Nonlinear filtering and smoothing: An introduction to martingales, stochastic integrals and estimation.* Courier Corporation, 2013.

[233] J.-F. Le Gall. *Brownian motion, martingales, and stochastic calculus*. Springer, 2016.

[234] L. Giulioni. "Stochastic model predictive control with application to distributed control systems". In: (2015).

[235] W. S. Levine, L. Grüne, et al. *Handbook of model predictive control*. Springer, 2018.

[236] R. Mohler and R. Rink. "Multivariable bilinear system control". In: *Proc. IFAC Symp. Multivariable Control Systems*. Vol. 14. 2. 1968.

[237] R. Rink and R. Mohler. "Completely controllable bilinear systems". In: *SIAM Journal on Control* 6.3 (1968), pp. 477–486.

[238] A Ruberti and R. Mohler. "Variable Structure Systems with Applications to Economics and Biology". In: *Proceedings of the Second US-Italy Seminar on Variable Structure Systems*. Springer. 1974.

[239] D. L. Elliott. "Bilinear systems". In: *Wiley Encyclopedia of Electrical and Electronics Engineering* (2001).

[240] W.-H. Steeb. "A note on Carleman linearization". In: *Physics Letters A* 140.6 (1989), pp. 336–338.

[241] E. Kaiser, J. N. Kutz, and S. L. Brunton. "Data-driven approximations of dynamical systems operators for control". In: *The Koopman Operator in Systems and Control* (2020), pp. 197–234.

[242] D. Elliott. *Bilinear control systems: matrices in action*. Vol. 169. Springer Science & Business Media, 2009.

[243] W. J. Rugh. *Nonlinear system theory*. Johns Hopkins University Press Baltimore, MD, 1981.

[244] J. Baillieul. "Geometric methods for nonlinear optimal control problems". In: *Journal of optimization theory and applications* 25.4 (1978), pp. 519–548.

[245] S. M. Rajguru, M. A. Ifediba, and R. D. Rabbitt. "Three-dimensional biomechanical model of benign paroxysmal positional vertigo". In: *Annals of biomedical engineering* 32.6 (2004), pp. 831–846.

[246] D. D'alessandro and M. Dahleh. "Optimal control of two-level quantum systems". In: *IEEE Transactions on Automatic Control* 46.6 (2001), pp. 866–876.

[247] G. Escobar et al. "An experimental comparison of several nonlinear controllers for power converters". In: *IEEE Control Systems Magazine* 19.1 (1999), pp. 66–82.

[248] H. Sira-Ramirez. "Sliding motions in bilinear switched networks". In: *IEEE transactions on circuits and systems* 34.8 (1987), pp. 919–933.

[249] S. Peitz. "Controlling nonlinear PDEs using low-dimensional bilinear approximations obtained from data". In: *arXiv preprint arXiv:1801.06419* (2018).

[250] J. Haddad and N. Geroliminis. "On the stability of traffic perimeter control in two-region urban cities". In: *Transportation Research Part B: Methodological* 46.9 (2012), pp. 1159–1176.

[251] M. B. Kane, J. Scruggs, and J. P. Lynch. “Model-predictive control techniques for hydronic systems implemented on wireless sensor and actuator networks”. In: *2014 American Control Conference*. IEEE. 2014, pp. 3542–3547.

[252] L. Hetel, M. Defoort, and M. Djemai. “Binary control design for a class of bilinear systems: Application to a multilevel power converter”. In: *IEEE Transactions on Control Systems Technology* 24.2 (2015), pp. 719–726.

[253] J. B. Rawlings, D. Q. Mayne, and M. Diehl. *Model predictive control: theory, computation, and design*. Vol. 2. Nob Hill Publishing Madison, WI, 2017.

[254] S. Boyd, N. Parikh, and E. Chu. *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.

[255] A. Rantzer. “Dynamic dual decomposition for distributed control”. In: *2009 American Control Conference*. IEEE. 2009, pp. 884–888.

[256] I Necoara and J. Suykens. “Interior-point lagrangian decomposition method for separable convex optimization”. In: *Journal of Optimization Theory and Applications* 143.3 (2009), pp. 567–588.

[257] J. V. Frasch, S. Sager, and M. Diehl. “A parallel quadratic programming method for dynamic optimization problems”. In: *Mathematical programming computation* 7.3 (2015), pp. 289–329.

[258] S. Richter, M. Morari, and C. N. Jones. “Towards computational complexity certification for constrained MPC based on Lagrange relaxation and the fast gradient method”. In: *2011 50th IEEE Conference on Decision and Control and European Control Conference*. IEEE. 2011, pp. 5223–5229.

[259] B. Houska, J. Frasch, and M. Diehl. “An augmented Lagrangian based algorithm for distributed nonconvex optimization”. In: *SIAM Journal on Optimization* 26.2 (2016), pp. 1101–1127.

[260] Y. Jiang et al. “Parallel MPC for linear systems with input constraints”. In: *IEEE Transactions on Automatic Control* (2020).

[261] Y. Jiang, C. N. Jones, and B. Houska. “A Time Splitting Based Real-Time Iteration Scheme for Nonlinear MPC”. In: *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE. 2019, pp. 2350–2355.

[262] F. Laine and C. Tomlin. “Parallelizing LQR Computation Through Endpoint-Explicit Riccati Recursion”. In: *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE. 2019, pp. 1395–1402.

[263] H. Deng and T. Ohtsuka. “A parallel Newton-type method for nonlinear model predictive control”. In: *Automatica* 109 (2019), p. 108560.

[264] S. Shin et al. “A parallel decomposition scheme for solving long-horizon optimal control problems”. In: *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE. 2019, pp. 5264–5271.

[265] I. Nielsen and D. Axehill. “A parallel structure exploiting factorization algorithm with applications to model predictive control”. In: *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE. 2015, pp. 3932–3938.

[266] Y. Wang and S. Boyd. “Fast model predictive control using online optimization”. In: *IEEE Transactions on control systems technology* 18.2 (2009), pp. 267–278.

[267] H. J. Ferreau, H. G. Bock, and M. Diehl. “An online active set strategy to overcome the limitations of explicit MPC”. In: *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal* 18.8 (2008), pp. 816–830.

[268] M. Diehl, H. G. Bock, and J. P. Schlöder. “A real-time iteration scheme for nonlinear optimization in optimal feedback control”. In: *SIAM Journal on control and optimization* 43.5 (2005), pp. 1714–1736.

[269] E. Zafiriou. “Robust model predictive control of processes with hard constraints”. In: *Computers & Chemical Engineering* 14.4-5 (1990), pp. 359–371.

[270] A. Bemporad et al. “The explicit linear quadratic regulator for constrained systems”. In: *Automatica* 38.1 (2002), pp. 3–20.

[271] D. M. Raimondo et al. “A robust explicit nonlinear MPC controller with input-to-state stability guarantees”. In: *IFAC Proceedings Volumes* 44.1 (2011), pp. 9284–9289.

[272] A. Bemporad et al. “The explicit linear quadratic regulator for constrained systems”. In: *Automatica* 38.1 (Jan. 2002), pp. 3–20.

[273] F. Borrelli, A. Bemporad, and M. Morari. *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.

[274] S. Wright, J. Nocedal, et al. “Numerical optimization”. In: *Springer Science* 35.67-68 (1999), p. 7.

[275] N. G. D. Robinson. “A second derivative SQP method: local convergence”. In: (2009).

[276] M. R. Hestenes. “Multiplier and gradient methods”. In: *Journal of optimization theory and applications* 4.5 (1969), pp. 303–320.

[277] M. J. Powell. “A method for nonlinear constraints in minimization problems”. In: *Optimization* (1969), pp. 283–298.

[278] R. T. Rockafellar. “Augmented Lagrangians and applications of the proximal point algorithm in convex programming”. In: *Mathematics of operations research* 1.2 (1976), pp. 97–116.

[279] R. Verschueren et al. “acados – a modular open-source framework for fast embedded optimal control”. In: *Mathematical Programming Computation* (2021). DOI: 10.1007/s12532-021-00208-8.

[280] P. E. Gill, W. Murray, and M. A. Saunders. “SNOPT: An SQP algorithm for large-scale constrained optimization”. In: *SIAM review* 47.1 (2005), pp. 99–131.

[281] G Di Pillo and L Grippo. “A new augmented Lagrangian function for inequality constraints in nonlinear programming problems”. In: *Journal of Optimization Theory and Applications* 36.4 (1982), pp. 495–519.

[282] S. Alwadani et al. “The difference vectors for convex sets and a resolution of the geometry conjecture”. In: *Open Journal of Mathematical Optimization* 2 (2021), pp. 1–18.

[283] S. Alwadani et al. “Resolvents and Yosida approximations of displacement mappings of isometries”. In: *Set-Valued and Variational Analysis* 29.3 (2021), pp. 721–733.

[284] B. Houska and Y. Jiang. “Distributed Optimization and Control with ALADIN”. In: *Recent Advances in Model Predictive Control: Theory, Algorithms, and Applications* (2021), 135–163.

[285] R. T. Rockafellar. “Augmented Lagrange multiplier functions and duality in nonconvex programming”. In: *SIAM Journal on Control* 12.2 (1974), pp. 268–285.

[286] B. Houska et al. “Convex optimization with ALADIN”. In: *Optimization Online preprint* (2017).

[287] B. Contesse. “Une caractérisation complete des minima locaux en programmation quadratique”. In: (1980).

[288] A. Forsgren, P. Gill, and W Murray. “On the identification of local minimizers in inertia-controlling methods for quadratic programming”. In: *SIAM journal on matrix analysis and applications* 12.4 (1991), pp. 730–746.

[289] Y. Wang, W. Yin, and J. Zeng. “Global convergence of ADMM in nonconvex nonsmooth optimization”. In: *Journal of Scientific Computing* 78.1 (2019), pp. 29–63.

[290] M. Herceg et al. “Enumeration-based approach to solving parametric linear complementarity problems”. In: *Automatica* 62 (2015), pp. 243–248. DOI: 10.1016/j.automatica.2015.09.019.

[291] P. Tøndel, T. Johansen, and A. Bemporad. “Evaluation of piecewise affine control via binary search tree”. In: *Automatica* 39.5 (2003), pp. 945–950.

[292] R. Verschueren et al. “A sparsity preserving convexification procedure for indefinite quadratic programs arising in direct optimal control”. In: *SIAM Journal on Optimization* 27.3 (2017), pp. 2085–2109.

[293] G. Frison. “Algorithms and methods for high-performance model predictive control”. PhD thesis. 2016.

[294] M. Kvasnica et al. “Multi-parametric toolbox (MPT)”. In: *International Workshop on Hybrid Systems: Computation and Control.* Springer. 2004, pp. 448–462.

[295] A. Engelmann et al. “ALADIN-—An open-source MATLAB toolbox for distributed non-convex optimization”. In: *Optimal Control Applications and Methods* 43.1 (2022), pp. 4–22.

[296] G. Frison et al. “BLASFEO: Basic linear algebra subroutines for embedded optimization”. In: *ACM Transactions on Mathematical Software (TOMS)* 44.4 (2018), pp. 1–30.

[297] G. Frison and M. Diehl. “HPIPM: a high-performance quadratic programming framework for model predictive control”. In: *IFAC-PapersOnLine* 53.2 (2020), pp. 6563–6569.

[298] F. Oldewurtel et al. “Stochastic model predictive control for building climate control”. In: *IEEE Transactions on Control Systems Technology* 22.3 (2013), pp. 1198–1205.

[299] G. Cimini et al. “Online model predictive torque control for permanent magnet synchronous motors”. In: *2015 IEEE International Conference on Industrial Technology (ICIT)*. IEEE. 2015, pp. 2308–2313.

[300] G. Cimini et al. “Embedded model predictive control with certified real-time optimization for synchronous motors”. In: *IEEE Transactions on Control Systems Technology* 29.2 (2020), pp. 893–900.

[301] R.-C. Li. “Relative perturbation theory: II. Eigenspace and singular subspace variations”. In: *SIAM Journal on Matrix Analysis and Applications* 20.2 (1998), pp. 471–492.

[302] Y. Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.

Ce document a été imprimé
au Centre d'impression EPFL,
imprimerie climatiquement
neutre, certifiée myClimate
depuis 2017

myclimate
neutral
Imprimerie

repro.epfl.ch