الجـــــــــــــــــــمـهـوريـــــة الجـــــــــــــزائـريـة الديمـــــــقـراطيـة الـشـــــــــعبيـة

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التـــــــــــــــــــــــــــعـليـم العـــــــالي و البـــــــــحث العـــــلـمـــي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جـــــــــــــامعة أبي بكـــر بـلـقـايد – تـلمســـــــــــــان –

Université Aboubakr Belkaïd – Tlemcen –
Faculté de TECHNOLOGIE

THESE

Présentée pour l'obtention du **grade** de **DOCTORAT 3ème Cycle**

En : Automatique

Spécialité : Automatique

Par : **DEKHICI Benaissa**

Sujet

Data-Driven Modeling, Order Reduction and Control of Anaerobic Digestion Processes

Soutenue publiquement, le **16 / 01 / 2024**, devant le jury composé de :

Mme CHOUKCHOU BRAHAM Amal	Professeur	Univ. Tlemcen	Président
M Boumediene BENYAHIA	Professeur	Univ. Tlemcen	Directeur de thèse
M Brahim CHERKI	Professeur	Univ. Tlemcen	Co- Directeur de thèse
M YADI Karim	Professeur	Univ. Tlemcen	Examinateur 1
M SARI Tawfik	Professeur	INRAE, Montpellier, France	Examinateur 2
M Jérôme Harmand	Doctor	INRAE, Narbonne, France	Invité

NB : les membres du jury doivent être classés conformément au P.V d'autorisation de soutenance

Contents

List of Figures

List of Tables

Abbreviations

AD	Anaerobic Digestion
COD	Chemical Oxygen Demand
VFA	Volatile Fatty Acids
CSTR	Continuously Stirred Tank Reactor
ADM1	Anaerobic Digestion Model No.1
AM2	Anaerobic Model 2
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
MPC	Model Predictive Control
DMD	Dynamic Mode Decomposition
DMDc	Dynamic Mode Decomposition with Control
EDMD	Extended Dynamic Mode Decomposition
EDMDc	Extended Dynamic Mode Decomposition with Control
SVD	Singular Value Decomposition
POD	Proper Orthogonal Decomposition
PCA	Principal Component Analysis
KMPC	Koopman Model Predictive Control
LMPC	Linear Model Predictive Control
PRBS	Pseudo-Random Binary Signal
RMSE	Relative Root Mean Square Error
RBF	Radial Basis Functions
IMQ-RBF	Multi-Quadric Radial Basis Function

Abstract

This research delves into the realms of data-driven modeling, order reduction, and control strategies within the context of Anaerobic Digestion (AD) processes. The study is centered on addressing pivotal challenges in this domain and delivering innovative contributions to the field. The primary objectives encompass streamlining the complexity of the Anaerobic Digestion Model No.1 (ADM1) for the specific purpose of control, as well as the exploration of suitable data-driven techniques to achieve precise modeling and prediction of AD systems. Furthermore, the research endeavors to extract kinetics reactions from simulated time-series AD data, develop robust predictive models for Chemostat dynamics under both Monod and Haldane kinetics through data-driven methodologies, and employ the Koopman Operator theory to enable data-driven modeling and control of the Chemostat system, relying solely on substrate measurements.

By adopting a data-driven approach, this research aims to provide profound insights into the intricacies of AD processes, thereby shedding light on their complex dynamics and advancing our comprehension beyond conventional models. It introduces an alternative modeling perspective exclusively grounded in data, augmenting our analytical capabilities within the realm of AD processes. The research rigorously evaluates and tests a variety of data-driven techniques, yielding intriguing results. Notably, the application of the Koopman Operator theory represents a significant contribution, particularly in scenarios where measurement resources are limited. This innovation holds the potential to pave the way for robust control strategies within AD systems, ultimately enhancing their sustainability and efficiency.

Keywords: Anaerobic Digestion, ADM1, Chemostat, AM2, ODE, Data-Driven Modeling, Order Reduction, Model Predictive Control, SVD, Koopman Operator, DMD.

Acknowledgements

I would like to take this opportunity to convey my deepest appreciation to the individuals and institutions who have played a pivotal role in my journey towards the attainment of my Ph.D. This achievement would not have been possible without their unwavering support, invaluable guidance, and continuous encouragement.

Foremost, I extend my profound gratitude to my esteemed mentors, Prof. Brahim CHERKI and Prof. Boumediene BENYAHIA. Their exceptional mentorship, insightful guidance, and profound expertise in the field of engineering have been instrumental in shaping my research endeavors. Their unwavering commitment to my academic and professional growth has been truly invaluable.

I am also indebted to Prof. Amal CHOUKCHOU-BRAHAM, whose assistance during the initial stages of my Ph.D. candidature test was pivotal. Without her support, I would not have embarked on this academic journey. I extend my heartfelt thanks for her generosity, modesty, and collaboration. Additionally, I express my gratitude to the esteemed members of the jury Pr. Tewfik SARI, Pr. Karim Yadi and Dr. Jérôme HARMAND for dedicating their time, providing invaluable insights, and offering constructive suggestions during the evaluation of my work. Their expertise and guidance have significantly contributed to the quality and rigor of my research.

As an Erasmus and Bilateral student, I am deeply thankful to the University of Trento for their unwavering support and the enriching academic environment they provided during my tenure in Trento. Special thanks go to Prof. Gianni ANDREOTTOLA and Prof. Luca FIORI, as well as all those who supported me during this phase. The opportunities and resources offered by the university have been invaluable in shaping my academic and research journey.

Lastly, I wish to express my heartfelt gratitude to my family, particularly my mother, DEKHICI Latifa, my brother, DEKHICI Samir, my three sisters, my wife, ROUABAH Siham, and my newborn son, DEKHICI Ahmed. Their boundless prayers, unwavering encouragement, and understanding throughout this journey have been a constant source of inspiration. Your unwavering belief in me has been a driving force behind my success.

Publications Related to The Thesis

Journal Publications:

- **Benaissa DEKHICI**, Boumediene BENYAHIA and Brahim CHERKI. "DATA-DRIVEN CONTROL OF THE CHEMOSTAT USING THE KOOPMAN OPERATOR THEORY", **U.P.B. Sci. Bull., Series C, Vol. 85, Iss. 2**,pp 137-150, 2023. https://www.scientificbulletin.upb.ro/rev_docs_arhiva/rezec0_330715.pdf

- **Benaissa DEKHICI**, Boumediene BENYAHIA, Brahim CHERKI, Luca FIORI, and Gianni ANDREOTTOLA. "Modeling the biogas production from HTC products in a continuous hybrid fixed bed bioreactor ." Submitted in the **ACM Transactions on Modeling and Computer Simulation**, Under review.

Refereed international conferences:

- **Benaissa DEKHICI**, Boumediene BENYAHIA and Brahim CHERKI. "Data-Driven Determination of the Governing Equations of the Continuously Stirred Tank Reactor ." In the **2023 10th International Conference on Electrical and Electronics Engineering (ICEEE 2023)**, Istanbul, Turkey, May 8-10, 2023.

- **Benaissa DEKHICI**, Boumediene Benyahia, Brahim Cherki. "Dynamic Mode Decomposition with Control for Data-driven Modeling of Anaerobic Digestion Process." **2022** 16^{th} **African Conference on Research in Computer Science and Applied Mathematics** CARI 2022, Oct 2022, Tunis, Tunisia. https://hal.science/hal-03696038

- **Benaissa DEKHICI**, Boumediene BENYAHIA and Brahim CHERKI, "Forecast of chemostat dynamics using data-driven approach," **2021 International Conference on Control, Automation and Diagnosis (ICCAD)**, Grenoble, France, 2021, pp. 1 -6, https://doi.org/10.1109/iccad52417.2021.9638749

- **Benaissa DEKHICI**, Boumediene BENYAHIA and Brahim CHERKI. "Model order reduction using Dynamic Mode Decomposition: Application to the Anaerobic Digestion Model N.1 (ADM1)" In the **2020 15th African Conference on Research in Computer Science and Applied Mathematics (CARI)**, Thiès, Senegal, Oct 2020. https://hal.science/hal-02928872

Chapter 1

General Introduction

1.1 Background Review

1.1.1 Anaerobic Digestion

Anaerobic digestion (AD), commonly referred to as methanization, stands out as a robust and economically viable technique for waste recovery. Particularly prominent in organic waste treatment and recovery within the agricultural and industrial sectors, AD utilizes the effectiveness of microorganisms operating in an oxygen-devoid environment to disintegrate complex organic matters. The remarkable outcome of this microbial collaboration is the production of biogas, predominantly composed of methane (CH_4) in the range of 60-70 % and carbon dioxide (CO_2) at 30-40 % [6]. This biogas, in turn, finds multifaceted applications, ranging from electricity and heat generation to being utilized as a biofuel.

The significance of AD transcends its waste management attributes, signaling as a repository of clean and renewable energy. In contrast to conventional energy sources that often contribute negatively to the environmental equilibrium, AD provides a greener alternative. Moreover, the process yields digestate, residual product with multiple uses. This digestate can serve as livestock bedding or function as an organic amendment, essentially a fertilizer, benefiting agricultural communities and farms [7].

Central to the optimization of the AD process is the sensible selection and monitoring of indicators that influence biogas production. These indicators are crucial for the functionality of the system. The evolving literature landscape places emphasis on the mathematical modeling of the AD process. This endeavor not only enhances our comprehension of the intricacies of process dynamics but also presents opportunities for optimizing digester performance [8]. As such, the endeavor of modeling not only furthers our grasp of the process intricacies but also serves as a catalyst for refining the process towards efficiency.

Throughout the entire AD process, a community of anaerobic microorganisms collaborates at each stage, working in harmony to degrade organic matter and produce biogas. This AD process unfolds through four primary phases: hydrolysis, acidogenesis, acetogenesis, and methanogenesis [9].

In the hydrolysis stage, intricate compounds undergo a transformation into sol-

uble constituents. This phase involves the enzymatic conversion of high-molecular-weight elements like organic polymers and lipids into simpler units such as fatty acids, monosaccharides, and amino acids, which are easily taken up by microbial metabolism to serve as an energy source. After this, the fermentation process follows, encompassing the acidogenesis and acetogenesis stages. Acidogenesis marks the transformation of the products generated from hydrolysis into intermediate components with lower molecular weights. At this stage, simple volatile fatty acids (VFAs) materialize from sugars and amino acids. In contrast, acetogenesis pertains to the fermentation process in which VFAs and alcohol are converted to acetate, hydrogen, and carbon dioxide. Lastly, a cluster of archaea microorganisms, known as methanogens, orchestrates the bacterial conversion of intermediate components into various simplified end products, primarily composed of CH_4 and CO_2 [6], [9].

In AD, microorganism communities are notably influenced by various factors, including shifts in key monitoring parameters such as pH, alkalinity, VFAs, as well as pivotal operational parameters like temperature, hydraulic retention time, substrate composition, and organic loading rate (OLR) [10]. These monitoring and operational parameters are pertinent in both the liquid and gas phases. Temperature has a considerable impact on microbial metabolic activities during fermentation [11], consequently influencing the rate of digestion and methane production [6]. The operational spectrum for AD encompasses three temperature ranges: psychrophilic ($4 - 15^oC$), mesophilic ($20 - 40^oC$), and thermophilic ($45 - 70^oC$). Among these, OLR assumes a central role as an operational parameter in AD systems along with the concentrations in the input and the dilution rate. The generation rate of biogas is closely linked to OLR, maintaining it within a suitable range promoting favorable biogas yields. However, a significant OLR surge beyond the appropriate threshold emerges as a primary catalyst for VFAs accumulation, ultimately leading to reduced biogas production. The significance of pH in AD cannot be overstated; it dictates the prevalence of microorganisms at each fermentation stage. Throughout the AD process, pH generally resides within the range of 6.0-8.0, its fluctuations contingent on the evolution of organic matter fermentation. Crucially, pH's role in the methanogenic phase is pivotal; a drop in pH triggers process inhibition [10]. The concentration of VFAs, acting as the principal methanogenic intermediate, serves as a widely suggested control and monitoring parameter for anaerobic digesters, given its reliable correlation with process imbalance in reactor accumulation. VFAs are valuable indicators for gauging system stress levels and diagnosing process issues. The measurement of biogas flow and composition holds paramount importance as they offer insight into overall digester performance [10],[12]. A low CH_4 content in biogas could signal methanogenic bacteria inhibition; hence, maintaining CH_4 concentration above 50 % is vital for efficient operation. Similarly, elevated hydrogen sulfide (H_2S) concentration in biogas can lead to AD inhibition, emission concerns, and corrosion problems [13]. Furthermore, recognizing that single indicators lack universality across digesting systems, a combination of factors such as VFA concentration, the ratio of bicarbonate alkalinity to total alkalinity, and the ratio of VFA to total alkalinity can collectively offer rapid, dependable, and complementary insights

into system metabolism, promoting early warnings [14]. Regular process monitoring is indispensable for comprehending system status and performance, enabling timely detection and response to substantial imbalances and disruptions [15]. Detecting process imbalances early on is critical, facilitating timely intervention to prevent process failure. Hence, meticulous monitoring and consistent control of AD parameters are pivotal to ensure a seamless process flow and maximize biogas production [6].

At both laboratory and industrial scales, the AD process unfolds within specialized anaerobic digesters. These digesters harbor a group of anaerobic bacteria responsible for managing the degradation of organic materials. The dynamics, behavior, and interplay among these bacteria have been extensively explored within the literature. The exploration of these intricacies serves a dual purpose: initially, to comprehend the intricacies of the process itself and subsequently, to fine-tune it for optimization and global adoption.

A chance to enhance the efficiency and performance of the AD process involves the careful design and upscaling of appropriate reactors [6],[7],[8],[16],[17]. The configuration of the reactor and the surrounding environmental conditions, encompassing factors such as retention time, temperature, feedstock, and stirring, exert a profound influence on the behavior and composition of the distinct bacterial groups accountable for the degradation of organic materials. The AD process, as we already explained in this work, is a multi-step procedure. The subsequent stages of this journey exhibit a direct correlation with the solids retention time (SRT) within the digester [16],[18], as well as the mechanisms governing the retention of microorganisms within it [19],[16],[18]. Consequently, the selection of appropriate technologies depends on objectives, initial investments, and the characteristics of the biodegradable waste in question, including factors like its solids content and texture (high/low solids, pumpable/wet or stackable/dry substrate). Among the plethora of technologies, several reactors are presented below which are commonly cited in the literature. For an in-depth exploration of the technologies employed for AD applications, spanning thermophilic, mesophilic, and more, along with their respective performances, readers are directed to references [6],[7],[8],[16],[17],[18].

There many configurations of bioreactors for AD including Batch reactors, Semi-Continuous Fermenters (Fed/Sequencing Batch Reactors), and Continuous Bioreactors. Continuous Bioreactors further divide into Free Cells digesters (like Continuous Stirred Tank Reactors (CSTR)), Contact digesters (addressing limitations of CSTR), Biofilm and Granules digesters (including Fixed Bed reactors, Upflow Anaerobic Sludge Blanket (UASB) reactors, and Mobile Support reactors), and Two-Stage Reactors [20],[16]. These systems vary in feeding modes, biomass retention, and efficiency. Batch Reactors fill and empty, while Semi-Continuous fermenters manage gradual filling and decantation. Continuous bioreactors operate with constant feeding, categorized by different biomass-substrate interactions. Biofilm and Granules digesters excel in handling shocks, and Two-Stage reactors optimize methane production through separate stages. These bioreactors cater to different needs and challenges in AD processes, offering options for various scale and efficiency requirements.

As indicated in [20],[16] , irrespective of the specific type of reactor chosen, a crucial factor that demands the attention of designers in biogas facilities is the dilution rate (D) given by:

$$D = \frac{\text{The feed rate of the bioreactor } (F)}{\text{volume of the digester } (V)} (\frac{1}{day})$$

This value also stands as the reciprocal of HRT and should be adequate to facilitate the growth of various bacterial species.

1.1.2 Modeling and Optimizing Anaerobic Digestion Systems

Modelling and optimizing AD processes is crucial for designing wastewater treatment and biogas plants, studying how the plant's behavior responds to operational parameters, monitoring performance, and assessing the viability of utilizing diverse substrates under varying conditions. In the literature, numerous models exist, tailored for specific applications or fermenters that use particular substrates. These models can be broadly classified into two categories: mechanistic models, which are grounded in biological, physical, and chemical principles, and empirical machine learning models, which employ mathematical and statistical equations to establish relationships between input and output variables based on measured process data [21], [22], [23].

Mechanistic AD models

Mechanistic models represent a category of models that utilize a defined set of differential equations to elucidate the biological and physico-chemical principles governing a given process [22], [23]. These models which will be described in details in the following chapters aim to characterize the behavior of bacteria in response to the substrate, particularly focusing on their growth and inhibition dynamics. The development of a mechanistic model involves three key elements. Firstly, a deep understanding of the interplay between process variables is essential. Secondly, model parameters are determined through empirical data collected from experiments. Lastly, data collected during the actual process are crucial for validation. If the model falls short of efficiency, it undergoes revisions informed by process knowledge[22], [24], [23]. Mechanistic models can be broadly categorized into three groups based on complexity.

The simplest mechanistic models involve single-step processes with a solitary bacterial population and a sole limiting substrate. These models offer limited descriptions of inhibition effects, similar to chemostat models employing specific bacterial growth functions like Monod or Haldane [25], [26], [23]. Models of intermediate complexity, such as the AM2 model by Bernard et al. [5], encompass a higher number of processes and bacterial populations while providing more accurate descriptions of inhibition factors. Complex models, exemplified by Batstone et al. [1] and Stemann et al. [27], account for numerous processes, specific bacterial populations, and offer in-depth descriptions of inhibition effects and relevant

chemical equilibria. Among these, the Anaerobic Digestion Model no. 1 (ADM1), initially developed by the IWA Task Group for Mathematical Modelling [1], stands out. It can describe the anaerobic degradation of various substrates, with a significant complexity that makes it challenging to use for design and control purposes. ADM1 involves estimating a large number of parameters, approximately a hundred, which becomes particularly challenging in complex plant operations due to limited available data in the literature.

Hence, researchers have explored simpler models, some focused on fewer processes or tailored for specific substrates. Notably, the AM2 model by Bernard et al. [5] strikes a balance between simplicity and accuracy, primarily serving as a tool for monitoring and controlling AD processes, rather than precise numerical simulations [21].

Machine Learning AD Models

In contrast to the previously discussed models, machine learning models aim to capture system behavior without relying on prior knowledge of underlying processes. This approach integrates various fields, including mathematical modeling, statistics, information theory, and data science [28],[22], [23]. Machine learning tools prove valuable in uncovering the inherent structure of a process and exploring correlations among its components, all without prior assumptions [29], [23].

The machine learning modeling journey starts with the accumulation of historical data, forming a dataset consisting of examples, each characterized by a set of attributes or variables. Leveraging this dataset during the training phase, the model learns and evolves. Following training, a testing phase ensues, where the model's effectiveness is evaluated through classification, prediction, or clustering of new examples, known as test data. The model's performance is quantified using a performance measure, which advances as the model refines its understanding during data training [29], [30], [31], [32], [33], [34], [35], [36], [37]. These studies collectively demonstrate the diverse applications and effectiveness of different machine learning modeling techniques in optimizing AD processes and predicting biogas production. Several practical challenges can be identified in machine learning models, including the need for high-quality data, the opacity of machine learning models, and model reliability. Machine learning relies on ample, high-quality data, including explanatory metadata. Developing robust machine learning algorithms and methods for interpreting neural network decisions is crucial. [23]

As a summuray, we highlight that while mechanistic models offer comprehensive descriptions of AD processes, they come with notable weaknesses. These include the need for precise knowledge of bioreactor kinetic and stoichiometric parameters and the challenge of fully understanding AD microbial and physicochemical processes, making digital implementation complex. These limitations hinder optimization and control law synthesis for fermentation using such models. In contrast, machine learning models prove more effective in revealing process behavior and relationships among factors. They excel in estimating optimal conditions when system dynamics are not explicitly known and in analyzing nonlinear and complex interactions among process variables, [23].

1.1.3 Data-Driven Dynamical Systems

Dynamical systems offer a mathematical framework that captures the intricate interactions between evolving quantities over time. They involve the analysis, prediction, and comprehension of systems described by differential equations or iterative mappings, portraying how a system's state evolves. This formulation encompasses a vast array of phenomena, spanning classical mechanics, fluid dynamics, climate science, finance, social systems, and more. The genesis of modern dynamical systems traces back to Poincaré's work on planetary motion chaos, building on centuries of mathematical modeling starting with Newton and Leibniz. Its history is extensive, captivating brilliant minds across fields for centuries, and addressing numerous challenging problems [28].

Dynamical systems form a comprehensive field bridging linear algebra, differential equations, topology, numerical analysis, and geometry. They play a pivotal role in engineering, physical, and life sciences, facilitating modeling and analysis [28], [38]. Contemporary dynamical systems are embracing data-driven approaches, blending big data and machine learning. As governing equations remain elusive for many complex problems, data-driven analysis gains importance. Classical fields like optics, turbulence and biology are also shifting towards data-driven methods. Climate prediction, disease spread understanding, cognition study from neural data, and turbulence control are areas to benefit from data-driven insights.

Beyond classical perspectives, the emerging Koopman operator theory [39] provides a new approach, based on evolving measurements of a system. This theory capitalizes on increasing data availability from complex systems. Koopman theory offers a way to represent nonlinear dynamics in a linear framework, revolutionizing prediction and control of strongly nonlinear systems. As data-driven discovery progresses, dynamical systems are experiencing a renaissance, contributing significantly to scientific and engineering advancements.

Objectives and Obstacles in Recent Dynamical Systems

Given that dynamical systems typically serve as models for real-world occurrences, several top-priority objectives are linked to the examination of these systems:

- **Forecasting Future States:** In various scenarios, like meteorology and climatology, the aim is to forecast a system's future state. However, making predictions for extended periods remains a difficult task.
- **Designing and Optimizing:** We might aim to adjust system parameters to enhance performance or stability, as seen in actions like adding fins to a rocket for optimization.
- **Estimating and Controlling:** In many instances, dynamic systems can be actively controlled through feedback, utilizing system measurements to guide adjustments that alter behavior. Here, it's crucial to estimate the complete system state from limited measurements for effective control.

- **Physical Understanding and Interpretability:** A core objective of dynamical systems is to offer intrinsic comprehension and meaningful interpretation of a system's behavior by studying trajectories and solutions to its governing equations.

Real-world systems often display nonlinearities and multifaceted behavior across various scales. There's inherent uncertainty in equations of motion, parameter specifications, and measurements. Certain systems are more vulnerable to this uncertainty, requiring probabilistic methods. Additionally, deriving basic equations from first principles is increasingly challenging and indeterminate in some cases.

In the initial chapter of this thesis, some recent data-driven methods for identifying and studying dynamical systems will be presented. The outlined data-driven approaches primarily tackle two fundamental obstacles encountered in modern dynamical systems:

1. **Nonlinearity:** The presence of nonlinearity remains a fundamental obstacle in the examination and management of dynamical systems, leading to intricate global dynamics. Unlike linear systems which can be comprehensively characterized through spectral decomposition of matrix $\mathbf{A}$, nonlinear systems lack a comparable unified framework for prediction, estimation, and control. This pursuit for a general nonlinear framework stands as a significant mathematical challenge in the twenty-first century [28]. A prevalent approach to nonlinear dynamical systems revolves around the geometry of subspaces near fixed points and periodic orbits. This geometric theory, pioneered by Poincaré, has revolutionized complex systems of modeling, aided by theoretical underpinnings like the Hartman–Grobman theorem [40]. This allows for the application of linear analysis techniques in proximity to fixed points or periodic orbits. Nonetheless, while the geometric perspective offers local linear models, global analysis remains predominantly qualitative and computational, thereby constraining the theory of nonlinear prediction, estimation, and control beyond fixed points and periodic orbits [28], [41].

2. **Unknown Dynamics:** A significant challenge stems from the absence of established governing equations for numerous contemporary systems. As researchers delve into intricate realms like neuroscience, epidemiology, and ecology, the scarcity of fundamental laws obstructs equation derivation. This extends even to systems with known equations, such as turbulence, AD and protein folding, where identifying patterns in high-dimensional systems for revealing core behaviors remains elusive. While classical analysis involved ideal approximations and Newtonian equations, complex systems necessitate a shift towards data-driven techniques for equation discovery. As models become more intricate, their accuracy falters, demanding automated model discovery to unveil underlying mechanisms. Latent variables further complicate matters, posing a major challenge in data-driven approaches by concealing pertinent dynamics [42], [28].

Unveiling unknown dynamics through data analysis and acquiring intrinsic coordinates to linearize nonlinear systems stand as vital objectives in recent dynamical systems research. Surmounting the obstacles posed by unknown dynamics and nonlinearity holds the potential to revolutionize our comprehension of intricate systems, offering significant advantages across diverse scientific and engineering disciplines.

Throughout this thesis, we will delve into these concerns and elucidate several emerging methodologies aimed at tackling these complexities. Notably, two pivotal strategies are shaping contemporary data-driven dynamical systems exploration:

- **Operator-theoretic representations:** To grapple with nonlinearity, operator-theoretic approaches are gaining prominence. We will demonstrate that it's feasible to portray nonlinear dynamical systems using linear, infinite-dimensional operators like the Koopman operator featured in Chapter 2, which leverages measurement functions.
- **Data-driven regression and machine learning:** In the face of burgeoning data availability and the study of systems resistant to conventional analysis, regression and machine learning emerge as indispensable tools for unraveling dynamical systems from data. This underpins several techniques expounded in this thesis, including Dynamic Mode Decomposition (DMD) in Section 2.3, Extended Dynamic Mode Decomposition (EDMD) in Section 2.4, data-driven Koopman methods for control in Section 2.5.

It's worth highlighting that numerous methods and viewpoints presented in this thesis are interconnected, and the ongoing research endeavors focus on further enhancing and revealing these connections.

1.2 Objectives and Contributions

The primary objectives of this research encompass several key aspects within the field of AD, each aiming to address critical challenges and contribute novel insights to the field:

- Reducing the ADM1 model with regards to control objective.
- Identifying the most suitable data-driven methods for accurate modeling and prediction of AD systems.
- Learning kinetics reactions from simulated time-series data of AD processes.
- Developing robust predictive models for Chemostat dynamics under both Monod and Haldane kinetics through data-driven approaches using time-series data.
- Data-driven modeling and controlling of the Chemostat dynamics only with substrate measurements using the Koopman Operator theory.

The contributions arising from this research encompass diverse dimensions, enriching the domain of AD in meaningful ways:

- By adopting data-driven methodologies, this study offers an in-depth understanding of AD processes. This alternative approach sheds light on complex dynamics, revealing hidden patterns and enhancing comprehension beyond traditional modeling paradigms.
- The introduction of an alternative modeling perspective, grounded solely in data, contributes a novel methodology to the AD field. This approach challenges conventional modeling techniques, offering new ways to conceptualize and analyze AD processes.
- By exploring and rigorously testing a spectrum of data-driven techniques, this research advances the applicability of such methods within the AD domain. The study's thorough evaluation of these approaches lays the groundwork for future investigations and implementations.
- The utilization of the Koopman Operator theory in data-driven modeling and control of the Chemostat system constitutes a notable contribution. By showcasing the effectiveness of this theory, particularly in scenarios with limited measurements, the study pioneers a promising avenue for robust control strategies within AD systems.

1.3 Organization of the Thesis

This dissertation is structured as follows:

In Chapter 2, we delve into the intricate realm of the Koopman Operator and its implications in data-driven methods for control and analysis. We begin by providing a comprehensive overview of the Koopman Operator Theory. Through illustrative examples, we illustrate its significance and potential applications. Furthermore, we explore how the Koopman Operator Theory can be harnessed for control purposes, highlighting its role in shaping system behavior. Within this chapter, we also uncover the concept of Dynamic Mode Decomposition (DMD). We elucidate the DMD algorithm, which allows us to dissect complex dynamics into simpler modes. Real-world applications across various domains showcase the versatility and practical utility of DMD. Building upon the foundation of DMD, we delve into the realm of Extended Dynamic Mode Decomposition (EDMD). We introduce an advanced perspective on mode decomposition, potentially uncovering deeper insights and patterns within dynamic systems. The power of data-driven control strategies takes center stage in this chapter. We delve into Model Predictive Control, a technique that leverages data to optimize system behavior over a predictive horizon. Moreover, we explore Nonlinear System Identification, like DMD with control (DMDc) and EDMD with control (EDMEc) highlighting its role in enhancing control strategies for complex systems. In essence, this chapter serves as an exploration of the Koopman Operator Theory and its practical applications through data-driven methods. By understanding and utilizing these techniques,

we gain valuable tools for analyzing, controlling, and optimizing dynamic systems across diverse domains.

In Chapter 3, we delve into the application of DMD a data-driven approach for model order reduction of the ADM1 model. The chapter begins by unveiling the most complicated and detailed model in AD systems, the ADM1 model. With 35 states and over 80 parameters, this high-dimensional model defies conventional mathematical analysis. Nevertheless, it offers intricate insights into the AD process. We provide a comprehensive overview of the ADM1 model, underscoring its capacity to encompass all intricacies of the AD process and the presentation of the model's mathematical equations. Then, we use DMD to achieve model order reduction for the ADM1 model. This involves distilling the model's complexity while retaining essential dynamics.

Chapter 4 Extends the utility of DMD, we employ DMDc to model the two step anaerobic model (AM2). First, we introduce the AM2 model, a streamlined AD model with a mere six states, four of which are primary. We furnish a comprehensive exposition of this model, starting with its biological and kinetic reactions, and concluding with the deduction of its mathematical equations, presented in the form of a set of Ordinary Differential Equations (ODEs). The DMDc data-driven approach then allows us to capture AM2 model dynamics, potentially uncovering valuable insights and facilitating effective control strategies.

In Chapter 5, we focus on data-driven modeling, forecasting and controlling of the Chemostat dynamics. This chapter starts with the underpinning mathematical principles that underlie bioprocesses and biotechnology. Initially, we introduce the simplified chemostat featuring merely two state variables, elucidating our approach to deriving ODEs for the chemostat system. Specifically, we investigate the Continuous Stirred Tank Reactors (CSTR) with both Monod and Haldane kinetics. Then, we underscore the versatility of data-driven methods in understanding such systems behavior. Harnessing the Koopman Operator Theory, we explore data-driven control techniques for the chemostat system. We introduce a Koopman model that represents input-output chemostat dynamics. Furthermore, we implement Model Predictive Control based on the Koopman model (KMPC). Through simulation results and discussions, we gain insights into the efficacy of these data-driven control strategies.

In the concluding section of this thesis, we offer a comprehensive analysis of the findings derived from the undertaken research, in alignment with the initial objectives established. These findings are presented along with insightful interpretations, thereby encapsulating the essence of the accomplished work.

Furthermore, the culmination of this thesis encompasses the projection of prospective avenues and future directions. This projection of perspectives and forthcoming work serves to highlight potential areas for further exploration, extending the scope of knowledge and contributing to the ongoing advancement of the field.

Chapter 2

The Koopman Operator and Data-Driven Methods

2.1 Introduction

In this chapter, we provide a comprehensive overview of the Koopman operator theory. While our focus primarily lies in developing a functional understanding of the Koopman operator for the purpose of this work, interested readers are encouraged to refer to [39] for a more extensive review of the Koopman operator and its applications. Our exploration begins by introducing the Koopman operator for nonlinear dynamical systems. We delve into the underlying concepts and principles, emphasizing its significance as a mathematical tool in the study of dynamical systems. Named after Bernard Koopman, who introduced it in the 1930s, the Koopman operator enables a linear representation of system dynamics, making analysis and problem-solving more accessible. We then proceed to discuss data-driven methods employed to approximately deduce the Koopman operator from experimental or simulation data of the dynamical systems. These techniques allow us to extract valuable insights and understand the behavior of complex systems using the available data. Although the Koopman operator is inherently infinite-dimensional, practical approximations can be made to make it more manageable for real-world applications. These approximations play a crucial role in leveraging the power of the Koopman operator in various fields, ranging from control systems to machine learning and beyond.

2.2 Overview of the Koopman Operator Theory

In this thesis, we consider the following autonomous continuous-time dynamical system:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}), \tag{2.1}$$

where $\mathbf{x}(t) \in \mathcal{M} \subseteq \mathbb{R}^n$ is the system's state that may exist on a submanifold $\mathcal{M}$ of an n-dimensional vector space $\mathbb{R}^n$, and $\mathbf{f(.)}$ represents a vector field that describes

the dynamics. Researchers strive to find a new vector of coordinates $\mathbf{s}$ of either

$$\mathbf{x} = \boldsymbol{\psi}(\mathbf{s}) \quad \text{or} \quad \mathbf{s} = \boldsymbol{\psi}(\mathbf{x}) \tag{2.2}$$

where we have simplified or potentially linearized dynamics:

$$\dot{\mathbf{s}} = \mathbf{L}\mathbf{s}. \tag{2.3}$$

The dynamics of $\mathbf{s}$ in these new linearizing coordinates are completely governed by the matrix $\mathbf{L}$. The eigen-decomposition of $\mathbf{L}$ can completely describe how the system will evolve in these coordinates in the future. The embeddings $\boldsymbol{\psi}$ might elevate the dynamics to a space of higher dimensions of $\mathbf{s}$ variables which enables the nonlinearities to be unraveled. In practical applications, we generally obtain *measurement* data that records the behavior of our system at discrete time points. This data is handled by a discrete-time dynamical system given by

$$\mathbf{x}_{k+1} = \mathbf{F}(\mathbf{x}_k) \tag{2.4}$$

where $\mathbf{x}_k = \mathbf{x}(t_k) = \mathbf{x}(k\Delta t)$ and $\mathbf{F}$ is denoted by the discrete-time *flow* map that evolves (2.1) such as:

$$\mathbf{F}(\mathbf{x}(t_0)) = \mathbf{x}(t_0 + t) + \int_{t_0}^{t_0+t} \mathbf{f}(\mathbf{x}(\boldsymbol{\tau}))\mathbf{d}\boldsymbol{\tau}, \tag{2.5}$$

Compared to the formulation of the continuous-time system in (2.1), the discrete-time dynamics are more general and also incorporate discontinuous and hybrid systems. In this situation, the objective is still to discover a linearizing coordinate transform in order to have

$$\mathbf{s}_{k+1} = \mathbf{K}\mathbf{s}_k. \tag{2.6}$$

where $\mathbf{K}$ is the discrete-time version of continuous-time matrix $\mathbf{L}$. To that end, B.O. Koopman in 1931 [43] introduced a way to view systems through infinite-dimensional linear operators $\mathcal{K}_t$, which can advance all measurement functions $g : \mathcal{M} \to \mathbb{C}$ with the flow of the dynamics, in order to have:

$$\mathcal{K}_t g(\mathbf{x}_k) = g(\mathbf{F}(\mathbf{x}_k)) = g(\mathbf{x}_{k+1}), \tag{2.7}$$

An alternative representation is given by:

$$\mathcal{K}_t g = g \circ \mathbf{F}, \tag{2.8}$$

where $\circ$ is the composition operation. The linear operator $\mathcal{K}_t$ acts on the Hilbert space $\mathcal{H}$ of scalar measurement functions g and provides a linear representation of the dynamics of the system. In the context of the Koopman operator, the scalar measurement function g is often referred to as an *observable function*. This terminology comes from the operator's origins in quantum mechanics. In the context of the Koopman operator, the observable function is a real-valued function that represents a physical quantity of interest in the system. However, it is important

to note that the terms *observables* or *observable functions* should not be confused with the term *observability* used in control theory. The Koopman operator $\mathcal{K}_t$ can be thought of as a way of trading nonlinear, finite-dimensional dynamics for linear, infinite-dimensional dynamics. This can be beneficial because linear differential equations can often be solved using the spectral representation, which can provide a more complete understanding of the system's behavior. However, because the Koopman operator is an infinite-dimensional operator, it can be challenging to work with in practice. Therefore, in practice, a sufficiently large, but finite, sum of modes is used to approximate the Koopman spectral solution. This approximation can provide an accurate representation of the dynamics of the system without the challenges associated with infinite-dimensional operators [39, 38].

When a dynamical system is sufficiently smooth, it is possible to define a continuous-time version of the Koopman dynamical system in (2.7):

$$\frac{d}{dt}g = \mathcal{K}g, \tag{2.9}$$

where the operator $\mathcal{K}$ describes the rate of change of the Koopman operators $\mathcal{K}_t$ with respect to the parameter t, also known as the continuous-time infinitesimal generator of the one-parameter of Koopman operator family [39, 38].

The eigenfunction $\boldsymbol{\psi}(\mathbf{x})$ of the discrete-time Koopman operator $\mathcal{K}_t$ in (2.7) corresponding to an eigenvalue λ is given by:

$$\boldsymbol{\psi}(\mathbf{x}_{k+1}) = \mathcal{K}_t\boldsymbol{\psi}(\mathbf{x}_k) = \lambda\boldsymbol{\psi}(\mathbf{x}_k). \tag{2.10}$$

where the Koopman eigenfunctions $\boldsymbol{\psi}(\mathbf{x})$ are actually the linearizing coordinates transform that allow the nonlinear system in (2.4) to be written as the linear form in (2.6). The Koopman eigenfunctions $\boldsymbol{\psi}(\mathbf{x})$ can be written in a continuous-time formulation as follows:

$$\frac{d}{dt}\boldsymbol{\psi}(\mathbf{x}) = \mathcal{K}\boldsymbol{\psi}(\mathbf{x}) = \lambda\boldsymbol{\psi}(\mathbf{x}). \tag{2.11}$$

Modern dynamical systems have a major applied difficulty in obtaining Koopman eigenfunctions from analytical formulations or from data. Globally linear representations of severely nonlinear systems are made possible by the discovery of these eigenfunctions. The time derivative of the Koopman eigenfunction $\boldsymbol{\psi}(\mathbf{x})$, when the chain rule is applied, results in

$$\frac{d}{dt}\boldsymbol{\psi}(\mathbf{x}) = \nabla\boldsymbol{\psi}(\mathbf{x}) \cdot \dot{\mathbf{x}} = \nabla\boldsymbol{\psi}(\mathbf{x}) \cdot \mathbf{f}(\mathbf{x}). \tag{2.12}$$

This leads to a partial differential equation (PDE) for the eigenfunction $\boldsymbol{\psi}(\mathbf{x})$ when taking in consideration equation in (2.11):

$$\nabla\boldsymbol{\psi}(\mathbf{x}) \cdot \mathbf{f}(\mathbf{x}) = \lambda\boldsymbol{\psi}(\mathbf{x}). \tag{2.13}$$

The given nonlinear PDE rises the possibility of the approximation of the Koopman eigenfunctions $\boldsymbol{\psi}(\mathbf{x})$ through regression using data or by solving for the Laurent series [38]. It is important to note that this approach assumes that the dynamics are both continuous and differentiable. In simpler terms, equations (2.10)

and (2.11) show that the Koopman operator can make the complex behavior of a system simpler and more understandable by transforming it into linear dynamics in terms of eigenfunction coordinates $\boldsymbol{\psi}(\mathbf{x})$. It also shows that certain functions, like conserved quantities (e.g., Hamiltonian energy function), can be expressed as eigenfunctions of the Koopman operator with zero eigenvalues [43, 38].

2.2.1 Illustrative Examples

A system that has a finite-dimensional Koopman operator.

If the right set of observables is selected, certain systems have a configuration that results in a closed Koopman subspace. In this case, we examine a sample system from Tu et al. [44] which has only one stationary point. This system is further analyzed in Brunton et al. [45]. Here is the system:

$$\begin{aligned} \dot{\mathbf{x}}_1 &= \boldsymbol{\mu}\mathbf{x}_1, \\ \dot{\mathbf{x}}_2 &= \boldsymbol{\lambda}(\mathbf{x}_2 - \mathbf{x}_1^2), \end{aligned} \tag{2.14}$$

By adding the nonlinear measurement $g = \mathbf{x}_1^2$ to the state $\mathbf{x}$, it is feasible to establish a Koopman invariant subspace with three dimensions. The dynamics of the system become linear when expressed in these coordinates:

$$\begin{bmatrix} \dot{\mathbf{z}}_1 \\ \dot{\mathbf{z}}_2 \\ \dot{\mathbf{z}}_3 \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{\mu} & 0 & 0 \\ 0 & \boldsymbol{\lambda} & -\boldsymbol{\lambda} \\ 0 & 0 & 2\boldsymbol{\mu} \end{bmatrix}}_{\mathcal{K}} \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \mathbf{z}_3 \end{bmatrix} \quad \textbf{for} \quad \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \mathbf{z}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_1^2 \end{bmatrix} \tag{2.15}$$

Where $\mathcal{K}$ represents the Koopman operator of the system in (2.14). This is highly connected to the process of linearizing a system using the Carleman linearization technique [46, 47, 48, 49, 50, 51]. It is obvious that since $\mathbf{z}_3 = \mathbf{x}_1^2$ than $\dot{\mathbf{z}}_3 = 2\mathbf{x}_1\dot{\mathbf{x}}_1 = 2\boldsymbol{\mu}\mathbf{x}_1^2 = 2\boldsymbol{\mu}\mathbf{z}_3$, and the set of observables ($\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_1^2$) create a subspace that is closed under differentiation to produce a Koopman operator with finite dimensions.

A system that has an infinite-dimensional Koopman operator.

The majority of systems lack a configuration that permits a Koopman operator with finite dimensions. Instead, let's consider this particular system:

$$\dot{\mathbf{x}} = \mathbf{x}^2, \tag{2.16}$$

If we attempt to create observables in a straightforward manner to generate a Koopman operator for this particular system, a logical option (inspired from the previous example) is to begin with the following:

$$\begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{x} \\ \mathbf{x}^2 \end{bmatrix} \tag{2.17}$$

Nevertheless, if we differentiate $\mathbf{z}_2 = \mathbf{x}^2$ we get $\dot{\mathbf{z}}_2 = 2\mathbf{x}\dot{\mathbf{x}} = 2\mathbf{x}^3$, we will be required to introduce $\mathbf{z}_3 = \mathbf{x}^3$, and as we add more observables, the differentiation process will lead to the exponent of each power being incremented by one.

$$\begin{bmatrix} \dot{\mathbf{z}}_1 \\ \dot{\mathbf{z}}_2 \\ \dot{\mathbf{z}}_3 \\ \dot{\mathbf{z}}_4 \\ \vdots \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots \\ 0 & 0 & 2 & 0 & \cdots \\ 0 & 0 & 0 & 3 & \cdots \\ 0 & 0 & 0 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \mathbf{z}_3 \\ \mathbf{z}_4 \\ \vdots \end{bmatrix} \quad \textbf{where} \quad \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \mathbf{z}_3 \\ \mathbf{z}_4 \\ \vdots \end{bmatrix} = \begin{bmatrix} \mathbf{x} \\ \mathbf{x}^2 \\ \mathbf{x}^3 \\ \mathbf{x}^4 \\ \vdots \end{bmatrix} \tag{2.18}$$

This will lead to an unbounded chain of higher powers being necessary to fully define the derivative of $\mathbf{z}$. Many systems require an endless number of observables, which causes the associated Koopman operator to be infinite in size. This is a frequent occurrence in various systems, and hence, the practical use of the Koopman operator depends on discovering accurate approximations with finite dimensions from the available data. In addition, and in such scenarios, the approach should be to seek out the eigenfunctions of the Koopman operator for the nonlinear dynamical system in (2.16).

Let us propose the following function:

$$\boldsymbol{\psi}(\mathbf{x}) = e^{-1/\mathbf{x}}, \tag{2.19}$$

We can easily remark that

$$\frac{d}{dt}\boldsymbol{\psi}(\mathbf{x}) = \mathbf{x}^{-2}\, e^{-1/\mathbf{x}}\, \dot{\mathbf{x}} = e^{-1/\mathbf{x}} = \boldsymbol{\psi}(\mathbf{x}), \tag{2.20}$$

where $\dot{\mathbf{x}} = \mathbf{x}^2$. Since $\boldsymbol{\psi}(\mathbf{x}) = e^{-1/\mathbf{x}}$ verifies the PDE in (2.13), $\boldsymbol{\psi}(\mathbf{x}) = e^{-1/\mathbf{x}}$ can be considered as a magic eigen-measurements of the Koopman operator (eigenfunction) that if we measure the nonlinear dynamical system in (2.16) on $\boldsymbol{\psi}(\mathbf{x})$ the system behavior will remain linear. Similar Koopman eigenfunctions can possibly be deduced analytically by solving the PDE in (2.13) using typical methods which involve recursively finding the components of a Taylor or Laurent series [38].

2.2.2 Koopman Operator Theory for Control

Initially, in [52], it was observed that the Koopman operator theory could be extended to actuated systems by interpreting stochastic forcing in random dynamical systems as actuation. However, the first Koopman-based control techniques were not introduced until more than ten years later, mainly driven by the algorithmic progress of DMD [53]. Koopman models have lately been used more frequently in conjunction with linear quadratic regulator (LQR) [45, 54], state-dependent LQR [55], and model predictive control (MPC) [56, 57], among other applications such as optimal control for switching control problems [58], Lyapunov-based stabilization [39, 59], eigenstructure assignment [60], and active learning [61]. MPC [62, 63, 64, 65, 66, 67] is particularly noteworthy as a primary enabler of Koopman-based control success, with a wide range of applications such as power grids [68], high-dimensional fluid flows [69], and electrical drives [70].

Koopman operator theory for control necessitates the separation of unforced dynamics from the influence of actuation. The earliest Koopman-based techniques were created for discrete-time systems, which are more broad and constitute a superset of those caused by continuous-time dynamics. These discrete-time dynamics are frequently more compatible with experimental measurements and actuation, and they may be preferable for numerical analysis [38].

Consider the controlled discrete-time dynamical system

$$\mathbf{x}_{k+1} = \mathbf{F}(\mathbf{x}_k, \mathbf{u}_k), \tag{2.21}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the state and $\mathbf{u} \in \mathcal{U} \subseteq \mathbb{R}^{n_u}$ is the control input of the system. For actuated systems, Koopman operator can be generalized by describing it as the Koopman operator related to uncontrolled dynamical system, which emerges in an extended state-space $\tilde{\mathbf{x}} := [\mathbf{x}^\top, \mathbf{u}^\top]^\top$ defined as a product of the original state-space and space for all control sequences, for instance, $\mathbb{R}^n \times \mathcal{L}(\mathcal{U})$, with $\mathcal{L}(\mathcal{U})$ is the space of all sequences $\mathbf{u} := (u_i)_{i=0}^{\infty}$ where $u_i \in \mathcal{U}$. The dynamics of the extended state $\tilde{\mathbf{x}}$ is given as :

$$\tilde{\mathbf{x}}_{k+1} = \tilde{\boldsymbol{F}}(\tilde{\mathbf{x}}_k) := (\mathbf{F}(\mathbf{x}_k, \mathbf{u}(0)), \mathcal{T}\mathbf{u}), \tag{2.22}$$

where $\mathcal{T} : \mathcal{L}(\mathcal{U}) \longrightarrow \mathcal{L}(\mathcal{U})$ is the left shift operator, where $(\boldsymbol{\mathcal{T}}\mathbf{u})(i) = \mathbf{u}(i+1)$, $i \in \mathbb{N}$. Let $g(\mathbf{x}, \mathbf{u}) : \mathbb{R}^n \times \mathcal{L}(\mathcal{U}) \to \mathbb{R}$ a scalar-valued observation functions, but unlike before, these functions now depend on both the state and input. Every observable function belongs to a Hilbert space $\mathcal{H}$ that has an infinite number of dimensions. The Koopman operator with inputs and control $\mathcal{K}_t : \mathcal{H} \longrightarrow \mathcal{H}$ associated with (2.22) which acts on the Hilbert space of observable functions is presented as:

$$\mathcal{K}_t g(\tilde{\mathbf{x}}_k) = g(\tilde{\boldsymbol{F}}(\tilde{\mathbf{x}}_k)) = g(\tilde{\mathbf{x}}_{k+1}), \tag{2.23}$$

The Koopman operator $\mathcal{K}_t$ in (2.23) is a linear operator that fully explains the nonlinear dynamical system in (2.21) as long as the state components of $\mathbf{x}_k$ are included in $\mathcal{H}$. In other words, the properties of the Koopman operator $\mathcal{K}_t$, such as its spectral properties, can give us insights into the spectral properties of the nonlinear dynamical system in (2.21). For a detailed version of how to extend the Koopman operator for controlled system, see [56, 71, 38, 39]. It is important to note that the definition of the Koopman operator assumes that $\mathcal{H}$ remains unchanged when acted upon by $\mathcal{K}_t$. Therefore, in a controlled environment, $\mathcal{H}$ will generally include functions that depend on $\mathbf{u}$ automatically.

2.3 Dynamic Mode Decomposition

Dynamic mode decomposition (DMD) was first introduced to the fluid dynamics community by P. Schmid [72, 73, 74, 75] as a data-driven algorithm to identify coherent spatio-temporal structures from high-dimensional time-series data, which is typical in fluid dynamics. Since then, DMD has become the standard algorithm to estimate the Koopman operator from data [76, 44, 75]. The first

link between DMD and the Koopman operator was established by Rowley et al. [76]. DMD is based on the efficient singular value decomposition (SVD) or proper orthogonal decomposition (POD) in fluid dynamics and, therefore, can perform scalable dimensionality reduction for high-dimensional data. SVD hierarchically orders modes based on the variance of the original data that each mode captures. These modes remain constant even when the data's order is changed over time. In contrast, DMD modes are linear combinations of SVD modes that are specially selected to extract spatially correlated structures that exhibit coherent linear behavior over time, indicated by oscillations with constant frequency with growth or decay. Therefore, DMD provides dimensionality reduction using a low-dimensional set of spatial modes and a linear model for the modes' amplitudes' time evolution. DMD combines SVD/POD in space with the Fourier transform in time, taking advantage of each method's strengths [77, 75].

DMD variants, including DMD with control and delay DMD, are related to subspace system identification methods that have been around for decades. Modern Koopman theory provides a new interpretation for these methods when applied to nonlinear systems. DMD and Koopman-based approaches are suited for high-dimensional systems, and DMD's wide use is due to its efficiency in processing high-dimensional spatio-temporal data. [38]

DMD is applicable to both experimental and simulated data and does not require knowledge of governing equations, making it highly versatile. Its linear algebra formulation also makes it easily extendable, leading to innovations in control, compressed sensing, and multi-resolution, among others. As a result, DMD has been widely used in various applications beyond fluid mechanics, such as neuroscience, robotics, and finance.[75, 39]

2.3.1 The DMD Algorithm

The DMD algorithm approximates the best-fit linear matrix operator that advances high-dimensional measurements of a system forward in time [44]. Therefore, DMD approximates the Koopman operator restricted to the measurement subspace given by direct measurements of the system's state. The goal of the DMD algorithm is to find a linear matrix operator $\mathbf{A}$ that best represents the state of a system, denoted by $\mathbf{x} \in \mathbb{R}^n$, as it evolves forward in time in accordance with the following linear dynamical system

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k, \tag{2.24}$$

where $\mathbf{x}_k = \mathbf{x}(k\Delta t)$ with $k = 1, 2, 3...m$ and Δt is a fixed time step that is chosen to be small enough to accurately capture the fastest changes or oscillations (i.e., the highest frequencies) in the dynamics of the system. DMD is basically a data-driven algorithm, and the matrix $\mathbf{A}$ is estimated using a collection of system snapshot pairs $\{\mathbf{x}(t_k), \mathbf{x}(t_k + \Delta t)\}_{k=1}^{m}$, where in the context of dynamic systems, a *snapshot* refers to a measurement or observation of the full state of a system at a specific point in time. This measurement is typically represented as a high-dimensional column vector, where each element of the vector represents a different aspect or variable of the system. In the standard DMD formulation [72], data are required

to be generated from a single trajectory with consistent sampling in time, so that $t_k = k\Delta t$. However, here we present the *exact* DMD algorithm presented in [44], since it works for unevenly spaced data and data joined from various time series. Hence, the moments t_k do not have to be consecutive or uniformly spaced, but for each snapshot $\mathbf{x}(t_k)$, there is a matching snapshot $\mathbf{x}(t_k + \Delta t)$ that is forwarded by one Δt in the future. The snapshots are organised into big data matrices $\mathbf{X}$ and $\mathbf{X}^{'}$:

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \mathbf{x}(t_1) & \mathbf{x}(t_2) & \dots & \mathbf{x}(t_{m-1}) \\ | & | & & | \end{bmatrix}, \tag{2.25}$$

$$\mathbf{X}^{'} = \begin{bmatrix} | & | & & | \\ \mathbf{x}(t_2) & \mathbf{x}(t_3) & \dots & \mathbf{x}(t_m) \\ | & | & & | \end{bmatrix}, \tag{2.26}$$

The dynamical system in (2.24) can be expressed in terms of these matrices and transformed into the following:

$$\mathbf{X}^{'} \approx \mathbf{A}\mathbf{X}, \tag{2.27}$$

The best-fitting matrix $\mathbf{A}$ builds a linear dynamical system that basically shifts snapshot measurements ahead in time, which may be expressed as the following optimization problem:

$$\mathbf{A} = \underset{\mathbf{A}}{\operatorname{argmin}} \|\mathbf{X}^{'} - \mathbf{A}\mathbf{X}\|_F = \mathbf{X}^{'}\mathbf{X}^{\dagger}, \tag{2.28}$$

Where $\dagger$ is the Moore-Penrose pseudo-inverse and $\|.\|_F$ is the Frobenius norm[1] of a matrix. Using the SVD of $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$ the pseudo-inverse of $\mathbf{X}$ can be calculated as $\mathbf{X}^{\dagger} = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^*$ where the matrices $\mathbf{U} \in \mathbb{C}^{n\times n}$, and $\mathbf{V}^{m\times m}$ are unitary which means $\mathbf{U}^*\mathbf{U} = \mathbf{I}$ and $\mathbf{V}^*\mathbf{V} = \mathbf{I}$, and * represents the complex-conjugate transpose. $\mathbf{\Sigma}^{n\times m}$ is a diagonal matrix containing the singular values. The columns of $\mathbf{U}$ are usually called the POD modes as in [78, 79].

Since $\mathbf{A}$ is an approximate representation of the Koopman operator constrained to a finite-dimensional subspace of linear measurements, we are frequently focused on the eigenvalues $\mathbf{\Lambda}$ and eigenvectors $\mathbf{\Phi}$ of $\mathbf{A}$ given:

$$\mathbf{A}\mathbf{\Phi} = \mathbf{\Phi}\mathbf{\Lambda}, \tag{2.29}$$

However, when dealing with high-dimensional data, the matrix $\mathbf{A}$ can become impractical to represent and compute due to its large size of $n \times n$ elements. Therefore, the DMD algorithm aims to find the leading spectral decomposition of $\mathbf{A}$ (i.e., eigenvalues and eigenvectors) without explicitly constructing it. Rather than calculating $\mathbf{A}$ using equation (2.28), we can project $\mathbf{A}$ onto the first r POD modes in $\mathbf{U}_r$, and estimate the pseudo-inverse using the rank-r SVD approximation

[1] The Frobenius norm of a matrix A is given by $\| A \|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2}$

$\mathbf{X} \approx \mathbf{U}_r \mathbf{\Sigma}_r \mathbf{V}_r^*$:

$$\tilde{\mathbf{A}} = \mathbf{U}_r^* \mathbf{A} \mathbf{U}_r \tag{2.30}$$
$$= \mathbf{U}_r^* \mathbf{X}^{'} \mathbf{X}^{\dagger} \mathbf{U}_r \tag{2.31}$$
$$= \mathbf{U}_r^* \mathbf{X}^{'} \mathbf{V}_r \mathbf{\Sigma}_r^{-1} \mathbf{U}_r^* \mathbf{U}_r \tag{2.32}$$
$$= \mathbf{U}_r^* \mathbf{X}^{'} \mathbf{V}_r \mathbf{\Sigma}_r^{-1}, \tag{2.33}$$

One can approximate the leading eigenvalues and eigenvectors of $\mathbf{A}$ by using the spectral decomposition of the considerably reduced matrix $\tilde{\mathbf{A}}$:

$$\tilde{\mathbf{A}} \mathbf{W} = \mathbf{W} \mathbf{\Lambda}, \tag{2.34}$$

The DMD eigenvalues of the diagonal matrix $\mathbf{\Lambda}$ are consistent with the eigenvalues of the high-dimensional matrix $\mathbf{A}$. The eigenvectors of $\tilde{\mathbf{A}}$ that form the columns of $\mathbf{W}$ act as a coordinate transformation to diagonalize the matrix. These columns can be interpreted as linear combinations of POD mode amplitudes that follow a single temporal pattern determined by the appropriate eigenvalue λ and act linearly.

Since $\tilde{\mathbf{A}}$ is a r rank where $r \ll n$ and it defines a reduced order linear model of the original dynamical system on POD coordinates:

$$\tilde{\mathbf{x}}_{k+1} = \tilde{\mathbf{A}} \tilde{\mathbf{x}}_k, \tag{2.35}$$

It allows for the reconstruction of the full-state system dynamics using:

$$\mathbf{x}_k = \mathbf{U} \tilde{\mathbf{x}}_{\boldsymbol{k}}, \tag{2.36}$$

The reduced system's eigenvectors $\mathbf{W}$ and the time-shifted data matrix $\mathbf{X}^{'}$ are used to reconstruct the eigenvectors of $\mathbf{A}$, that represent the DMD modes $\mathbf{\Phi}$ by using the following formula:

$$\mathbf{\Phi} = \mathbf{X}^{'} \mathbf{V} \mathbf{\Sigma}^{-1} \mathbf{W}, \tag{2.37}$$

The DMD algorithm is summarized in Algorithm 1. Tu et al. [44] have shown that the DMD modes correspond to the eigenvectors of the full matrix $\mathbf{A}$, provided that certain conditions are met (these conditions include the assumption that the data matrices $\mathbf{X}$ and $\mathbf{X}^{'}$ have full column rank, and that the snapshots are collected at equally spaced time intervals). Under these conditions, the DMD modes can be computed efficiently and accurately, and can provide a useful low-dimensional representation of the high-dimensional dynamics of the system. Numerous open-source implementations of the DMD algorithm are available, which can be used to perform DMD analysis on various types of data. These implementations may offer different features and capabilities, and can be used for various applications in fields such as fluid dynamics, image processing, and machine learning. Some popular open-source DMD implementations include PyDMD, DMDc, and DMDLIVE, which are available as Python packages and can be easily installed and used. Additionally, several software packages such as MATLAB and SciPy also offer built-in DMD functions that can be used for DMD analysis [75, 53, 80].

Algorithm 1: The Exact DMD Algorithm as in [75]

Data: Collect and arrange snapshots data from a system into matrices

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}(t_1) & \mathbf{x}(t_2) & \dots & \mathbf{x}(t_{m-1}) \end{bmatrix}, \quad \mathbf{X}' = \begin{bmatrix} \mathbf{x}(t_2) & \mathbf{x}(t_3) & \dots & \mathbf{x}(t_m) \end{bmatrix}$$

Step 1: Compute the reduced SVD of the matrix $\mathbf{X}$:

$$\mathbf{X} \approx \mathbf{U}_r \mathbf{\Sigma}_r \mathbf{V}_r^*$$

Step 2: Define the reduced order linear matrix $\tilde{\mathbf{A}}$:

$$\tilde{\mathbf{A}} = \mathbf{U}_r^* \mathbf{X}' \mathbf{V}_r \mathbf{\Sigma}_r^{-1}$$

Step 3: Compute the eigen-decomposition of $\tilde{\mathbf{A}}$:

$$\tilde{\mathbf{A}} \mathbf{W} = \mathbf{W} \mathbf{\Lambda},$$

Result: Compute the DMD modes of the big matrix $\mathbf{A}$:

$$\mathbf{\Phi} = \mathbf{X}' \mathbf{V} \mathbf{\Sigma}^{-1} \mathbf{W}$$

After the DMD modes and eigenvalues have been calculated, the system state may be expressed in terms of the DMD expansion:

$$\mathbf{x}_k = \sum_{j=1}^{r} \phi_j \lambda_j^{k-1} b_j = \mathbf{\Phi} \mathbf{\Lambda}^{k-1} \mathbf{b}, \tag{2.38}$$

where ϕ_j and λ_j are DMD modes and DMD eigenvalues respectively (eigenvectors and eigenvalues of $\mathbf{A}$), and b_j are the DMD mode amplitudes. The mixture of the DMD amplitudes $\mathbf{b}$ can be deduced using the first snapshot $\mathbf{x}_1$ of the big matrix $\mathbf{X}$:

$$\mathbf{b} = \mathbf{\Phi}^{\dagger} \mathbf{x}_1, \tag{2.39}$$

By incorporating the continuous eigenvalues $\omega = log(\lambda)/\Delta t$, the spectral expansion in (2.38) may be transformed to continuous time and written as follows:

$$\mathbf{x}(t) = \sum_{j=1}^{r} \phi_j e^{\omega_j t} b_j = \mathbf{\Phi} \exp(\mathbf{\Omega} t) \mathbf{b}, \tag{2.40}$$

where the continuous-time eigenvalues ω_j are represented by the diagonal matrix $\mathbf{\Omega}$. The data matrix $\mathbf{X}$ may therefore be written as:

$$\mathbf{X} \approx \begin{bmatrix} | & & | \\ \phi_1 & \dots & \phi_r \\ | & & | \end{bmatrix} \begin{bmatrix} b_1 & & \\ & \ddots & \\ & & b_r \end{bmatrix} \begin{bmatrix} e^{\omega_1 t_1} & \dots & e^{\omega_1 t_m} \\ \vdots & \ddots & \vdots \\ e^{\omega_r t_1} & \dots & e^{\omega_r t_m} \end{bmatrix} \tag{2.41}$$

2.3.2 Domain applications

The versatility of DMD lies in its ability to extract dynamic features from time-varying data without requiring prior knowledge of the underlying system's equations. This leads to an the application of DMD to increasingly diverse array of problems, including finance [81], plasma physics [82] and robotics [83]. It originated in fluid dynamics research [73, 72, 75], where it has been applied to study mixing, acoustics, combustion, and various flow geometries, such as jets [84, 85, 86], cavity flows [87, 88] , boundary layers [89, 90, 91], wakes[92, 93, 94, 95], and shock-turbulent boundary layer interactions[96]. DMD has also been applied to investigate epidemiological systems [97, 53], where modal frequencies often correspond to yearly or seasonal fluctuations, and the phase of DMD modes provides insight into how disease fronts propagate spatially. In neuroscience, DMD has the potential to transform the analysis of neural recordings, as evidenced by a recent study that identified dynamically relevant features in electrocorticography (ECoG) data of sleeping patients [98, 99, 100]. DMD has also been used in video processing to separate foreground and background objects, providing a flexible platform for real-time video separation [101, 102, 103].

2.4 Extended Dynamic Mode Decomposition

The Dynamic Mode Decomposition method presented in Section 2.3 is widely used for approximating the Koopman operator from data [76, 44, 75, 38], but it has limitations in identifying nonlinear changes of coordinates needed for strongly nonlinear systems because it relies on linear measurements. DMD alone may not be sufficient to capture crucial features of nonlinear systems, such as multiple fixed points and transients. To address this limitation, Williams et al.[3, 104, 105] introduced the Extended Dynamic Mode Decomposition (EDMD) method. The EDMD is a popular approach for identifying and representing Koopman embeddings from data. EDMD improves DMD by using an augmented vector that includes nonlinear measurements of the system's state. This approach allows the best-fit linear DMD regression to be performed on the augmented vector, enabling better approximation of the Koopman operator for strongly nonlinear systems.

More specifically, let $\varphi_k(\mathbf{x})$ be a set of nonlinear observable functions or measurements of the state $\mathbf{x}$. The augmented state vector $\mathbf{s}$ is defined as:

$$\mathbf{s} = \boldsymbol{\varphi}^\top(\mathbf{x}) = \begin{bmatrix} \varphi_1(\mathbf{x}) \\ \varphi_2(\mathbf{x}) \\ \vdots \\ \varphi_p(\mathbf{x}) \end{bmatrix} \tag{2.42}$$

where $\mathbf{s} \in \mathbb{R}^p$ is a vector that contains the original state $\mathbf{x}$ as well as nonlinear measurements such that $p \gg n$. Two data matrices are produced in the same way as previously described in DMD:

$$\mathbf{S} = \begin{bmatrix} | & | & & | \\ \mathbf{s}_1 & \mathbf{s}_2 & \dots & \mathbf{s}_{m-1} \\ | & | & & | \end{bmatrix}, \tag{2.43}$$

$$\mathbf{S}^{'} = \begin{bmatrix} | & | & & | \\ \mathbf{s}_2 & \mathbf{s}_3 & \dots & \mathbf{s}_{m} \\ | & | & & | \end{bmatrix}, \tag{2.44}$$

with $\mathbf{s}_k = \boldsymbol{\varphi}(\mathbf{x}(t_k))$. A best-fit linear matrix operator $\mathbf{A_S}$ that projects $\mathbf{S}$ into $\mathbf{S}^{'}$ is created, much like in DMD:

$$\mathbf{A_S} = \underset{\mathbf{A_S}}{\operatorname{argmin}} \|\mathbf{S}^{'} - \mathbf{A_S}\mathbf{S}\|_F = \mathbf{S}^{'}\mathbf{S}^{\dagger}, \tag{2.45}$$

Then, this regression may be expressed using the initial data matrices $\mathbf{X}$ in (2.25) and $\mathbf{X}^{'}$ in (2.26) used in DMD:

$$\mathbf{A_S} = \underset{\mathbf{A_S}}{\operatorname{argmin}} \|\boldsymbol{\varphi}^{\top}(\mathbf{X}^{'}) - \mathbf{A_S}\boldsymbol{\varphi}^{\top}(\mathbf{X})\|_F = \boldsymbol{\varphi}^{\top}(\mathbf{X}^{'})(\boldsymbol{\varphi}^{\top}(\mathbf{X}))^{\dagger}, \tag{2.46}$$

When dealing with large data sets, kernel methods [104, 39, 56, 106, 107] are often used to perform regression analysis as the resulting augmented vector $\mathbf{s}$ can be significantly larger than the original state $\mathbf{x}$. By using an enriched library $\boldsymbol{\varphi}$, a larger basis is created to approximate the Koopman operator. Recent studies have shown that, when the number of snapshots is in the limit of infinite, the EDMD operator converges to the Koopman operator projected onto the subspace spanned by the original data $\boldsymbol{\varphi}$ [108]. Even though it is recommended to use validation and cross-validation techniques to prevent EDMD models from overfitting in some specific situations [38, 75].

2.5 Data-Driven Control

The Koopman operator framework is particularly significant for engineering applications in control [39, 109], as it provides new potential for nonlinear system control by overcoming theoretical and computational limits caused by nonlinearity. Some of these constraints are solved by nonlinear control approaches such as feedback linearization and sliding mode control. However, these techniques frequently fail to extend beyond a restricted class of systems, and defining stability and robustness requirements, for example, can be a difficult task [38]. Koopman-based techniques use a linear approach that takes advantage of established theoretical and computational methods. These methods have proven to be effective in tackling various demanding applications, such as robotics [110, 61, 111, 54], chemical processes [112], power grids [68, 113], traffic [114], biology [115], logistics [116] and fluid dynamics [39, 69]. This is accomplished by Koopman analysis, which represents nonlinear dynamics within a linear framework that covers the whole basin of attraction of stable or unstable equilibria or periodic points,

without the need for linearization. Consequently, the Hartman-Grobman theorem can be extended using Koopman analysis in a globally linear fashion [40]. In addition, the Koopman operator operates on observables, making it suitable for data-driven (model-free) methods that have been highly developed in recent years [53, 117, 56, 71, 118, 55, 119, 58, 61]. The resultant models have demonstrated the ability to uncover crucial information about global stability properties [120, 121] , observability [122], controllability [123], and sensor/actuator placement [124, 125] for the fundamental nonlinear system. Koopman operator theory shares a strong connection with Carleman linearization [46], which likewise incorporates finite-dimensional dynamics into infinite-dimensional linear systems. For many years, Carleman linearization has been employed to obtain truncated linear (and bilinear) state estimators [47] and to investigate the stability, observability, and controllability of the underlying nonlinear system [38, 48]. However, Carleman linearization is limited to polynomial (or analytical) systems, while the Koopman operator framework can be used for general nonlinear systems, even those with discontinuities, as it does not depend on the analyticity of the vector field.

2.5.1 Model Predictive Control

MPC, or model-based control, is a highly successful control scheme that has become increasingly popular over the last two decades [62, 63, 64, 65, 66, 67, 126]. It is favored for its versatility, adaptability to modeling discrepancies and disturbances, and its use of customized cost functions and constraints. MPC has become the standard advanced control method in process industries [66] and has gained attraction in the aerospace industry [126]. At its core, MPC is a control scheme that solves an optimization problem to determine the sequence of control inputs $\{\hat{\mathbf{u}}_0,\ \hat{\mathbf{u}}_1, \ldots, \hat{\mathbf{u}}_{\boldsymbol{N-1}}\}$ that will minimize a cost function J while satisfying a set of constraints. The optimization is performed over a finite time horizon $T = N\Delta t$, and only the first control input $\hat{\mathbf{u}}_0^{opt}$ in the sequence is applied to the system. The process then repeats, with a new measurement being taken and a new optimization problem being solved to determine the next control input in the sequence. One of the key advantages of MPC is its ability to incorporate customized cost functions and constraints. This means that the control inputs can be tailored to meet specific performance objectives, such as minimizing energy consumption, or maintaining product quality. The constraints can also be tailored to meet specific requirements, such as limits on process variables or equipment capacities. Figure 2.1 depicts the schematic of the receding horizon MPC framework. MPC can handle both linear and nonlinear systems, but the most critical part of MPC is identifying a dynamical model that accurately represents the system behavior in the presence of actuation. If the model is linear, the optimization problem is typically convex and can be solved efficiently. However, nonlinear systems require more complex optimization techniques, which can be computationally expensive and limit their real-time applicability. Koopman-based controls have benefited from the MPC framework's success, as its adaptive nature enables compensation for discrepancies in the model and disturbances in the system. Several versions of MPC, such as nonlinear MPC [64, 67], robust MPC [127], and explicit MPC [128],

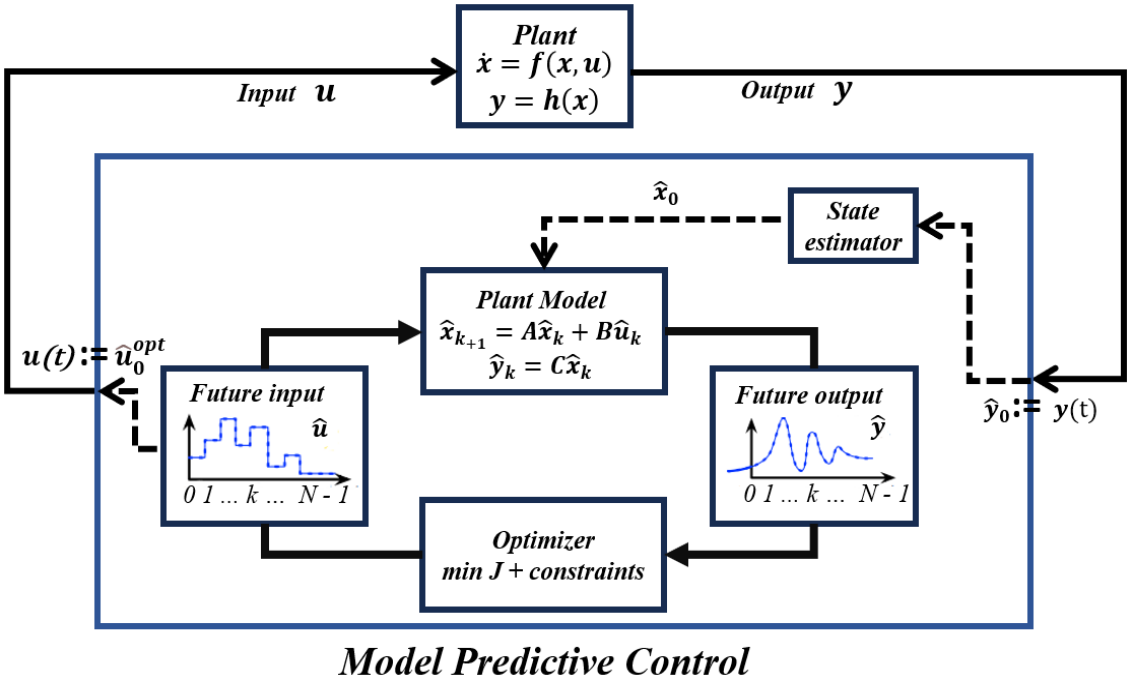

Figure 2.1: Schematic of the receding horizon model predictive control framework.

are available but are computationally expensive, limiting their use in real-time applications. However, combining Koopman-based models with linear MPC can significantly broaden the scope of linear MPC to address nonlinear systems.

The optimization problem, which is based on a receding-horizon approach, can be described as follows. For linear MPC, the goal is to minimize the quadratic objective function.

$$\min_{\hat{\mathbf{u}}(.|\mathbf{y})\in\mathbb{U}} J = \min_{\hat{\mathbf{u}}(.|\mathbf{y})\in\mathbb{U}} \sum_{k=0}^{N-1} \|\hat{\mathbf{y}}_{\boldsymbol{k}} - \mathbf{y}_k^{Ref}\|_{\mathbf{Q}}^2 + \|\hat{\mathbf{u}}_{\boldsymbol{k}}\|_{\mathbf{R}}^2 + \|\Delta\hat{\mathbf{u}}_{\boldsymbol{k}}\|_{\mathbf{R}_\Delta}^2 \tag{2.47}$$

subject to the discrete-time linear dynamical model

$$\begin{aligned} \hat{\mathbf{x}}_{k+1} &= \mathbf{A}\hat{\mathbf{x}}_k + \mathbf{B}\hat{\mathbf{u}}_k, \\ \hat{\mathbf{y}}_k &= \mathbf{C}\hat{\mathbf{x}}_k, \end{aligned} \tag{2.48}$$

and under the presence of constraints on state and input

$$\mathbf{y}_{Min} \leq \hat{\mathbf{y}}_k \leq \mathbf{y}_{Max}, \tag{2.49}$$

$$\mathbf{u}_{Min} \leq \hat{\mathbf{u}}_k \leq \mathbf{u}_{Max}, \tag{2.50}$$

where $\Delta\hat{\mathbf{u}}_{\boldsymbol{k}}$ is the control input rate given by $\Delta\hat{\mathbf{u}}_{\boldsymbol{k}} := \hat{\mathbf{u}}_{\boldsymbol{k}} - \hat{\mathbf{u}}_{\boldsymbol{k-1}}$. The state transition matrix in model (2.48) is represented by $\mathbf{A}$, while $\mathbf{B}$ denotes the control matrix, and $\mathbf{C}$ stands for the measurement matrix. The calculation of each component in the cost function (2.47) involves determining the weighted norm of a vector, i.e.,$\|\hat{\mathbf{y}}\|_Q^2 = \mathbf{y}^\top\mathbf{Q}\mathbf{y}$. Weight matrices $\mathbf{Q} \in \mathbb{R}^{n\times n}$, $\mathbf{R} \in \mathbb{R}^{u_n\times u_n}$, and

$\mathbf{R}_\Delta \in \mathbb{R}^{u_n \times u_n}$, used to penalize deviations of predicted output $\mathbf{y}$ from a reference trajectory $\mathbf{y}^{Ref}$, inputs, and input rates, respectively, are positive semi-definite and determine their respective significance levels. After receiving the measurement $\mathbf{y}$, we establish the control sequence to be resolved over the moving time horizon as $\hat{\mathbf{u}}(0, \ldots, N-1|\mathbf{y}) := \{\hat{\mathbf{u}}_0, \hat{\mathbf{u}}_1, \ldots, \hat{\mathbf{u}}_{\boldsymbol{N-1}}\}$. The current output of the plant, known as the measurement $\mathbf{y}$, is utilized to assess the initial condition $\hat{\mathbf{x}}_0$ when resolving the optimization problem.

There are two primary research efforts in the integration of Koopman theory and MPC, which are roughly classified into discrete or continuous input approaches. For the latter, a Koopman-MPC framework has been proposed, which leverages the Koopman operator to capture the dynamics of the system. In this approach, the output measurements are lifted into a higher-dimensional space using a nonlinear transformation, which enables the use of linear models for the system dynamics. Specifically, the lifted measurements are mapped to a set of observables that form a basis for the Koopman operator. The Koopman operator is a linear operator that preserves the dynamics of the system, and it can be used to model the evolution of the system in the augmented space. To construct the Koopman-based model, a set of snapshots of the system's output are used to estimate the Koopman operator matrix. This is typically done using a linear least-squares regression problem, which minimizes the difference between the model predictions and the observed output. The resulting Koopman-based model is then employed in the MPC optimization process to generate the control inputs. Figure 2.2 illustrates the MPC controller based on the Koopman operator.

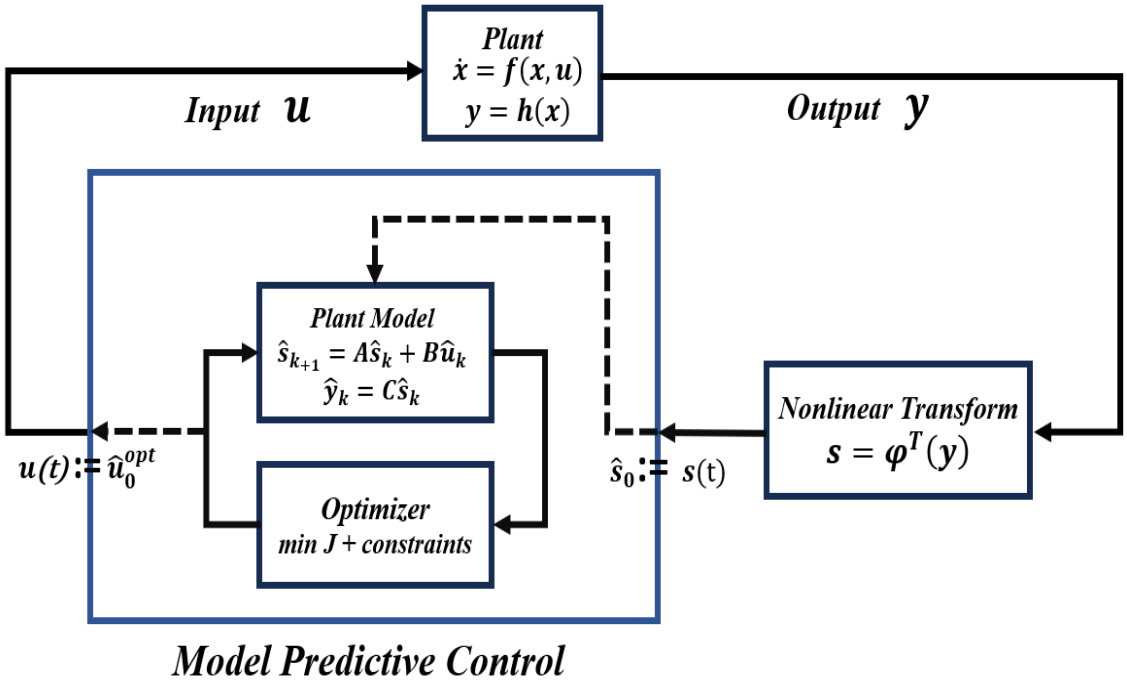

Figure 2.2: Schematic of the MPC framework incorporating a model based on the Koopman operator.

Besides the goal of achieving increased predictive power via a Koopman-based

model, this approach further provides the possibility to readily incorporate nonlinear cost functions and constraints in a linear fashion by incorporating these directly in the set of observables. This is because the observables, which are the inputs to the Koopman operator, can be chosen to be any nonlinear function of the system's output. Therefore, by selecting appropriate observables, the nonlinear cost functions and constraints can be incorporated in a linear manner, making it easier to solve the MPC optimization problem [117, 38, 56].

2.5.2 Nonlinear System Identification for Control

Dynamic Mode Decomposition with Control

The DMD algorithm was extended by Proctor et al. [53] to include the effect of actuation and control, which is called DMD with control (DMDc). The reason for this extension is that when DMD is naively applied to data from a system with actuation, it often gives inaccurate results because the effects of internal dynamics are mixed with the effects of actuation. DMDc solves this problem by including the actuation signal in the analysis to separate the effects of internal dynamics from those of actuation and control. This approach has been used to study the spread of diseases [97] and can also work on heavily sub-sampled or compressed data [129]. As an alternative, if the actuation signal is measured, a new DMD regression algorithm may be developed to extract the effects of internal dynamics from actuation and control.

Using the following measurement data matrices of snapshots :

$$\mathbf{X} = \begin{bmatrix} | & & | \\ \mathbf{x}_1 & \dots & \mathbf{x}_{m-1} \\ | & & | \end{bmatrix}, \quad \mathbf{X}^{'} = \begin{bmatrix} | & & | \\ \mathbf{x}_2 & \dots & \mathbf{x}_{m} \\ | & & | \end{bmatrix}, \quad \boldsymbol{\Upsilon} = \begin{bmatrix} | & & | \\ \mathbf{u}_2 & \dots & \mathbf{u}_{m} \\ | & & | \end{bmatrix} \tag{2.51}$$

where $\mathbf{X}$ and $\mathbf{X}^{'}$ are the same as in DMD presented in Section 2.3 while $\boldsymbol{\Upsilon}$ is a matrix showing the history of the actuation input. The DMDc technique in this case aims to find the best-fit linear operators $\mathbf{A}$ and $\mathbf{B}$ that roughly fulfill the following dynamics:

$$\mathbf{x}_{k+1} \approx \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k, \tag{2.52}$$

Using the data matrices in (2.51) the dynamics in (2.52) may be written as

$$\mathbf{X}^{'} \approx \mathbf{A}\mathbf{X} + \mathbf{B}\boldsymbol{\Upsilon}, \tag{2.53}$$

Again, the matrix $\mathbf{A}$ and the corresponding DMD modes and eigenvalues may be obtained via a least-squares regression approach as in DMD Algorithm 1. However, we can easily distinguish two cases to consider when using this algorithm: when the input data matrix represented by $\mathbf{B}$ is known, the algorithm can be modified in order to find the DMD modes. In this modified version, $\mathbf{X}^{'}$ is replaced with $\mathbf{X}^{'} - \mathbf{B}\boldsymbol{\Upsilon}$, where:

$$(\mathbf{X}^{'} - \mathbf{B}\boldsymbol{\Upsilon}) \approx \mathbf{A}\mathbf{X}, \tag{2.54}$$

However, if $\mathbf{B}$ is unknown, it is necessary to simultaneously identify $\mathbf{A}$ and $\mathbf{B}$. In this instance, we can reformulate the dynamics in (2.53) as:

$$\mathbf{X}^{'} \approx \begin{bmatrix} \mathbf{A} & \mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \boldsymbol{\Upsilon} \end{bmatrix} \tag{2.55}$$

The following least-squares optimization problem is resolved to simultaneously compute the matrices $\mathbf{A}$ and $\mathbf{B}$:

$$\min_{\begin{bmatrix} \mathbf{A} & \mathbf{B} \end{bmatrix}} \left\| \mathbf{X}^{'} - \begin{bmatrix} \mathbf{A} & \mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \boldsymbol{\Upsilon} \end{bmatrix} \right\|_2^2, \tag{2.56}$$

in which the solution is stated as:

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \end{bmatrix} = \mathbf{X}^{'} \begin{bmatrix} \mathbf{X} \\ \boldsymbol{\Upsilon} \end{bmatrix}^{\dagger}, \tag{2.57}$$

In the original paper by Proctor et al. [53], the formulation of the solution in (2.57) was written as:

$$\mathbf{G} = \mathbf{X}^{'} \boldsymbol{\Omega}^{\dagger} \tag{2.58}$$

where $\mathbf{G} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \end{bmatrix}$ is an augmented matrix containing the unknown operators $\mathbf{A}$ and $\mathbf{B}$, and $\boldsymbol{\Omega} = \begin{bmatrix} \mathbf{X} \\ \boldsymbol{\Upsilon} \end{bmatrix}$ is an augmented matrix that contains the state and input data matrices $\mathbf{X}$ and $\boldsymbol{\Upsilon}$. The DMDc algorithm in this case where $\mathbf{B}$ is unknown is presented in Algorithm 2. DMDc is a method that was initially used in epidemiology to improve intervention efforts [130]. However, it has been used with MPC to control nonlinear systems since then [56, 57, 131]. Surprisingly, the DMDc method works well even for strongly nonlinear systems because it uses both linear and nonlinear measurements of the system. DMDc offers a flexible regression framework that can handle measurements of an actuated system and identify accurate and efficient models [75, 38].

Extended Dynamic Mode Decomposition with Control

The DMDc algorithm is a straightforward and effective way to identify systems using numerical methods. This involves approximating the Koopman operator with a linear model that progresses through linear observables and actuation variables in the best possible way. EDMD is a method similar to DMD, except that it employs nonlinear observables. Williams et al. [117] introduced an EDMD extension for controlled systems (EDMDc). This method estimates an unforced system's Koopman operator and accounts for the effects of inputs on system dynamics and data. In this approach, inputs are treated as system parameters that change over time. The Koopman operator is then represented as an operator that varies with these parameters, taking inspiration from a type of models called linear parameter varying (LPV) models [38]. Korda and Mezić [56] generalized this method to identify system matrices $\mathbf{A}$ and $\mathbf{B}$ in a higher-dimensional observable space. This helps to clarify the unforced dynamics and control on observables, making

Algorithm 2: The DMDc Algorithm

Data: Collect and arrange snapshots data from a system into matrices

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & \dots & \mathbf{x}_{m-1} \end{bmatrix}, \quad \mathbf{X}^{'} = \begin{bmatrix} \mathbf{x}_2 & \dots & \mathbf{x}_m \end{bmatrix}, \quad \boldsymbol{\Upsilon} = \begin{bmatrix} \mathbf{u}_1 & \dots & \mathbf{u}_m \end{bmatrix}$$

Step 1: Construct the augmented matrix of the state and input snapshots $\boldsymbol{\Omega}$:

$$\boldsymbol{\Omega} = \begin{bmatrix} \mathbf{X} \\ \boldsymbol{\Upsilon} \end{bmatrix}$$

Step 2: Compute the SVD of the matrix $\boldsymbol{\Omega}$:

$$\boldsymbol{\Omega} \approx \tilde{\mathbf{U}}\tilde{\boldsymbol{\Sigma}}\tilde{\mathbf{V}}^{*}$$

Step 3: Decompose the matrix $\tilde{\mathbf{U}} \in \mathbb{R}^{(n+p)\times n}$ into $\tilde{\mathbf{U}}_1 \in \mathbb{R}^{n\times n}$ and $\tilde{\mathbf{U}}_2 \in \mathbb{R}^{p\times n}$ where n and p are the numbers of the states and control inputs, respectively such as:

$$\tilde{\mathbf{U}} = \begin{bmatrix} \tilde{\mathbf{U}}_1 \\ \tilde{\mathbf{U}}_2 \end{bmatrix}$$

Result: Compute the $\mathbf{A}$ and $\mathbf{B}$ matrices :

$$\mathbf{A} = \mathbf{X}^{'}\tilde{\mathbf{V}}\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\mathbf{U}}_1^{*}$$
$$\mathbf{B} = \mathbf{X}^{'}\tilde{\mathbf{V}}\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\mathbf{U}}_2^{*}$$

it less ambiguous. The Koopman operator is defined as an independent operator on the extended state $\tilde{\mathbf{x}} := [\mathbf{x}^\top, \mathbf{u}^\top]^\top$ similar to (2.23), with observables that may involve nonlinear functions of both the state and input i.e. $g(\mathbf{x}, \mathbf{u})$. However, for practical purposes, certain simplifications are made to enable the control problem to be formulated in a convex manner. It is assumed that the observables are composed of nonlinear functions of the state, but linear functions of the control input i.e. $\mathbf{g}(\mathbf{x}, \mathbf{u}) := [\varphi_1(\mathbf{x}) \dots \varphi_p(\mathbf{x}), u_1 \dots u_q]^\top \in \mathbb{R}^{p+q}$, by focusing solely on the state observables $\varphi(\mathbf{x})$ and limiting the dynamics of interest to them, the resulting linear evolution equation to be established can be expressed as follows:

$$\mathbf{s}_{k+1} \approx \mathbf{A}\mathbf{s}_k + \mathbf{B}\mathbf{u}_k, \tag{2.59}$$

where $\mathbf{s} \in \mathbb{R}^p$ is the vector-valued observable described in section 2.4 as :

$$\mathbf{s} := \boldsymbol{\varphi}^\top(\mathbf{x}) = \begin{bmatrix} \varphi_1(\mathbf{x}) \\ \varphi_2(\mathbf{x}) \\ \vdots \\ \varphi_p(\mathbf{x}) \end{bmatrix} \tag{2.60}$$

The (time-shifted) data matrices in the augmented space $\mathbf{S} = \boldsymbol{\varphi}^\top(\mathbf{X})$ and $\mathbf{S}^{'} = \boldsymbol{\varphi}^\top(\mathbf{X}^{'})$ are assessed given the data $\mathbf{X}, \mathbf{X}^{'}$ and $\boldsymbol{\Upsilon}$ similarly to DMDc as in (2.51). The least-squares regression problem yields the system matrices $\mathbf{A}$ and $\mathbf{B}$ is given by:

$$\min_{[\mathbf{A} \quad \mathbf{B}]} \left\| \mathbf{S}^{'} - \begin{bmatrix} \mathbf{A} & \mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{S} \\ \boldsymbol{\Upsilon} \end{bmatrix} \right\|_2^2, \tag{2.61}$$

where the solution is provided as

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \end{bmatrix} = \mathbf{S}^{'} \begin{bmatrix} \mathbf{S} \\ \boldsymbol{\Upsilon} \end{bmatrix}^{\dagger} = \boldsymbol{\varphi}^\top(\mathbf{X}^{'}) \begin{bmatrix} \boldsymbol{\varphi}^\top(\mathbf{X}) \\ \boldsymbol{\Upsilon} \end{bmatrix}^{\dagger}, \tag{2.62}$$

In many cases, the state $\mathbf{x}$ is considered as part of the basis or space of observables e.g., $\mathbf{s} = [\mathbf{x}^\top, \boldsymbol{\varphi}(\mathbf{x})]^\top$. To estimate it, one can choose the suitable elements from the observable vector $\mathbf{x}$, in order to have $\mathbf{x} = \mathbf{C}\mathbf{s}$ with $\mathbf{C} = [\mathbf{I}_{\mathbf{n}\times\mathbf{n}}\mathbf{0}]$ and $\mathbf{C}$ is the measurements matrix. If the state vector is not considered as an observable, one alternative approach is to solve a similar least-squares problem to obtain an approximation of the measurement matrix:

$$\min_{\mathbf{C}} \|\mathbf{X} - \mathbf{C}\mathbf{S}\|_2^2, \tag{2.63}$$

It has also been demonstrated that the state may be calculated using techniques such as multidimensional scaling [132].

When full-state measurements are not attainable, and only input-output data is accessible, it is common to use *Time-delay coordinates* which are an important and versatile type of measurement functions for systems that exhibit long-term memory effects. They are particularly significant in practical situations where full-state measurements are not feasible [38], and they have shown to outperform models that use monomials as observables in control applications [119]. This technique is usually employed in system identification [133]. To that end, we will examine a series of measurements involving a scalar input-output pair, namely $u(t)$ and $x(t)$. Using these measurements, we can create a vector of delayed inputs $\mathbf{u}_k := [u_k, \ u_{k-1}, \ u_{k-2}, \ldots, \ u_{k-m}]^\top$ and outputs $\mathbf{s}_k := \mathbf{g}(\mathbf{x}_k) := [x_k, \ x_{k-1}, \ x_{k-2}, \ldots, \ x_{k-m}]^\top$, respectively. We can represent a dynamical system as the following:

$$\mathbf{s}_{k+1} = \mathbf{A}\mathbf{s}_k + \mathbf{B}\mathbf{u}_k, \tag{2.64}$$

$$y_k = [1 \quad 0 \quad \ldots \quad 0]\, \mathbf{s}_k = x_k, \tag{2.65}$$

The initial component of $\mathbf{s}_k$ is used to retrieve the present state x. For causality to be maintained, both the system matrix $\mathbf{A}$ and control matrix $\mathbf{B}$ must have an upper triangular structure, or else current states will be influenced by future states and inputs. In this case, the input $\mathbf{u}_k$ is formed using past inputs, despite the system actually having only one input. Therefore, it is advisable to include previous inputs in $\mathbf{s}$, i.e., $\mathbf{s}_k := \mathbf{g}(\mathbf{x}_k, \mathbf{u}_{k-1}) := [x_k, \ (x,u)_{k-1}, \ (x,u)_{k-2}, \ldots, \ (x,u)_{k-m}]^\top$ to make the current actuation value u_k appear as a single input to the system [56, 134, 39]:

$$\mathbf{s}_{k+1} = \hat{\mathbf{A}}\mathbf{s}_k + \hat{\mathbf{b}}\, u_k, \tag{2.66}$$

$$y_k = [1 \quad 0 \quad \ldots \quad 0]\, \mathbf{s}_k = x_k, \tag{2.67}$$

The delay observables $\mathbf{s}_k := \mathbf{g}(\mathbf{x}_k, \mathbf{u}_{k-1})$ has always been used with EDMDc when full-measurements is absent and only input-output data is accessible [38, 56]. If the delay information is absent, EDMDc may not be successful even after projecting onto a higher-dimensional observable space. To derive a unified equation, we can merge equations (2.64) and (2.65):

$$s_{1,k+1} = y_{k+1} = \begin{bmatrix} \mathbf{CA} & \mathbf{CB} \end{bmatrix} \begin{bmatrix} \mathbf{s}_k \\ \mathbf{u}_k \end{bmatrix} \tag{2.68}$$

This statement is similar to the autoregressive-exogenous (ARX) models used in linear system identification [133]. In ARX models, the present output is expressed as a combination of past measurements and inputs in a linear manner. EDMDc provides a linear model of the controlled dynamics, using a nonlinear transformation of the state within the space of observables. It can be used together with any model-based control approach, much like DMDc. When combined with MPC using a quadratic cost function, the resulting optimization problem can be proven to be a convex quadratic programming problem, as per a study cited in reference [56, 38]. The computational cost of this approach is comparable to that of linear MPC, and it is not affected by the number of observables. Additionally, it performs better than MPC using a model based on either local linearization or Carleman linearization.

2.6 Conclusion

In this chapter we offered a thorough examination of the Koopman Operator Theory and various well-known data-driven methodologies. We began with a review of the Koopman Operator Theory, outlining its essential concepts and importance in comprehending complicated dynamic systems. The Dynamic Mode Decomposition (DMD) approach is then shown, a strong algorithm that approximates the Koopman Operator from data generated by complicated nonlinear dynamical systems. The Dynamic Mode Decomposition (DMD) approach is then shown, a strong algorithm that approximates the Koopman Operator form data generated by complicated nonlinear dynamocal systems. The DMD method and domain applications that proved its efficiency in extracting coherent structures and modes from high-dimensional data were described. The promise of this method has been emphasized in a variety of areas, including fluid dynamics, neurobiology, and video analysis. Furthermore, we investigated the Extended Dynamic Mode Decomposition (EDMD), which builds on the basis of DMD to address nonlinearity and high-dimensional data issues. EDMD expands DMD's capabilities by including new features, allowing for more accurate representation of complex systems and improved predictive capabilities. We looked at Data-Driven Control in the context of control systems by employing the Koopman Operator Theory. Using the Koopman model for control purposes brings up intriguing opportunities for optimizing system behavior. Model Predictive Control (MPC) was established as a major application, allowing for real-time control by repeatedly optimizing control inputs based on projected system behavior. Furthermore, we investigated Nonlinear System Identification for Control utilizing DMD with control and EDMD

with control, demonstrating their use in detecting nonlinear controlled systems and building control-oriented models. These data-driven approaches give vital insights into system dynamics, allowing the creation of effective control strategies for a wide range of applications such as robots, autonomous vehicles, and process control. Finally, the Koopman Operator and data-driven methods have revolutionized the field of system analysis and control. Their ability to extract meaningful information from data, detect system behavior, and assist predictive control has opened up new research and practical application possibilities in a variety of disciplines. As data-driven approaches progress, we may expect even greater advances in understanding and managing complex dynamic systems in the future, leading to more efficient, adaptive, and intelligent control methodologies.

Chapter 3

Model Order Reduction of The ADM1 Model Using the DMD Method

3.1 Introduction

This chapter investigates the complexities of the ADM1, a sophisticated and intricate model. The properties and components of the ADM1 model are described, allowing for a better understanding of AD dynamics. The chapter also gives the ADM1 model's mathematical equations, demonstrating its resilience in describing the complicated biochemical interactions seen in anaerobic systems. The examination of this model reveals that it is a useful tool for academics and practitioners in understanding and optimizing AD processes. Next, we apply the DMD method to reduce the high-dimensional complexity of the ADM1 model. This approach allows us to obtain a simplified representation of the system.

3.2 The Anaerobic Digestion Model Number.1

3.2.1 Description of the ADM1 model

The ADM1 model is a detailed model presented in a report prepared by the IWA Task Group for Mathematical Modeling of Anaerobic Digestion Processes [1]. The model explains the conversion of complex organic substrates into methane and carbon dioxide and inert byproducts. Figure 3.1 provides a summary of the model's focus on substrates and the corresponding conversion processes. The model includes disintegration of complex solids into carbohydrates, proteins, long chain fatty acids (LCFA) and inert substances, and then hydrolysis of these products to sugars, amino acids and LCFA. These substances are then fermented to produce volatile organic acids and molecular hydrogen. Methane is produced through different mechanisms, including cleavage of acetate to methane and reduction of carbon dioxide by molecular hydrogen. The ADM1 model uses state variables to describe the behavior of soluble and particulate components, organic species and

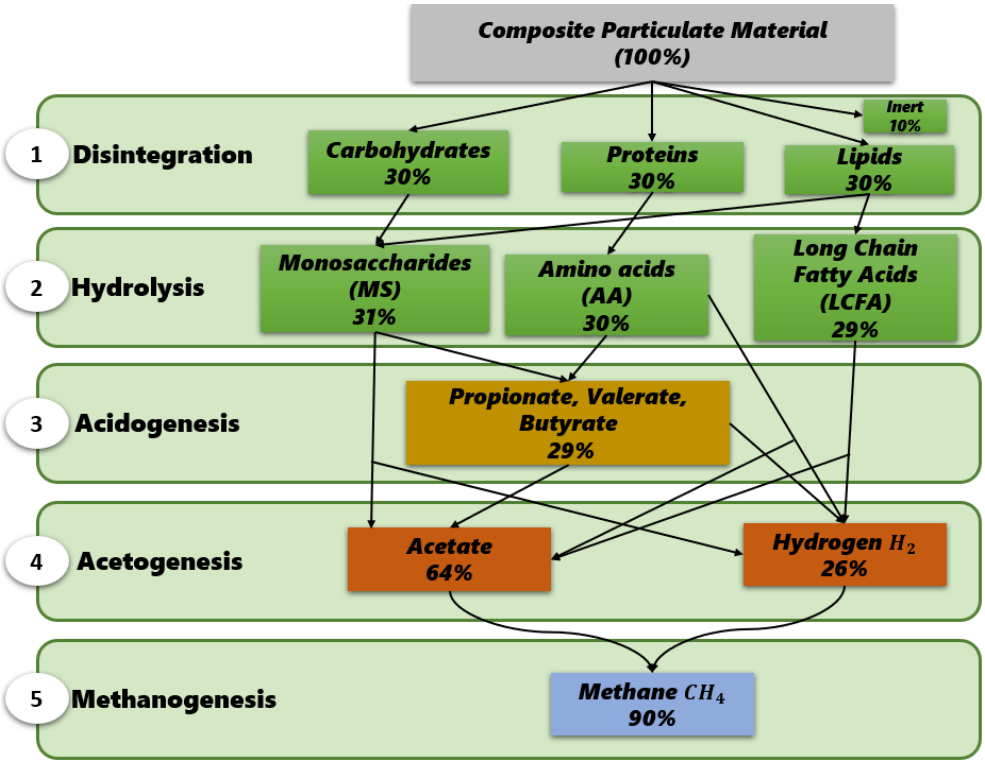

Figure 3.1: Reaction paths and COD flux as described in ADM1. [1]

molecular hydrogen are described in terms of chemical oxygen demand (COD). Nitrogenous and inorganic carbon species are described in terms of their molar concentrations. The ADM1 solves for the hydrogen ion concentration, and thereby the pH, by ensuring chemical neutrality in solution. Particulate species consist of either active biomass species or particulate substances that are incapable of directly passing through bacterial cell walls. The model employs several kinetic expressions that describe the conversion rates in terms of substrate concentrations and rate constants. The ADM1 model also considers the inhibition caused by extreme pH, accumulation of molecular hydrogen, and elevated free ammonia concentrations. The liquid-gas mass transfer of gaseous components is described by mass transfer relationships, requiring separate mass balances for the liquid and gas phases.

3.2.2 Mathematical equations of the ADM1 model

The ADM1 has 35 dimensions in its state space and can be divided into four categories: soluble components "S", particulate components "X", gas components, and ion components. By applying the principle of mass conservation as in (5.1), the equations describing the system can be expressed in the following manner:

- **Soluble components**

$$[S_{su},\ S_{aa},\ \ S_{fa},\ \ S_{va},\ \ S_{bu},\ \ S_{pro},\ \ S_{ac},\ \ S_{H_2},\ \ S_{CH_4},\ \ S_{IC},\ \ S_{IN},\ \ S_I\ ]$$

$$\frac{d}{dt}S_i = \frac{F}{V_{liq}}(S_{i,in} - S_i) + \sum_{j=1}^{19} \rho_j v_{ij} \tag{3.1}$$

- **Particulate components**
  $[X_{su},\ X_{aa},\ \ X_{fa},\ \ X_{ch},\ \ X_{pr},\ \ X_{pro},\ \ X_{ac},\ \ X_{H_2},\ \ X_{li},\ \ X_c,\ \ X_{c4},\ \ X_I\ ]$

$$\frac{d}{dt}X_i = \frac{F}{V_{liq}}(X_{i,in} - X_i) + \sum_{j=1}^{19} \rho_j v_{ij} \tag{3.2}$$

- **Gas components**
  $[S_{gas,H_2},\ \ S_{gas,CH_4},\ \ \ S_{gas,CO_2}\ ]$

$$\frac{d}{dt}S_{gas,i} = -\frac{F_{gas}}{V_{liq}}S_{gas,i} + \rho_{T,i}\frac{V_{liq}}{V_{gas}} \tag{3.3}$$

- **Ion components**
  $[S_{cat},\ \ S_{an},\ \ \ S_{va^-},\ \ S_{bu^-},\ \ S_{pro^-},\ \ S_{ac^-},\ \ S_{HCO_3^-},\ \ S_{NH_3}\ ]$

$$\begin{aligned}
\frac{d}{dt}S_{cat} &= -\frac{F}{V_{liq}}(S_{cat,in} - S_{cat}) \\
\frac{d}{dt}S_{an} &= -\frac{F}{V_{liq}}(S_{an,in} - S_{an}) \\
\frac{d}{dt}S_{i^-} &= -\rho_{A,i^-}
\end{aligned} \tag{3.4}$$

The ADM1 model consists of a total of 28 different processes, which fall into three categories: 19 biochemical processes that are characterized by their reaction rates ρ_j, 9 physical processes that are characterized by their acid-base rates ρ_{A,i^-} and gas transfer rates $\rho_{T,i}$. These processes are important for converting complex organic substrates into methane, carbon dioxide, and other byproducts. It should be noted that these processes are essential components of the ADM1 model, and their accurate representation is necessary to understand the behavior of the system being analyzed. In this context, the way in which biomass grows is described using functions of the Monod type, and the way in which it decays is described using first-order equations:

$$\rho_j = \frac{k_{m,i} \cdot S_i}{K_{S,i} + S_i} \cdot X_i \tag{3.5}$$

$$\rho_j = k \cdot X_i \tag{3.6}$$

The equations include various parameters such as maximum uptake rate constant ($k_{m,i}$), half saturation constant ($K_{S,i}$), and decomposition rate constant (k) of component i, along with rate coefficient (v_{ij}) which represents the effect of the j^{th} process on the i^{th} component. The liquid and gas volumes of the reactor are

represented by V_{liq} and V_{gas} respectively. The concentrations of the i^{th} component in the reactor are denoted by S_i, X_i, $S_{gas,i}$, and S_{i^-}. Additionally, the flow rate and concentration of each component of the organic matter feed are represented by F, $S_{i,in}$ and $S_{i,in}$.

The following is an explicit presentation of the pH value:

$$pH = -\log_{10}(S_{H^+}) \tag{3.7}$$

$$S_{H^+} = -\frac{\theta}{2} + \frac{1}{2}\sqrt{\theta^2 + 4K_\omega} \tag{3.8}$$

$$\theta = S_{cat^+} - S_{an^-} - \frac{S_{va^-}}{208} - \frac{S_{bu^-}}{160} - \frac{S_{pro^-}}{112} - \frac{S_{ac^-}}{64} - S_{HCO_3^-} + S_{NH_4^+} \tag{3.9}$$

where K_ω is a parameter that pertains to the transition of hydrogen ions. When considering the total COD, it involves adding up all the organic components.

$$\begin{aligned} COD_{Total} = {} & S_{su} + S_{aa} + S_{fa} + S_{va} + S_{bu} + S_{pro} + S_{ac} + S_{H_2} + S_{CH_4} \\ & + S_{IC} + S_{IN} + S_I + X_{su} + X_{aa} + X_{fa} + X_{ch} + X_{pr} + X_{pro} \\ & + X_{ac} + X_{H_2} + X_{li} + X_c + X_{c4} + X_I, \end{aligned} \tag{3.10}$$

The ADM1 takes also into account the inhibition coefficients of inorganic nitrogen, hydrogen, and ammonia nitrogen given by the following:

$$\begin{aligned} I_{IN,lim} &= \frac{1}{1 + K_{I,IN}/S_{IN}} \\ I_{H_2} &= \frac{1}{1 + X_{H_2}/K_{I,H_2}} \\ I_{NH_3} &= \frac{1}{1 + S_{NH_3}/K_{I,NH_3}} \end{aligned} \tag{3.11}$$

To use the ADM1 model as an accurate simulation of the AD process, it is important to first identify the critical parameters that are highly sensitive to its performance, such as S_{in}, X_{in}, and certain rate constants. These parameters, along with the corresponding inhibition parameters $K_{I,IN}$, K_{I,H_2}, and K_{I,NH_3}, should be identified when considering a biogas plant feed with a particular type of organic matter. Once these parameters have been identified and calibrated, the model can be used as a reliable simulation of the AD process.

The ADM1 is a complex system that includes a large number of state variables and time constants ranging from fractions of a second to several months. Simulation results [135, 1, 136, 137, 138] have shown that the time constants of soluble, particulate, and gas components are much longer than those of the ion components. This means that the soluble, particulate, and gas components can be considered as slow states, while the ion components are considered fast states. Furthermore, it has been observed that some of the slow states have low sensitivities to the substrate feed. This means that changes in the substrate feed have a minimal impact on these slow states. This information is important in designing

and optimizing the AD process, as it allows researchers and engineers to identify the critical components of the system that require attention in order to achieve optimal performance. ADM1 is a model that takes into account the degradation of organic matter from composite organic to gaseous products. However, due to its high dimensionality and numerous parameters (35 state variables and more than 130 parameters), it is challenging to use the model for control applications. Therefore, a reduced model must be developed. However, the reduction of ADM1 should not affect the model structure, as this could interfere with the way that inputs, states, and outputs of different model components are linked and interact. This is important because ADM1 is applied in various types of digesters, with different organic feeds and under a wide range of operational conditions.

3.3 Model Order Reduction of the ADM1 Model Using DMD Method

In this section, the ADM1 model presented in Section 3.2 is reduced using a naive application of the DMD method explained in Chapter 2 (Section 2.3). The core idea behind the model reduction procedure is depicted in Figure 3.2. The DMD method can be applied on data generated from either a real AD plant or the simulation of ADM1 model, which we will be referencing to a virtual anaerobic plant or bioreactor. The virtual plant used in this study consisted in the benchmark version from Lund University [136, 139], where the implementation and validation of the differential and algebraic equations of the ADM1 model were carried out (all the parameter, initial condition and constants of the ADM1 model used in this work can be found in [136]). Using MATLAB, the ADM1 model is simulated after applying a constant dilution rate in order to generate snapshots of data, over a simulation period of 100 days. Since ADM1 has 35 state variables, each snapshot of data contains 35 elements. The sampling time is taken equal to 1 day, so we have finally 100 state snapshots available for the DMD approach. The state measurements vector (containing Soluble, Particulate and Ion and Gas components) and data matrices are represented respectively by (3.12),(3.13) and (3.14) as follows:

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{x}(S_{su}, t_k) \\ \mathbf{x}(S_{aa}, t_k) \\ \vdots \\ \mathbf{x}(X_i, t_k) \\ \vdots \\ \mathbf{x}(S_{gas,CO_2}, t_k) \end{bmatrix} \tag{3.12}$$

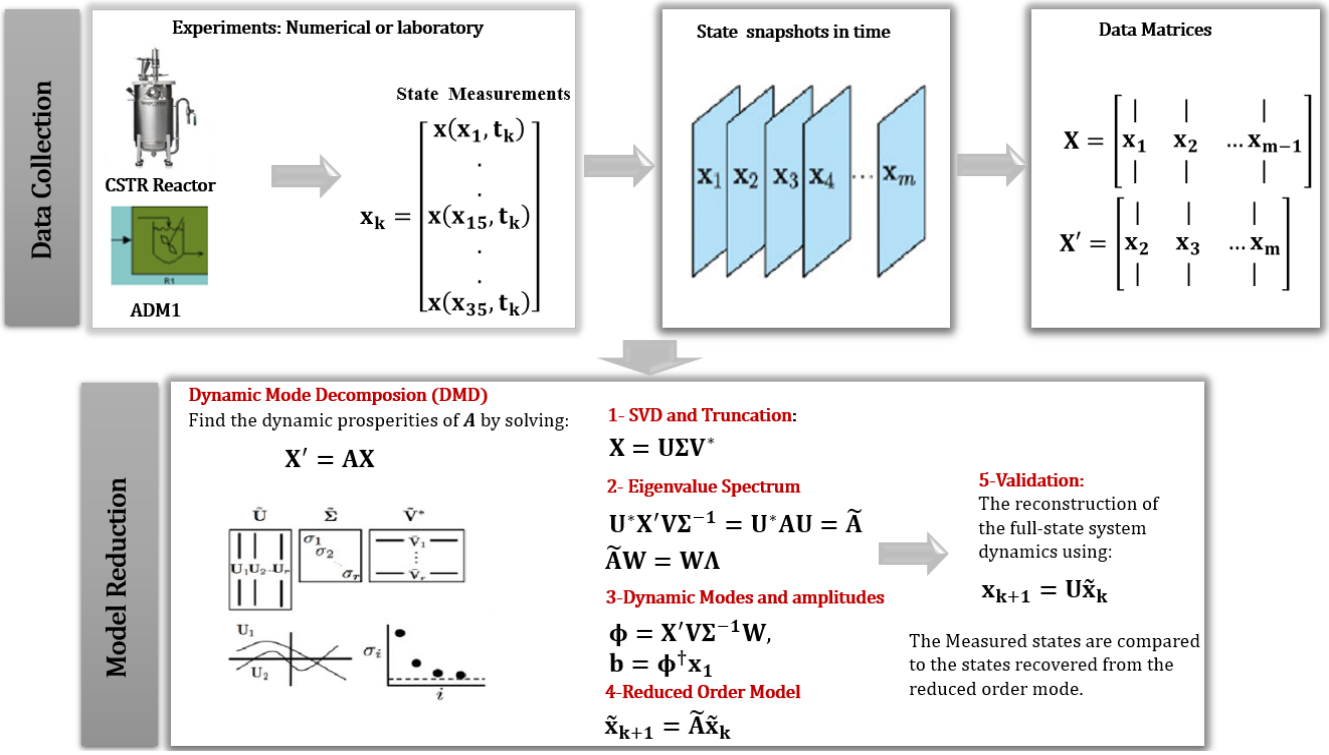

Figure 3.2: The ADM1 Model Reduction Procedure Using DMD method:
Data Collection: *Generate data either through simulations or experiments. Collect state measurements data at different time points and organize them into state snapshots. Reshape the collected data snapshots into data matrices.*
Model Reduction: *Apply Singular Value Decomposition (SVD) on the data matrix X. Examine the resulting sigma matrix Σ. Determine the appropriate truncation order based on the singular values. Deduce the Dynamic Mode Decomposition (DMD) modes and amplitudes. Construct the reduced-order model using Proper Orthogonal Decomposition (POD) modes. Utilize the POD modes to reconstruct the dynamics of the full-state system for validation purposes.*

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_{100-1} \\ | & | & & | \end{bmatrix}, \tag{3.13}$$

$$\mathbf{X}' = \begin{bmatrix} | & | & & | \\ \mathbf{x}_2 & \mathbf{x}_3 & \dots & \mathbf{x}_{100} \\ | & | & & | \end{bmatrix}, \tag{3.14}$$

In our analysis, each column of the matrix $\mathbf{X}$, denoted as $\mathbf{x}_k$, represents the measured data of 35 variables from the ADM1 model at different time points t_k, ($k = 1...99$). Thus, the size of matrix $\mathbf{X}$ is (35×99). To analyze the data using MATLAB, we employed the DMD algorithm 1. By applying the SVD to our data matrix $\mathbf{X}$, we examined the sigma matrix Σ, which contains the singular values of the ADM1 system as shown in Table 3.1. To create a reduced-order model, we retained only the singular values corresponding to the desired reduced order of the global linear model. The remaining singular values, associated with dynamics that have less influence on the overall system behavior, were truncated. In this case, we decided to retain only 7 modes, highlighted in red in Table 3.1. It is important to mention that in [140], the authors employed a similar method based on Principal Component Analysis (PCA) for reducing the complexity of the ADM1 model. They proposed three potential reduced models of the ADM1, starting with 7, 8, and 9 state variables. This parallel can be drawn with the singular values matrix Σ in Table 3.1, where we can observe the flexibility to choose either 8 or 9 values instead of solely 7.

$$\tilde{A} = \begin{bmatrix} 1.0000 & 0.0017 & 0.0925 & 0.1151 & -0.0051 & -2.5154 & 1.2853 \\ 0.0004 & 0.9801 & -2.2847 & -3.0447 & 1.8790 & 65.4570 & -11.5858 \\ -0.0003 & 0.0112 & 0.7896 & -0.3413 & 1.2606 & 7.7779 & 11.2188 \\ 0 & 0.0022 & 0.0424 & 0.9697 & 0.2909 & -0.2709 & 3.6992 \\ 0 & -0.0002 & -0.0137 & 0.0220 & 0.7279 & -0.6853 & -2.4070 \\ 0 & -0.0001 & -0.0029 & 0.0022 & 0.0151 & 0.9111 & 0.3044 \\ 0 & 0 & 0.0020 & -0.0025 & 0.0444 & 0.1197 & 0.0717 \end{bmatrix} \tag{3.15}$$

One significant advantage of the DMD method is its capability to generate a *linear global model* while allowing us to choose the desired level of linear order approximation. By applying the DMD method, we successfully reduced the ADM1 model from 35 states to just 7 states , which correspond to the dominant dynamics of the AD process. To verify the accuracy of our selection of dominant modes in the reduced linear model with the given constant coefficients matrix $\tilde{A}$ in (3.15), we reconstructed the original data matrix $\mathbf{X}$ generated from the simulation of ADM1 model over a period of 100 days. In order to illustrate the effectiveness of the method, we present Figures 3.3, 3.4, and 3.5, which show the comparison of a subset of original state variables of ADM1 and the states generated from the reduced DMD model: 6 soluble components in Figure 3.3, 6 particulate components in Figure 3.4, and 3 gas components and 3 ion components in Figure 3.5. We compare the original data with the data recovered from the reduced model using the DMD algorithm 1. The comparison was conducted for two time intervals: the first 100 days (0...100 days) where measurements were available, and the

σ_i	1	2	3	4	5	6	7	8	9	...	35
1	343.3	0	0	0	0	0	0	0	0	$\cdots$	0
2	0	16.9	0	0	0	0	0	0	0	$\cdots$	0
3	0	0	1.29	0	0	0	0	0	0	$\cdots$	0
4	0	0	0	0.42	0	0	0	0	0	$\cdots$	0
5	0	0	0	0	0.07	0	0	0	0	$\cdots$	0
6	0	0	0	0	0	0.015	0	0	0	$\cdots$	0
7	0	0	0	0	0	0	0.006	0	0	$\cdots$	0
8	0	0	0	0	0	0	0	0.0025	0	$\cdots$	0
9	0	0	0	0	0	0	0	0	0.0017	$\cdots$	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	0
35	0	0	0	0	0	0	0	0	0	0	0

Table 3.1: The matrix Sigma Σ, where singular values are reported in the diagonal and, where the 7^{th} order is the truncation target.

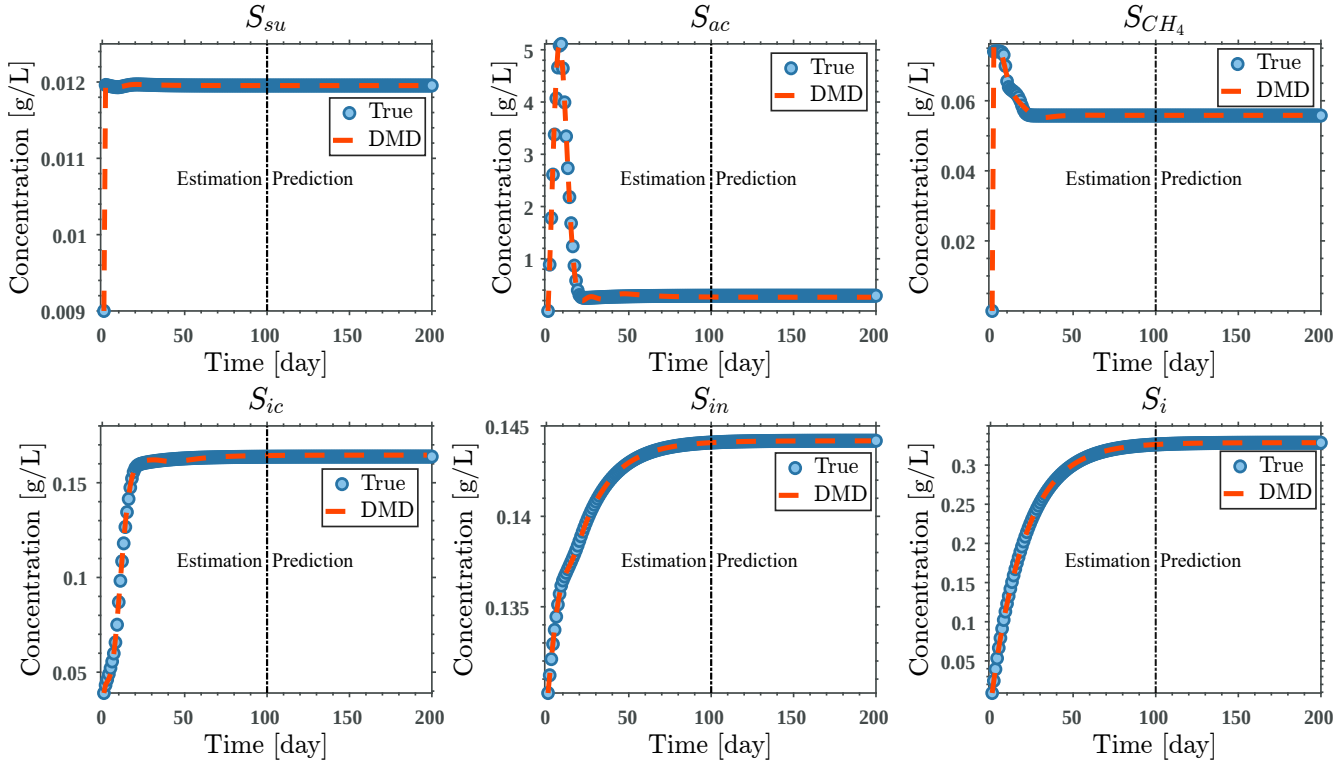

Figure 3.3: DMD applied on the ADM1 data: Soluble Components results comparison

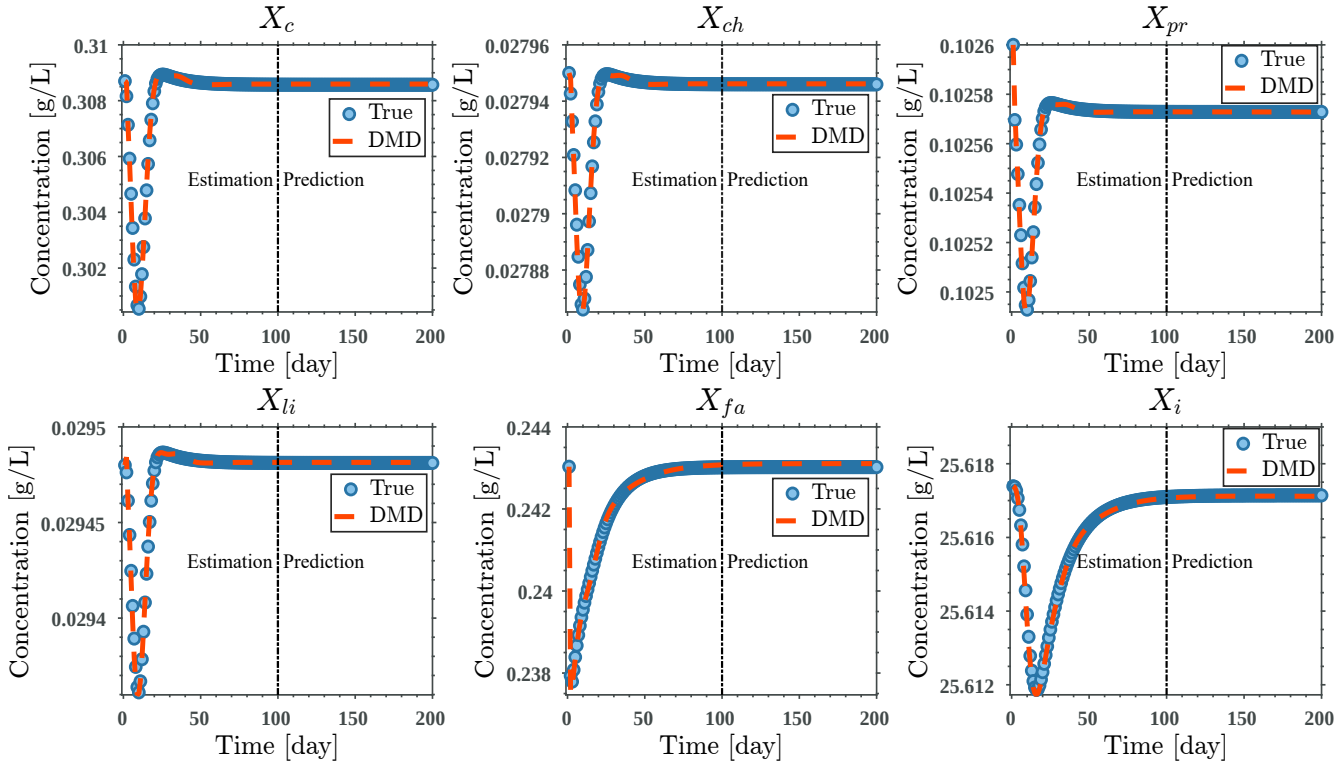

Figure 3.4: DMD applied on the ADM1 data: Particulate Components results comparison

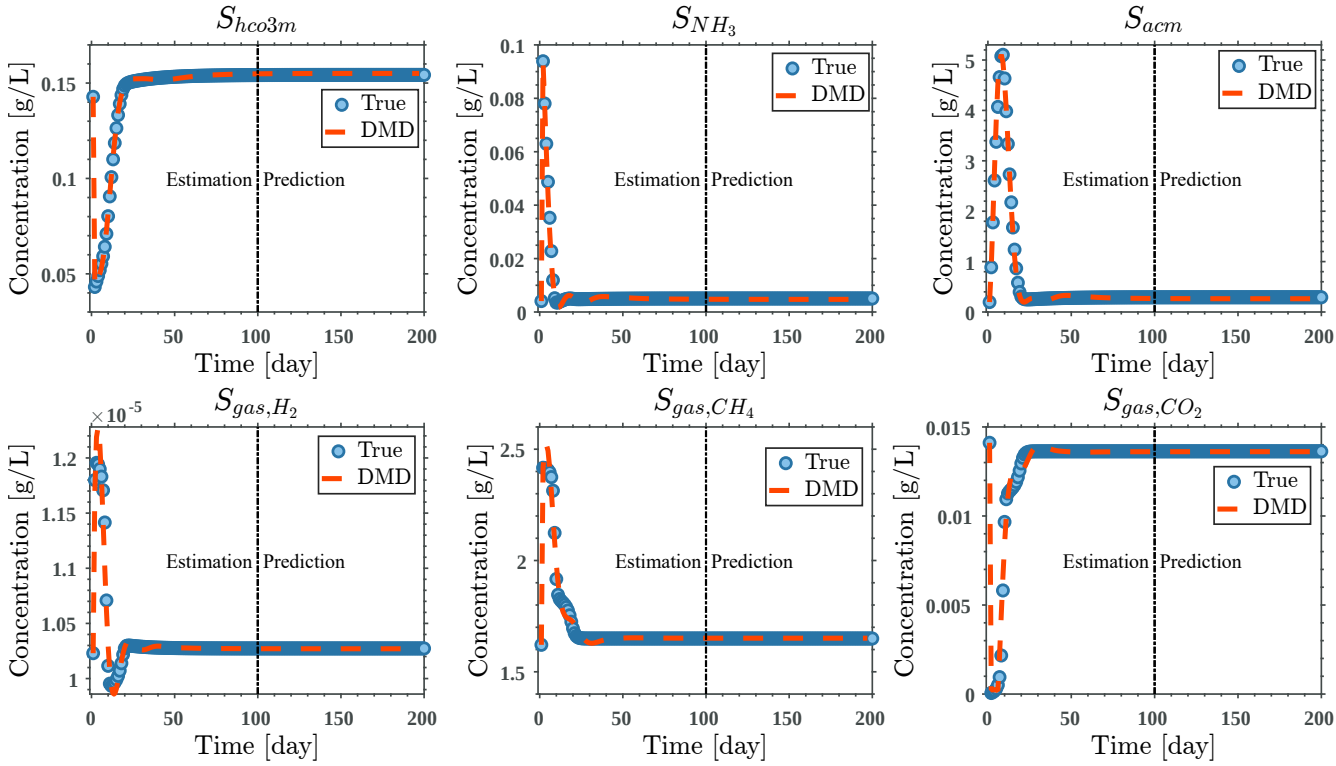

Figure 3.5: DMD applied on the ADM1 data: Gas and ions Components results comparison

subsequent 100 days (101...200 days) where no data was collected. These initial results obtained using the DMD method have been highly satisfactory, particularly considering that this is a data-driven technique without any prior knowledge of the underlying dynamics of the complex ADM1 model. The fact that the DMD approach can successfully capture and represent the dominant dynamics of the system without relying on detailed knowledge of the model is remarkable. Furthermore, it is important to highlight that similar positive results were obtained for other variables in addition to the ones presented. This indicates the robustness and effectiveness of the DMD method in capturing the essential dynamics of the system across multiple variables. These findings suggest that the reduced linear model derived through the DMD method holds promise for various applications. It provides a simplified representation of the complex ADM1 model while maintaining a high level of accuracy in capturing the system's behavior. These results open up possibilities for utilizing the reduced model in process control applications and further exploring its potential in optimizing and controlling anaerobic digestion processes.

3.4 Conclusion

In conclusion, there is a gradual progress towards comprehensive AD models, culminating in the ADM1, which represents a highly detailed model for AD processes. Emphasizing the significance of this mathematical model as an approximation of real-world systems, we delve into the characteristics and properties of the ADM1 model. Understanding these mathematical representations is crucial as they serve as the foundation for generating data that can be employed as input for our subsequent data-driven techniques. The DMD method was employed to extract a simplified linear model from the intricate ADM1 system, reducing 35 state variables to 7 while retaining the primary dynamics. Moreover, the initial ADM1 state variables were successfully recovered using the linear reduced model, showcasing the potential of this approach for modeling and predicting system behavior. The success of the DMD method in producing satisfactory results without relying on explicit knowledge of the ADM1 model's dynamics underscores its value as a valuable tool in data-driven modeling and analysis. It offers a practical approach to understanding and leveraging the behavior of complex systems based solely on observational data.

Chapter 4

Modeling the AM2 Model Using the DMDc Technique

4.1 Introduction

The chapter looks at the AM2 classical modeling approach, which provides a detailed explanation of the biological and kinetic reactions involved using the mass balance law. The AM2 model's mathematical equations were refined, revealing insight on the measurement and modeling of the anaerobic process. We employ the DMDc technique for the data-driven modeling of the AM2 model as an alternative modeling approach for AD systems modeled by the AM2. This method facilitates the construction of a data-driven model for the AM2 system, which can be valuable for system analysis and control design.

4.2 The Two Step Anaerobic Model

4.2.1 Description of the model: biological and kinetics reactions

The Anaerobic Model 2 (AM2) proposed by Bernard et al [5], is a simple model which represents AD in two steps (Acidogenesis-Methanogenesis). It is a model developed within the framework of the European Research Project called AMOCO and it is derived from the law of mass balance [25]. The primary purpose of developing the AM2 model is to facilitate the design of monitoring and control systems for AD processes. It is not intended as a numerical simulation tool for studying process behavior. The model primarily focuses on describing the AD of soluble substrates or those with minimal particulate content [21], [141]. The AM2 model describes the dynamics of six state variables: two consortia of bacteria X_1 and X_2, two types of substrates S_1 and S_2, Total Inorganic Carbon component C and Total Alkalinity component Z.

The two steps and biological reaction schemes of the AM2 model are given by the following :

1. **Acidogenesis:** The process of acidogenesis involves a group of acidogenic

bacteria X_1, which breaks down the organic substance (S_1) into two main products: Volatile Fatty Acids (VFA) (S_2) and carbon dioxide (CO_2). This degradation occurs at a reaction rate denoted as $\mu_1(S_1)X_1$:

$$k_1 S_1 \xrightarrow{\mu_1(S_1)X_1} X_1 + k_2 S_2 + k_4 CO_2 \tag{4.1}$$

2. **Methanogenesis:** At this step, the group of methanogenic bacteria X_2, converts the substance S_2 into methane (CH_4) and carbon dioxide (CO_2). This transformation occurs at a specific reaction rate denoted as $\mu_2(S_2)X_2$:

$$k_3 S_2 \xrightarrow{\mu_2(S_2)X_2} X_2 + k_6 CH_4 + k_5 CO_2 \tag{4.2}$$

The state S_2 primarily consists of acetate, propionate, and butyrate and is assumed to behave like acetate. The Total COD is the sum of S_1 and S_2, i.e. $COD_{Total} = S_1 + S_2$ [5]. One of the primary challenges in AD is the possibility of VFA (S2) accumulation within the bioreactor during the process of methanogenesis. This accumulation has the potential to disrupt the stability of the process by causing acidification in the bioreactor. To address this inhibitory effect, a Haldane equation is employed to model the specific growth rate of methanogens (the specific growth rate of X_2 on S_2), which is represented as $\mu_2(S_2)$:

$$\mu_2(S_2) = \mu_{2max} \frac{S_2}{\frac{S_2^2}{K_i} + S_2 + K_2} \tag{4.3}$$

where:

- μ_{2max} is the maximum growth rate of X_2 on S_2.
- K_2 the half-saturation constant associated with the Substrate S_2.
- K_i is the inhibition constant linked to the substrate S_2.

The acidogenesis specific growth rate $\mu_1(S_1)$ (the specific growth rate of X_1 on S_1) is modeled by a Monod type equation given by

$$\mu_1(S_1) = \mu_{1max} \frac{S_1}{S_1 + K_1}, \tag{4.4}$$

where:

- μ_{1max} is the maximum growth rate of X_1 on S_1.
- K_1 the half-saturation constant associated with the Substrate S_1.

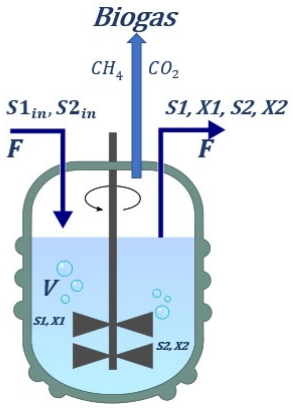

Figure 4.1: Schematic diagram of an CSTR modeled by AM2 model

4.2.2 Mathematical Equations of The AM2 Model

Figure 4.1 depicts a simplified representation of an anaerobic CSTR modeled by AM2 model. The diagram illustrates the bioreactor's continuous operational mode. Furthermore, by applying the mass balance law in (5.1), the AM2 model is derived and its original mathematical equations as proposed in [5] are:

$$\dot{S}_1 = D(S_{1in} - S_1) - k_1\mu_1(S_1)X_1, \tag{4.5}$$
$$\dot{X}_1 = (\mu_1(S_1) - \alpha D)X_1, \tag{4.6}$$
$$\dot{S}_2 = D(S_{2in} - S_2) + k_2\mu_1(S_1)X_1 - k_3\mu_2(S_2)X_2, \tag{4.7}$$
$$\dot{X}_2 = (\mu_2(S_2) - \alpha D)X_2, \tag{4.8}$$
$$\dot{C} = D(C_{in} - C) - q_C + k_4\mu_1(S_1)X_1 + k_5\mu_2(S_2)X_2, \tag{4.9}$$
$$\dot{Z} = D(Z_{in} - Z), \tag{4.10}$$

Where the states of the AM2 model are:

1. S_1: is the substrate concentration of carbonic substance (g/L).
2. X_1: is the population concentration of acidogenic bacteria (g/L).
3. S_2: is the substrate concentration of VFA $(mmol/L)$.
4. X_2: is the population concentration of methanogenic bacteria (g/L).
5. C : is the concentration of the total inorganic carbon $(mmol/L)$.
6. Z : is the concentration of the Total Alkalinity $(mmol/L)$.

And the AM2 model parameters are:

- $\alpha \in [0, 1]$: is the fraction of bacteria in the liquid phase.
- k_1: is the yield for substrate degradation (g/g).

- k_2: is the yield for VFA production $(mmol/g)$
- k_3: is the yield for VFA consumption $(mmol/g)$
- k_4: is the yield for CO_2 production $(mmol/g)$
- k_5: is the yield for CO_2production $(mmol/g)$
- q_C: is the CO_2 flow rate $(mmol/L$ per $day)$

The dilution rate $D = \frac{F}{V}$ $(1/day)$ considered as the control input. In addition, S_{1in}, S_{2in}, C_{in} and Z_{in} are respectively the input concentrations of the substrates S_1, S_2, C and Z. When it comes to modeling the biogas production, the flow rate of biogas (q_m), which includes methane (CH_4) and carbon dioxide (CO_2), can be represented mathematically based on the current values of the state variables:

$$q_m = k_6 \mu_2(S_2) X_2, \tag{4.11}$$

where k_6 is the yield for biogas production $(mmol/g)$.

Next, for the purpose of our work and for simplifications, we disregard the variables C and Z, and denote $\xi = [S_1, X_1, S_2, X_2]^\top$ as the vector of state variables. This will not pose any issues for the mathematical analysis of the fourth-order model (4.5)-(4.8), as it is independent of the C and Z variables. The variables C and Z can be added later if needed.

Considering the state vector ξ, the fourth-order model (4.5)-(4.8) used in this study is expressed as a general nonlinear state space model in a matrix form given below:

$$\dot{\xi} = Kr(\xi) - D\xi + F \tag{4.12}$$

where

$$r(\xi) = \begin{bmatrix} \mu_1(S_1)X_1 \\ \mu_2(S_2)X_2 \end{bmatrix}, K = \begin{bmatrix} -k_1 & 0 \\ 1 & 0 \\ k_2 & -k_3 \\ 0 & 1 \end{bmatrix}, F = \begin{bmatrix} DS_{1in} \\ 0 \\ DS_{2in} \\ 0 \end{bmatrix}, D = \begin{bmatrix} D & 0 & 0 & 0 \\ 0 & \alpha D & 0 & 0 \\ 0 & 0 & D & 0 \\ 0 & 0 & 0 & \alpha D \end{bmatrix} \tag{4.13}$$

4.3 Data-Driven Modeling of the AM2 Model using DMDc Method

In this section, we expand the application of the DMD method to derive a linear controlled model for the AM2 model discussed in Section 4.2. The DMDc method is a nonlinear system identification technique that relies solely on data, as explained in Section 2.5.2. The main distinction between DMD and DMDc lies in the incorporation of control. While DMD is capable of producing a linear operator that describes the system's behavior, it cannot differentiate the influence of control on the system. In contrast, DMDc can generate a discrete controlled

linear model, which is useful for identifying and controlling nonlinear systems. First, we generate the training data by considering the AM2 model in (4.5)-(4.8), and rewritten in the following as a virtual system:

$$\begin{cases} \dot{S}_1 = u(S_{1in} - S_1) - k_1\mu_1(S_1)X_1 \\ \dot{X}_1 = (\mu_1(S_1) - \alpha u)X_1 \\ \dot{S}_2 = u(S_{2in} - S_2) + k_2\mu_1(S_1)X_1 - k_3\mu_2(S_2)X_2 \\ \dot{X}_2 = (\mu_2(S_2) - \alpha u)X_2 \end{cases} \tag{4.14}$$

where u is the control input of the system and it is the dilution rate D (day^{-1}). The parameters used for this simulation are given in the following Table 4.1 which were taken from the original paper of the AM2 model [5] just to give a practical sense of identification process.

Parameter	Value	Unit
S_{1in}	20	g/L
S_{2in}	150	$mmol/g$
μ_{1max}	1.5	day^{-1}
μ_{2max}	0.74	day^{-1}
k_1	42.14	–
k_2	116.5	$mmol/g$
k_3	268	$mmol/g$
K_1	7.1	g/L
K_2	9.28	$mmol/g$
K_i	256	$mmol/g$
α	1	$mmol/g$

Table 4.1: Parameters used in the simulation of the AM2 model using MATLAB as in [5].

We create the training data matrices presented in (4.15), (4.16) and (4.17) by spanning 100 days, incorporating a discrete control input u (Figure 4.3). The control input takes the form of a Pseudo-Random Binary Signal (PRBS) [142], characterized by deterministic behavior with properties similar to white noise, and serves to represent various potential interventions.

$$\mathbf{X} = \begin{bmatrix} S_1^1 & S_1^2 & , ..., & S_1^{100-1} \\ X_1^1 & X_1^2 & , ..., & X_1^{100-1} \\ S_2^1 & S_2^2 & , ..., & S_2^{100-1} \\ X_2^1 & X_2^2 & , ..., & X_2^{100-1} \end{bmatrix}, \tag{4.15}$$

$$\mathbf{X}' = \begin{bmatrix} S_1^2 & S_1^3 & , ..., & S_1^{100} \\ X_1^2 & X_1^3 & , ..., & X_1^{100} \\ S_2^2 & S_2^3 & , ..., & S_2^{100} \\ X_2^2 & X_2^3 & , ..., & X_2^{100} \end{bmatrix}, \tag{4.16}$$

$$\Upsilon = \begin{bmatrix} u_1 & u_2 & ,..., & u_{100-1} \end{bmatrix} \tag{4.17}$$

The initial column of matrix $\mathbf{X}$ in (4.15) represents the state measurements of the AM2 model in (4.14), in the first day. These measurements are obtained by applying the control input (u_1) from the input matrix $\boldsymbol{\Upsilon}$ in the same day. Moreover, the matrix $\mathbf{X}'$ in (4.16) is identical to $\mathbf{X}$, but it is shifted one time step ($\Delta t = 1day$) into the future. Once the data has been gathered, it is processed using the DMDc Algorithm 2. The purpose of running this algorithm is to create a linear state-space model for the full-state AM2 system. This model aims to represent the behavior and dynamics of the original system. To assess the accuracy and reliability of the DMDc model generated, a validation process is carried out. This validation involves applying the same control input to both the original AM2 system and the identified DMDc model, but over a a longer time period. By doing this, we can observe and compare how both systems respond and check if the DMDc model accurately captures the behavior of the real system. Figure 4.4 presents the results of both the training and validation phases of the DMDc algorithm. The figure displays two parts: the Training phase (identification) represented with a grey background and the Validation phase (prediction) represented with a white background. In the Training phase, the DMDc model is created, and in the Validation phase, the model's predictive capabilities are assessed. The purpose of this analysis is to identify a linear model by comparing the measured data from the CSTR with the data generated by the DMDc model. This comparison is achieved by applying the same PRBS control inputs shown in Figure 4.3. The excellent fit of the *DMDc global linear state-space input-output model* to the measured data of the AD system is easily noticeable. To thoroughly test the DMDc linear model's capabilities, we deliberately introduced instability by applying a PRBS control input. This allowed us to assess how well the DMDc model could accurately respond to such challenging conditions. The results of this evaluation are quite promising. The Linear DMDc model, characterized by the matrices A and B given in (4.18) , exhibits precise predictions of the real system's measured data. Notably, it efficiently tracks and adapts to any sudden changes in the behavior of the AM2 dynamics, considering the variable control input. The DMDc global linear model's accuracy in capturing the real system's dynamics is remarkable and noteworthy. It is important to stress that the DMDc method operates solely on data, making it a fully data-driven approach. By using only input-output data, the method constructs a comprehensive global linear state-space model. This model can then be easily employed for various tasks, such as state-estimation and control. This data-driven nature not only simplifies the model construction but also ensures its applicability to real-world systems without the need for extensive system knowledge or complex mathematical models. The ability to use input-output data directly makes the DMDc approach highly accessible and versatile for practical applications in engineering and control systems.

$$A = \begin{bmatrix} 0.0873 & 0.1929 & -0.0033 & -0.7343 \\ 0.0224 & 1.1087 & 0.0006 & -0.0493 \\ -0.5018 & -0.7024 & 0.2263 & -3.4821 \\ 0.0146 & 0.4848 & 0.0051 & 0.7353 \end{bmatrix}, B = \begin{bmatrix} 9.1095 \\ -0.2337 \\ 32.5205 \\ -0.2903 \end{bmatrix} \tag{4.18}$$

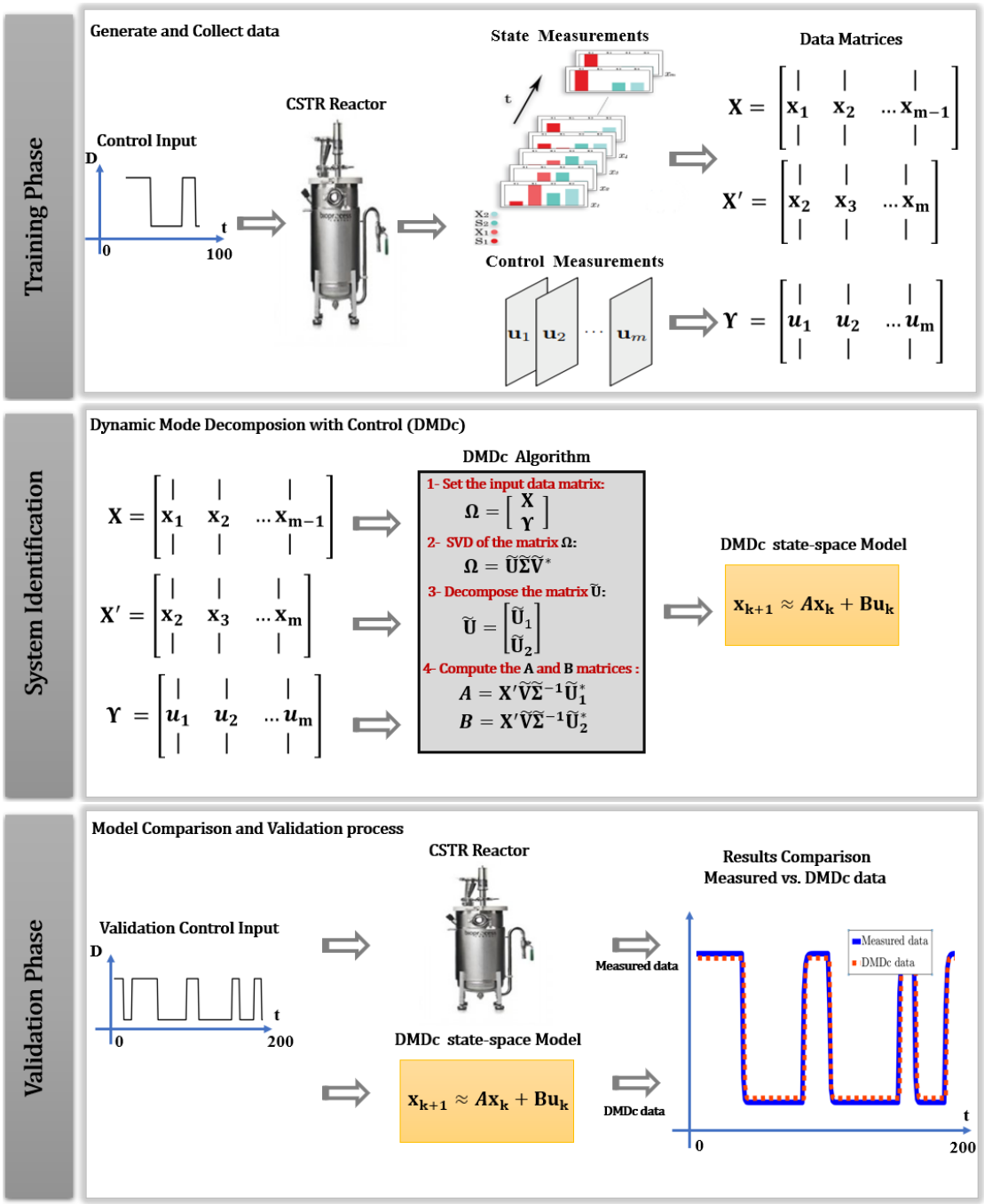

Figure 4.2: The nonlinear system identification process using DMDc applied to the AM2 model: **Training Phase** : *Generate data collect data by applying a PRBS control input to the CSTR modeled by AM2 model.* **System Identification Phase**: *Use the data matrices X, X' and Υ in (4.15), (4.16) and (4.17) as an input to the DMDc Algorithm 2 and generate as a output a DMDc state-space model.* **Validation Phase**: *Apply a PRBS control input for a longer time period to both original CSTR reactor and DMDc model. Compare the results between the measured and DMDc data in corder to validate the model.*

In Figure 4.2, the three step system identification process used in this study is shown.

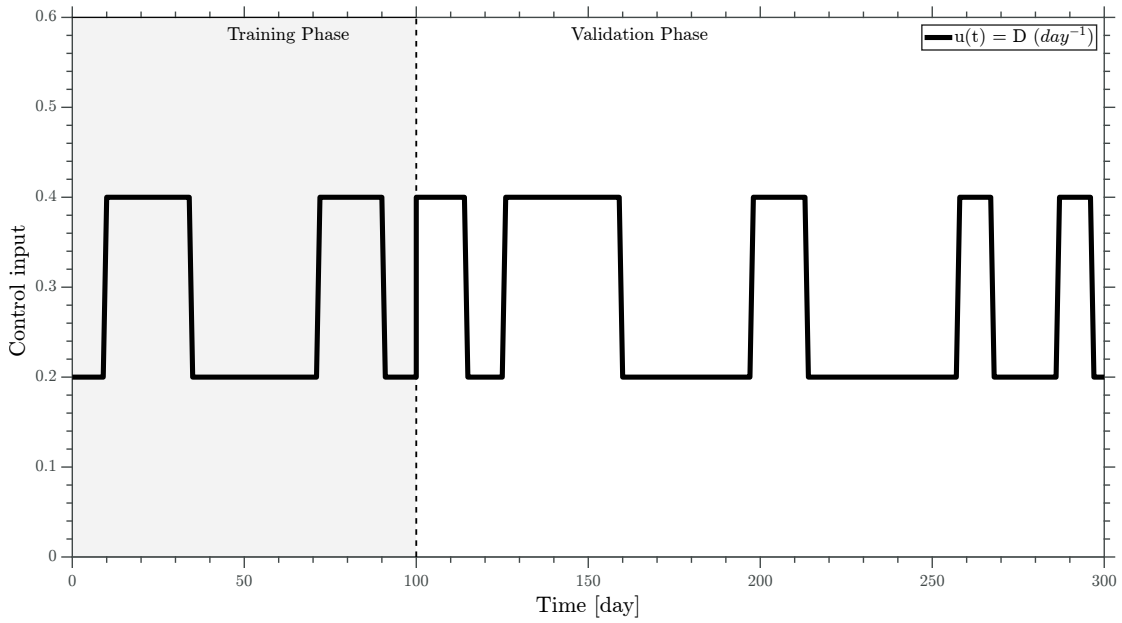

Figure 4.3: PRBS Control Input for Training and Validation

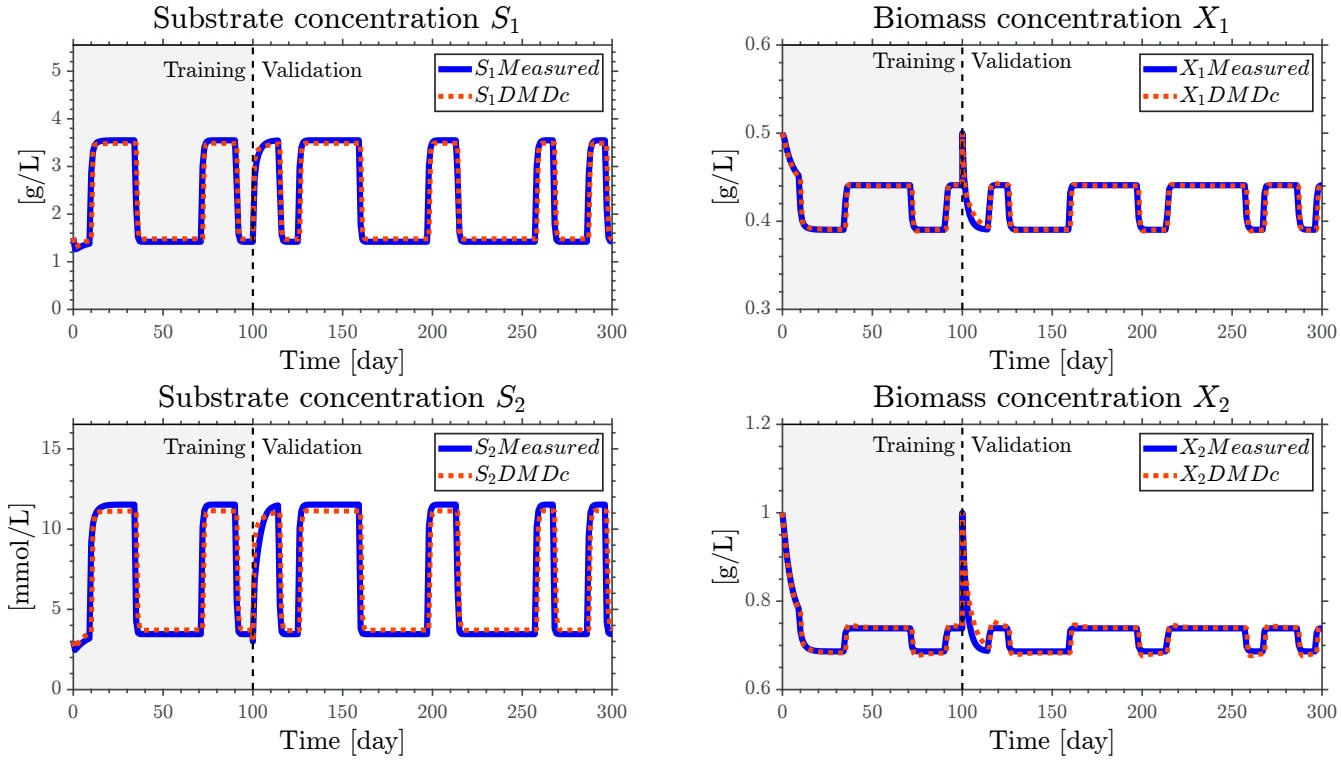

Figure 4.4: DMDc applied on the AM2 model for modeling and identification by applying a variable dilution rate.

4.4 Conclusion

In summary, this chapter provides in-depth investigation of the AM2 model within the framework of AD processes. A data-driven approach was applied to the AM2 in order to deduce a global linear state-space input-output model for controlled AD processes. Demonstrating the accuracy of the DMDc algorithm, the results illustrated the DMDc model's ability to predict AM2 dynamics even under varying control inputs, setting the stage for future utilization of this linear model for controlling AD systems through established linear control techniques. In short, this chapter presented the effectiveness of DMDc in constructing simplified models for AM2 model and highlighted their potential in enhancing control and prediction capabilities. By bridging the gap between theoretical models and real-world applications, these techniques offer new avenues for tackling intricate systems in various fields.

Chapter 5

Data-Driven Modeling, Forecasting and Controlling of the Chemostat Dynamics

5.1 Introduction

In this chapter, we began by looking at the Chemostat's basic model, which gives a basic understanding of the bioprocesses system dynamics. The model was derived from the well-known mass balance law as it is the basics of bioprocesses mathematical modeling. Then, we explore the broader application of the Koopman operator theory and its data-driven approximation. We demonstrate its efficacy in the data-driven modeling and controlling of a CSTR. This data-driven techniques offers a powerful framework for understanding the system dynamics and designing control strategies based on available data.

5.2 The Simple Mathematical Model of the Chemostat

The Chemostat also known as a Continuous Stirred-Tank Reactor (CSTR) is a continuous culture system used in microbiology and biotechnology to maintain a stable environment for microorganism growth. It consists of a bioreactor where a culture of microorganisms is continuously supplied with nutrients at a controlled flow rate while an equal volume of the culture is removed to maintain a constant volume. By regulating the nutrient supply and dilution rate, the growth rate and biomass concentration of the microorganisms can be controlled. The chemostat allows researchers to study microbial growth dynamics, nutrient utilization, and various biochemical processes under controlled conditions, making it valuable for research and industrial applications [2], [25]. Figure 5.1 provides a basic diagram illustrating a fully mixed CSTR. The typical way to describe the dynamic behavior of a specific population of microorganisms, labeled as X, that resides in a CSTR and relies on a particular limiting substrate, labeled as S, often involves employing equations derived from the well-established mass balance principle [2], [17], given

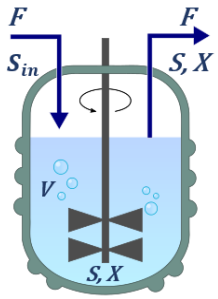

Figure 5.1: A basic schematic representation of a totally mixed Chemostat or CSTR [2]

by:

$$\begin{bmatrix}\text{Time variation}\\ \text{of the mass of}\\ \text{the component}\\ \text{in the tank}\end{bmatrix} = \begin{bmatrix}\text{Mass of the}\\ \text{component}\\ \text{that goes in}\\ \text{the tank}\end{bmatrix} - \begin{bmatrix}\text{Mass of the}\\ \text{component}\\ \text{that goes out}\\ \text{of the tank}\end{bmatrix} \pm \begin{bmatrix}\text{Mass of the}\\ \text{component}\\ \text{produced or}\\ \text{consumed}\end{bmatrix} \tag{5.1}$$

The assumption of complete mixing within the CSTR implies that the concentrations are uniformly distributed throughout the tank, meaning that the concentrations at the outlet of the reactor is the same as the concentrations at any other point within the tank.

According to (5.1), the equation representing the mass balance for the substrate S in the CSTR depicted in Figure 5.1 is expressed as follows:

$$\frac{dS}{dt} = \frac{F}{V}S_{in} - \frac{F}{V}S - k\mu(S)X \tag{5.2}$$

The mass balance of the biomass X in the CSTR is similarly given by:

$$\frac{dX}{dt} = -\frac{F}{V}X + \mu(S)X \tag{5.3}$$

Where S is the substrate concentration (g/L) , X is the biomass concentration (g/L), Sin is the input substrate concentration (g/L), k is the biomass substrate consumption yield coefficient. F and V are the flow rate (L/day) of the CSTR and its volume (L) respectively. D is the dilution rate given by $D = \frac{F}{V}$ $(1/day)$. It is important to note that S_{in} and D are regarded usually as controlled variables or inputs in the CSTR. The specific growth rate of the biomass on the substrate, denoted as $\mu(S)$, represents the rate at which the biomass increases in relation to the substrate concentration. This growth rate is influenced by various factors and can be represented by different mathematical functions.

The typical mathematical representation of the CSTR is commonly expressed as a system of interconnected ODEs, as follows:

$$\begin{cases}\dot{S} &= D(S_{in} - S) - k\mu(S)X\\ \dot{X} &= (\mu(S) - D)X\end{cases} \tag{5.4}$$

In our work, we suppose that the initial conditions S_0 and X_0 of system (5.4) are in the box $[S_0^-, S_0^+] \times [X_0^-, X_0^+] \subset \Sigma$ where the set Σ is defined by :

$$\Sigma = \{(S, X) \in \mathbb{R}^2_+ : 0 < S < S_{in} \quad \text{and} \quad X > 0\}$$

The literature widely acknowledges and establishes the well-known fact that the state variables will remain positive for all future time instances [2], [25], [17]. In relation to the kinetics $\mu(S)$ of the CSTR in (5.4), we employ two commonly used and widely accepted models to describe the growth rate of the biomass:

- **Monod Model**:

$$\mu(S) = \frac{\mu_M S}{K + S} \tag{5.5}$$

  where:

  - μ_M is the maximum growth rate.
  - K the half-saturation constant .

  The Monod model, named after Jacques Monod [143], considers the substrate concentration S as the sole limiting factor for microbial growth. It assumes that the growth rate of the biomass X is directly proportional to the substrate concentration S, with a saturation effect. This model is particularly effective for studying microorganisms that exhibit nutrient-dependent growth patterns.

  Upon substituting equation (5.5) into equation (5.4), we obtain a CSTR that exhibits Monod-type kinetics. This modified CSTR, referred to as CSTRm, is described by the following:

$$\begin{cases} \dot{S} &= D(S_{in} - S) - k\frac{\mu_M S}{K+S} X \\ \dot{X} &= (\frac{\mu_M S}{K+S} - D)X \end{cases} \tag{5.6}$$

- **Haldane Model:**

$$\mu(S) = \frac{\mu_M S}{K_s + S + \frac{S^2}{K_i}} \tag{5.7}$$

  where:

  - μ_M is the maximum growth rate in the absence of inhibition.
  - K_s the half-saturation constant.
  - K_i the inhibition constant.

  The Haldane model, developed by J.B.S. Haldane [144], takes into account additional factors such as substrate inhibition and toxic effects. It offers a more comprehensive representation of the complex dynamics observed in microbial growth, considering both substrate utilization and potential inhibitory effects. This model is particularly useful when studying systems with varying substrate concentrations and potential inhibitory compounds.

By substituting equation (5.7) into equation (5.4), we obtain a CSTR with Haldane-type kinetics (referred to as CSTRh), which can be expressed as:

$$\begin{cases} \dot{S} &= D(S_{in} - S) - k\frac{\mu_M S}{K_s + S + \frac{S^2}{K_i}} X \\ \dot{X} &= (\frac{\mu_M S}{K_s + S + \frac{S^2}{K_i}} - D)X \end{cases} \tag{5.8}$$

By employing these well-established models of the CSTRm in (5.6) and CSTRh in (5.8), researchers and scientists gain valuable insights into the behavior and kinetics of microorganisms in different environments and under varying substrate conditions. These models have widespread applications in fields such as bioprocess engineering, environmental microbiology, and pharmaceutical research, enabling better understanding, control, and optimization of biological systems. The behavior of the two CSTR models, is directly influenced by the kinetics models presented in equations (5.5) and (5.7). This relationship has been demonstrated in previous studies [2], [25], [17]. Moreover, it is important to note that the Monod model in equation (5.5) is a monotonic function and has a bounded behavior, whereas the Haldane model in equation (5.7) is non-monotonic and does not exhibit such constraints. This distinction is illustrated in Figure 5.2, emphasizing the contrasting dynamic characteristics of the two models.

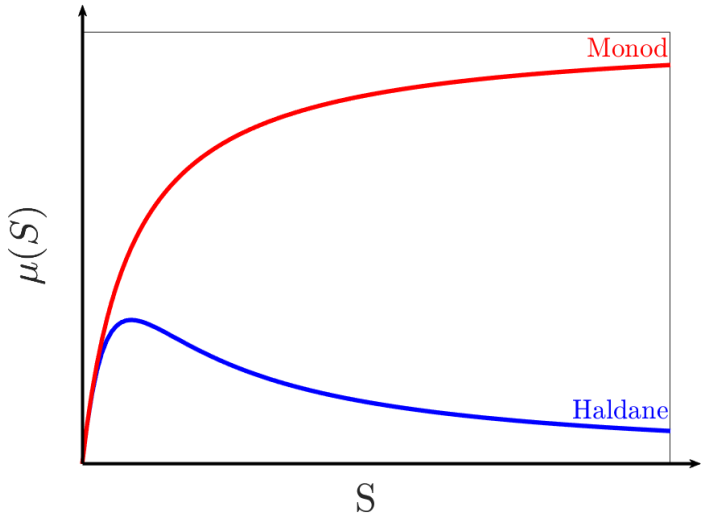

Figure 5.2: The graphs of Monod and Haldane functions

5.3 Data-Driven Modeling and Forecasting of the Chemostat Dynamics

This section focuses on forecasting Chemostat dynamics using the EDMDc data-driven approach, which was introduced in Chapter 2, Section 2.5.2. The goal is to create a data-driven model based on the Koopman operator theory , as discussed in Chapter 2, Section 2.2. This model enables the prediction of future states for

the nonlinear dynamical system of the chemostat presented in Section 5.2, solely relying on measurements of the system's states and inputs. The presented predictor exhibits a linear structure and offers various applications, including diagnostics, state estimation, future state prediction, and control of nonlinear Chemostat systems. An essential aspect of this approach is that it is entirely data-driven, simplifying the process through nonlinear data transformation (embedding) and a linear least squares problem in the embedded space. This simplicity allows for the straightforward handling of large data sets, making the method efficient and practical. In addition, by employing the EDMDc data-driven technique, we can develop a predictive model for the chemostat dynamics, leveraging the Koopman operator theory. The resulting linear structure facilitates versatile uses in understanding, predicting, and controlling nonlinear chemostat systems, making it a valuable tool in various applications.

To simplify numerical control and data storage, the continuous-time dynamical system of the Chemostat described in equation (5.4) is transformed into a discrete-time dynamics model. This conversion allows us to represent the system's behavior at specific time intervals rather than continuously. By discretizing the model, we can work with discrete data points, making it more manageable for computational purposes and enabling easier implementation of prediction and control algorithms. Additionally, this discrete-time representation facilitates data storage since it requires storing values at specific time instances, rather than continuously recording data points. Denote $\mathbf{x} = [S, X]^\top$ the state vector and $u \in [u_{Min}, u_{Max}]$ is the control input where it is considered to be the dilution rate, e.i. $D \in [D_{Min}, D_{Max}]$. Hence, the nonlinear system (5.4) can be seen as:

$$\dot{\mathbf{x}} = f(\mathbf{x}, u), \tag{5.9}$$

Using the Runge-Kutta fourth order approach as in [4], the discrete-time model of chemostat is given by:

$$x_{k+1} = x_k + \frac{T_d}{6}(k_1 + 2k_2 + 2k_3 + k_4), \tag{5.10}$$

where k_1, k_2, k_3, k_4 are given as:

$$\begin{cases} k_1 = f(t_k, x_k, u(t_k)), \\ k_2 = f(t_k + \frac{T_d}{2}, x_k + \frac{T_d}{2}k_1, u(t_k + \frac{T_d}{2})) \\ k_3 = f(t_k + \frac{T_d}{2}, x_k + \frac{T_d}{2}k_2, u(t_k + \frac{T_d}{2})) \\ k_4 = f(t_k + T_d, x_k + T_d k_3, u(t_k + T_d)) \end{cases} \tag{5.11}$$

where t_k is the time instant and T_d is the sampling time.

The discrete-time dynamical model (5.10) can be rewritten as follows:

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, u_k), \tag{5.12}$$

where $\mathbf{x}_k = [S_k, X_k]^\top$, $\mathbf{x}_k \in \Sigma \subset \mathbb{R}^2_+$ and $u \in \mathcal{U} \subset \mathbb{R}^1$.

In this scenario, we consider the system (5.12) for which we have access to full-state measurements, meaning we can directly observe the system's states, denoted as $y_k = \mathbf{x}_k$. To analyze the system's behavior more effectively, we employ a

technique called embedding, which involves mapping the original state space into a higher-dimensional space using nonlinear observable functions. By transforming the data into this higher-dimensional state space, we can apply the EDMDc algorithm to process the augmented data effectively. The main objective is to approximate the Koopman operator, which is a linear operator representing the Chemostat's underlying dynamics in the new embedded space. Through the EDMDc algorithm, we can derive the matrices A, B, and C of the linear Koopman model in least-squares sense. These matrices are crucial for the construction of the linear Koopman model that accurately describes the Chemostat's behavior in the augmented space. The obtained model can then be used for analysis, prediction, and control of the Chemostat's dynamics. Figure 5.3 describes the process of identifying a linear dynamical model of the chemostat using operational data from either the CSTRm in (5.6) or CSTRh in (5.8) within the framework of the Koopman operator theory.

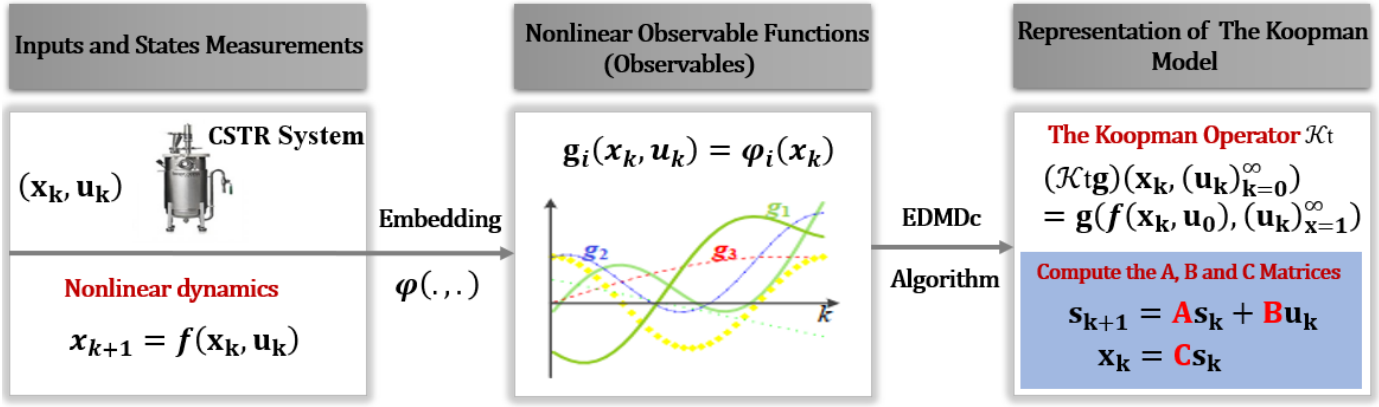

Figure 5.3: By using operational data from the Chemostat (CSTRm or CSTRh), a linear dynamical model of the chemostat can be identified. This process proceeds in two steps: **Step 1:** *The measured states of the system are lifted or embedded to a higher-dimensional state space with the help of nonlinear observable functions e.g., common basis functions in the existing literature, the thin plate spline radial basis function , the Gauss basis function, the polynomial basis function and the Fourier basis function [3], [4].* **Step 2:** *The EDMDc algorithm is applied to the lifted data. This algorithm allows for the approximation of the Koopman operator, which characterizes the underlying chemostat dynamics in the higher-dimensional space. In this step, the matrices A, B, and C are obtained using a least-squares approach, which further refines the model's accuracy.*

5.3.1 Application to the CSTR with Monod Kinetics

First, we perform the numerical simulation to apply the data-driven approach to the chemostat with Monod kinetics or CSTRm presented in (5.6). To simplify,

the latter can be rewritten as:

$$\begin{aligned} \frac{dx_1}{dt} &= u(S_{in} - x_1) - k\mu_{max}\frac{x_1}{x_1 + K}x_2 \\ \frac{dx_2}{dt} &= \mu_{max}\frac{x_1}{x_1 + K}x_2 - ux_2 \end{aligned} \tag{5.13}$$

where x_1, x_2 are the substrate and the biomass concentrations S, X, respectively and the control input u is the dilution rate D. The parameters used are given in the following table:

Parameter	Value	Unit
S_{in}	2	g/L
μ_M	1	day^{-1}
K	0.2	g/L
k	1	$--$

Table 5.1: CSTRm parameters

The CSTRm model described in equation (5.13) and with the parameter values presented in Table 5.1 exhibits a maximum of two equilibrium points. The first equilibrium point is termed the "undesired washout equilibrium" denoted as $E_0(2; 0)$, where the chemostat experiences bacterial washout. The second equilibrium point, referred to as the "desired positive equilibrium" $E_1(0.2; 1.8)$, corresponds to the normal functioning state of the chemostat, where the substrate S is degraded by biomass X. The existence and stability of these equilibria depend on the values of the operating parameters D and S_{in}. To illustrate the behavior of the system around the positive equilibrium, Figure 5.4 showcases the dynamics of the CSTRm model. In this region, the chemostat operates near the desired positive equilibrium, and the figure highlights how the system behaves under these conditions. For more comprehensive insights into the analysis of equilibria and their stability, readers are encouraged to refer to the source [25]. This reference provides detailed information on the mathematical analysis of the equilibrium points and their stability properties in the context of the Chemostat model, enabling a deeper understanding of the system's behavior and performance characteristics.

Data Collection

To collect the data, we discretize the scaled dynamics of the CSTRm in (5.13) with a descritization period $T_d = 0.1$ day. Subsequently, we conduct 200 individual trajectories of the system over 1000 sampling periods, applying a PRBS control input that is uniformly distributed within the range $[0.01, 0.5]$. In order to initiate these trajectories, we randomly generate initial conditions $x_1(0)$ and $x_2(0)$ from a uniform distribution within the box $[0, 2]^2$. This approach ensures a diverse set of starting points for the trajectories, providing a comprehensive representation of the system's behavior. The collected data is then organized into the following

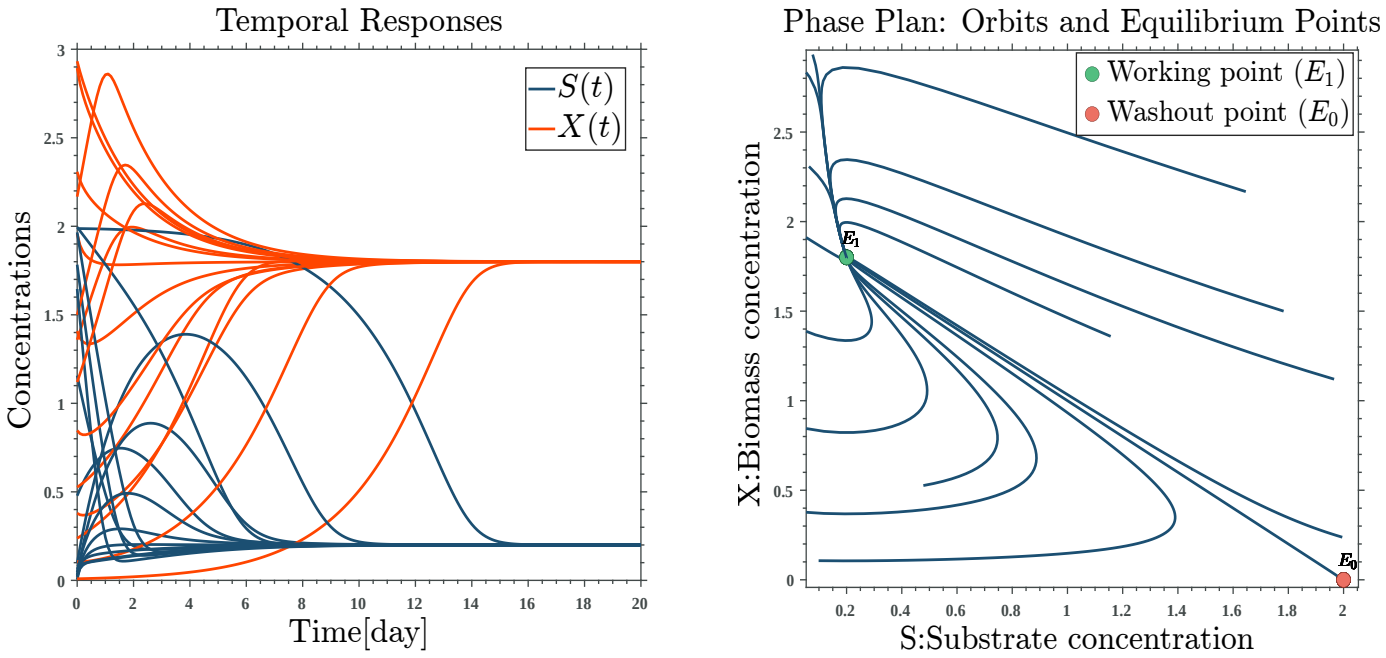

Figure 5.4: Numerical simulation of system (5.13) around the positive equilibrium. Right: the solutions $S(t)$ and $X(t)$ with respect to time, Left: the phase plan $X(t)$ with respect to $S(t)$ for different initial conditions.

three matrices:

$$\mathbf{X} = \begin{bmatrix} x_1^1 & x_1^2 & , ..., & x_1^{200000-1} \\ x_2^1 & x_2^2 & , ..., & x_2^{200000-1} \end{bmatrix}, \tag{5.14}$$

$$\mathbf{X}' = \begin{bmatrix} x_1^2 & x_1^3 & , ..., & x_1^{200000} \\ x_2^2 & x_2^3 & , ..., & x_2^{200000} \end{bmatrix}, \tag{5.15}$$

$$\boldsymbol{\Upsilon} = \begin{bmatrix} u_1 & u_2 & , ..., & u_{200000-1} \end{bmatrix} \tag{5.16}$$

Matrix $\mathbf{X}$ of size $2\times 2\cdot 10^5$ comprises the system's state measurements, capturing the state variables x_1 and x_2 over the specified trajectory periods. Matrix $\mathbf{X}'$ of size $2 \times 2 \cdot 10^5$ contains the shifted states matrix. Lastly, matrix $\boldsymbol{\Upsilon}$ of size $1 \times 2 \cdot 10^5$ contains the control inputs applied to the system during the trajectories. By employing this data collection approach, we can obtain a significant amount of information about the CSTRm model's behavior and dynamics.

Basis Functions Selection and Embedding

There are many basis functions that can be used as embedded functions or obeservables in order to augment the data matrices $\mathbf{X}$ and $\mathbf{X}'$. The Radial Basis Functions (RBF) are one of the most effective and commonly-used basis functions in the approximations of the Koopman operator from data, as recommended in [3, 4, 56]. In this work, we choose the Thin Plate Spline RBFs [145] given by:

$$\varphi(x) = \| x - x_c \|^2 \, log(\| x - x_c \|)) \tag{5.17}$$

where $x_c = x_0$ is the center and it is randomly selected with consistent distribution over $[0, 2]^2$. We choose the embedded functions φ_i to be equal to the state itself ($\varphi_1 = x_1$, $\varphi_2 = x_2$) and 30 RBFs, therefore, the dimension of the augmented state-space is $N = N_{RBF} + 2 = 32$. The lifted data matrices $\mathbf{S}$ and $\mathbf{S}'$ are given as:

$$\mathbf{S} = \boldsymbol{\varphi}(\mathbf{X}) = \begin{bmatrix} x_1^1 & x_1^2 & , ..., & x_1^{200000-1} \\ x_2^1 & x_2^2 & , ..., & x_2^{200000-1} \\ \varphi_3^1(x) & \varphi_3^2(x) & , ..., & \varphi_3^{200000-1}(x) \\ \vdots & \vdots & , ..., & \vdots \\ \varphi_{32}^1(x) & \varphi_{32}^2(x) & , ..., & \varphi_{32}^{200000-1}(x) \end{bmatrix}, \tag{5.18}$$

$$\mathbf{S}' = \boldsymbol{\varphi}(\mathbf{X}') = \begin{bmatrix} x_1^2 & x_1^3 & , ..., & x_1^{200000} \\ x_2^2 & x_2^3 & , ..., & x_2^{200000} \\ \varphi_3^2(x) & \varphi_3^3(x) & , ..., & \varphi_3^{200000}(x) \\ \vdots & \vdots & , ..., & \vdots \\ \varphi_{32}^2(x) & \varphi_{32}^3(x) & , ..., & \varphi_{32}^{200000}(x) \end{bmatrix}, \tag{5.19}$$

The matrices $\mathbf{S}$, $\mathbf{S}'$ and $\boldsymbol{\Upsilon}$ (not lifted input matrix) are used as an input for the EDMDc method presented in Chapter 2 Section 2.5.2 in order to construct the $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ matrices for the linear Koopman model of CSTRm as already illustrated in Figure 5.3.

Model Comparison

The study compares the performance of the linear Koopman model with two other predictors based on local linearizations of the CSTRm model. The local linearizations are conducted around a given initial condition x_0 and the desired positive equilibrium $E_1(0.2; 1.8)$.

The figures (Figure 5.6 and Figure 5.7) illustrate the results, demonstrating the significantly higher accuracy of the proposed Koopman model compared to the predictors based on local linearizations. To generate these predictions, a PRBS control input $u = D$ presented in Figure 5.5 is applied, with uniformly distributed in the range $[0.1, 0.43]$. The simulations start from two different initial conditions, $x_0^1 = [1; 0.5]$ (Figure 5.6) and $x_0^2 = [0.3; 1.5]$ (Figure 5.7), with the latter chosen close to the equilibrium E_1.

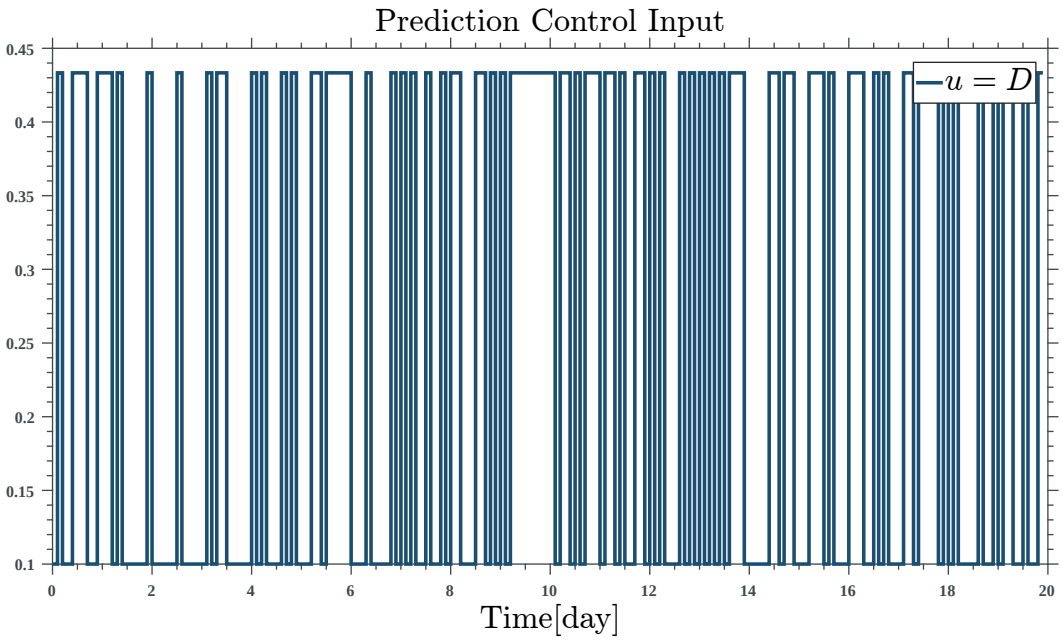

Figure 5.5: PRBS Control Input for Validation

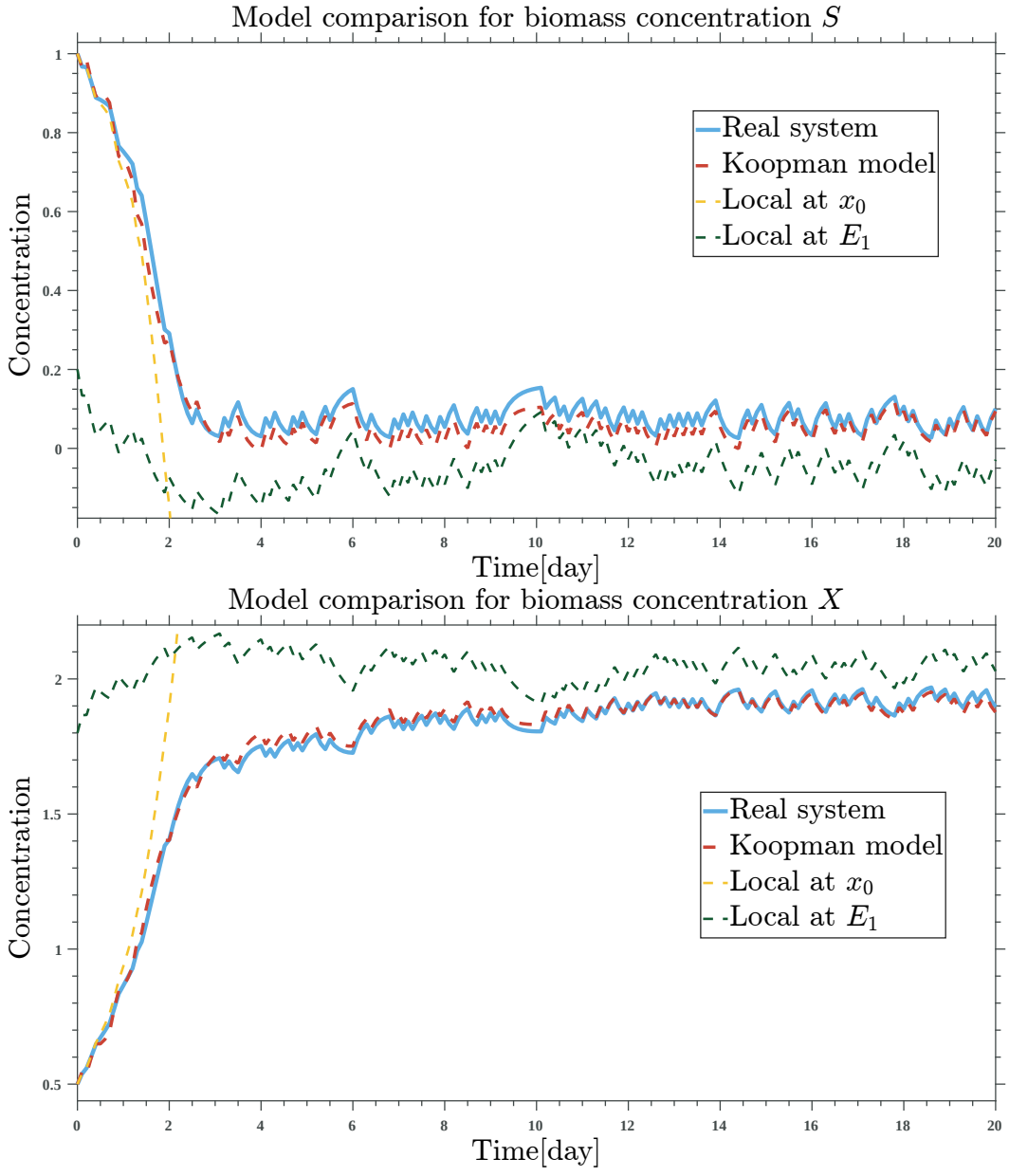

Figure 5.6: Prediction comparison for the CSTRm at initial condition $x_0^1 = [1; 0.5]$

.

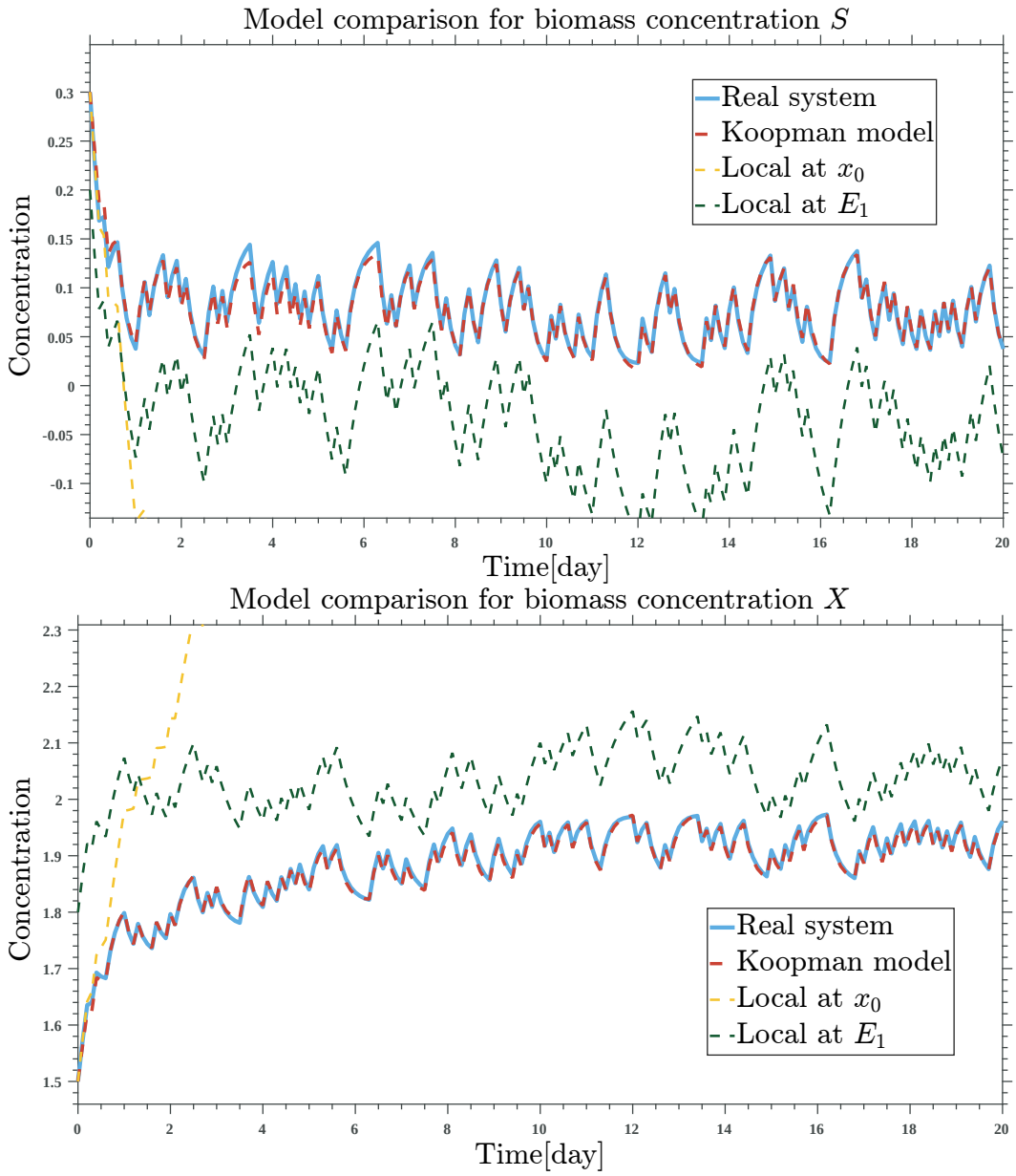

Figure 5.7: Prediction comparison for the CSTRm at initial condition $x_0^2 = [0.3; 1.5]$.

The evaluation metric used is the Relative Root Mean Square Error (RMSE)

$$RMSE = 100 \times \sqrt{\frac{\Sigma_k \parallel x_{Prediction}(kT_h) - x_{True}(kT_h) \parallel_2^2}{\Sigma_k \parallel x_{True}(kT_h) \parallel_2^2}} \tag{5.20}$$

calculated for 100 randomly chosen initial conditions. The results, presented in Table. 5.2, clearly demonstrate that the embedding-based Koopman model outperforms the predictors based on local linearizations across various initial conditions. Remarkably, even when the initial conditions are close to the equilibrium E_1, the linear Koopman model proposed in this study remains superior. Overall, the findings highlight the high accuracy and effectiveness of the linear Koopman model

Predictor	Average RMSE
Koopman Model	**2.030 %**
Local linearization at x_0	*545.449%*
Local linearization at E_1	*23.605%*

Table 5.2: Prediction comparison- average RMSE (5.20) over 100 randomly chosen initial conditions

in capturing the true dynamics of the chemostat system. The proposed Koopman model proves to be a robust and reliable predictor, demonstrating its superiority over local linearization-based predictors for accurate trajectory predictions, even in cases where the initial conditions are in the vicinity of the equilibrium point.

5.3.2 Application to the CSTR with Haldane Kinetics

Next, and by following the same simulation procedure for the CSTRm, we apply the numerical simulations to the chemostat with Haldane kinetics or CSTRh given in (5.8) and rewritten as:

$$\begin{aligned}\frac{dx_1}{dt} &= u(S_{in} - x_1) - k\mu_{max}\frac{x_1}{\frac{x_1^2}{K_i} + x_1 + K_s}x_2 \\ \frac{dx_2}{dt} &= \mu_{max}\frac{x_1}{\frac{x_1^2}{K_i} + x_1 + K_s}x_2 - ux_2\end{aligned} \tag{5.21}$$

and the parameters considered in this case are depicted in Table 5.3 below:

Parameter	Value	Unit
S_{in}	2	g/L
μ_M	5	day^{-1}
K_s	0.5	g/L
K_i	0.2	g/L
k	1	$--$

Table 5.3: CSTRh parameters

As opposed to the CSTRm in (5.13) and with the presence of the quadratic term in the denominator the CSTRh presented in (5.21) has at most three equilibrium points: $E_0(2;0)$ denoted the undesired washout equilibrium, $E_1(0.0574;1.9426)$ denoted the desired working equilibrium and $E_2(1.7426;0.2574)$ denoted the hidden equilibrium. In a manner similar to CSTRm, the existence and stability of equilibria in CSTRh are influenced by the values of operating parameters D and Sin. However, CSTRh exhibits a certain sensitivity to variations in the dilution rate D, and the presence of three equilibrium points directly hinges on the specific value of D. This characteristic introduces complexities when employing a data-driven approach to construct a linear Koopman model capable of predicting the

dynamical behavior of the Chemostat. To visually demonstrate the system's behavior, Figure 5.8 offers insights into the dynamics of the CSTRh model in (5.21). Operating in proximity to the desired positive equilibrium, the chemostat functions appropriately. However, in certain instances, if the system commences from initial conditions that are close to the undesired washout equilibrium, the latter might attract and lead to a malfunction of the chemostat system. Furthermore, the presence of a hidden equilibrium becomes evident, which could potentially give rise to future challenges. The figure effectively depicts how the system behaves under these specified conditions. The sensitivity of CSTRh to the dilution rate and the existence of multiple equilibria pose challenges when attempting to develop a linear Koopman model using a data-driven approach. Understanding the system's behavior in various scenarios is crucial, as it can lead to unforeseen issues that might hinder the predictive capabilities of the chemostat model.

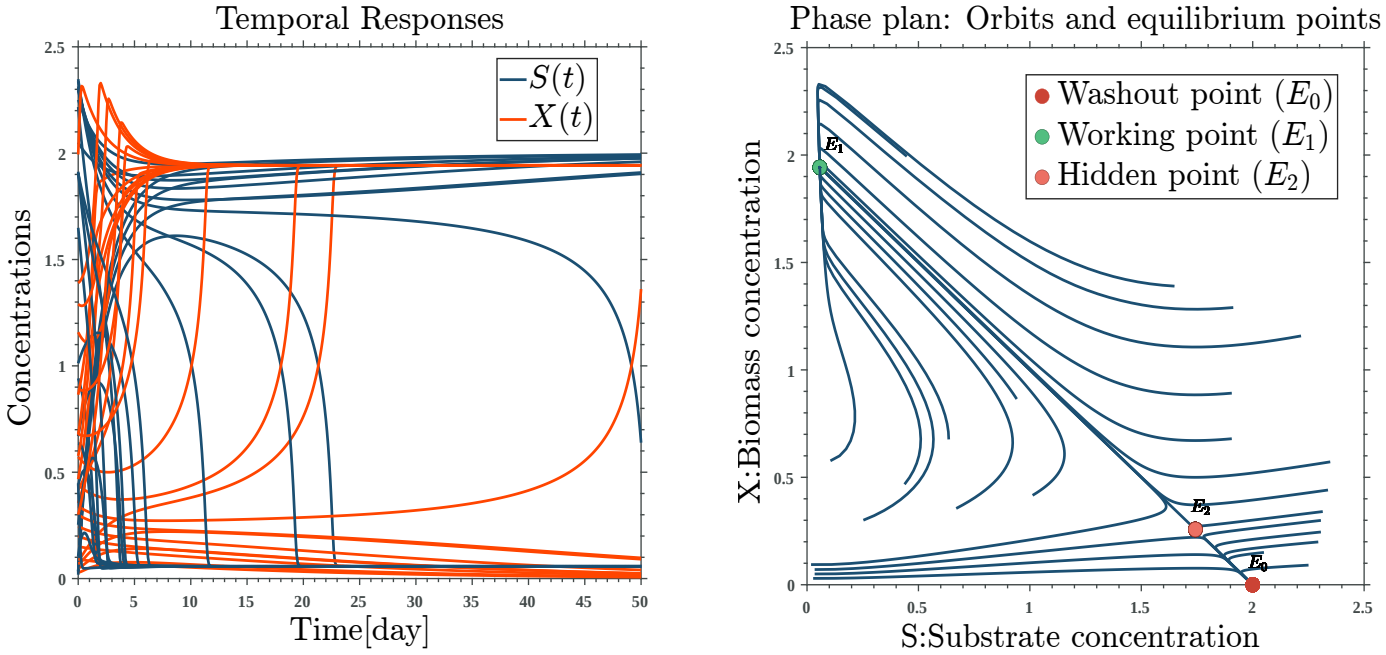

Figure 5.8: Numerical simulation of system (5.21) around the equilibrium points. Right: the solutions $S(t)$ and $X(t)$ with respect to time, Left: the phase plan $X(t)$ with respect to $S(t)$ for different initial conditions.

Data Collection

The data collection strategy was done in the same manner for the CSTRm in (5.13). The only difference is that we used a PRBS control input uniformly distributed within a broader range of $[0.01, 1]$. And the same data matrices $\mathbf{X}$ in (5.14) , $\mathbf{X}^{'}$ in (5.15) and $\boldsymbol{\Upsilon}$ in (5.16) are gathered.

Basis Functions Selection and Embedding

Similarly, the Thin Plate Spline RBF in (5.17) (one can use different RBF in order to compare) was used as embedding functions to augment the data matrices $\mathbf{X}$ and

$\mathbf{X}^{'}$. So, we choose the embedded functions φ_i to be equal to the state itself ($\varphi_1 = x_1$, $\varphi_2 = x_2$) and 50 RBFs, compared to 30 RBFs for augmenting the CSTRm data. Therefore, the dimension of the augmented state-space is $N = N_{RBF} + 2 = 52$. It is worth noting that there is no such technique to chose the number of RBFs used to augment the datasets, however, in our work, we used trial-and-error to find the desired number.

Model Comparison

The research evaluates and contrasts the effectiveness of two models in predicting the CSTRh (5.21) behavior: the construed linear Koopman model and the local linearization-based model. Similar to the CSTRm model comparison procedure, the local linearization model is performed twice, once around a specific initial condition x_0 and another around the desired positive equilibrium E_1. The study aims to understand which approach yields better results in capturing and predicting the dynamics of the system. The validation control input is shown in Figure 5.9, which is uniformly distributed in the range $[0.2, 0.7]$. Figures 5.10 and 5.11 show

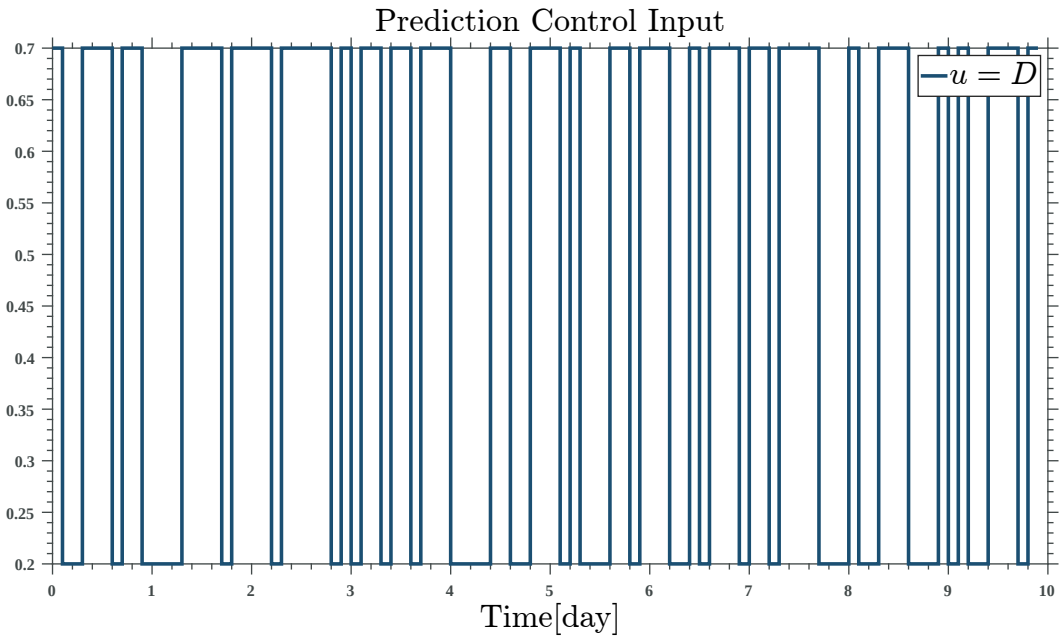

Figure 5.9: PRBS Control Input for Validation

the results, indicating the significantly higher accuracy of the proposed Koopman model compared to local linearization-based predictors. The simulations begin with two different initial conditions, $x_0^1 = [1.01; 1.5]$ presented in Figure 5.10 and $x_0^2 = [0.2; 1.9]$ presented in Figure 5.11, with x_0^2 selected near the equilibrium E_1 in accordance to the model comparison strategy of the CSTRm model. Using the RMSE in (5.20) for evaluation across 100 random initial conditions, Table 5.4 confirms the superiority of the embedding-based Koopman model over local linearization-based predictors.

Predictor	Average RMSE
Koopman Model	**2.741 %**
Local linearization at x_0	37×10^6 %
Local linearization at E_1	*16.475%*

Table 5.4: Prediction comparison- average RMSE (5.20) over 100 randomly chosen initial conditions

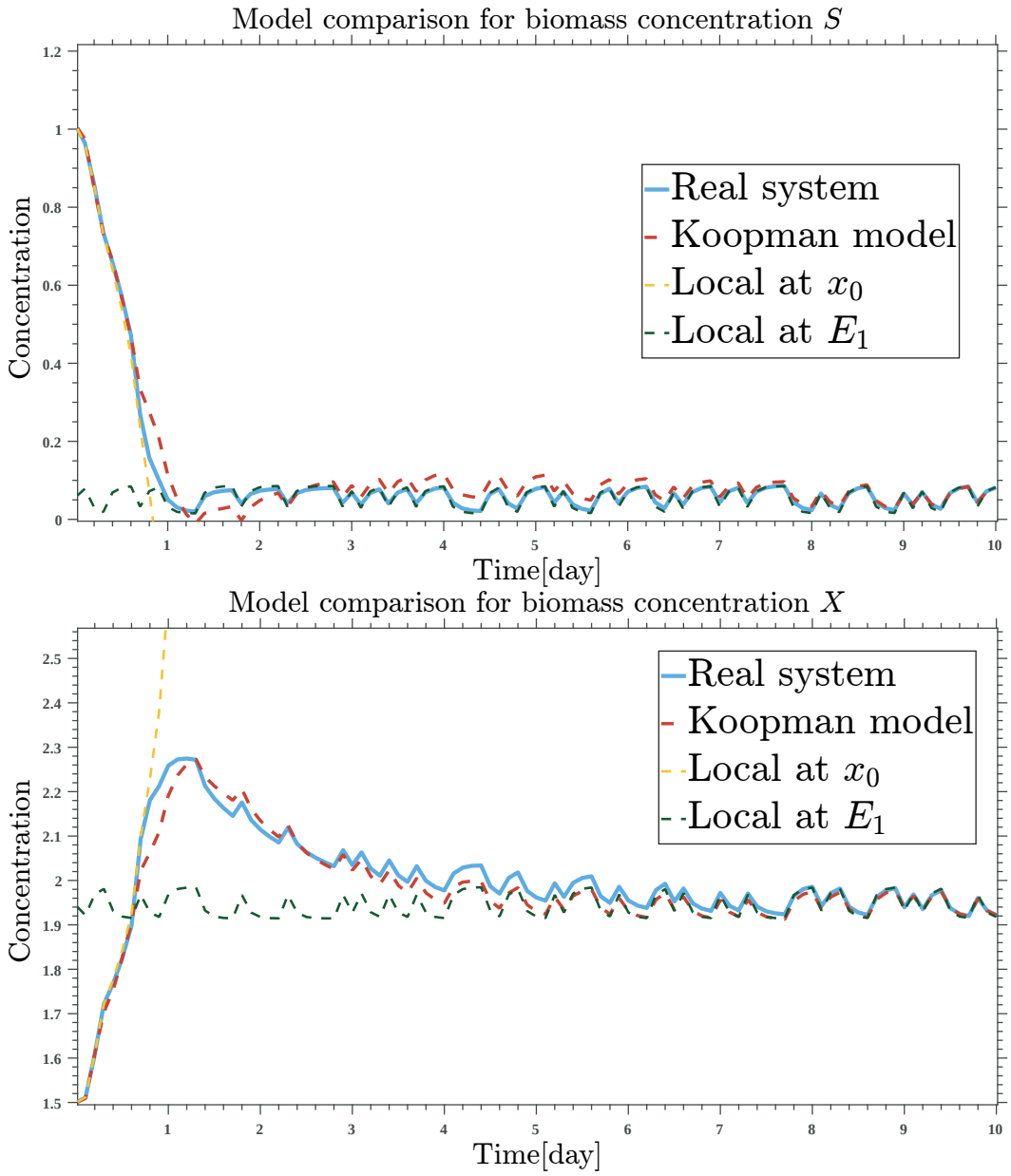

Figure 5.10: Prediction comparison for the CSTRh at initial condition $x_0^1 = [1.01; 1.5]$.

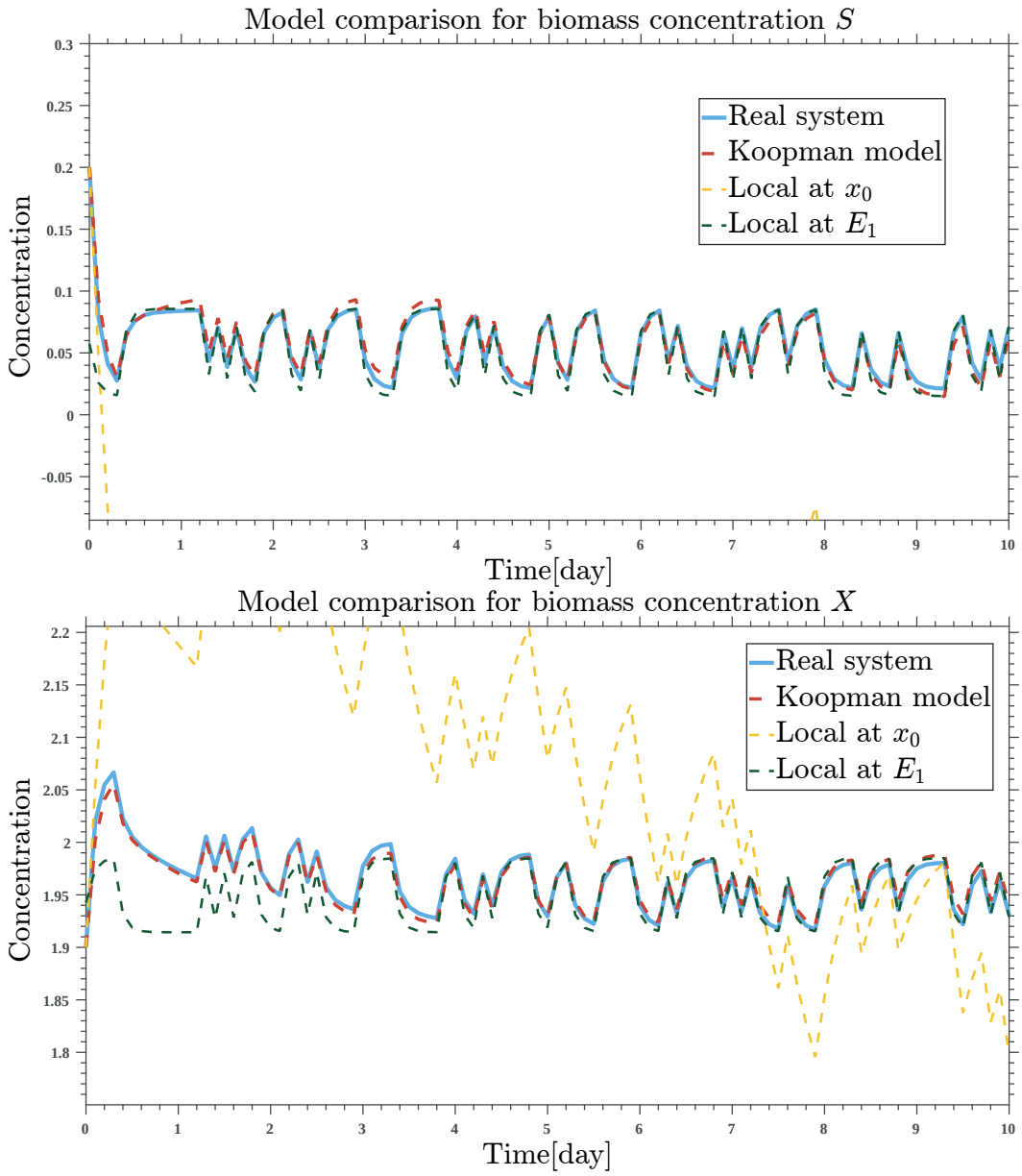

Figure 5.11: Prediction comparison for the CSTRh at initial condition $x_0^2 = [0.2; 1.9]$.

Notably, even when initial conditions approach E_1, the linear Koopman model excels. However, it's worth noting that this time the local linearization model performs particularly well around E_1 in the steady state compared to the transient state (this can be clearly noticed in Figure 5.11), where the Koopman model outperforms it. Overall, these findings highlight the Koopman model's accuracy and effectiveness in capturing the true dynamics of the chemostat system with Haldane kinetics, demonstrating its superiority over local linearization-based predictors.

5.4 Data-Driven Control of the Chemostat Using the Koopman Operator Theory

In this section, we explore the application of the Koopman operator approach to develop a linear model for a simple chemostat with Monod kinetics, comprising one substrate and one biomass (as given in (5.6)). The primary objective is to establish a data-driven control strategy for the chemostat. However, instead of having access to full-state measurements, we are constrained to only measure a specific output, which in our case is the substrate concentration ($y = S$). To address this limitation, we create a linear model using the Koopman operator solely based on the available input-output data. This model will serve as the foundation for developing a MPC scheme, as discussed in 2 Section 2.1. This MPC approach based on the Koopman model will be referred to as KMPC. The section outlines the strategy used to construct the Koopman model for the input-output chemostat, enabling its application in the MPC control scheme. Throughout the discussion, we present more theoretical aspects of implementing MPC based on the Koopman model (KMPC). To illustrate the effectiveness of the KMPC approach, numerical examples will be provided. Firstly, we compare the constructed Koopman model for the input-output chemostat with a local linearization model, both evaluated at a given initial condition x_0). Subsequently, we apply both Linear MPC and KMPC for the chemostat, considering two scenarios: one involving a large space of initial conditions, and the other focusing on the invariant manifold (where trajectories originate from initial conditions x_1^0 and x_2^0 generated randomly with a uniform distribution, subject to the constraint $x_1^0 + x_2^0 = Sin$). The section aims to demonstrate the advantages and capabilities of using the Koopman operator approach and the derived linear model in the context of MPC Control for the chemostat system.

5.4.1 Koopman model for the input-output chemostat

We examine the discrete-time representation of the nonlinear input-output dynamical system for the chemostat in the following manner:

$$\begin{aligned} x_{k+1} &= f(x_k, u_k), \\ y_k &= h(x_k), \end{aligned} \tag{5.22}$$

where the state of the system at time step k is denoted as x_k, where $x_k = [S_k \quad X_k]^\top$. The successor state of the system is x_{k+1}, and $x_k \in \Sigma \subset \mathbb{R}_+^2$, The control input at time step k is denoted as u_k, where $u_k = D$ is the dilution rate. The system's transition mapping is represented by the function f, and the measured output at time step k is denoted as y_k. In our specific case, the measured output is the substrate concentration S_k, so we have $h(x_k) = S_k$, where $h : \mathbb{R}_+^2 \to \mathbb{R}_+$. We are specifically seeking a straightforward model that exhibits a linear structure, making it well-suited for linear control design approaches, such as MPC.

We make the assumption that the Koopman model takes the form of a controlled linear dynamical system given by:

$$\begin{aligned} s_{k+1} &= As_k + Bu_k, \\ \hat{y}_k &= Cs_k, \end{aligned} \tag{5.23}$$

With $\hat{y}_k$ representing the predicted output y_k in (5.22). Importantly, in (5.23), the control input u_k remains unchanged from (5.22), allowing for linear constraints to be efficiently applied to the control inputs.

To create a valid linear model for the chemostat described in equation (5.22), it is necessary to adhere to the steps outlined in the Koopman operator framework as discussed in Chapter 2 Section 2.5.2. The approach involves utilizing the EDMDc method to approximate the Koopman Operator for the input-output dynamical system of the chemostat. However, there is a notable difference: the selection of lifting functions depends on both the present measured output and a collection of past measured outputs and inputs. This approach, known as *Time-delayed embedding*, entails incorporating multiple consecutive measurements of the output into a single data point. While this technique has long been a powerful tool in system identification theory, it has more recently been integrated into the Koopman operator's approximations as well (as seen in references [56], [39], [38], for example). Let us consider the following set of data matrices:

$$\mathbf{X} = [\boldsymbol{\zeta}_1, .., \boldsymbol{\zeta}_K], \quad \mathbf{Y} = [\boldsymbol{\zeta}_1^+, .., \boldsymbol{\zeta}_K^+], \quad \mathbf{U} = [u_1, .., u_K] \tag{5.24}$$

In this scenario, the matrices consist of a sequence of samples from a series with a length of $n_d + 1$ (where n_d represents the number of delays), where

$$\begin{aligned} \boldsymbol{\zeta}_i &= [y_{i,n_d}^\top \quad \hat{u}_{i,n_d-1}^\top \quad y_{i,n_d-1}^\top \quad \dots \quad \hat{u}_{i,0}^\top \quad y_{i,0}^\top]^\top \in \mathbb{R}^{(n_d+1)n_h+n_d} \\ \boldsymbol{\zeta}_i^+ &= [y_{i,n_d+1}^\top \quad \hat{u}_{i,n_d}^\top \quad y_{i,n_d}^\top \quad \dots \quad \hat{u}_{i,1}^\top \quad y_{i,1}^\top]^\top \in \mathbb{R}^{(n_d+1)n_h+n_d} \\ u_i &= \hat{u}_{i,n_d} \end{aligned} \tag{5.25}$$

$(\hat{u}_{i,j})_{j=0}^{n_d}$ represents a series of inputs that generate a vector $(y_{i,j})_{j=0}^{n_d+1}$ consisting of successive measurements for the output.

The data matrices $\mathbf{X}, \mathbf{Y}, \mathbf{U}$ given in (5.24) are augmented with the help of the following embedded function or obsevables:

$$\boldsymbol{\varphi}(\boldsymbol{\zeta}) = [\varphi_1(\boldsymbol{\zeta}), \dots, \varphi_N(\boldsymbol{\zeta})]^\top, \tag{5.26}$$

Then, new sets of augmented data matrices are obtained as follows:

$$\mathbf{X}_{embed} = [\boldsymbol{\varphi}(\boldsymbol{\zeta}_1), .., \boldsymbol{\varphi}(\boldsymbol{\zeta}_K)], \mathbf{Y}_{embed} = [\boldsymbol{\varphi}(\boldsymbol{\zeta}_1^+), .., \boldsymbol{\varphi}(\boldsymbol{\zeta}_K^+)], \mathbf{U} = [u_1, .., u_K] \tag{5.27}$$

By having $\mathbf{X}_{embed}, \mathbf{Y}_{embed}$ and the input matrix $\mathbf{U}$, we can find the linear matrices $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$ of (5.23) by solving the following least squares problems:

$$\min_{\mathbf{A},\mathbf{B}} \parallel \mathbf{Y}_{embed} - \mathbf{A}\mathbf{X}_{embed} - \mathbf{B}\mathbf{U} \parallel_F, \quad \min_{\mathbf{C}} \parallel [y_{1,n_d} \dots y_{K,n_d}] - \mathbf{C}\mathbf{X}_{embed} \parallel_F, \tag{5.28}$$

The linear Koopman model in (5.23) starts from the initial condition:

$$s_0 = \boldsymbol{\varphi}(\boldsymbol{\zeta_0}) \tag{5.29}$$

Where

$$\boldsymbol{\zeta_0} = [y_0^\top \quad \hat{u}_{-1}^\top \quad y_{-1}^\top \quad \dots \quad \hat{u}_{-n_d}^\top \, y_{-n_d}^\top]^\top \tag{5.30}$$

is the vector of n_d input measurements and $n_d + 1$ are the latest measurements of the output.

5.4.2 Model predictive control based on the Koopman model (KMPC)

In the preceding section, we introduced an approach enabling us to construct a model for the input-output chemostat as a linear dynamic system (described by equation (5.23)). In the upcoming section, we will employ MPC utilizing this linear model to control the initial nonlinear chemostat system described in (5.22). This concept is illustrated in Figure 5.12.

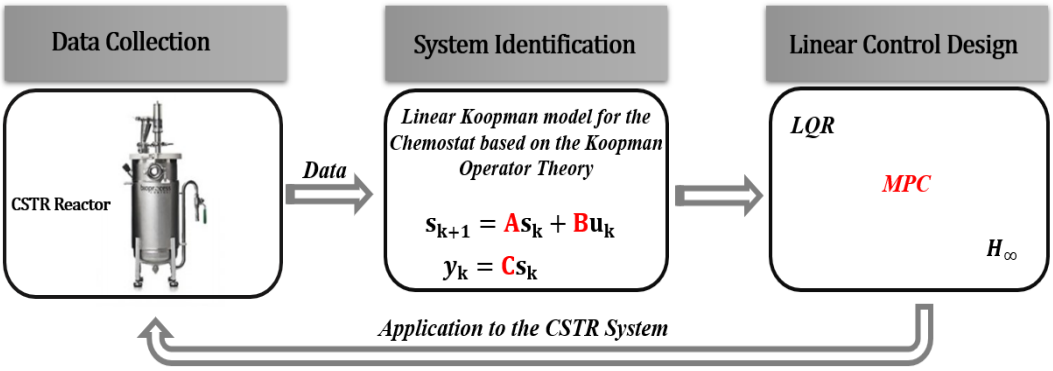

Figure 5.12: Linear forecast for the nonlinear controlled dynamical system of the cheomstat. This forecast can then be used in linear control design as linear MPC.

MPC is a popular technique for controlling complex systems with constraints as already shown in Chapter 2, Section 2.5.1. In MPC, we calculate the control inputs by solving a problem that aims to minimize a cost, which is set by the user, over a future time horizon. This happens at every step during the system's operation. Linear MPC deals with relatively simple optimization problems, allowing for quick control input assessment, often by solving a Quadratic Program (QP). On the other hand, nonlinear MPC handles more complex optimization challenges. This involves solving a non-convex optimization problem at each step, which can be computationally intensive. To manage this complexity, local optimization methods are usually used.

The KMPC method follows a strategy known as "receding horizon control" Here's how it works:

- First, the linear model (5.23) is utilized to predict how the system will evolve over a defined prediction horizon.

- Using these predictions, an optimal sequence of control inputs is calculated. This sequence is designed to minimize the given cost function within the prediction horizon.
- However, only the first element of this calculated sequence is applied to the actual system. This generates a new output, and the entire process is repeated.

In the following, we'll introduce some notations and mathematical background related to this technique. The key advantage of using the Koopman linear model (5.23) is that even though we start with a nonlinear dynamical system, the resulting MPC problem becomes convex QP. The so called "*dense form*" approach [56] is employed, which ensures that the solution to the complex QP is not affected by the potentially large dimension of the variable "**s**" of the Koopman model in (5.23). This simplification enables the QP to be efficiently solved using existing optimized solvers for linear MPC, such as qpOASES [146].

The optimization problem tackled by KMPC in every step of the closed-loop process can be outlined as follows:

$$\begin{aligned}(u_i^\star)_{i=0}^{N_h-1}, (y_i^\star)_{i=0}^{N_h} = \arg\min\, & J((u_i)_{i=0}^{N_h-1}, (y_i)_{i=0}^{N_h}) \\ \text{s.t.} \quad & s_{i+1} = As_i + Bu_i, \qquad i = 0, ..., N_h \\ & y_i = Cs_i \\ & W_i^y y_i + W_i^u u_i \leq v_i, \qquad i = 0, ..., N_h - 1 \\ & W_{N_h} y_{N_h} \leq v_{N_h} \\ & s_0 = \boldsymbol{\varphi}(\boldsymbol{\zeta_k}),\end{aligned} \tag{5.31}$$

where N_h represents the prediction horizon, and $(u_i)_{i=0}^{N_h-1}$ and $(y_i)_{i=0}^{N_h}$ form sequences of input and output values spanning the duration of N_h. The matrices $W^y_{i=0,...,N_h-1}, W^u_{i=0,...,N_h-1}\, W_{N_h}$ and the vector v_i define constraints for the state and input variables. The present state $\boldsymbol{\zeta_k}$ of the delayed-state $\boldsymbol{\zeta}$ at the time instant k is expressed as follows:

$$\boldsymbol{\zeta_k} = [y_k^\top \;\; u_{k-1}^\top \;\; y_{k-1}^\top \;\; \dots \;\; u_{k-n_d}^\top \, y_{k-n_d}^\top]^\top \tag{5.32}$$

The cost function J has a convex quadratic form given by:

$$\begin{aligned}J((u_i)_{i=0}^{N_h-1}, (y_i)_{i=0}^{N_h}) = \; & y_{N_h}^\top Q_{N_h} y_{N_h} + q^\top y_{N_h} \\ & + \sum_{i=1}^{N_h-1} (y_i)^\top Q_i y_i) + u_i^\top R_i u_i + q_i^\top y_i + r_i^\top u_i \\ & + u_0^\top R u_0 + r_0^\top u_0,\end{aligned} \tag{5.33}$$

The optimization problem above incorporates matrices $R_{i=0,...,N_h-1}, Q_{i=0,...,N_h}$. These matrices are real, symmetric, and positive semi-definite, serving as components of the cost function. The function J defined by these matrices can accommodate various control objectives, such as achieving time-varying reference tracking. It's worth noting that nonlinear constraints and objectives in the original state x_k can

be managed by introducing nonlinear functions into the embedding variables (as discussed in [56]).

Once the optimal input sequence $(u_i^\star)_{i=0}^{N_h-1}$ is computed, its first element $u_0^\star$ is applied to the system, producing a new measured output value $(h(x_k))$. This update refreshes the current state $\boldsymbol{\zeta_k}$, and the entire process is repeated in a receding horizon manner. The entire process of the KMPC approach in closed-loop operation is outlined in Algorithm 3, and the KMPC framework for identification and control of the chemostat is visualized in Figure 5.13.

Algorithm 3: Closed-loop operation of KMPC

for k =0,1,... **do**
  Set $s_0 = \boldsymbol{\varphi}(\boldsymbol{\zeta_k})$
  Solve (5.31) to get an optimal solution $(u_i^\star)_{i=1}^{N_h}$
  Apply $u_1^\star$ to the nonlinear system of the chemostat (5.22)
end for

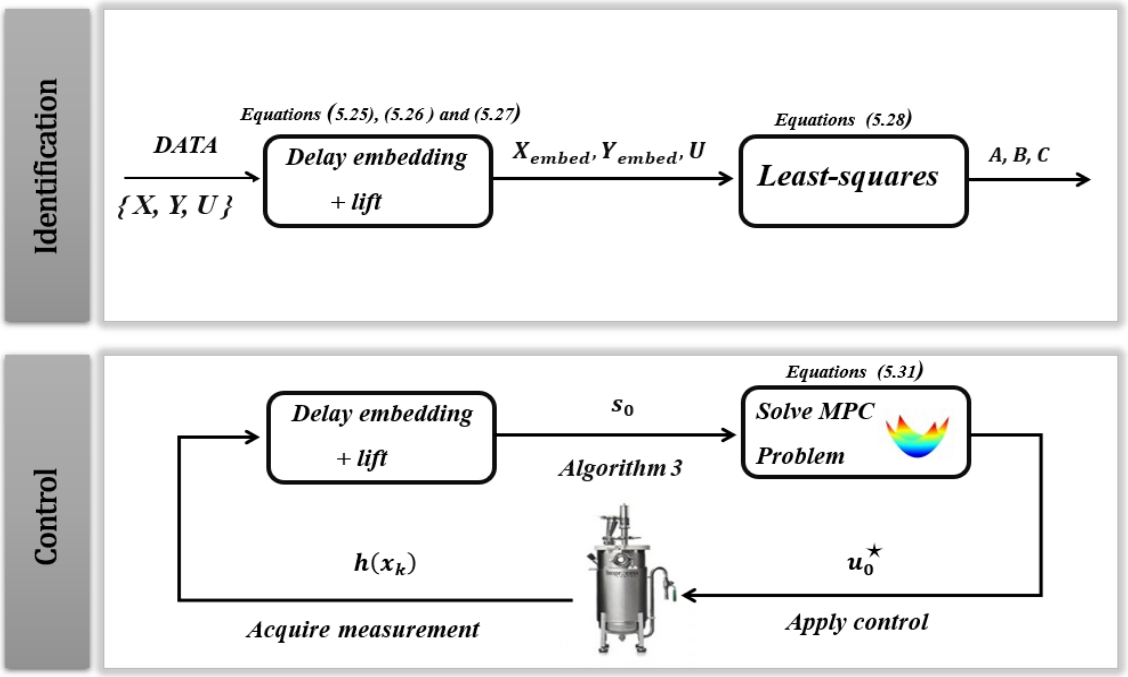

Figure 5.13: Schematic representation of KMPC framework for identification and closed-loop control of nonlinear system of the chemostat.

5.4.3 Simulation results and discussions

In this section we extend the proposed data-driven technique to control the nonlinear input-output dynamical system of the chemostat (5.22) with a Monod kinetics

as in (5.6)) rewritten bellow:

$$\begin{aligned} \frac{dx_1}{dt} &= u(S_{in} - x_1) - k\mu_{max}\frac{x_1}{x_1 + K}x_2 \\ \frac{dx_2}{dt} &= \mu_{max}\frac{x_1}{x_1 + K}x_2 - ux_2 \end{aligned} \tag{5.34}$$

where the parameters used in this case for simulating the CSTRm model are given in the Table 5.5 below:

Parameter	Value	Unit
S_{in}	20	g/L
μ_M	0.6	day^{-1}
K	3	g/L
k	1	$--$
$u = D$	$[0.01, 0.5]$	$day^{-}1$

Table 5.5: Parameters of the chemostat dynamics with Monod kinetics CSTRm

The control of the chemostat dynamics is always restricted in the large space of initial conditions. Rather, it is usually controlled in the invariant manifold (i.e., the trajectories begin from initial conditions $x_1(0), x_2(0)$ generated randomly with uniform distribution where $x_1(0) + x_2(0) = S_{in}$). In our work, we address both cases. The aim is to construct a MPC controller based on the Koopman operator framework presented in Section 5.4.1 which means that we do not have full-state measurements anymore, and we have access to only input-output data of the dynamical system of the chemostat (5.22). First, we compare the prediction accuracy of the output of the linear Koopman model (5.22) with the output of the true dynamics of the chemostat and a model based on local linearization of the chemostat at a given initial condition x_0. Then we use the MPC controller for the feedback control of the chemostat based only on input-output data available, without any information about the chemostat model. We establish a comparison between MPC controller based on the Koopman model presented in this study (KMPC) and a local linearization-based MPC (LMPC).

In terms of collecting the set of data as in (5.24), we will use the same strategy as before by considering the chemostat model in (5.22) as a virtual bioreactor generating this data and we use it for constructing the Koopman model (5.23). We are going to start with the general and difficult case (control in a large space of the initial conditions) then use the same strategy for the special case (control in the invariant manifold).

The case of a large space of initial conditions

In the first instance, we try to model and control our system in a larger interval of the initial condition. The idea is to consider the dynamical system of the CSTRm in (5.34) as a virtual bioreactor generating the set of data presented in (5.24) (the

data can be collected from a real chemostat system, if available). Then, we use this data for the construction of the Koopman model (5.23). To collect the data in (5.24), the discretization period is chosen to be $T_d = 0.1day$ (since the dynamics of the chemostat are evolving slowly due to the consumption of the substrate by the biomass) and we simulate 200 trajectories over 1000 sampling periods with a random control input ($u = D$) signal equally distributed. Furthermore, we choose that the trajectories begin from initial conditions $x_1(0), x_2(0)$ generated randomly with uniform distribution where $x_1(0) \in [10, 20[$ and $x_2(0) \in [2, 10]$. We choose the number of delays $n_d = 1$ and the embedded lifting functions $\boldsymbol{\varphi}_i = S_k$ taken to form the Time-Delayed vector $\boldsymbol{\zeta} \in \mathbb{R}^3$ in (5.25). In this work, we choose the Inverse Multi-Quadric Radial Basis Function (IMQ-RBF) given by:

$$\varphi(x) = \frac{1}{\sqrt{1+ \parallel x - x_c \parallel^2}} \tag{5.35}$$

where x_c is the center and it is randomly selected with consistent distribution over $[0, 20]^3$. In order to augment the sets of data in (5.24) we used 60 IMQ-RBFs, therefore, the dimension of the augmented state-space is $N = 63$.

Koopman model construction and comparison

Figure 5.14 shows the accuracy of the output forecast of the Koopman model (5.23) constructed only from input-output data generated from two randomly chosen initial conditions ($x_0^1 = [10, 10]^\top, x_0^2 = [12, 5]^\top$), and how it fits the real system of the chemostat compared to the local linerization-based model. In Table 5.6 we demonstrate the superiority of the Koopman model prediction accuracy over the linerization-based model for longer prediction times and several initial conditions. This is done by the RMSE (5.20) averaged over thirty-days forecast horizon over one hundred randomly selected initial conditions (in both Figure 5.14 and Table 5.6, we have applied a PRBS control input $u = [0.1, 0.35]$ for each initial condition anew). In Table 5.7, we illustrate that for one hundred samples of x_1 and x_2 starting from one hundred randomly chosen initial conditions (large space of the initial conditions), we always have $x_1 + x_2 = S_{in}$.

Model	**Average RMSE**
Koopman Model	**11.68 %**
Local linearization at x_0	4.72×10^4 %

Table 5.6: Forecast comparison-RMSE (5.20) for 100 randomly chosen initial conditions

Feedback Control of the Chemostat

The control objective is to track a given substrate concentration reference y_{Ref}, which means that we need to minimize the following objective function of the

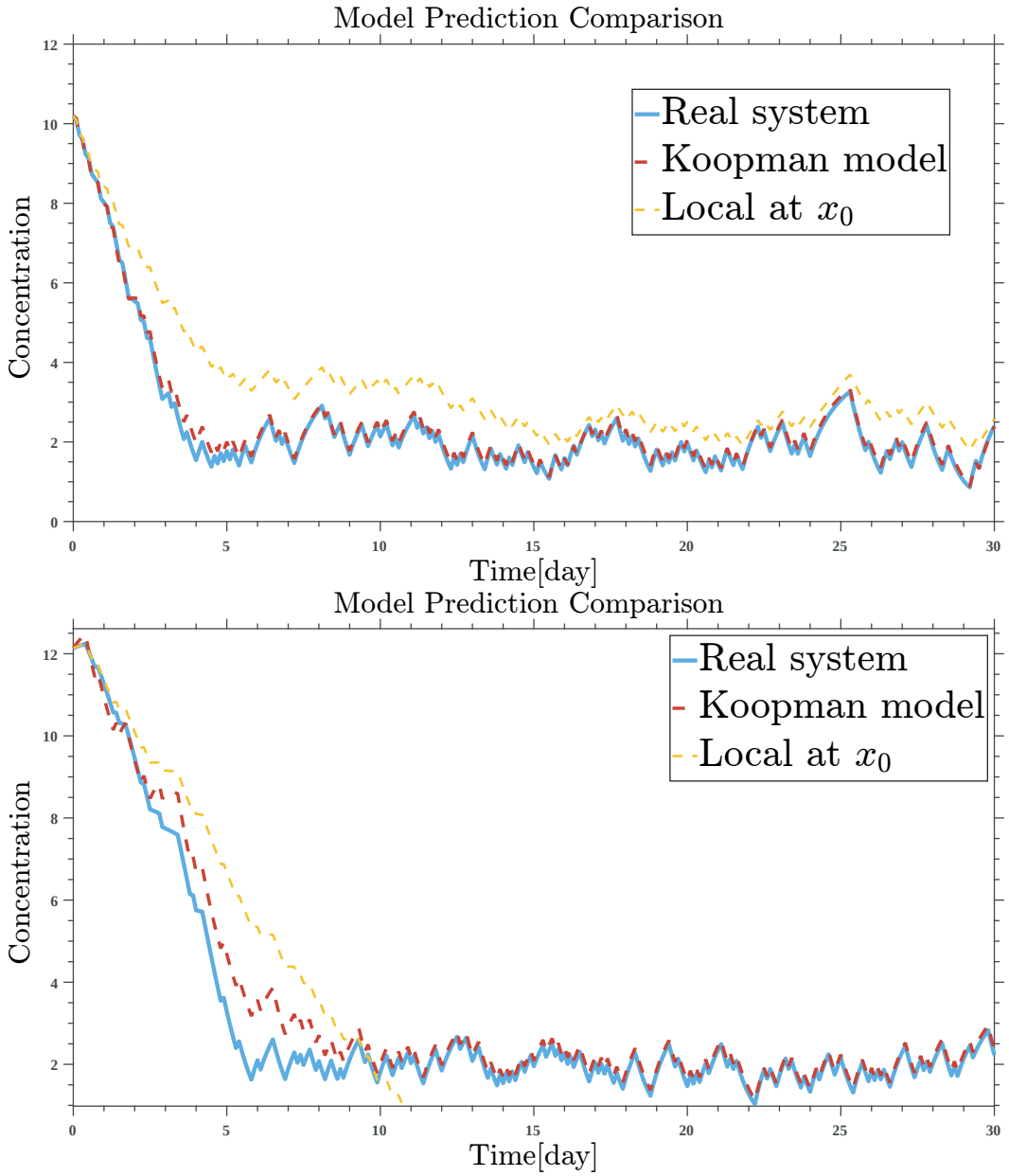

Figure 5.14: The case of larger space of the initial condition - Model prediction comparison of the chemostat - $y_k = S_k$. Up: initial condition $x_0[10, 10]^\top$. Down: initial condition $x_0[12, 5]^\top$

Day	$x_1(0)$	$x_2(0)$	x_1	x_2	$x_1 + x_2$
1	12.3341	3.2453	2.5118	17.4881	20
2	10.2993	7.4390	2.0463	17.9536	20
3	18.4034	3.3054	2.0040	17.9959	20
4	19.3560	8.8671	1.5382	18.4617	20
⋮	⋮	⋮	⋮	⋮	⋮
57	13.2423	6.5340	1.9446	18.0553	20
58	10.8691	6.0239	2.8650	17.1349	20
59	17.8812	3.5623	1.9103	18.0896	20
60	17.6888	3.3508	1.8316	18.1684	20
⋮	⋮	⋮	⋮	⋮	⋮
97	14.2234	7.0464	1.5124	18.4875	20
98	17.7434	7.9064	2.9247	17.0752	20
99	17.9543	6.4402	1.5996	18.4003	20
100	12.0606	5.5628	2.3772	17.6227	20

Table 5.7: Table of initial condition $x_1(0), x_2(0)$ and x_1, x_2 over averaged one hundred days forecast horizon.

MPC problem:

$$\begin{aligned} J &= (y_{N_h} - y_{Ref})^\top Q_{N_h} (y_{N_h} - y_{Ref}) \\ &+ \sum_{i=1}^{N_h - 1} (y_i - y_{Ref})^\top Q (y_i - y_{Ref}) + u_i^\top R u_i, \end{aligned} \tag{5.36}$$

Where cost functions matrices were chosen as $Q = Q_{N_h} = 10$ and $R = 0.01$ with a prediction horizon $N_h = 10$(i.e.,one day). We establish a comparison between MPC controller based on the Koopman model presented in this study (Koopman-MPC) and a local linearization-based MPC (Linear-MPC) in two cases. First, we track a reference constant ($y_{Ref} = S_k^{Constant}$) where we want to either maximize or minimize the output which is the substrate concentration S_k. With no constraints imposed on the output but we have imposed constraints on the control input $u \in [0.2, 3]$, simulation results are presented in Figure 5.15 and 5.16, we can see that the tracking performance in the first case are almost identical. Both KMPC and LMPC reach the desired output in the same time and manner Although the LMPC is slightly superior to the KMPC (this is clearly noticeable in the simulation where the reference tracking is $y_{Ref} = 10$ and the initial condition is $x_0 = [5, 17]^\top$). This is due to the fact that the LMPC controller needs a model for the local linearization computation and full measurements of the state. On the other hand, the KMPC controller has the advantage of being entirely *data-driven* with only *output measurements* being required. In the second case, we track a time-varying function $y_{Ref}(t) = 17cos(2\pi t/300)$ with constraints imposed on the output where $y \in [2, 15]$. The simulation results are shown in Figure 5.17; we can observe a good reference tracking without any violation of the output constraints

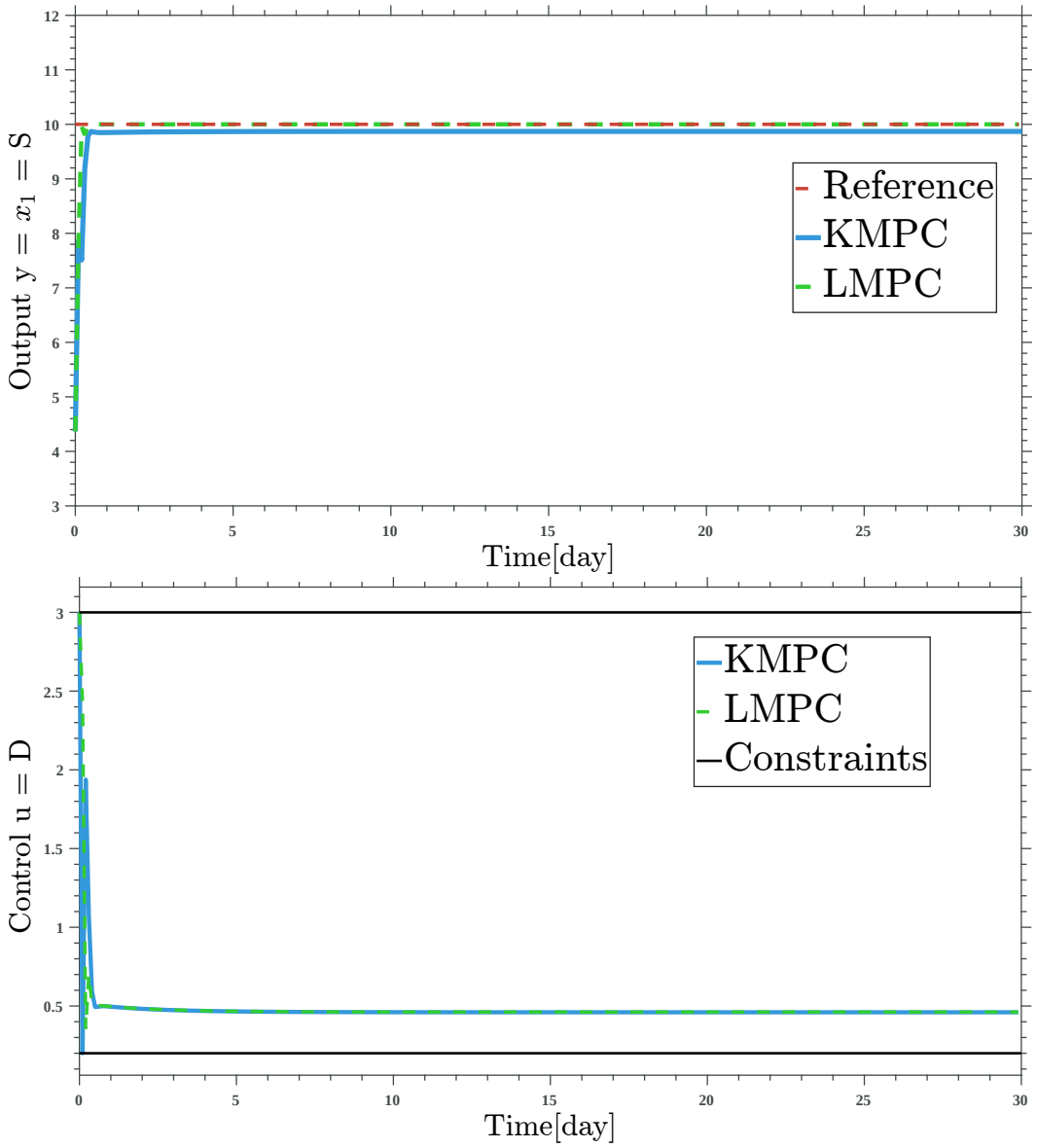

Figure 5.15: Control in larger space of the initial condition -Feedback control of the chemostat- constant reference tracking $y_{Ref} = 10$ with no state constraints and constrained control input $u \in [0.2, 3]$. Initial condition $x_0 = [5, 17]^\top$.

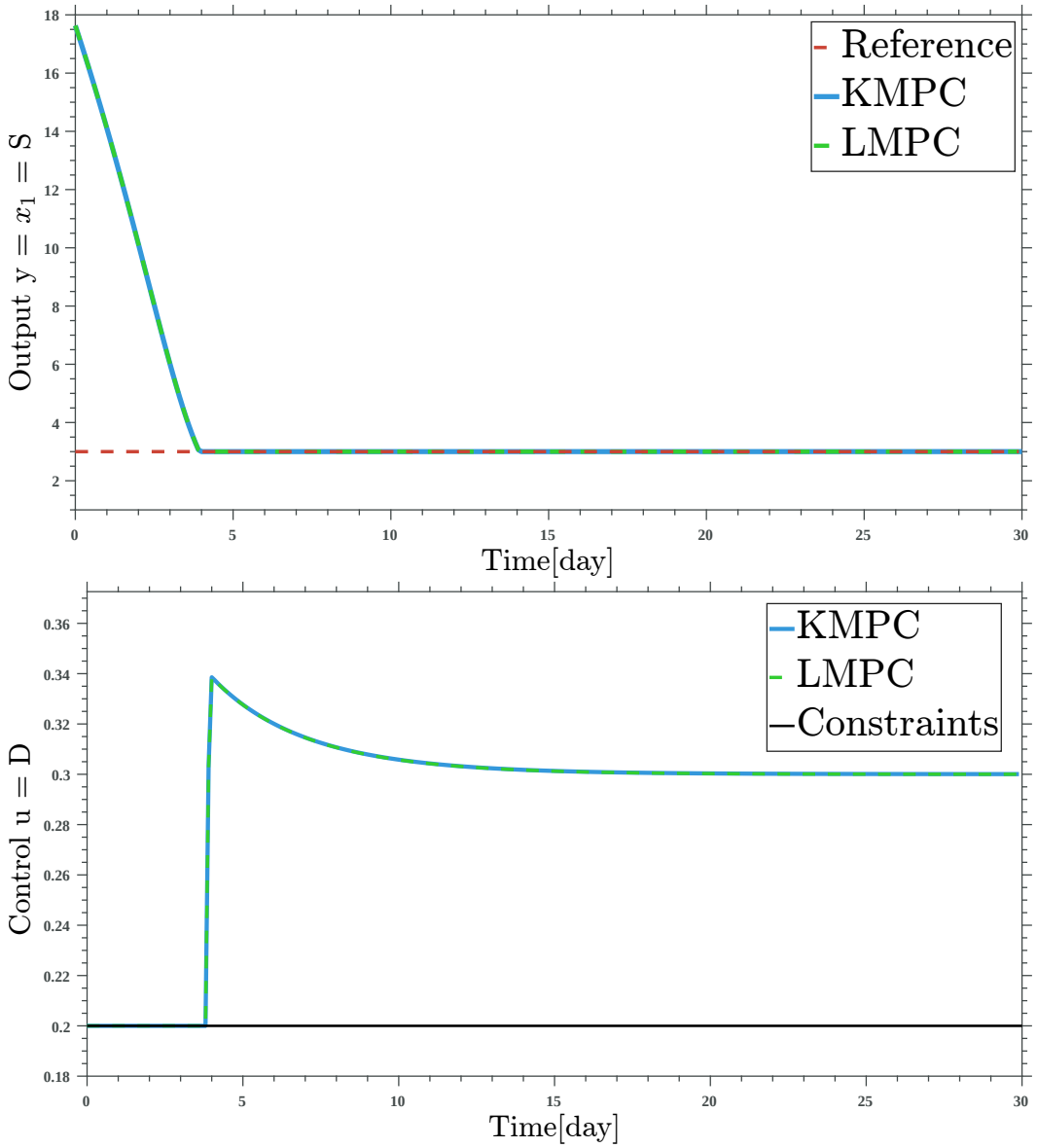

Figure 5.16: Control in larger space of the initial condition -Feedback control of the chemostat- constant reference tracking $y_{Ref} = 3$ with no state constraints and constrained control input $u \in [0.2, 3]$. Initial condition $x_0 = [18, 7]^\top$.

with a constrained control input applied ($u \in [0.2, 3]$) for the KMPC controller, however, the LMPC controller turns out to be infeasible and therefore it stops before continuing the entire simulation period (in predictive control, infeasibility is a common issue that occurs as a result of different trial-and-error or theoretically backed-up methods, for more details see [65],[67]). The inaccuracy of the forecast of the local linearization-based model over long prediction horizons is causing the infeasibility of the LMPC controller. The KMPC controller, nevertheless, is not only *data-driven* but it can also transcend the infeasibility problem and continue the simulation until the end. In terms of the underlying optimization issue, all

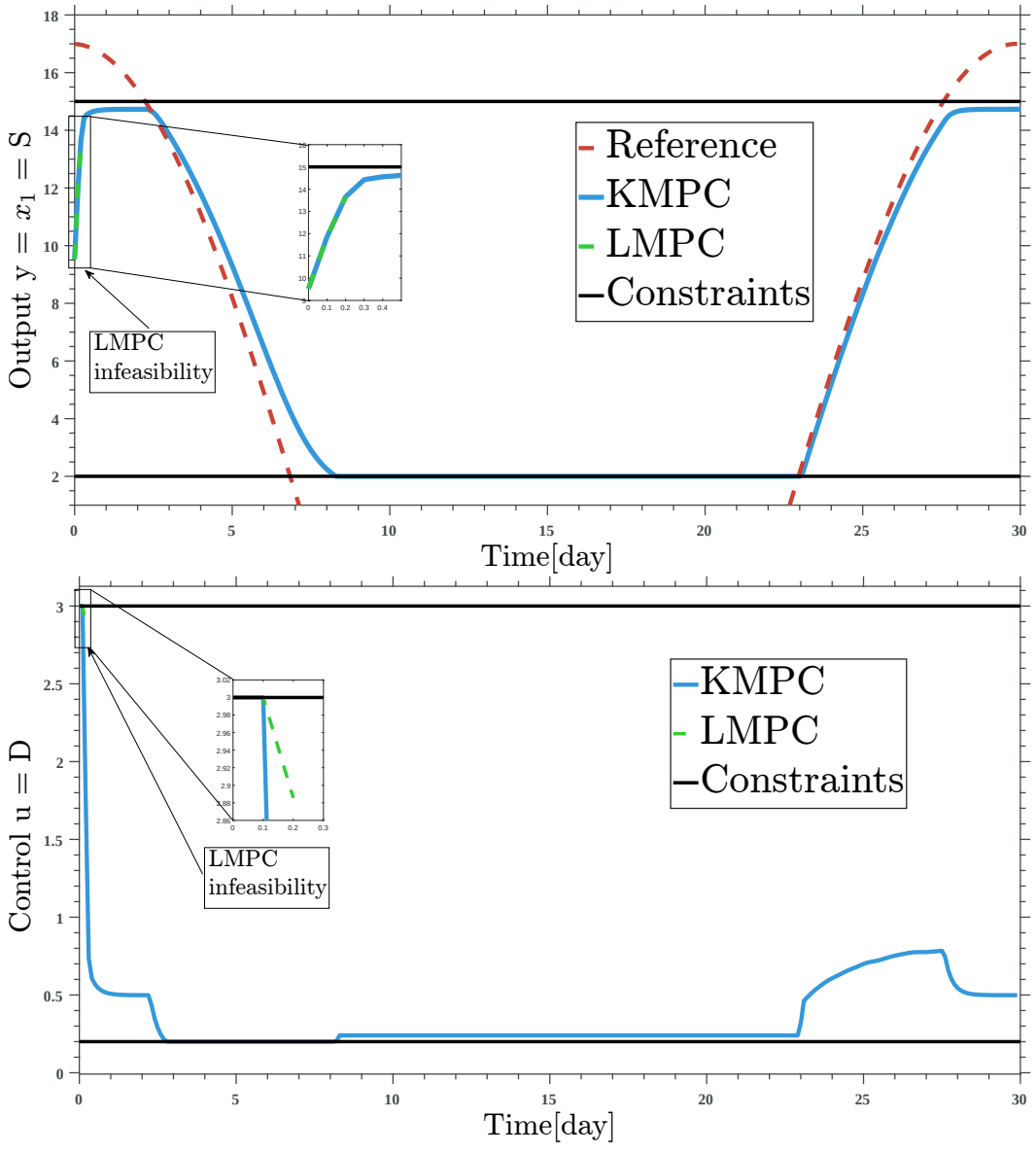

Figure 5.17: Control in larger space of the initial condition-Feedback control of the chemostat- Time-varying reference tracking with constraints imposed on the output ($y \in [2, 15]$) and constrained control input ($u \in [0.2, 3]$) , initial condition $x_0 = [10, 10]^\top$.

the defining data are pre-computed offline with the KMPC controller, whereas in the case of the LMPC controller, all the data are continuously re-computed at each iteration. This makes KMPC faster than the LMPC when it comes to computational speed. We should stress out that these results are gained for a large space of initial conditions where the control of the chemostat is usually limited.

The case of the invariant manifold

Now, we try to model and control our system in the invariant manifold. Furthermore, we choose that the trajectories begin from initial conditions $x_1(0), x_2(0)$ generated randomly with uniform distribution where $x_1(0) + x_2(0) = S_{in}$ in this case we are sure that $x_1 + x_2 = S_{in}$ for all time t (this was illustrated by the Table 5.8, where we have taken one hundred samples of x_1 and x_2 to validate it). For that purpose, we take $x_1(0) \in [2, 20[$ and $x_2(0) = S_{in} - x_1(0)$. We used the same $n_d = 1$ and the embedding lifting functions $\boldsymbol{\varphi}_i = S_k$ as the first case, we have also used 60 IMQ-RBFs (5.35) with the same centers selection.

Day	$x_1(0)$	$x_2(0)$	x_1	x_2	$x_1 + x_2$
1	6.2014	13.7986	2.5118	17.4881	20
2	2.5387	17.4613	2.0463	17.9536	20
3	17.1261	2.8739	2.0040	17.9959	20
4	18.8408	1.1592	1.5382	18.4617	20
⋮	⋮	⋮	⋮	⋮	⋮
56	17.1339	2.8661	1.8459	18.1540	20
57	7.8361	12.1639	1.9446	18.0553	20
58	3.5644	16.4356	2.8650	17.1349	20
59	16.1861	3.8139	1.9103	18.0896	20
⋮	⋮	⋮	⋮	⋮	⋮
97	9.6021	10.3979	1.5124	18.4875	20
98	15.9382	4.0618	2.9247	17.0752	20
99	16.3177	3.6823	1.5996	18.4003	20
100	5.7091	14.2909	2.3772	17.6227	20

Table 5.8: Table of initial condition $x_1(0), x_2(0)$ and x_1, x_2 over one hundred days.

Koopman model construction and comparison

As in Section 5.4.3, we compare the Koopman model (5.23) with the local linearization-based model, generated from two randomly chosen initial conditions ($x_0^1 = [13.07, 6.93]^\top$, $x_0^2 = [9, 11]^\top$). Illustrated in Figure 5.18, we can remark a clear superiority of the proposed Koooopman model in terms the prediction accuracy over the local linearization-based model. Table 5.9 shows the averaged RMSE (5.20) forecast comparison over one hundred randomly chosen initial conditions. It is apparent

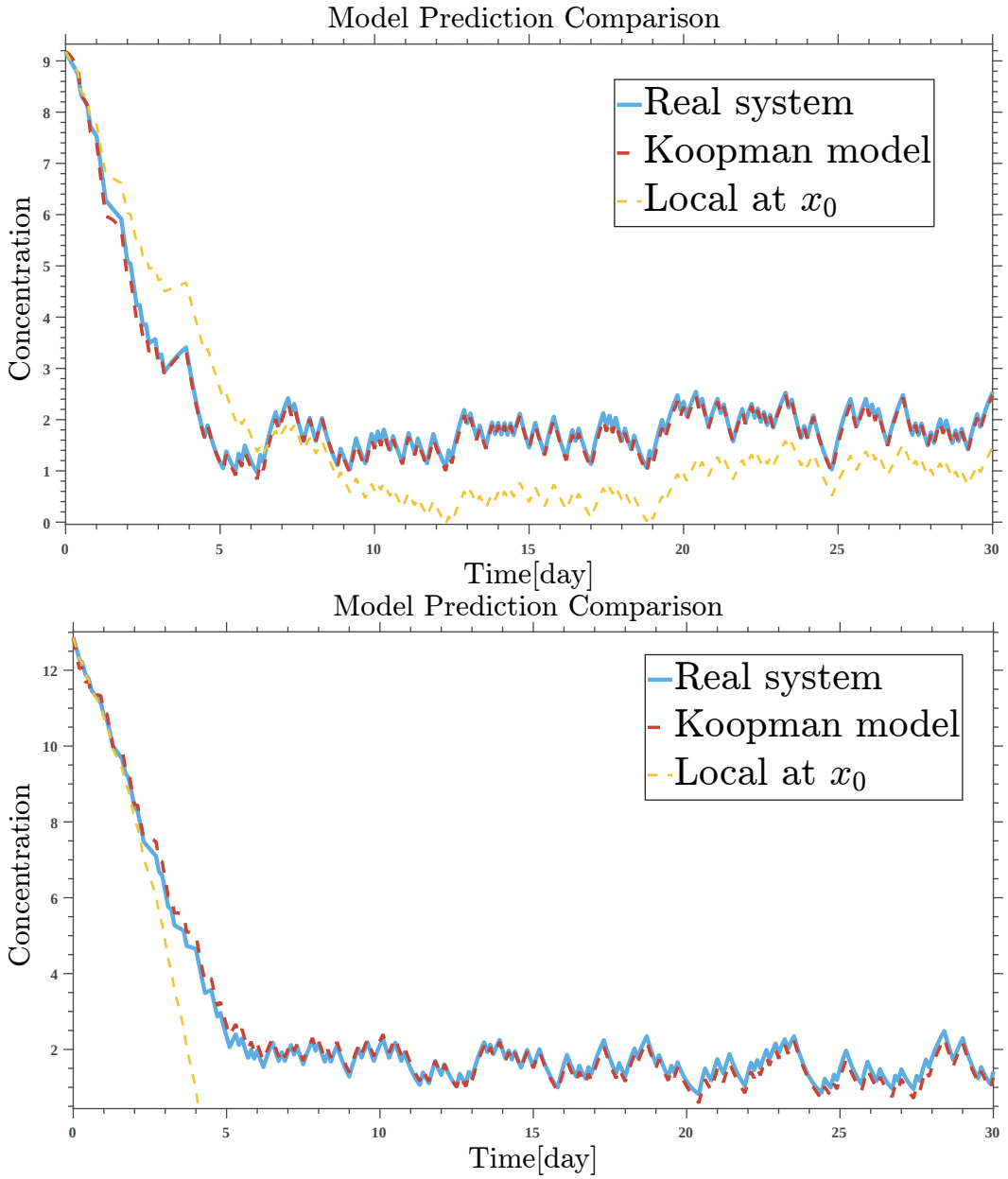

Figure 5.18: The case of the invariant manifold- Model prediction comparison of the chemostat - $y_k = S_k$. Up: initial condition $x_0^2 = [9, 11]^\top$ Down: initial condition $x_0^1 = [13.07, 6.93]^\top$.

that even in this case, the Koopman model (5.23) has the lowest value of the RMSE compared to the local linearization-based model over longer prediction horizons (based on the value of the averaged RMSE given in Table 5.9 and Table 5.6, the Koopman model generated in the invariant manifold is more accurate compared to the one generated in the large space of initial conditions).

Model	**Average RMSE**
Koopman Model	**7.79 %**
Local linearization at x_0	2.83×10^2 %

Table 5.9: Forecast comparison-RMSE (5.20) for 100 randomly chosen initial conditions

Feedback control of the chemostat

The control objective is still the same as in Section 5.4.3, so, we will use the previous cost functions and prediction horizon ($Q = Q_{N_h} = 10, R = 0.01, N_h = 10$) for the reference tracking purpose. However, in the case of constant reference tracking, we forced the system to track a piece-wise constant reference.

In Figure 5.19 and 5.20 , simulation results are presented, where first we have a comparison between KMPC controller and LMPC controller for the constant reference tracking, we can observe a similarity of the prediction accuracy in this case without any violation of the constraints (in the control input), nonetheless, LMPC performed a bit better when it comes to maximizing the output (we have seen the same results in Figure 5.15). For the time-varying reference tracking, the KMPC controller shows its unique performance while we apply constraints in the output ($y \in [2, 15]$) and the control input ($u \in [0.2, 3]$). On the other hand, the LMPC controller was incapable of continuing the whole simulation process. The KMPC controller is completely data-driven and needs only output measurements which have the advantages of any other controller especially controllers where the model of the system is needed, in our case LMPC.

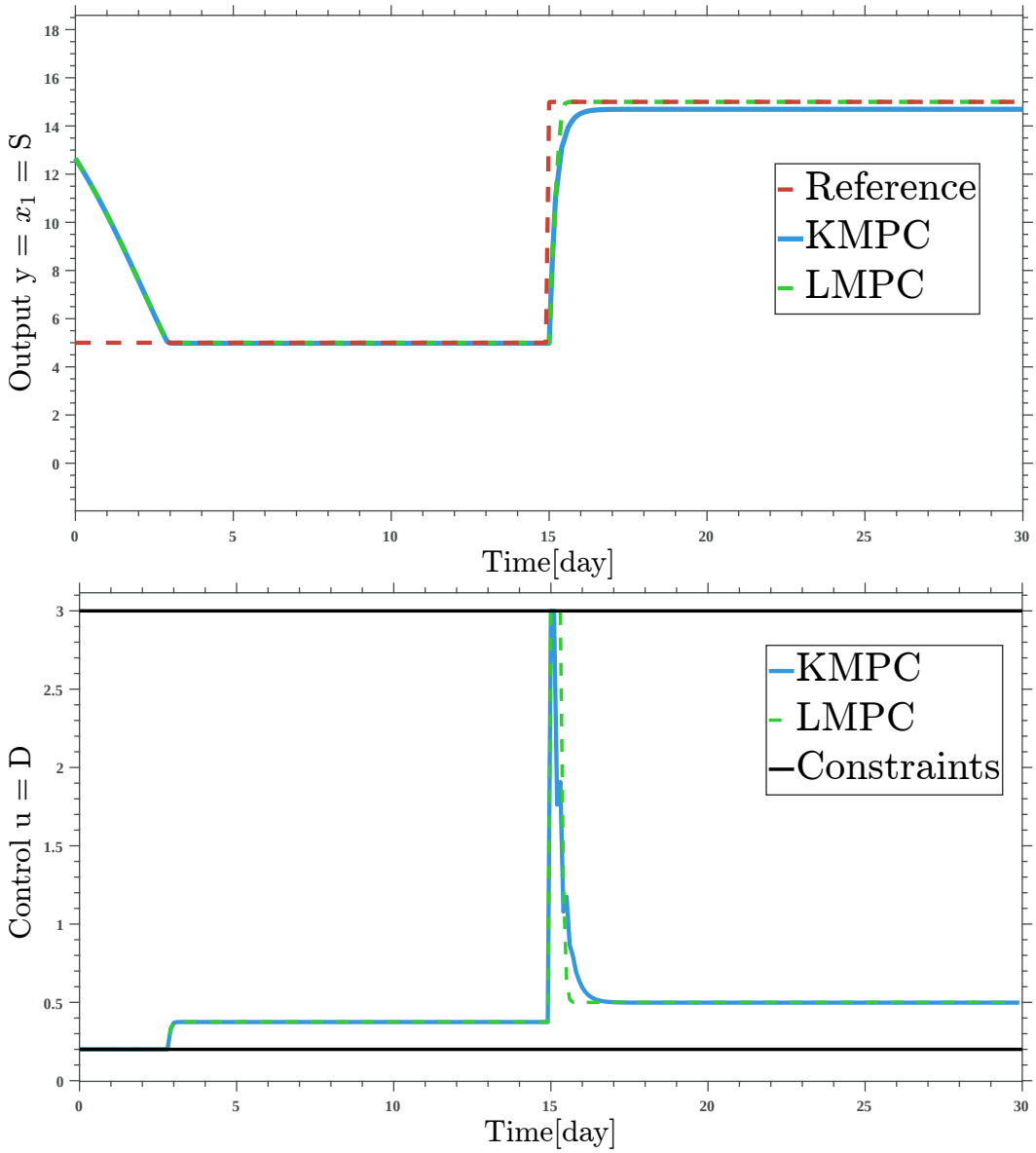

Figure 5.19: Control in the invariant manifold-Feedback control of the chemostat. Top: Constant reference tracking in parts with no state constraints and control input with constraints ($u \in [0.2, 3]$),$x_0 = [13, 7]^\top$.

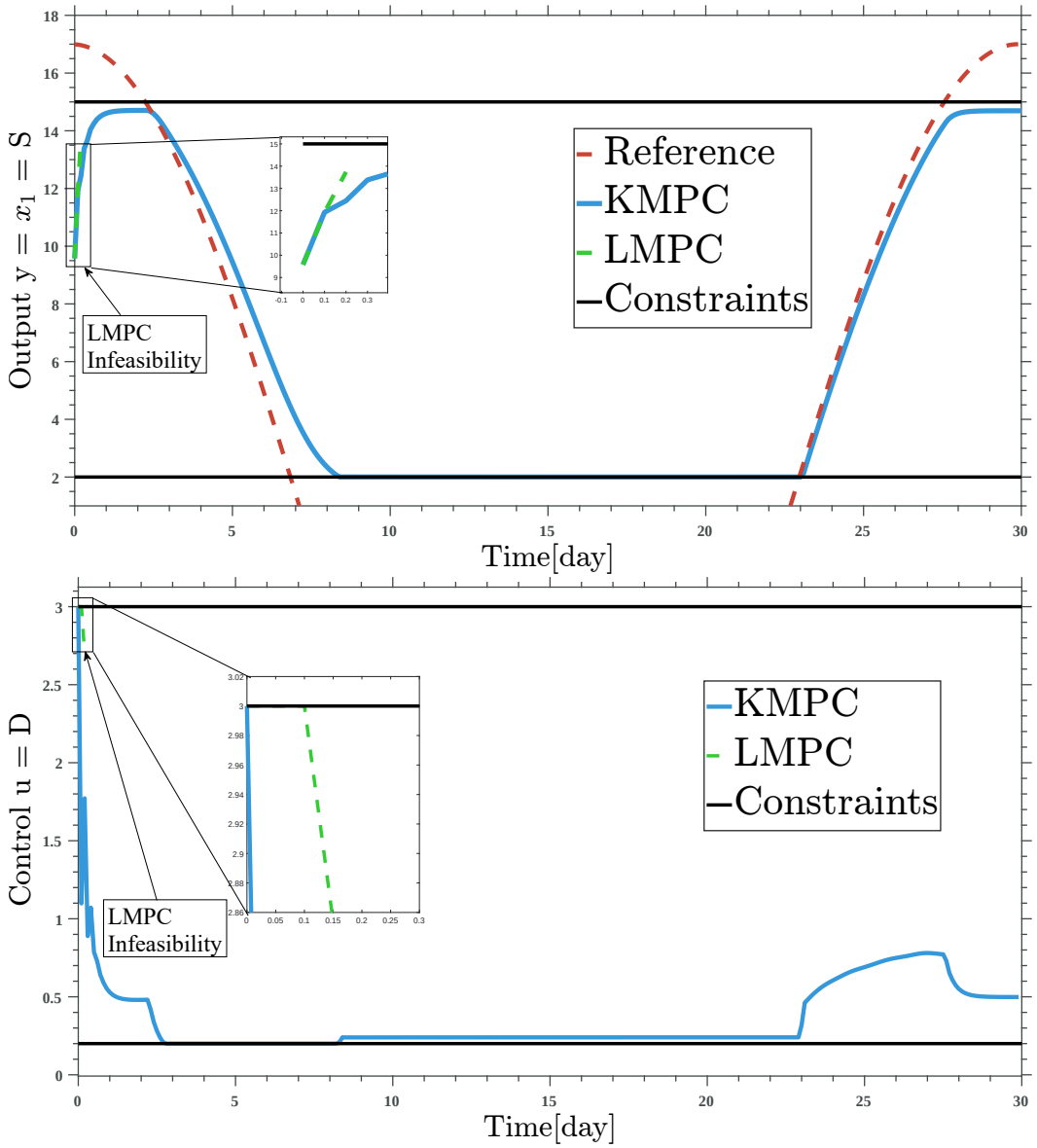

Figure 5.20: Control in the invariant manifold-Feedback control of the chemostat. Time-varying reference tracking with constraints imposed in the output ($y \in [2, 15]$) and control input with constraints($u \in [0.2, 3]$), $x_0 = [9, 10]^\top$.

5.5 Conclusion

In conclusion, this chapter highlighted the complexity and uncertainty associated with the chemostat dynamical model due to poorly known kinetics functions and uncertain variables. We've utilized both EDMD and EDMDc methods to predict how the chemostat will behave in both CSTRm and CSTRh scenarios. Furthermore, we've examined two cases: one where we can measure all aspects of the system's state, and another, more intricate situation involving input-output data. The application of KMPC was introduced as a solution, projecting the nonlinear model into a higher-dimensional space for effective linearization and control. This

approach allows for linear MPC to be applied to the nonlinear chemostat model, offering improved control performance over classical model-based MPC methods.

Chapter 6

General Conclusions

6.1 Summary of Findings and Interpretations

Throughout the course of this thesis, our primary aim revolved around the exploration of data-driven methodologies for forecast, simulation, and regulation of AD processes. Our endeavor, guided by the principles of reduced order modeling, encompassed a thorough investigation into the deployment of Ordinary Differential Equation (ODE) representations as a pertinent framework to address the aforementioned applications. The pertinence of ODE representations stems from their inherent linkage to fundamental physical laws, including Newton's and Lagrange's principles, and the principles of mass balances within the context of bioprocesses. These models naturally emerge and manifest interpretability, particularly within the phase space, thereby harboring a multitude of regularization properties.

This dissertation can be split in two main parts:
In the first part, we conducted a comprehensive exploration of the Koopman Operator Theory and several established data-driven methodologies. Starting with a thorough overview of the Koopman Operator Theory, we demonstrated its fundamental concepts and its pivotal role in unraveling intricate dynamic systems. Subsequently, we delved into the DMD approach, a robust algorithm capable of approximating the Koopman Operator using data from complex nonlinear dynamical systems. This method's efficacy was showcased through domain applications, demonstrating its ability to extract coherent structures and modes from high-dimensional data. The broad applicability of DMD was highlighted across diverse fields, including fluid dynamics, neurobiology, and video analysis. We then explored the EDMD, an extension that builds upon DMD to address nonlinear and high-dimensional data challenges. EDMD enhances the capabilities of DMD, offering improved representation of complex systems and enhanced predictive potential. Our investigation extended to the realm of Data-Driven Control, where we tackled the Koopman Operator Theory. Utilizing the Koopman model for control presented intriguing opportunities for optimizing system behavior. MPC emerged as a prominent application, enabling real-time control by iteratively optimizing control inputs based on projected system behavior. Furthermore, we delved into Nonlinear System Identification for Control by introducing DMD with control and EDMD with control, showcasing their utility in detecting nonlinear controlled

systems and building control-oriented models. Ultimately, the integration of the Koopman Operator and data-driven methodologies has catalyzed a transformation in system analysis and control. Their capacity to extract meaningful insights from data, discern system behavior, and facilitate predictive control has opened new avenues for research and practical applications across disciplines. As data-driven approaches continue to evolve, we anticipate further developments in comprehending and managing complex dynamic systems, leading to more efficient, adaptable, and intelligent control techniques.

In the second part of this thesis, we provided a comprehensive exploration of various models within the context of AD processes. Our journey commenced with an examination of the ADM1, which represents a highly detailed model for AD processes. Emphasizing the significance of mathematical models as approximations of real-world systems, we delve into the characteristics and properties of these models. Understanding these mathematical representations is crucial as they serve as the foundation for generating data that can be employed as input for our subsequent data-driven techniques. Subsequently, we delved into the intricacies of the AM2 model, which not only presented the intricate biological and kinetic reactions at play but also refined its mathematical equations. This refinement offered valuable insights into both measurement and modeling aspects, shedding light on the anaerobic process, and culminating in exploring the most fundamental type of bioreactors. The Chemostat, a simple yet fundamental bioreactor model, laying the groundwork for comprehending the dynamics of bioprocess systems. Derived from the foundational principles of mass balance, this model forms the cornerstone of mathematical modeling in bioprocessing. By studying and analyzing these bioreactor models, we gain valuable insights into the underlying dynamics and behavior of the systems. Leveraging insights from these models, we can further enhance the efficiency and sustainability of critical biological processes across diverse applications, spanning wastewater treatment, biogas generation, and beyond. This knowledge is essential for subsequent steps, where data-driven techniques are employed to extract patterns, identify correlations, and develop advanced control and optimization strategies. Thus, the mathematical models we explored in this study serve as critical building blocks for the data-driven approaches presented in the first part of this thesis. Lastly, we focused on the potential and power of data-driven methodologies presented in part one in addressing the complexities of AD systems. Through a focused exploration of the ADM1, AM2, and chemostat models, we have demonstrated how these techniques can be used to create simplified yet accurate representations of intricate dynamics. The ADM1 model showcased the application of the DMD method, allowing us to refine a complex system into a concise linear framework. This reduction not only enhances our understanding of the system but also opens avenues for predicting its behavior. Similarly, the AM2 model benefited from the DMDc algorithm, constructing a versatile linear model capable of predicting dynamics under varying control inputs. This not only advances our ability to model and control AD processes but also sets the stage for future applications in diverse contexts. Addressing the challenging chemostat system, we employed EDMD and EDMDc methods, effectively predicting behavior in different scenarios. The introduction of Koopman Model Predictive

Control (KMPC) demonstrated how data-driven techniques can bridge the gap between nonlinear models and linear control strategies, leading to improved control performance. In essence, in this work we shed light on the transformative potential of data-driven strategies, simplifying complex models and enhancing control and prediction capabilities. By forging connections between theoretical insights and practical applications, these approaches offer innovative solutions to intricate challenges across various domains. The integration of data-driven methodologies marks a promising trajectory for shaping the future of AD systems and beyond, driving progress, efficiency, and adaptability in the management of complex dynamic processes.

In these concluding remarks, we emphasize the critical insight gained from this study: the judicious use of data-driven methods necessitates great attention to align the techniques with the specific characteristics of the provided observations. The belief in the power of complex artificial intelligence models to continuously unravel the complexities of biotechnology is unfounded. Beyond subjective and cost-related assessments of AI models, ascertaining their generalizability presents intricate challenges. In light of this perspective, we advocate for the regularization of AI models by integrating prior knowledge concerning established biological and mathematical constraints. This regularization serves as a pivotal strategy to attain consistency and coherence when employing such techniques.

6.2 Limitations and Future Directions

Due to time and logistic constraints, our study has some limitations which we can summarise in the following:

Regarding the application of the DMD method to reduce the ADM1 model's state variables from 35 to 7, an inherent challenge arises due to the absence of prior knowledge about which specific states contribute significantly to the dynamic behavior of the AD process. This lack of biological insights underscores the complexity of the reduction task. The DMD technique hinges upon data-driven principles, extracting dominant coherent structures embedded within the data. While it is established that the ADM1 model can indeed be reduced to 9, 8, and 7 states through SVD and PCA [140] as we have already mentioned, the inherent nature and significance of these states remain elusive. The SVD technique inherently organizes singular values hierarchically, with a progression from most to least important. Moreover, the linear operator A produced by the DMD method, while capturing aspects of nonlinear dynamical behavior, falls short in providing a comprehensive representation of the entire ADM1 model. This limitation underscores the need for future research to address the integration of biological and biochemical constraints at the core of the DMD approach.

One promising avenue for enhancing the DMD method involves the integration of known physical principles or constraints, as exemplified by the Physics Informed DMD (PIDMD) framework [147]. PIDMD extends the scope of DMD by incorporating established physical laws into the analysis. This integration serves to refine the decomposition process, yielding more meaningful and insightful interpretations

of the underlying system dynamics. By honoring governing equations or physical principles, PIDMD is adept at extracting coherent modes while preserving the intrinsic physics of the system. Notably, Physics Informed DMD finds utility across diverse domains, ranging from fluid dynamics to structural mechanics and even biological systems like AD processes. This integration of established physics within the DMD framework equips researchers and practitioners with a potent tool to gain deeper insights, better model complex systems, and ultimately bridge the gap between data-driven techniques and the inherent underlying dynamics of intricate systems like AD.

When considering the application of the Koopman operator theory and its data-driven approximations, such as EDMD and its controlled variant EDMDc, a noteworthy challenge emerges in the selection of observables or the embedded functions. As previously discussed, this selection process offers a range of choices, encompassing functions from polynomials to RBFs – the latter being the focus of our investigation. This selection intricacy significantly impacts the accurate approximation of the Koopman operator for nonlinear systems. A crucial observation from our study reveals the sensitivity of the approximated Koopman operator to the choice of RBFs, highlighting the potential divergence in outcomes even for the same system. With regards to this challenge, the determination of the number of embedded functions, thereby shaping the dimensionality of the augmented state space, remains a challenging endeavor. This is underscored by our varying utilization of RBFs; for instance, we employed 30 RBFs to approximate the Koopman operator of the CSTRm, and 50 RBFs for the CSTRh. Notably, this dimensionality is not fixed and fluctuates based on the selected embedded functions and their count.

To tackle this issue, researchers have employed neural networks (NN) and deep neural networks (deep NN) [148], [149], [150] in order to facilitate the learning of Koopman eigenfunctions and embedded functions directly from time series data. Noteworthy is their parsimonious and interpretable nature, systematically embedding dynamics within a low-dimensional manifold characterized by Koopman eigenfunctions. These techniques identify nonlinear coordinates that globally render the dynamics as linear, employing a modified auto-encoder framework. Significantly, these methods extend Koopman representations to encompass systems exhibiting continuous spectra. They effectively leverage the capabilities and adaptability of deep learning, while simultaneously preserving the intrinsic interpretability of Koopman embeddings, thereby advancing the boundaries of data-driven approximations into unexplored territories such as AD processes.

Bibliography

[1] D. Batstone, J. Keller, I. Angelidaki, S. Kalyuzhnyi, S. Pavlostathis, A. Rozzi, W. Sanders, H. Siegrist, and V. Vavilin, "The iwa anaerobic digestion model no 1 (adm1)," *Water Sci. Technol.*, vol. 45, no. 10, pp. 65–73, 2002.

[2] W. P. Smith H, *The theory of the chemostat: Dynamics of Microbial Competition*, vol. 13. Cambridge University Press, 1995.

[3] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, "A data-driven approximation of the koopman operator: Extending dynamic mode decomposition.," *Journal of Nonlinear Science*, 2015.

[4] B. Chen, Z. Huang, R. Zhang, W. Liu, H. Li, J. Wang, Y. Fan, and J. Peng, "Data-driven koopman model predictive control for optimal operation of high-speed trains," *IEEE Access*, vol. 9, pp. 82233–82248, 2021.

[5] O. Bernard, Z. Hadj-Sadok, D. Dochain, A. Genovesi, and J.-P. Steyer, "Dynamical model development and parameter identification for an anaerobic wastewater treatment process," *Biotechnology and bioengineering*, vol. 75, no. 4, pp. 424–438, 2001.

[6] K. B. Cantrell, T. Ducey, K. S. Ro, and P. G. Hunt, "Livestock waste-to-bioenergy generation opportunities," *Bioresource Technology*, vol. 99, no. 17, pp. 7941–7953, 2008.

[7] J. B. Holm-Nielsen, T. Al Seadi, and P. Oleskowicz-Popiel, "The future of anaerobic digestion and biogas utilization," *Bioresource technology*, vol. 100, no. 22, pp. 5478–5484, 2009.

[8] J. Lauwers, L. Appels, I. P. Thompson, J. Degrève, J. F. Van Impe, and R. Dewil, "Mathematical modelling of anaerobic digestion of biomass and waste: Power and limitations," *Progress in Energy and Combustion Science*, vol. 39, no. 4, pp. 383–402, 2013.

[9] K. R. Manchala, Y. Sun, D. Zhang, and Z.-W. Wang, "Chapter two - anaerobic digestion modelling," in *Advances in Bioenergy* (Y. Li and X. Ge, eds.), vol. 2, pp. 69–141, Elsevier, 2017.

[10] C. O. Onwosi, I. E. Eke, V. C. Igbokwe, J. N. Odimba, J. K. Ndukwe, K. O. Chukwu, G. O. Aliyu, and T. N. Nwagu, "Towards effective management of

digester dysfunction during anaerobic treatment processes," *Renewable and Sustainable Energy Reviews*, vol. 116, p. 109424, 2019.

[11] A.-A. Issah, T. Kabera, and F. Kemausuor, "Biogas optimisation processes and effluent quality: A review," *Biomass and Bioenergy*, vol. 133, p. 105449, 2020.

[12] K. Boe, D. J. Batstone, J.-P. Steyer, and I. Angelidaki, "State indicators for monitoring the anaerobic digestion process," *Water Research*, vol. 44, no. 20, pp. 5973–5980, 2010.

[13] C. Wolf, S. McLoone, and M. Bongards, "Biogas plant control and optimization using computational intelligence methods biogasanlagenregelung und -optimierung mit computational intelligence methoden," *at - Automatisierungstechnik*, vol. 57, no. 12, pp. 638–649, 2009.

[14] L. Li, Q. He, Y. Wei, Q. He, and X. Peng, "Early warning indicators for monitoring the process failure of anaerobic digestion system of food waste," *Bioresource technology*, vol. 171, pp. 491–494, 2014.

[15] S. Theuerl, C. Herrmann, M. Heiermann, P. Grundmann, N. Landwehr, U. Kreidenweis, and A. Prochnow, "The future agricultural biogas plant in germany: A vision," *Energies*, vol. 12, no. 3, 2019.

[16] K. C. Draa, *Observation and control of anaerobic digestion processes for improved biogas production.* PhD thesis, Université de Lorraine; Université du Luxembourg, 2017.

[17] D. Bastin, G., *On-line estimation and adaptive control of bioreactors*, vol. 1. Process Measurment and Control Elsevier, 1990.

[18] L. Appels, J. Lauwers, J. Degrève, L. Helsen, B. Lievens, K. Willems, J. Van Impe, and R. Dewil, "Anaerobic digestion in global bio-energy production: Potential and research challenges," *Renewable and Sustainable Energy Reviews*, vol. 15, no. 9, pp. 4295–4301, 2011.

[19] H. Bouallagui, Y. Touhami, R. Ben Cheikh, and M. Hamdi, "Bioreactor performance in anaerobic digestion of fruit and vegetable wastes," *Process Biochemistry*, vol. 40, no. 3, pp. 989–995, 2005.

[20] J. Hess, *Modélisation de la qualité du biogaz produit par un fermenteur méthanogène et stratégie de régulation en vue de sa valorisation.* Theses, Université Nice Sophia Antipolis, Dec. 2007.

[21] E. Ficara, S. Hassam, A. Allegrini, A. Leva, F. Malpei, and G. Ferretti, "Anaerobic digestion models: a comparative study," *IFAC Proceedings Volumes*, vol. 45, no. 2, pp. 1052–1057, 2012. 7th Vienna International Conference on Mathematical Modelling.

[22] F. Walid, S. El Fkihi, H. Benbrahim, and H. Tagemouati, "Modeling and optimization of anaerobic digestion: a review," in *E3S Web of Conferences*, vol. 229, p. 01022, EDP Sciences, 2021.

[23] M. Alvi, D. Batstone, C. K. Mbamba, P. Keymer, T. French, A. Ward, J. Dwyer, and R. Cardell-Oliver, "Deep learning in wastewater treatment: A critical review," *Water Research*, p. 120518, 2023.

[24] A. Donoso-Bravo, J. Mailier, C. Martin, J. Rodríguez, C. A. Aceves-Lara, and A. V. Wouwer, "Model selection, identification and validation in anaerobic digestion: A review," *Water Research*, vol. 45, no. 17, pp. 5347–5364, 2011.

[25] J. Harmand, C. Lobry, A. Rapaport, and T. Sari, *The chemostat: mathematical theory of microorganism cultures*, vol. 1. John Wiley and Sons, Ltd,Chemical Engineering series / Chemostat and bioprocesses, 2017.

[26] S. P. Graef and J. F. Andrews, "Stability and control of anaerobic digestion," *Journal (Water Pollution Control Federation)*, vol. 46, no. 4, pp. 666–683, 1974.

[27] S. Sotemann, N. Ristow, M. Wentzel, and G. Ekama, "A steady state model for anaerobic digestion of sewage sludges," *Water SA*, vol. 31, no. 4, pp. 511–528, 2005.

[28] S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control.* Cambridge University Press, 2019.

[29] K. G. Liakos, P. Busato, D. Moshou, S. Pearson, and D. Bochtis, "Machine learning in agriculture: A review," *Sensors*, vol. 18, no. 8, 2018.

[30] S. Sathish and S. Vivekanandan, "Parametric optimization for floating drum anaerobic bio-digester using response surface methodology and artificial neural network," *Alexandria Engineering Journal*, vol. 55, no. 4, pp. 3297–3307, 2016.

[31] H. Akbas and G. Ozdemır, "Prediction modeling of biogas production with classification and regression tree at wastewater treatment plants," *The Eurasia Proceedings of Science Technology Engineering and Mathematics*, no. 4, pp. 212 – 217, 2018.

[32] L. Wang, F. Long, W. Liao, and H. Liu, "Prediction of anaerobic digestion performance and identification of critical operational parameters using machine learning algorithms," *Bioresource Technology*, vol. 298, p. 122495, 2020.

[33] P. Kazemi, J.-P. Steyer, C. Bengoa, J. Font, and J. Giralt, "Robust data-driven soft sensors for online monitoring of volatile fatty acids in anaerobic digestion processes," *Processes*, vol. 8, no. 1, 2020.

[34] H. Abu Qdais, K. Bani Hani, and N. Shatnawi, "Modeling and optimization of biogas production from a waste digester using artificial neural network and genetic algorithm," *Resources, Conservation and Recycling*, vol. 54, no. 6, pp. 359–363, 2010.

[35] J. Dalmau, J. Comas, I. Rodríguez-Roda, E. Latrille, and J.-P. Steyer, "Selecting the most relevant variables for anaerobic digestion imbalances: Two case studies," *Water environment research : a research publication of the Water Environment Federation*, vol. 82, pp. 492–8, 06 2010.

[36] D. Güçlü, N. Yılmaz, and U. G. Ozkan-Yucel, "Application of neural network prediction model to full-scale anaerobic sludge digestion," *Journal of Chemical Technology & Biotechnology*, vol. 86, no. 5, pp. 691–698, 2011.

[37] T. Beltramo, M. Klocke, and B. Hitzmann, "Prediction of the biogas production using ga and aco input features selection method for ann model," *Information Processing in Agriculture*, vol. 6, no. 3, pp. 349–356, 2019.

[38] S. L. Brunton, M. Budišić, E. Kaiser, and J. N. Kutz, "Modern koopman theory for dynamical systems," *SIAM Review*, vol. 64, no. 2, pp. 229–340, 2022.

[39] A. Mauroy, I. Mezic, and Y. Susuki, *The Koopman operator in systems and control concepts, methodologies, and applications*. Springer, 2020.

[40] Y. Lan and I. Mezić, "Linearization in the large of nonlinear systems and koopman operator spectrum," *Physica D: Nonlinear Phenomena*, vol. 242, no. 1, pp. 42–53, 2013.

[41] I. M. M. budišić, R. Mohr, "Applied koopmanism," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 22, pp. 047510–047543, 2012.

[42] J. E. Marsden and T. S. Ratiu, *Introduction to Mechanics and Symmetry*. Springer, 2nd ed., 1999.

[43] B. O. Koopman, "Hamiltonian systems and transformation in hilbert space.," *Proceedings of the National Academy of Sciences of the USA*, vol. 17, no. 5, p. 315–318, 1931.

[44] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz., "On dynamic mode decomposition: theory and applications," *Journal of Computational Dynamics*, vol. 1, pp. 391–421, 2014.

[45] S. L. Brunton, B. W. Brunton, J. L. Proctor, and J. N. Kutz, "Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control," *PLOS ONE*, vol. 11, pp. 1–19, 02 2016.

[46] T. Carleman, "Application de la théorie des équations intégrales linéaires aux systèmes d'équations différentielles non linéaires," *Acta Math*, vol. 59, p. 63–87, 1932.

[47] A. Amini, Q. Sun, and N. Motee, "Carleman state feedback control design of a class of nonlinear control systems," *IFAC-PapersOnLine*, vol. 52, no. 20, pp. 229–234, 2019. 8th IFAC Workshop on Distributed Estimation and Control in Networked Systems NECSYS 2019.

[48] K. Loparo and G. Blankenship, "Estimating the domain of attraction of nonlinear feedback systems," *IEEE Transactions on Automatic Control*, vol. 23, no. 4, pp. 602–608, 1978.

[49] J. Minisini, A. Rauh, and E. P. Hofer, "Carleman linearization for approximate solutions of nonlinear control problems: Part 1–theory," in *Proc. of the 14th Intl. Workshop on Dynamics and Control*, pp. 215–222, 2007.

[50] A. Rauh, J. Minisini, and E. P. Hofer, "Carleman linearization for approximate solutions of nonlinear control problems: Part 2–applications," in *Proc. of the 14th Intl. Workshop on Dynamics and Control, Moscow-Zvenigorod, Russia*, pp. 266–273, 2007.

[51] K. Kowalski and W.-H. Steeb, *Nonlinear dynamical systems and Carleman linearization*. World Scientific, 1991.

[52] I. Mezić and A. Banaszuk, "Comparison of systems with complex behavior," *Physica D: Nonlinear Phenomena*, vol. 197, no. 1, pp. 101–133, 2004.

[53] J. L. Proctor, S. L. Brunton, and J. N. Kutz, "Dynamic mode decomposition with control," *SIAM Journal on Applied Dynamical Systems*, vol. 15, no. 1, pp. 142–161, 2016.

[54] G. Mamakoukas, M. L. Castaño, X. Tan, and T. D. Murphey, "Local koopman operators for data-driven control of robotic systems," *Robotics: Science and Systems XV*, 2019.

[55] E. Kaiser, J. N. Kutz, and S. L. Brunton, "Data-driven discovery of koopman eigenfunctions for control," *Machine Learning: Science and Technology*, vol. 2, p. 035023, jun 2021.

[56] M. Korda and I. Mezic., "Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control," *Automatica*, vol. 93, pp. 149–160, 2018.

[57] E. Kaiser, J. N. Kutz, and S. L. Brunton, "Sparse identification of nonlinear dynamics for model predictive control in the low-data limit," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 474, no. 2219, p. 20180335, 2018.

[58] S. Peitz and S. Klus, "Koopman operator-based model reduction for switched-system control of pdes," *Automatica*, vol. 106, pp. 184–191, 2019.

[59] B. Huang, X. Ma, and U. Vaidya, "Feedback stabilization using koopman operator," in *2018 IEEE Conference on Decision and Control (CDC)*, pp. 6434–6439, 2018.

[60] M. Hemati and H. Yao, "Dynamic mode shaping for fluid flow control: New strategies for transient growth suppression," *8th AIAA Theoretical Fluid Mechanics Conference*, 2017.

[61] I. Abraham and T. D. Murphey, "Active learning of dynamics for data-driven control using koopman operators," *IEEE Transactions on Robotics*, vol. 35, pp. 1071–1083, 2019.

[62] J. L. Garriga and M. Soroush, "Model predictive control tuning methods: A review," *Industrial Engineering Chemistry Research*, vol. 49, no. 8, pp. 3505–3515, 2010.

[63] J. H. Lee, "Model predictive control: Review of the three decades of development," *International Journal of Control, Automation and Systems*, vol. 9, pp. 415–424, 2011.

[64] F. Allgöwer and A. Zheng, *Nonlinear model predictive control*, vol. 26. Birkhäuser, 2012.

[65] M. Morari and J. H. Lee, "Model predictive control: past, present and future," *Computers and Chemical Engineering*, vol. 23, no. 4, pp. 667–682, 1999.

[66] S. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, no. 7, pp. 733–764, 2003.

[67] F. Allgower, R. Findeisen, Z. K. Nagy, *et al.*, "Nonlinear model predictive control: From theory to application," *Journal-Chinese Institute Of Chemical Engineers*, vol. 35, no. 3, pp. 299–316, 2004.

[68] M. Korda, Y. Susuki, and I. Mezić, "Power grid transient stabilization using koopman model predictive control," *IFAC-PapersOnLine*, vol. 51, no. 28, pp. 297–302, 2018. 10th IFAC Symposium on Control of Power and Energy Systems CPES 2018.

[69] H. Arbabi, M. Korda, and I. Mezic, "A data-driven koopman model predictive control framework for nonlinear partial differential equations," *2018 IEEE Conference on Decision and Control (CDC)*, pp. 6409–6414, 2018.

[70] S. Hanke, S. Peitz, O. Wallscheid, S. Klus, J. Böcker, and M. Dellnitz, "Koopman operator-based finite-control-set model predictive control for electrical drives," *arXiv preprint arXiv:1804.00854*, 2018.

[71] J. L. Proctor, S. L. Brunton, and J. N. Kutz., "Generalizing koopman theory to allow for inputs and control," *SIAM Journal on Applied Dynamical Systems*, vol. 17, pp. 909–930, 2016.

[72] P. J. Schmid, "Application of the dynamic mode decomposition to experimental data," *Exp. Fluids*, vol. 50, no. 4, pp. 1123–1130, 2011.

[73] P. Schmid and J. Sesterhenn, "Dynamic Mode Decomposition of numerical and experimental data," in *APS Division of Fluid Dynamics Meeting Abstracts*, vol. 61 of *APS Meeting Abstracts*, p. MR.007, Nov. 2008.

[74] P. J. SCHMID, "Dynamic mode decomposition of numerical and experimental data," *Journal of Fluid Mechanics*, vol. 656, p. 5–28, 2010.

[75] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic mode decomposition: data-driven modeling of complex systems*. SIAM, 2016.

[76] C. W. ROWLEY, I. MEZIĆ, S. BAGHERI, P. SCHLATTER, and D. S. HENNINGSON, "Spectral analysis of nonlinear flows," *Journal of Fluid Mechanics*, vol. 641, p. 115–127, 2009.

[77] K. Chen, J. Tu, and C. Rowley, "Variants of dynamic mode decomposition: Boundary condition, koopman, and fourier analyses.," *J Nonlinear Sci*, vol. 22, p. 887–915, 2012.

[78] K. Taira, S. L. Brunton, S. T. M. Dawson, C. W. Rowley, T. Colonius, B. J. McKeon, O. T. Schmidt, S. Gordeyev, V. Theofilis, and L. S. Ukeiley, "Modal analysis of fluid flows: An overview," *AIAA Journal*, vol. 55, no. 12, pp. 4013–4041, 2017.

[79] K. Taira, M. S. Hemati, S. L. Brunton, Y. Sun, K. Duraisamy, S. Bagheri, S. T. M. Dawson, and C.-A. Yeh, "Modal analysis of fluid flows: Applications and outlook," *AIAA Journal*, vol. 58, no. 3, pp. 998–1022, 2020.

[80] N. Demo, M. Tezzele, and G. Rozza, "Pydmd: Python dynamic mode decomposition," *Journal of Open Source Software*, vol. 3, no. 22, p. 530, 2018.

[81] J. Mann and J. N. Kutz, "Dynamic mode decomposition for financial trading strategies," *Quantitative Finance*, vol. 16, no. 11, pp. 1643–1655, 2016.

[82] R. Taylor, J. N. Kutz, K. Morgan, and B. A. Nelson, "Dynamic mode decomposition for plasma diagnostics and validation," *Review of Scientific Instruments*, vol. 89, no. 5, p. 053501, 2018.

[83] E. Berger, M. Sastuba, D. Vogt, B. Jung, and H. B. Amor, "Estimation of perturbations in robotic behavior using dynamic mode decomposition," *Advanced Robotics*, vol. 29, no. 5, pp. 331–343, 2015.

[84] G. Bellani, *Experimental Studies of Complex Flows through Image-Based Techniques*. PhD thesis, KTH, Fluid Physics, 2011. QC 20110519.

[85] P. J. Schmid, D. Violato, and F. Scarano, "Decomposition of time-resolved tomographic PIV," *Experiments in Fluids*, vol. 52, pp. 1567–1579, June 2012.

[86] O. Semeraro, G. Bellani, and F. Lundell, "Analysis of time-resolved piv measurements of a confined turbulent jet using pod and koopman modes," *Experiments in Fluids*, vol. 53, no. 5, pp. 1203–1220, 2012.

[87] J. Basley, L. R. Pastur, N. Delprat, and F. Lusseyran, "Space-time aspects of a three-dimensional multi-modulated open cavity flow," *Physics of Fluids*, vol. 25, no. 6, p. 064105, 2013.

[88] J. Basley, L. R. Pastur, F. Lusseyran, T. M. Faure, and N. Delprat, "Experimental investigation of global structures in an incompressible cavity flow using time-resolved PIV," *Experiments in Fluids*, vol. 50, pp. 905–918, Apr. 2011.

[89] Y. Mizuno, D. Duke, C. Atkinson, and J. Soria, "Investigation of wall-bounded turbulent flow using dynamic mode decomposition," *Journal of Physics: Conference Series*, vol. 318, p. 042040, dec 2011.

[90] C. M. Ostoich, D. J. Bodony, and P. H. Geubelle, "Interaction of a mach 2.25 turbulent boundary layer with a fluttering panel using direct numerical simulation," *Physics of Fluids*, vol. 25, p. 110806, 2013.

[91] T. Sayadi, P. Schmid, J. Nichols, and P. Moin, "Reduced-order representation of near-wall structures in the late transitional boundary layer," *Journal of Fluid Mechanics*, vol. 748, pp. 278–301, June 2014.

[92] J. H. Tu, C. W. Rowley, E. Aram, and R. Mittal, "Koopman spectral analysis of separated flow over a finite-thickness flat plate with elliptical leading edge," in *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2011.

[93] C. Pan, D. Yu, and J. Wang, "Dynamical mode decomposition of gurney flap wake flow," *Theoretical and Applied Mechanics Letters*, vol. 1, no. 1, p. 012002, 2011.

[94] S. Bagheri, "Koopman-mode decomposition of the cylinder wake," *Journal of Fluid Mechanics*, vol. 726, p. 596–623, 2013.

[95] R. Dunne and B. J. McKeon, "Dynamic stall on a pitching and surging airfoil," *Experiments in Fluids*, vol. 56, no. 8, pp. 1–15, 2015. This article belongs to a Topical Collection of articles entitled Extreme Flow Workshop 2014. Guest editors: I. Marusic and B. J. McKeon.

[96] M. Grilli, P. J. Schmid, S. Hickel, and N. A. Adams, "Analysis of unsteady behaviour in shockwave turbulent boundary layer interaction," *Journal of Fluid Mechanics*, vol. 700, p. 16–28, 2012.

[97] J. L. Proctor and P. A. Eckhoff, "Discovering dynamic patterns from infectious disease data using dynamic mode decomposition," *International Health*, vol. 7, pp. 139–145, 02 2015.

[98] B. W. Brunton, L. A. Johnson, J. G. Ojemann, and J. N. Kutz, "Extracting spatial–temporal coherent patterns in large-scale neural recordings using dynamic mode decomposition," *Journal of Neuroscience Methods*, vol. 258, pp. 1–15, 2014.

[99] M. Agrawal, S. Vidyashankar, and K. Huang, "On-chip implementation of ecog signal data decoding in brain-computer interface," in *2016 IEEE 21st International Mixed-Signal Testing Workshop (IMSTW)*, pp. 1–6, 2016.

[100] S. Tirunagari, N. Poh, K. Wells, M. Bober, I. Gorden, and D. Windridge, "Movement correction in dce-mri through windowed and reconstruction dynamic mode decomposition," *Machine Vision and Applications*, vol. 28, no. 3-4, pp. 393 – 407, 2017.

[101] E. N.B., S. Brunton, and J. N. Kutz, "Compressed dynamic mode decomposition for background modeling," *J Real-Time Image Proc*, vol. 16, p. 1479–1492, 2019.

[102] J. Grosek and J. N. Kutz, "Dynamic Mode Decomposition for Real-Time Background/Foreground Separation in Video," *arXiv e-prints*, p. arXiv:1404.7592, Apr. 2014.

[103] S. D. Pendergrass, J. N. Kutz, and S. L. Brunton, "Streaming gpu singular value and dynamic mode decompositions," 2016.

[104] M. O. Williams, C. W. Rowley, and I. G. Kevrekidis, "A kernel-based method for data-driven koopman spectral analysis," *Journal of Computational Dynamics*, vol. 2, no. 2, pp. 247–265, 2015.

[105] S. Klus, P. Koltai, and C. Schütte, "On the numerical approximation of the perron-frobenius and koopman operator," *Journal of Computational Dynamics*, vol. 3, no. 1, pp. 51–79, 2016.

[106] K. Fujii and Y. Kawahara, "Dynamic mode decomposition in vector-valued reproducing kernel hilbert spaces for extracting dynamical structure among observables," *Neural networks : the official journal of the International Neural Network Society*, vol. 117, pp. 94–103, 2018.

[107] S. Das and D. Giannakis, "Koopman spectra in reproducing kernel hilbert spaces," *Applied and Computational Harmonic Analysis*, vol. 49, no. 2, pp. 573–607, 2020.

[108] M.Korda and I.Mezic, "On convergence of extended dynamic mode decomposition to the koopman operator," *Journal of Nonlinear Science*, vol. 28, pp. 687–710, 2018.

[109] S. Otto and C. Rowley, "Koopman operators for estimation and control of dynamical systems," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, pp. 59–87, May 2021. Publisher Copyright: © 2021 by Annual Reviews. All rights reserve.

[110] I. Abraham, G. D. L. Torre, and T. D. Murphey, "Model-based control using koopman operators," *ArXiv*, vol. abs/1709.01568, 2017.

[111] D. Bruder, B. Gillespie, C. D. Remy, and R. Vasudevan, "Modeling and control of soft robots using the koopman operator and model predictive control," *CoRR*, vol. abs/1902.02827, 2019.

[112] A. Narasingam and J. S.-I. Kwon, "Koopman lyapunov-based model predictive control of nonlinear chemical process systems," *AIChE Journal*, vol. 65, no. 11, p. e16743, 2019.

[113] M. Netto and L. Mili, "A robust data-driven koopman kalman filter for power systems dynamic state estimation," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 7228–7237, 2018.

[114] E. Ling, L. Ratliff, and S. Coogan, "Koopman operator approach for instability detection and mitigation in signalized traffic," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, p. 1297–1302, IEEE Press, 2018.

[115] A. Hasnain, N. Boddupalli, S. Balakrishnan, and E. Yeung, "Steady state programming of controlled nonlinear systems via deep dynamic mode decomposition," in *2020 American Control Conference (ACC)*, pp. 4245–4251, 2020.

[116] J. Hogg, M. Fonoberova, I. Mezić, and R. Mohr, "Koopman mode analysis of agent-based models of logistics processes," *PLOS ONE*, vol. 14, pp. 1–37, 09 2019.

[117] M. O. Williams, M. S. Hemati, S. T. M. Dawson, I. G. Kevrekidis, and C. W. Rowley, "Extending data-driven koopman analysis to actuated systems," *In IFAC Symposium on Nonlinear Control Systems (NOLCOS)*, 2016.

[118] A. Surana, "Koopman operator based observer synthesis for control-affine nonlinear systems," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 6492–6499, 2016.

[119] E. Kaiser, J. N. Kutz, and S. L. Brunton, "Discovering conservation laws from data for control," *2018 IEEE Conference on Decision and Control (CDC)*, pp. 6415–6421, 2018.

[120] A. Sootla and A. Mauroy, "Properties of isostables and basins of attraction of monotone systems," in *2016 American Control Conference (ACC)*, pp. 7365–7370, 2016.

[121] A. Mauroy and I. Mezić, "Global stability analysis using the eigenfunctions of the koopman operator," *IEEE Transactions on Automatic Control*, vol. 61, no. 11, pp. 3356–3369, 2016.

[122] U. Vaidya, "Observability gramian for nonlinear systems," in *2007 46th IEEE Conference on Decision and Control*, pp. 3357–3362, 2007.

[123] D. Goswami and D. A. Paley, "Global bilinearization and controllability of control-affine nonlinear systems: A koopman spectral approach," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 6107–6112, 2017.

[124] S. Sinha, U. Vaidya, and R. Rajaram, "Operator theoretic framework for optimal placement of sensors and actuators for control of nonequilibrium dynamics," *Journal of Mathematical Analysis and Applications*, vol. 440, no. 2, pp. 750–772, 2016.

[125] H. Sharma, U. Vaidya, and B. Ganapathysubramanian, "A transfer operator methodology for optimal sensor placement accounting for uncertainty," *Building and Environment*, vol. 155, pp. 334–349, 2019.

[126] U. Eren, A. Prach, B. Koçer, S. Rakovic, E. Kayacan, and B. Açikmese, "Model predictive control in aerospace systems: Current state and opportunities," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 7, pp. 1541–1566, 2017.

[127] M. N. Zeilinger, D. M. Raimondo, A. Domahidi, M. Morari, and C. N. Jones, "On real-time robust model predictive control," *Automatica*, vol. 50, no. 3, pp. 683–694, 2014.

[128] A. Alessio and A. Bemporad, "A survey on explicit model predictive control," *Nonlinear Model Predictive Control: Towards New Challenging Applications*, pp. 345–369, 2009.

[129] Z. Bai, E. Kaiser, J. L. Proctor, J. N. Kutz, and S. L. Brunton, "Dynamic mode decomposition for compressive system identification," *CoRR*, vol. abs/1710.07737, 2017.

[130] J. L. Proctor, S. L. Brunton, B. W. Brunton, and J. N. Kutz, "Exploiting sparsity and equation-free architectures in complex systems," *European Physical Journal Special Topics*, vol. 223, Dec. 2014.

[131] S. Yu, E. Sheng, Y. Zhang, Y. Li, H. Chen, and Y. Hao, "Efficient nonlinear model predictive control of automated vehicles," *Mathematics*, vol. 10, no. 21, 2022.

[132] Y. Kawahara, "Dynamic mode decomposition with reproducing kernels for koopman spectral analysis," in *Advances in Neural Information Processing Systems* (D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, eds.), vol. 29, Curran Associates, Inc., 2016.

[133] L. Ljung., *System identification. In Signal Analysis and Prediction*. Springer, 1998.

[134] E. Kaiser, J. N. Kutz, and S. L. Brunton, *Data-Driven Approximations of Dynamical Systems Operators for Control*, pp. 197–234. Cham: Springer International Publishing, 2020.

[135] D. Batstone, J. Keller, I. Angelidaki, S. Kalyuzhnyi, S. Pavlostathis, A.Rozzi, W. Sanders, H. Siegrist, and V. Vavilin, "The iwa anaerobic digestion model no 1 (adm1)," *Water Sci. Technol.*, vol. 45, pp. 65–73, 2002.

[136] C. Rosén and U. Jeppsson, *Aspects on ADM1 Implementation within the BSM2 Framework*, vol. 7224 of *TEIE*. Department of Industrial Electrical Engineering and Automation, Lund Institute of Technology, 2005.

[137] W. J. Parker, "Application of the adm1 model to advanced anaerobic digestion," *Bioresource Technology*, vol. 96, no. 16, pp. 1832–1842, 2005.

[138] L. Xue, D. Li, and Y. Xi, "Nonlinear model predictive control of anaerobic digestion process based on reduced adm1," in *2015 10th Asian Control Conference (ASCC)*, pp. 1–6, IEEE, 2015.

[139] C. Rosen, D. Vrecko, K. Gernaey, M.-N. Pons, and U. Jeppsson, "Implementing adm1 for plant-wide benchmark simulations in matlab/simulink," *Water Science and Technology*, vol. 54, no. 4, pp. 11–19, 2006.

[140] C. García-Diéguez, O. Bernard, and E. Roca, "Reducing the anaerobic digestion model no. 1 for its application to an industrial wastewater treatment plant treating winery effluent wastewater," *Bioresource Technology*, vol. 132, pp. 244–253, 2013.

[141] B. Benyahia, T. Sari, B. Cherki, and J. Harmand., "Bifurcation and stability analysis of a two step model for monitoring anaerobic digestion processes.," *Journal of Process Control*, vol. 22, no. 6, pp. 1008–1019, 2012.

[142] L. Ljung, *System Identification: Theory for the User*. Prentice Hall information and system sciences series, Prentice Hall PTR, 1999.

[143] J. Monod, "La technique de culture continue theorie et application," *Ann. Inst.Pasteur*, vol. 79, p. 390–410, 1950.

[144] J. B. S. Haldane, *Enzymes*. Longmans, London, 1930.

[145] V. S. Zuzana Majdisova, "Radial basis function approximations:comparison and applications," *Applied Mathematical Modelling*, vol. 51, pp. 728 – 743, 2017.

[146] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "qpoases :a parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, pp. 327–363, 2014.

[147] P. J. Baddoo, B. Herrmann, B. J. McKeon, J. Nathan Kutz, and S. L. Brunton, "Physics-informed dynamic mode decomposition," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 479, no. 2271, p. 20220576, 2023.

[148] B. Lusch, S. L. Brunton, and J. N. Kutz, "Data-driven discovery of Koopman eigenfunctions using deep learning," in *APS Division of Fluid Dynamics Meeting Abstracts*, APS Meeting Abstracts, p. M1.006, Nov. 2017.

[149] B. Lusch, J. N. Kutz, and S. L. Brunton, "Deep learning for universal linear embeddings of nonlinear dynamics," *Nature Communications*, vol. 9, p. 4950, Nov. 2018.

[150] E. Yeung, S. Kundu, and N. O. Hodas, "Learning deep neural network representations for koopman operators of nonlinear dynamical systems," *CoRR*, vol. abs/1708.06850, 2017.