Data-driven development of advanced controllers for complex reaction systems with minimal prior information

by

Shahdab Mohamedimran Pathan

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Chemical Engineering

Department of Chemical and Materials Engineering
University of Alberta

Abstract

In the realm of complex reactive systems where full knowledge of ongoing reactions is unattainable, the adoption of data-driven inferential models based on mixture spectra has gained significant traction. Spectra-based online monitoring has shown promise due to the rapidity, non-invasiveness, non-destructiveness, and cost-effectiveness in spectral analysis. This study aims to develop advanced controllers for such complex reactive systems in the absence of ground truth information and subsequently compare their performances. To achieve this objective, a comprehensive suite of tools, including spectral deconvolution, Bayesian networks, neural ordinary differential equations (ODEs), long short-term memory (LSTM), model predictive control (MPC), and reinforcement learning are employed to transform spectra into actionable control strategies. The initial phase of the research focuses on establishing a model-based control framework through the utilization of spectral deconvolution and Bayesian networks, particularly in scenarios where ground truth knowledge is limited or the system dynamics are complex. Spectral deconvolution untangles pseudo component spectra and their corresponding concentration profiles from mixture spectra. These deconvoluted spectra serve as the Bayesian network's inputs, effectively identifying potential reaction networks within the system. Concurrently, neural ODEs leverage the concentration profiles obtained from spectral deconvolution to extract rate law parameters and facilitate step-ahead concentration predictions. This holistic approach results in a comprehensive rate-law-based kinetic model that captures the reaction system's dynamics. Two modeling approaches are employed and compared: a data-driven LSTM and a physics-driven grey-box model utilizing Neural ODE. While the LSTM model operates as a black box, providing step-ahead concentration predictions, the Neural ODE model represents a grey-box approach incorporating first principles, also generating step-ahead predictions. The aim is to evaluate

the performance of these approaches, contrasting the efficacy of the data-driven black box model (LSTM) with the physics-driven grey-box model (Neural ODE). In the latter phase of the study, reinforcement learning-based techniques are leveraged to design a model-free controller with a focus on optimizing the selectivity of desired products, like in MPC with neural ODE as model/environment.

For future work, the focus will be on leveraging spectra corresponding to specific wavenumber ranges that are indicative of the functional groups associated with target products. This strategy diverges from previous approaches, such as the deconvolution pathway that emphasized modeling the kinetics. Instead, the plan is to adopt a model that utilizes mixture spectra as inputs. This model, in its control segment, will be designed to incentivize the agent or controller to prioritize selectivity towards certain products and/or wavenumber ranges. This methodology enables the system to refine its control strategy by relying solely on spectral data. This is particularly beneficial in situations where a comprehensive understanding of the system's dynamics is not available, thus circumventing the need to develop detailed kinetic models. In conclusion, this work harnesses a range of advanced modelling and control methodologies to translate spectral data into actionable control strategies for complex reactive systems. The efficacy of the developed controllers is demonstrated through a simulation environment of a CSTR aimed at maximizing the selectivity of a desired species, thereby achieving the desired overall system performance.

Preface

This thesis is the original work of Shahdab Pathan. The part of this work that focuses on obtaining deconvolved data and reaction networks has been adapted from Dr. Anajana Puliyanda's work under the supervision of Dr. Vinay Prasad. The sections on Neural ODE and other subsequent sections are original works by Shahdab Pathan.

Acknowledgements

I am extremely grateful for the guidance and support of Dr. Prasad throughout my research journey. His conduct has been exceptional and unwavering, making it possible for me to navigate through challenging situations.

Dr. Anjana Puliyanda and Dr. Karthik Srinivasan have played a crucial role in my academic and personal development with their unwavering support and guidance. I have learned valuable lessons from their dedication to fostering curiosity and perseverance, and I will always cherish their endless patience towards me.

I would also like to express my gratitude to Dr. Abhishek Yadav for his encouragement and belief in my potential. His insights laid the foundation for my research and motivated me to pursue a master's degree.

The support of my friends Nisarg, Jhanvi, and Arpit Patel has been invaluable during my time away from home. Their friendship and encouragement have made this journey more joyous.

Ishant Godariya has been a constant support since my undergraduate days, and I am grateful to him for that. I'd also like to thank Devavrat Thosar, Abhijit Bhakte, Yash Kothari and Kevin Mao for their constant support and wisdom that immensely contributed to my personal and academic growth.

Above all, I cannot express enough gratitude to my parents and my sister. Their endless love, support, and sacrifices have been the cornerstone of my strength and perseverance.

To all who have contributed to my journey, big and small, I am eternally grateful for your support. Your encouragement has been the driving force that propelled me forward and inspired me to reach new heights. This journey has impacted my academic learning and taught me valuable life lessons that I will carry with me throughout my life.

Table of Contents

List of Figures

List of Tables

List of Symbols

k, k_1, k_2, k_3	Reaction rate constants
E_a, E_{a1}, E_{a2}, E_{a3}	Activation energies
T	Temperature
C_i, C_A, C_B, C_C, C_D	Reactant/product concentrations
R	Gas constant
$Rh(111)$	Rhodium surface
t	Time
V	Reactor volume
r	Reaction rate
τ	Update rate
$C_{A0}, C_{B0}, C_{C0}, C_{D0}$	Inlet concentrations
k_0, k_{01}, k_{02}, k_{03}	Pre-exponential factors
$F(t)$	Flow rate
C_{t+n}	Future concentration
A	Absorbance
ε	Molar absorptivity
l	Light path length
c	Species concentration
a, b	Reaction orders
WF	Flow weight factor
C'	Modified concentration
X'	Data matrix

S	Scores matrix
L	Loadings matrix
E	Residual matrix
x_{ijk}	Data array entry
a_{if}, b_{jf}, c_{kf}	Factorization components
e_{ijk}	Factorization error
D	Result matrix
$\mathcal{X}$	Data tensor
$\mathbf{W}, \mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_N$	Decomposition matrices
λ	Regularization parameter
$\mathbf{G}$	Estimated tensor
$\mathbf{T}$	Target tensor
$P(X_1, X_2, \ldots, X_N)$	Probability distribution
$Pa(X_i)$	Node parents
$AIC(M_k)$	Akaike Information Criterion
$BIC(M_k)$	Bayesian Information Criterion
$\frac{d\mathbf{h}(t)}{dt}$	State vector rate
f	Dynamics function
θ	Network parameters
$W^{(1)}$	First weight matrix
$W^{(2)}$	Second weight matrix
H_t	Hidden state
$\frac{dC'}{dt}$	Modified rate change
$\hat{C}_{t+n}$	Predicted future concentration
L_{conc}	Concentration loss function
L_{rate}	Rate loss function
$L_{weights}$	Weights loss function
L_{total}	Total loss function
X_t	Input data
$b^{(1)}$	First layer bias

C_t	Cell state
f_t	Forget gate activation
i_t	Input gate activation
$\tilde{C}_t$	Candidate state
h_t	Hidden state
o_t	Output gate activation
$V^{\pi}(s)$	Policy value function
$\pi(a \mid s)$	Action probability
$P(s', r \mid s, a)$	Transition probability
$R(s, a, s')$	Reward function
a_t	Action
s_t	Current state
μ_θ	Policy network
$Q(s_t, a_t \mid \theta^Q)$	Q-function
$\theta^Q, \theta^{Q'}$	Network parameters
τ	Soft update parameter
r	Reward
$state_3, state_4$	Species concentrations
sp	Setpoint values
$J(x(t), u)$	Cost function
ϵ	Convergence threshold
J	Cost function
N	Prediction horizon
x_0	Initial state
u_0	Initial control
γ	Discount factor
Q, R	Cost weights
P	Terminal weight
F	Flow rate
$S(t)$	FTIR spectrum

$c_i(t)$	Species concentration
S_i	Component spectrum

Abbreviations

CRNN: Chemical Reaction Neural Network

LSTM: Long Short-Term Memory

ML: Machine Learning

MPC: Model Predictive Control

RL: Reinforcement Learning

FTIR: Fourier-Transform Infrared Spectroscopy

NMR: Nuclear Magnetic Resonance

GC: Gas Chromatography

LC: Liquid Chromatography

DRNN: Deep Recurrent Neural Network

DRO: Deep Reaction Optimizer

DDPG: Deep Deterministic Policy Gradient

ESC: Extremum Seeking Control

DFT: Density Functional Theory

SOAP: Smooth Overlap of Atomic Positions

RNN: Recurrent Neural Network

CSTR: Continuous Stirred-Tank Reactor

NARX: Nonlinear Autoregressive Exogenous Model

ANN: Artificial Neural Networks

ODE: Ordinary Differential Equation

NIST: National Institute of Standards and Technology

HPLC: High-Performance Liquid Chromatography

EVs: EigenValues

ALS: Alternating Least Squares

PLS: Partial Least Squares

DAG: Directed Acyclic Graph

UV-vis: Ultraviolet–visible spectroscopy

PCA: Principal Component Analysis

EM: Expectation Maximization

AIC: Akaike Information Criterion

BIC: Bayesian Information Criterion

HC: Hill Climbing

^{1}H NMR: proton nuclear magnetic resonance

MMHC: Maximum Minimum Hill Climbing

SNR: Signal-to-Noise Ratio

EFA: Evolving Factor Analysis

PARAFAC: Parallel Factor Analysis

MCR: Multivariate Curve Resolution

SMCR: Self-modeling Multivariate Curve Resolution

JNTF: Joint Non-negative Tensorial Factorization

LOF: Lack of Fit

PC: Pseudo-component

SMILES: Simplified Molecular Input Line Entry System

NODE: Neural Ordinary Differential Equation

PINN: Physics-Informed Neural Networks

PID: Proportional-Integral-Derivative

DP: Dynamic Programming

MDPs: Markov Decision Processes

MC: Monte Carlo

TD: Temporal-Difference

DQN: Deep Q-Networks

PPO: Proximal Policy Optimization

SAC: Soft Actor-Critic

TD3: Twin Delayed DDPG

A2C: Advantage Actor-Critic

SARSA: State-Action-Reward-State-Action

MFPC: Model-Free Predictive Control

MFLC: Model-Free Learning Control

Chapter 1

Introduction

1.1 Background and motivation

The pursuit of sustainable and efficient production processes in chemical engineering has necessitated a shift towards more complex and often renewable feedstocks. This transition is motivated by economic, environmental, and technological factors, aiming to reduce reliance on finite natural resources and mitigate carbon emissions. However, integrating such feedstocks introduces significant challenges, particularly in the context of chemical reactors, where the variability and complexity of these materials complicate process control, optimization, and scale-up efforts.

Despite the environmental and economic incentives for adopting renewable feedstocks, the industry's existing infrastructure and technological paradigms heavily favour conventional fossil fuels. The development of technologies for converting bio-based feedstocks to chemicals remains a daunting task[1], hampered by the intrinsic variability of these materials and the lack of reliable kinetic models that account for hydrodynamics and transport phenomena essential for process scaling, specifically in a reactor.[2] The need for comprehensive and accurate reactor models is paramount, as they underpin the efficiency and reliability of the entire production process.

The inherent variability in complex feedstocks leads to numerous operational challenges, including inconsistent input quality, process inefficiencies, and increased production of off-spec products. Addressing these issues requires a deep understanding of the kinetics of

chemical reactions within reactors, a task often hindered by the cost and complexity of pilot-scale experiments. Moreover, the scalability of processes designed for complex feedstocks could be improved with technical hurdles, such as plugging, choking, and corrosion, further complicating the transition to sustainable production methods.[3]

This backdrop sets the stage for this research, which seeks to bridge the gap between the potential of renewable feedstocks like biomass and other such reaction systems comprising complex and unknown interactions among components and the practicalities of their utilization in industrial chemical processes. By focusing on data-driven control strategies for complex reaction systems, this work aims to come up with data-driven and, at the same time, inferential models to contribute to development of more flexible, efficient, and sustainable production processes that can be controlled to increase the yield or selectivity of desired components while minimizing the undesired ones at the same time.

1.1.1 Reaction Kinetics Modeling

Reaction kinetics is a fundamental pillar in understanding the rate at which chemical reactions proceed, affecting industries ranging from pharmaceutical development to environmental management. This area of study focuses on how variables like temperature and reactant concentrations impact reaction rates, anchored by key concepts such as reaction rate, order, and activation energy. The exploration of reaction kinetics has evolved significantly, transitioning from hands-on experimental methods to sophisticated computational models. This shift has allowed for more efficient and scalable approaches to understanding complex chemical systems. Petzold and Zhu[4] pioneered a model simplification technique that reduces the complexity of chemical systems without trading off accuracy. Their method optimizes computational resources while maintaining a high level of modeling precision. Building on this computational approach, Toch et al.[5] emphasized the importance of combining statistical analysis with kinetic modeling. By starting with a qualitative analysis of experimental data to inform rate equations, they managed to maintain both the physical integrity and statistical significance of their models, demonstrating the value of a balanced approach.

Kemmler et al.[6] showcased a different angle by focusing on real-time parameter estimation during chemical reactions. Their use of calorimetric data and simulation optimization

revealed the potential for models to adapt and improve as more data becomes available, highlighting the dynamic nature of reaction kinetics modeling.

Westbrook and Dryer[7] merged the study of chemical kinetics with combustion modeling, revealing the complex structure of reaction mechanisms across different hydrocarbon fuels. Their work emphasized the importance of robust models validated through empirical data to accurately model combustion processes. The progression in kinetic studies from traditional experimental techniques to advanced computational modeling demonstrates the ongoing effort to refine our understanding of chemical processes, ensuring models are both computationally efficient and grounded in physical reality.

1.1.2 Complex Reaction Network Modeling

The kinetic modeling of complex reaction systems is pivotal in chemical engineering, especially for optimizing chemical processes across a variety of industries. This literature review synthesizes seminal contributions to the field, spotlighting methodological advancements and innovations that tackle the multifaceted challenges of modeling these systems.

James et al.[8] pioneered a structural approach for analyzing chemical reactions, focusing particularly on monomolecular systems. Their methodology, which blends qualitative and quantitative analyses through geometric interpretations and matrix transformations, simplifies complex reaction dynamics and enhances the precision of kinetic modeling and the predictability of reaction behaviours. By extending their analysis to pseudo monomolecular systems, they demonstrated the framework's broad applicability, bridging theoretical models with practical applications and thus enriching the toolkit for addressing catalysis and enzyme chemistry challenges.

Building upon these theoretical foundations, Dryer et al.[9] delved into modeling chemical reactions within flow reactors. Their study critically examined the plug flow assumption and developed strategies to manage uncertainties in reaction initialization, thereby improving kinetic model development and validation. This work is distinguished by its detailed examination of axial and radial gradients and complex kinetics, significantly enriching our understanding of reactor dynamics. This study thus serves as a baseline for modeling complex reactions and integrating reactor dynamics effectively.

Zhang et al.[10] introduced a systematic approach to model development in complex chemical systems, emphasizing the integration of experimental evaluations with model construction. Characterized by an iterative refinement process, this methodology starts with establishing a comprehensive reaction network from preliminary experiments. This network forms the basis for generating and refining simplified models through targeted experiments, marking a significant stride in model development and reactor design optimization.

Frenklach et al.[11] presented a transformative methodology for converting experimental data into predictive models, focusing primarily on combustion chemistry. This approach notably emphasizes collaborative data analysis by integrating response surface techniques and robust control theory within a collaborative data processing framework. The methodology significantly boosts model predictions by directly incorporating uncertainties.

Complementary to these methods, Okino et al.[12] investigated the simplification of mathematical models for chemical reaction systems, advocating for model reduction to navigate computational demands and parameter uncertainty. Alongside contribution from Prickett et al.[13] this body of work expands the computational toolkit for analyzing complex chemical systems, automating the generation of complex reaction networks, exploring reaction systems as mathematical formalisms, and employing stochastic methods for systematic uncertainty analysis.

Experimental and mathematical combinations have been reliable and less computationally intensive for modeling complex reaction systems. However, atomistic and molecular simulation studies have also been conducted to predict equilibrium behaviour in non-ideal environments and simulate chemical reactions. Neurock et al.[14] and Turner et al.[15] introduced Monte Carlo methods in molecular interactions, reaction kinetics, and the discovery of new chemical pathways. These computational simulations have significant potential in elucidating the complexities of chemical reactions and their mechanisms.

Collectively, these studies represent a paradigm shift toward integrative, precise, and predictive models of chemical kinetics. The field has significantly evolved from structural analysis and model-building techniques to kinetic extraction from simulation data, enhancing the understanding of chemical reactions and facilitating the optimization of chemical processes.

1.1.3 Machine learning-based modeling of complex system

Machine learning (ML) has become an important tool in modeling complex reaction systems across various fields, offering predictive power and insights into system dynamics, efficiencies, and optimization strategies. These models typically involve using ML techniques to predict, simulate, and understand the behaviour of complex chemical reactions, where complexity arises due to the involvement of numerous reactants, products, intermediates, and pathways. ML models, including neural networks, regression models, and decision trees, have been employed to tackle various challenges in this domain.

Traditional deterministic approaches to modeling complex reaction systems rely on an understanding of mechanisms, facing challenges with nonlinear, high-dimensional systems when such mechanisms are unclear. Machine learning (ML) approaches offer a data-driven alternative that identifies patterns and optimizes conditions without explicit mechanistic knowledge. Hybrid methods merge these strategies, incorporating mechanistic insights into ML models to enhance predictability and efficiency while ensuring adherence to chemical principles. This integration improves accuracy and interpretability and reduces the data needed for practical training, presenting a robust solution for modeling the intricacies of chemical reactions by combining the precision of traditional models with the adaptability of ML techniques.

Kayala et al.'s [16] reaction predictor is one example that leverages machine learning to systematically predict complex chemical reactions, utilizing a two-stage framework that interprets reactions as interactions between approximate molecular orbitals. Initially, it filters the vast array of potential reactions at the molecular orbital level, then employs ranking models to prioritize likely productive reactions. The standout feature of the work is the mechanistic pathway predictor, which employs a constrained tree-search algorithm to suggest steps from reactants to products, marking an approach in multistep reaction prediction, thus automating and systematizing reaction prediction beyond traditional rule-based methods, thereby enhancing synthesis planning and chemical research.

Meuwly [17] highlighted the role of ML in advancing the understanding and prediction of chemical reactions, from small molecule dynamics to complex reaction networks. ML

techniques, such as Bayesian inference, Gaussian processes, and neural networks, deliver alternatives to traditional computational methods, potentially enabling more accurate predictions of reaction rates, pathways, and outcomes. These approaches facilitate a deeper analysis of biological reactions, improve experimental chemistry practices, and bridge the gap between theoretical predictions and experimental data.

Stocker et al.[18] utilized a machine-learning framework to analyze chemical reaction spaces in their research. They examined the Rad-6 database, which contains over 10,000 molecules and 32,515 reactions. By employing Kernel Ridge Regression and the Smooth Overlap of Atomic Positions (SOAP) kernel, they were able to accurately predict atomization and reaction energies, highlighting the adaptation of ML strategies to the unique structures of reaction networks. The study also identified the need for intensive kernels and training sets to make precise predictions, which facilitated the extraction of simplified reaction networks for complex processes such as methane combustion, providing a streamlined approach to understanding and optimizing chemical reactions.

Blurock [19] presented a machine-learning approach to categorize complex chemical reaction mechanisms into distinct reactive phases without the influence of human bias. By utilizing a conceptual clustering method to analyze reaction sensitivity values, the study successfully segmented the Hochgreb and Dryer aldehyde combustion mechanism into three primary phases: the initial aldehyde reaction, an intermediate phase with dwindling aldehyde presence, and a final phase leading to end products. This technique harnessed normalized sensitivity constants and a hierarchy of clusters to automatically distinguish between different stages of the reaction based on the dynamics of each reaction's significance. This approach offered an impartial, data-driven analysis that enhanced understanding and potentially streamlined reaction mechanisms by identifying critical reactions within each phase, utilizing machine learning to illuminate chemical reaction progression systematically.

Ulissi et al. [20] presented a novel method for optimizing reaction networks in heterogeneous catalysis. This approach combined density functional theory (DFT) calculations with surrogate models to efficiently handle the complexity of surface reaction networks. This approach significantly reduced computational resources by focusing on potential rate-limiting steps with high-accuracy DFT calculations and utilizing surrogate models based on estab-

lished scaling relations and Gaussian process regression for the broader network. The syngas reaction over Rh(111) was used to demonstrate this method, revealing a more probable reaction mechanism with fewer DFT calculations. The methodology was characterized by its probabilistic framework for mechanism elucidation, simplicity, and potential for generalization to multisite or multicatalyst models. This work represented a significant step forward in computational catalysis, providing a practical tool for catalyst design and understanding experimental data.

In the work of Margraf et al., [21], the application of ML in comprehending and navigating catalytic reaction networks was explored. The integration of ML for both the direct computational construction of these networks and the interpretation of experimental data was highlighted. Through ML, efficiency was notably enhanced by the approximation of computational chemistry calculations, forecasting reaction behaviours, and pinpointing critical reaction pathways and intermediates. The study was distinguished by the proposal of a dual approach: a bottom-up method that constructs reaction networks from atomic-level simulations and a top-down method that formulates effective kinetic models from experimental data, leaving the need for detailed knowledge of the catalyst structure.

In their work, Fooshie et al.[22] built upon Kayala et al.'s[16] reaction predictor prototype to introduce a deep learning-based system for predicting chemical reactions, focusing on elementary reactions in organic chemistry. This approach leveraged a curated dataset of over 11,000 reactions to accurately predict reaction outcomes and pathways by identifying and pairing electron sources and sinks. The Reaction Predictor outperformed traditional methods with an 80% top-5 recovery rate on challenging benchmark reactions. What set this system apart was its ability to operate at the elementary reaction level, meaning that its predictions were interpretable and useful for identifying novel reactions and side products. The methodology included enhancements such as combinatorial reaction generation and the use of recurrent neural networks (RNNs), specifically long-short-term memory (LSTM) architectures, for direct operation on (Simplified Molecular Input Line Entry System)SMILES strings. These enhancements aimed to refine prediction accuracy and broaden reactant context comprehension.

Zhou et al.[23] developed the Deep Reaction Optimizer (DRO), which was aimed at en-

hancing the efficiency of chemical reaction optimization through the use of deep reinforcement learning. The experimental conditions were iteratively adjusted based on previous results, leading to a 71% reduction in optimization steps. Incorporating chemistry domain knowledge, the DRO was showcased for its strong generalizability and learning capability. New insights into reaction mechanisms were also provided, particularly in microdroplet chemistry. This approach was recognized as a significant advancement in the field of chemical reaction optimization, merging advanced machine-learning techniques with practical chemical understanding. These research methodologies thus demonstrate the integration of physics-based insights into data-driven models, whether through experimental work or modeling efforts, resulting in improved model performances.

1.1.4 Control of complex reaction systems

In exploring the domain of control strategies for complex reaction systems, various innovative approaches are highlighted across several studies, focusing on the challenges and solutions in optimizing these systems amidst uncertainties and nonlinear dynamics.

Komives et al.[24] shed light on the progress in bioreactor control, transitioning from model-based strategies in chemical processes to their adaptation for biological systems. The complexity of bioprocess variables like pH, temperature, and dissolved oxygen necessitates advanced control and monitoring strategies. The integration of diverse models and technologies, such as artificial neural networks and metabolic flux analysis, has marked substantial advancements in bioreactor state estimation and control, aiming at elevating process efficiency and stability.

An examination of control strategies for bioreactors and complex reaction systems [25, 26, 27, 28] highlights the efficacy of MPC and Extremum Seeking Control (ESC) in managing nonlinear dynamics. These approaches underscore the necessity of nuanced control strategies to address the precision required in handling the intricacies of chemical reactions and bioprocesses.

Vaidyanathan[29] explored the dynamics of the Brusselator chemical reaction system, a model for autocatalytic reactions demonstrating complex dynamics like nonlinear oscillations and bifurcations. The study introduced an adaptive control strategy for achieving

anti-synchronization between identical Brusselator systems with unknown parameters. Further, the exploration of adaptive control strategies and neural network models for CSTR dynamics by Knapp et al.[30] presents a case for the benefits of these controllers over traditional methods. Leveraging neural networks demonstrates significant improvements in error convergence and control performance, especially in complex chemical process environments. This adaptability is important in ensuring efficient control across a broad spectrum of operating conditions.

Al Seyab et al.[31] delve into training Differential Recurrent Neural Networks (DRNNs) for nonlinear MPC applications, showcasing the potential of DRNNs in predicting future dynamics and enhancing real-time control of complex processes.

In their study, Pan et al. [32] investigated the application of constrained reinforcement learning to optimize bioreactor operations. The primary objective was to maximize biomass production while maintaining operational constraints. By integrating constraints into the reinforcement learning framework, the research developed control strategies that improve performance and guaranteed the safety and stability of bioprocesses.

Petsagkourakis et al.[33] and Alhazmi et al.[34] discussed the utilization of reinforcement learning, which has gained traction as control choice recently and specifically, Policy Gradient methods and Deep Deterministic Policy Gradient (DDPG) RL, to optimize bioprocesses and complex chemical reactions within CSTRs. These approaches highlight the adaptability and efficiency of RL in overcoming traditional control limitations, emphasizing the shift towards model-free control solutions in chemical engineering.

The studies conducted in the field of control engineering have shown an ongoing evolution towards more data-driven and adaptive methodologies. Merging optimal control theory by means of MPC and RL techniques can lead to improved efficiency, robustness, and adaptability of chemical engineering and biotechnology processes. However, few studies have been conducted on the usage of complex yet inferential models that have great potential when combined with control approaches like MPC and RL, which can yield robust control.

1.2 Thesis Objectives

The main goal of this research is to use a hybrid modeling approach that is inferential to compare it with a black-box model and establish a baseline comparison. Later, the hybrid model is used to apply control strategies such as MPC and RL, with the aim of maximizing the desired species concentration and minimizing the undesired one by comparing them in setpoint tracking and disturbance rejection scenarios.

1.3 Thesis Outline

Chapter 2 of this thesis discusses the different modeling methods used, starting from the generation of FTIR spectra and ending with the creation of a reaction kinetics model. The hybrid modeling approaches employed include Joint Non-Tensorial Factorization, Bayesian Networks, and Neural Ordinary Differential Equations, which is compared with Long Short-Term Memory networks (LSTMs) as a black-box approach.

In Chapter 3, a comparison is made between the control methodologies of Model Predictive Control (MPC) and Reinforcement Learning (RL) for both setpoint tracking and disturbance rejection scenarios.

Finally, Chapter 4 presents a conclusions for various modeling and control methodologies used, laying a foundation for further research in the future.

Chapter 2

Data-driven modeling of complex reaction systems

Data-driven modeling is a rapidly evolving methodology, gaining widespread acceptance across numerous scientific and engineering disciplines. When integrated with mechanistic insights, this approach can yield complex yet interpretable models that encapsulate a dynamic understanding of processes. Research in this domain has spanned a diverse array of systems, including but not limited to biomass conversion, food processing, hydrocarbon combustion, drug delivery mechanisms, and tropospheric chemical interactions. Typically, the methodologies adopted in these studies have relied on statistical techniques, curve fitting, and experimentation under specific operating conditions to construct empirically fitted models that, while effective, often lack universality across different systems and operating conditions.

One critical aspect distinguishing recent advances from earlier studies is the push toward generalisability across a broad spectrum of operating conditions. This leap in model applicability is crucial for developing predictive tools that are not only accurate within narrow confines but are also robust and reliable under varied and yet untested scenarios. Another differentiating factor is the leveraging of cutting-edge computational techniques, such as machine learning and deep learning, supported by improving hardware to distill insights from vast datasets accompanied by experimental validations. These methodologies enable the identification of underlying patterns and relationships that traditional modeling approaches

may overlook, thereby enhancing the predictive capabilities and interpretability of the models. [35, 36, 37, 38]

Moreover, the incorporation of real-time data analytics and adaptive algorithms allows for the continuous refinement of models, ensuring they remain relevant and accurate as new data becomes available. This dynamic aspect of data-driven modeling is particularly advantageous for systems subject to rapid changes or those operating in unpredictable environments.

In addition to enhancing model generalizability and leveraging computational advancements, there is an increasing emphasis on the transparency and explainability of data-driven models. As these models find broader application in critical decision-making processes, the ability for users to understand and trust model predictions becomes paramount. Efforts to integrate domain-specific knowledge into the modeling process, therefore, serve not only to improve model performance but also to make these tools more accessible and interpretable to a wider audience.

The evolution of data-driven modeling indicates reliability in scientific and engineering research, characterized by models that are not only more predictive and robust across diverse conditions but also more transparent and interpretable. As these methodologies continue to mature, they promise to unlock novel insights and innovations across a wide range of fields, from environmental science to healthcare and beyond.

This chapter describes the framework adopted for the study of data-driven modeling, aimed at developing robust and interpretable models with a keen emphasis on generalizability across various operating conditions. Recognizing the critical need for models that not only provide predictive accuracy but also offer insights into the underlying processes

2.1 Modeling framework for data-driven approaches

In tackling the complexity of reactive systems, full identification of all reactions and species involved is often difficult to achieve or unattainable. Consequently, the research leans towards employing data-driven inferential models for system analysis. Emphasis has thus been placed on leveraging methodologies conducive to the surveillance of reaction mechanisms. Various techniques exist for reaction monitoring, including spectroscopy, ion signatures, se-

lected reaction monitoring, or tracking critical performance indicators.[39, 40] Chemometrics emerges as a pivotal branch within this domain, facilitating the examination of intricate data sets to extract meaningful insights on chemical reactions based on their spectral signatures. Spectroscopy stands out as an essential instrument for assessing reaction kinetics due to its non-invasive nature and the depth of chemical understanding it provides. Its inherent value is in the real-time observation of chemical transactions, offering precise insights into molecular structures, dynamics, and properties. When applied to reaction monitoring, chemometrics employs a diverse array of strategies that allow for the concurrent evaluation of numerous variables, thus augmenting the accuracy and distinctiveness of experimental outcomes. Tools such as multivariate analysis are crucial within chemometrics, enabling the extraction of relevant information from dense data sets by identifying patterns and correlations not visible through conventional means. This approach not only clarifies overlapping spectral signals but also boosts detection sensitivity and supports kinetic reaction modeling. [41]

The choice of system and monitoring objectives dictates the suitability of these indicators for effective surveillance. In this context, the system in question relies on Fourier Transform Infrared (FTIR) spectrometry for its monitoring needs. FTIR spectra is the most commonly used tool for the identification of unknown materials as it helps with the identification of functional groups that are present in the system being monitored.[42] Studies focusing on organic compounds and biomass focusing on a combination of series and parallel reactions utilize FTIR spectra for monitoring and modeling purposes.

FTIR, UV-vis spectroscopy, Raman spectroscopy, mass spectroscopy, and nuclear magnetic resonance (NMR) are among the spectroscopic instruments most commonly used to track chemical composition changes, offering the advantage of rapid data collection. Techniques like gas chromatography (GC) and liquid chromatography (LC) provide high specificity for distinguishing between similar compounds, the ability to quantify substances, low detection limits, and the straightforward separation of components in complex mixtures. The integration of spectroscopic and chromatographic techniques is gaining traction as an effective strategy for the real-time monitoring chemical processes by leveraging the unique benefits of both methods.

Aside from the tools that spectroscopic monitoring provides, an abundance of IR spectra

of pure compounds is available on commercial and open-source platforms. Further qualitative monitoring of change in peaks of compounds over time and for specific wavenumbers provide insights into changes happening in a reaction with an underlying assumption that the mixture spectra is resultant of a linear combination of the pure component spectra of the species involved weighted by the fraction or the concentration of the species present in the reaction system.

A set of methods has been adopted in a sequence that is followed by FTIR spectra generation for a synthetic system to obtain the pseudo-component spectral signatures and concentration profiles that lay a foundation for the kinetic models that will be developed. Figure 2.1 shows the workflow that has been adapted from [43], which discusses the methodology to move from a mixture of FTIR spectra to pseudo-component concentration and spectral profiles[44] that help in the creation of the kinetic model, which is one of the aims of the study.

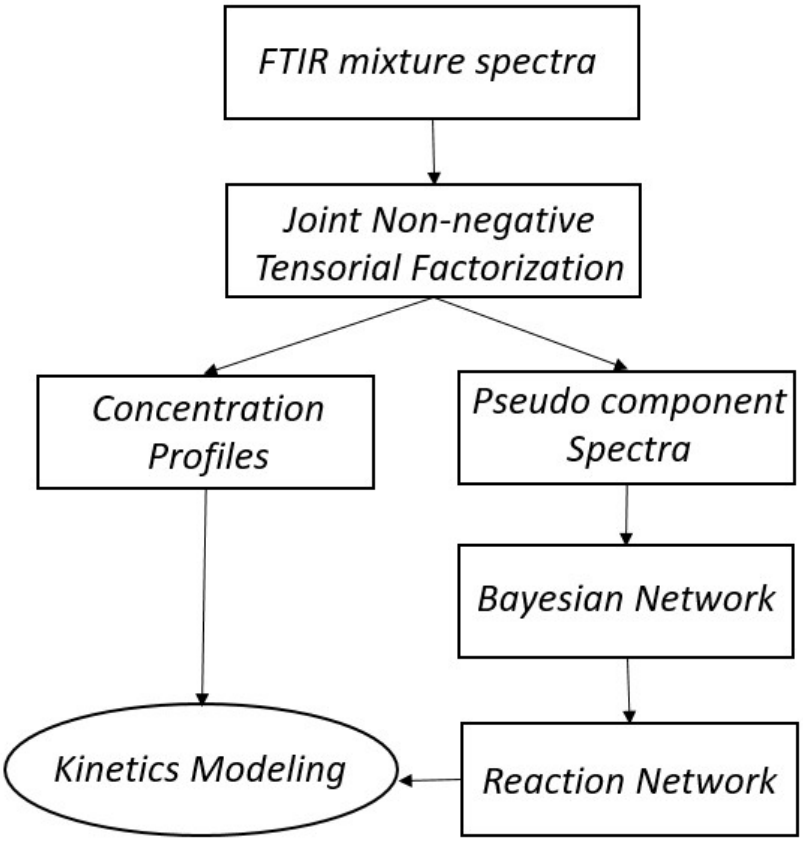

Figure 2.1: Mixture spectra to model generation pathway. Adapted from [43]

The workflow depicted in (2.1) is explained step by step:

1. **FTIR Mixture Spectra:** The process initiates with acquiring FTIR spectra from the mixture in the reaction system.

2. **Joint Non-negative Tensorial Factorization:** This technique decomposes a mixture of FTIR spectra into constituent components, yielding interpretable concentration profiles and pseudo-component spectra.

3. **Concentration Profiles:** Derived from factorization, these profiles chart the concentration changes of various species, important for understanding reaction kinetics.

4. **Pseudo Component Spectra:** These simplified spectra represent the individual or pseudo-components within the mixture, explaining the underlying chemical behaviour.

5. **Bayesian Network:** Utilizing Bayesian inference, this network models the probabilistic relationships in component spectra, resulting in a probabilistic reaction network.

6. **Kinetics Modeling:** The pseudo-component concentration profiles and Bayesian network are used to constrain the kinetics model used to predict reaction rates.

2.1.1 Generating synthetic FTIR Spectra for reactions in a Continuous Stirred Tank Reactor

Understanding the underlying truth of chemical reactions allows us to validate the predictions from our framework. This task becomes challenging for complex systems where the precise details about species, their reaction pathways, and kinetics might be elusive or difficult to determine. In this study, we generate synthetic spectroscopic data based on the species' pure component FTIR profiles. This data follows a reaction template from the Denbigh reaction system, with the FTIR profiles sourced from the NIST database [45], which has a network of reactions characterized by both series and parallel pathways.

As illustrated in Figure 2.2, the chosen network comprises two parallel reactions and two series pathways. The initial reaction (from ethane to chloroethane) produces an intermediate. The subsequent parallel reactions (from chloroethane to acetic acid and chloroethane to

ethanol) are competitive, modeling a scenario where one pathway yields a desired product while the other leads to an undesired outcome. Following the acquisition of pure IR spectra for these components from the NIST database, a kinetic model becomes essential to capture the dynamic interplay of reactant, intermediate, and product species.

A standard approach to model the reaction kinetics within a Continuous Stirred Tank Reactor (CSTR) involves employing rate laws where the rate constant follows the Arrhenius equation, augmented with terms to account for inflow and outflow within the CSTR. The employed rate law is based on the power law model:

$$r = k[A]^x[B]^y \tag{2.1}$$

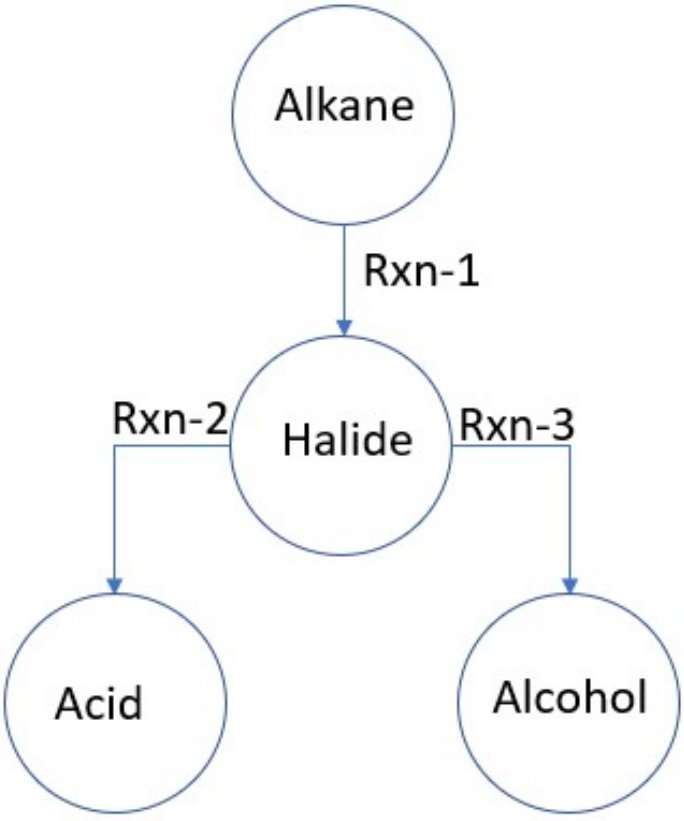

Figure 2.2: Denbigh reaction network

where x and y represent the reaction orders, and k is the rate constant defined as:

$$k = k_0 \exp\left(-\frac{E_a}{RT}\right) \tag{2.2}$$

Here, k_0 denotes the pre-exponential factor, E_a the activation energy, R the gas constant, and T the temperature. The rate constants thus characterize the reactions occurring within the CSTR:

$$k_1 = k_{01} \exp\left(-\frac{E_{a_1}}{RT(t)}\right) \tag{2.3}$$

$$k_2 = k_{02} \exp\left(-\frac{E_{a_2}}{RT(t)}\right) \tag{2.4}$$

$$k_3 = k_{03} \exp\left(-\frac{E_{a_3}}{RT(t)}\right) \tag{2.5}$$

The governing differential equations for the CSTR are formulated as follows:

$$\frac{dC_A}{dt} = \frac{F(t)}{V} \cdot (C_{A0} - C_A) - k_1 C_A \tag{2.6}$$

$$\frac{dC_B}{dt} = \frac{F(t)}{V} \cdot (C_{B0} - C_B) + k_1 C_A - k_2 C_B^2 - k_3 C_B \tag{2.7}$$

$$\frac{dC_C}{dt} = \frac{F(t)}{V} \cdot (C_{C0} - C_C) + k_2 C_B^2 \tag{2.8}$$

$$\frac{dC_D}{dt} = \frac{F(t)}{V} \cdot (C_{D0} - C_D) + k_3 C_B \tag{2.9}$$

In these equations, $F(t)$ is the time-dependent flow rate, V the reactor volume, and C_{i0} the inlet concentrations of the respective species. The parameters of the model are presented in table 2.1.

Parameter	Description	Value	Units
k_{01}	Pre-exponential factor for k_1	1×10^{-2}	s^{-1}
k_{02}	Pre-exponential factor for k_2	1.5×10^{-2}	$m^3\ mol^{-1}\ s^{-1}$
k_{03}	Pre-exponential factor for k_3	5×10^{-2}	s^{-1}
E_{a_1}	Activation energy for k_1	50,000	J/mol
E_{a_2}	Activation energy for k_2	10,000	J/mol
E_{a_3}	Activation energy for k_3	75,000	J/mol
R	Universal gas constant	8.314	J/(mol K)

Table 2.1: Rate constants and parameters

The simulation of the CSTR dynamics involves incorporating random signals for both flow and temperature, as illustrated in figure 2.3. These random fluctuations allow for the exploration of different operational conditions within predefined ranges: temperature values ranging from 523 to 1023 K, and flow rates fluctuating between 0.01 and 0.05 m^3/s across 4001 timesteps. The generation of multi-level signals occurs by subdividing the temperature and flow ranges into subranges, and selecting a subrange for each temperature and flow data point to undergo random signal generation within that range. Subsequently, the ranges are switched. The nature of these fluctuations results in alterations that take place within specific temperature and flow intervals before transitioning to a different operational state. This method produces a dataset that accentuates significant concentration variations resulting from the combined impacts of flow and temperature fluctuations.

Having simulated the concentration profiles within the CSTR, we can now obtain FTIR mixture spectra. These spectra result from the linear combination of pure component spectra, weighted by the concentration of each component at specific timesteps. This process adheres to Beer's Law, expressed as:

$$A = \varepsilon l c \tag{2.10}$$

Here, A represents the absorbance, ε denotes the molar absorptivity, l signifies the path length of the light through the sample, and c is the concentration of the absorbing species. It is important to note the assumption of a constant path length, which significantly affects the measurements.

Consequently, the FTIR spectrum of the mixture at any given time t can be formulated as:

$$S(t) = \sum_i c_i(t) \cdot S_i \tag{2.11}$$

where $S(t)$ is the mixture's FTIR spectrum at time t, $c_i(t)$ is the concentration of the i^{th} species, and S_i represents the pure component FTIR spectrum of the i^{th} species.

Equation 2.11 yields the spectra displayed in figure 2.4 for the simulated system. Changes in the spectral peaks reveal variations in the concentrations of specific components, corresponding to particular wavenumbers that denote the presence of different functional groups. This demonstrates IR spectra generation for a hypothesized system using rate law, which

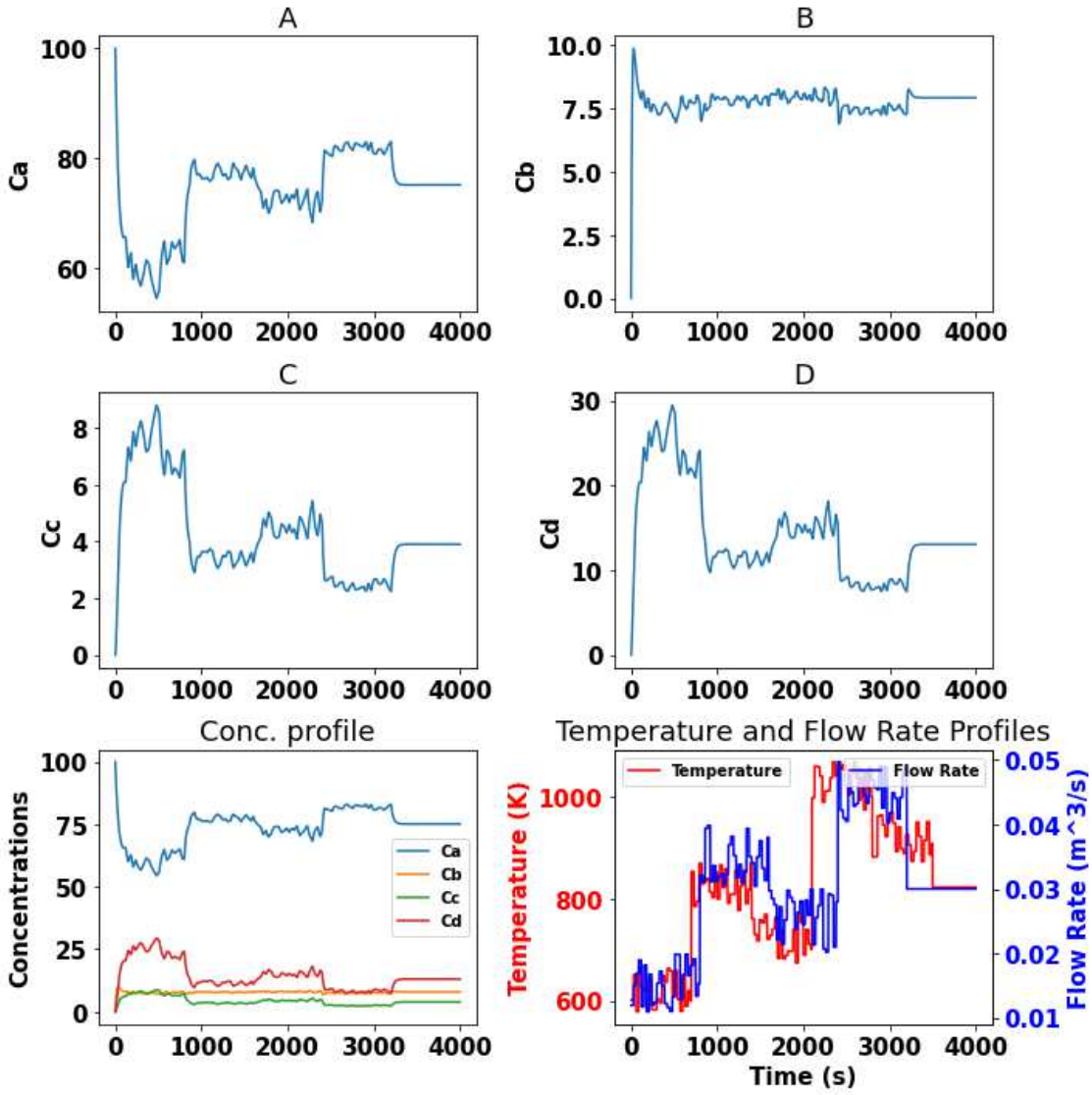

Figure 2.3: Simulated concentration profiles for the ODE system

will be subsequently used in coming sections.

2.1.2 Spectral Deconvolution

The first step in analyzing spectroscopic data involves pre-processing to eliminate noise that could potentially disrupt subsequent analysis. Such noise generally arises from various sources, including inaccuracies in instrument calibration, variations in the process, and uncertainties in measurements. To counter these challenges, the data undergoes a series of corrections and adjustments. These include baseline correction to normalize the data, scaling

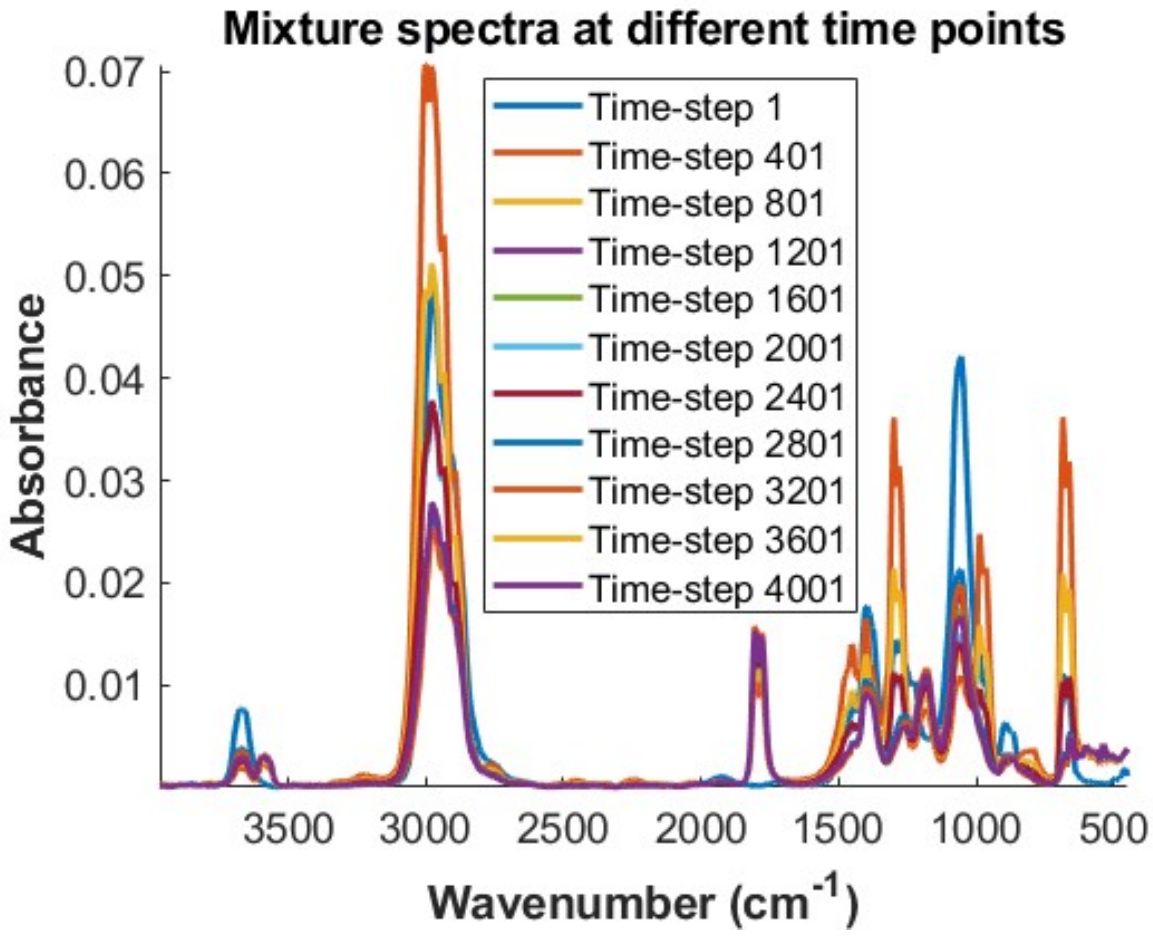

Figure 2.4: FTIR mixture spectra

to ensure uniformity across the dataset, and the identification and elimination or correction of any outliers using well-established techniques. Following these adjustments, data derived from different spectroscopic techniques can be enhanced. Specifically, for this study, given the synthetic nature of the data, artificial white gaussian noise is introduced to the signal, maintaining a signal-to-noise ratio (SNR) of 100%.

The mixture of FTIR spectra, now embedded with noise, is subjected to various analytical methods. These range from baseline corrections and filtering to more sophisticated chemometric tools such as Evolving Factor Analysis (EFA), Parallel Factor Analysis (PARAFAC), and Multivariate Curve Resolution (MCR). These tools are crucial for generating additional information necessary for Joint Tensor Factorization (JNTF) to derive approximations of component spectra and concentration profiles. The necessity for such analytical methods stems from the complex nature of spectral data, which is typically high-dimensional, non-

causal, lacks full rank, contains noise, and may have missing values.[46, 47]. After gathering the FTIR mixture spectra data, a suite of chemometric analysis techniques can be employed to gain deeper insights into the system under study. These methods allow for the identification of present compounds and the formulation of hypotheses regarding the reaction network. This analytical approach is elaborately discussed in [48, 49]. The sections henceforth focus on discussing various tools used in chemometrics and a useful tool for this study.

2.1.2.1 Evolving Factor Analysis

Evolving Factor Analysis (EFA)traces its roots back to 1985 with a pioneering report by Gampp et al.[50] This technique focuses on observing how the importance or 'rank' of a dataset changes against an ordered variable, often employing Principal Component Analysis (PCA) on datasets that expand with each measurement. This approach is instrumental in fields like analytical chemistry, aiding in deciphering complex data from techniques that adhere to Beer's law, such as High-Performance Liquid Chromatography(HPLC).[51, 52, 53]

EFA starts with the initial data point—for instance, the first spectrum obtained—and calculates the eigenvalues (EVs) successively for each subset of the data matrix X_i, including the first $i = 1 \dots P$ rows. This iterative process is encapsulated by the equation:

$$X' = SL + E \tag{2.12}$$

where X' represents the approximated data matrix, S (of dimension $i \times N$) is the scores matrix, L (dimension $N \times N$) the loadings matrix, and E the residual matrix capturing noise or errors. The core of this methodology lies in accurately determining the unknown number of factors, N. This typically involves initially calculating all Q eigenvectors of the data matrix, with S set to dimensions $i \times Q$ and L to $Q \times Q$.

As data is progressively added—row by row—the eigenvalues are recalculated with each new addition, reflecting the growing complexity of the data matrix. This step-by-step addition and analysis through PCA enable the identification of changes in the data's dimensionality, offering insights into the components present, their occurrences, and concentration profiles within the sample.

The broad applicability of EFA is evident from its use in various domains, including phase equilibrium studies in solution chemistry, chromatography, and mixture characterization. It facilitates the characterization of variance in data into principal components and iteratively monitors these components across experimental data.[54] A significant change in a principal component above a baseline is indicative of the characteristics, occurrences, and concentrations of components.

However, despite its versatility, EFA faces challenges, especially in systems with highly nonlinear compound interactions and data exhibiting heteroscedasticity—where variance is unequal across the data range. These limitations underscore the need for careful application and interpretation of EFA results.

2.1.2.2 PARAFAC decomposition

Introduced by Harshman and further developed by Carroll and Chang, Parallel Factor Analysis (PARAFAC) is a canonical decomposition technique that extends Principal Component Analysis (PCA) to tackle higher-dimensional data through a trilinear decomposition approach. This method breaks down a data matrix into its fundamental trilinear components, demonstrating a significant leap in handling complex datasets by mapping them onto a structure defined by:

$$x_{ijk} = \sum_{f=1}^{F} (a_{if} b_{jf} c_{kf} + e_{ijk}) \tag{2.13}$$

where e_{ijk} encapsulates the residual error. Utilizing the Alternating Least Squares (ALS) technique for iterative estimation of parameters, PARAFAC distinguishes itself by using residual, leverage, or triple cosine similarity analyses, albeit demanding precise input on the number of components.

The versatility of PARAFAC has been showcased across various applications in literature, from unravelling data variance and achieving unique decomposition in fluorescence data to modeling regression problems, tackling datasets with missing elements, and addressing constrained issues like variable non-negativity. Its generalizability, simplicity, and robust modeling capabilities stand out distinctly against alternative methods such as Partial Least Squares (PLS), and principal component regression. One of PARAFAC's hallmark advan-

tages lies in its solution's uniqueness, ensuring that even with minimal noise and the right component count, the derived solution is both accurate and immune to the initial parameter settings. [55]

Moreover, PARAFAC's ability to handle constrained problems and datasets with missing values further underscores its adaptability and potential as a soft sensing tool for calibration and analytical endeavours. Its capability to sidestep the rotation issue typically encountered in decompositions offers a simpler, more interpretable mode post-deconvolution, enhancing the model's robustness. Notably, the method's approach to scaling and centering data, as well as effectively modeling non-negativity constraints on variables, has been proven to not only preserve but also enhance the interpretability and predictive power of the model.

Despite its strengths, PARAFAC's computational demands, attributed to the high dimensionality of variables, pose a significant challenge, necessitating advancements in data compression techniques, optimization of ALS iterations, and refinement of convergence criteria to mitigate its intensive computational requirements.

2.1.2.3 Multivariate curve resolution

Multivariate Curve Resolution (MCR) is an important technique in chemometrics and analytical chemistry for deciphering complex datasets, especially in spectroscopic and chromatographic analyses. MCR aims to deconvolve a data matrix D, representing a mixture's spectral data, into matrices of pure component concentration profiles (C) and pure spectra (S), according to the bilinear model :

$$D = CS^T + E \tag{2.14}$$

Here, E captures the discrepancies between observed data and its approximation by the model, signifying the essence of MCR in distilling complex mixtures into their constituents. This decomposition is refined iteratively using the Multivariate Curve Resolution-Alternating Least Squares (MCR-ALS) algorithm, which integrates chemically significant constraints like non-negativity and unimodality, ensuring solutions are not just mathematically sound but chemically valid as well.[56]

MCR encounters challenges like accurately estimating mixture components, resolving potential rotational ambiguities, and managing data complexities such as signal overlaps. Nevertheless, its robustness against noise and capability to accommodate deviations from ideal bi-linearity through preprocessing methods and constraints highlight MCR's evolution to meet analytical demands. Its ability to handle non-idealities through multi-set analysis and constraint incorporation underscores MCR's invaluable role in analytical science, making it a key player in extracting meaningful insights from complex, multi-dimensional datasets.

A testament to MCR's utility is its application in the online monitoring of bitumen conversion using infrared spectroscopy, where it has been used to autonomously resolve spectra and track the concentration profile of changing species in complex mixtures[57, 58]. This example illustrates MCR's potential for real-time, efficient monitoring of chemical reactions in intricate mixtures, showcasing its significance in modern analytical methodologies.

2.1.2.4 JNTF factorization

The Joint Non-negative Tensor Factorization (JNTF) technique, an advancement in chemometrics, is employed to distill complex, multi-dimensional datasets into interpretable components while adhering to non-negativity constraints.[48] Specifically, the method's utility shines in the fusion of diverse data types, such as FTIR and ^1H NMR spectra, to unravel pseudo component spectra that encapsulate the intricate dynamics within reacting environments. Central to the JNTF methodology is the formulation of an objective function aimed at minimizing the reconstruction error:

$$\text{minimize} \quad ||\mathcal{X} - [\![\mathbf{W}; \mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_N]\!]||_F^2 + \lambda \cdot R(\mathbf{W}, \mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_N) \tag{2.15}$$

subject to $\mathbf{W}, \mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_N \geq 0$, where regularization plays an important role. This approach not only constrains solution ambiguity but also enhances the decomposition's fidelity to the physical reality, allowing for a direct interpretation in terms of concentrations without requiring a priori constraints.

JNTF workflow begins with FTIR mixture spectra where the multi-dimensional data in

tensor form is readied for analysis. Following this, in the Initialization step, factor matrices are initialized for each mode of the tensor. These matrices serve to encapsulate latent features that capture the underlying patterns within the data.

Moving on to iterative optimization, an objective function is defined to evaluate the correspondence between the original tensor and its reconstructed form based on the factor matrices. This function typically includes a reconstruction error term and regularization terms to enforce constraints on the factors. Iteratively, the factor matrices are updated to minimize this objective function, often using optimization techniques like gradient descent or alternating least squares.

Convergence is then monitored in the convergence step, where the algorithm checks if the changes in factor matrices between iterations have fallen below a predefined threshold. Once convergence is achieved, the algorithm proceeds to post-processing. Here, the factor matrices are normalized to ensure the non-negativity and interpretability of the factors. Additionally, optional analysis may be performed on the factor matrices to gain deeper insights into the latent features and underlying patterns captured by them. Thus resulting in pseudo-component concentration and spectral profiles.

The innovation of this study lies in its data-driven strategy for identifying pseudo-components and elucidating reaction networks, leveraging core consistency diagnostics and Bayesian structure learning. The latter infers causal relationships among pseudo-components, facilitating the hypothesis generation regarding reaction networks. An adjacency matrix, derived from the Bayesian networks, further informs the construction of kinetic models, reflecting the connectivity among various pseudo-components and underscoring potential conversion pathways not evident in isolated analyses.

This approach to tensor decomposition can deal with data artifacts such as missing observations and non-Gaussian noise. The efficacy of JNTF in species identification and the automated discovery of reaction mechanisms holds promise for revolutionizing automation and control within chemical analysis, making it a cornerstone technique in modern analytical science. Further, some papers discuss semi-supervised machine learning techniques to obtain auxiliary information to get started with JNTF decomposition or rather bypass it.[59]

JNTF offers unique advantages in decomposing complex data matrices into interpretable

components, providing insights into the underlying structure of multi-dimensional datasets and serving as reliable tools in chemometrics.

2.1.2.5 Transition to Application: Spectral Deconvolution

Moving forward, we transition from the theoretical foundations of these decomposition techniques to their practical implementation in spectral deconvolution. The JNTF decomposition as an initial step necessitates identifying the number of components involved in the reaction, which is done by running Lack of Fit (LOF) with core-consistency diagnostics.

Lack of fit refers to the measure of how well the factorization model captures the original data. It assesses the extent to which the reconstructed tensor closely resembles the input tensor. By comparing the original data tensor with the reconstructed tensor based on the factor matrices, one can quantify the discrepancy, which helps in assessing the goodness of fit and selecting an appropriate rank.

Core consistency check, on the other hand, is a diagnostic tool used to assess the consistency of the factorization results across different modes of the tensor.[60] It examines the core tensor, which represents the interactions between the latent factors along each mode, to ensure that it exhibits consistent patterns across different mode combinations. Inconsistencies in the core tensor can indicate that the chosen rank may not adequately capture the underlying structure of the data.

$$\text{Core Consistency} = 100 \left(1 - \frac{\|\mathbf{G} - \mathbf{T}\|_F^2}{\|\mathbf{T}\|_F^2}\right) \tag{2.16}$$

where, $\|\mathbf{G} - \mathbf{T}\|_F^2$ is the squared Frobenius norm of the difference between the estimated tensor and the target tensor, quantifying the error or inconsistency.

Combination of this methods determines the rank to be four as depicted in figure 2.5. This determination is based on a threshold whereby 98% of the variance in the data is captured. This information is then employed to initialize the Non-negative Tensor Factorization, which leverages Alternating Least Squares (ALS) to compute the best estimate of the rank CP model. This results in pseudo-component spectra and concentration profiles. The Lack of Fit and Core Consistency plots, presented here using a subset of FTIR data, are illustrated

in Figure 2.5. The concentration and spectral profiles, obtained after 1000 iterations, yield the results displayed in Figure 2.6, with a sum of squared errors (SSE) loss of 0.00048.

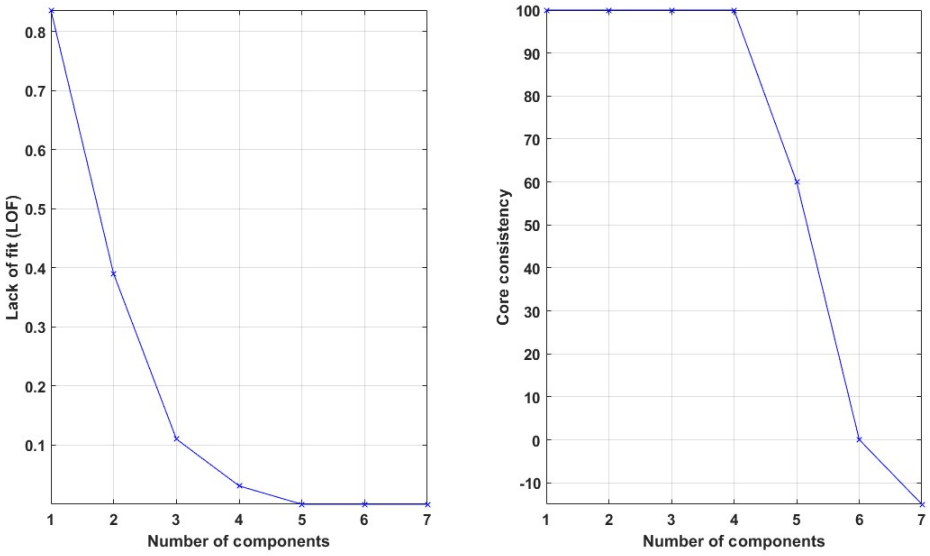

Figure 2.5: Chemical rank determination

The pseudo-component spectra are depicted in Figure 2.7. A side-by-side comparison of the profiles reveals that peaks, indicative of the presence of functional groups, appear within the same wavenumber range for each species.

When analyzing the pseudo-component spectra in conjunction with the pure component spectra of ethane, distinct spectral characteristics emerge. Specifically, a peak observed near $3000\,\mathrm{cm}^{-1}$ coupled with the absence of a peak around $2000\,\mathrm{cm}^{-1}$ strongly suggests the presence of an alkane functional group. This is attributed to the C-H stretching vibrations typical of alkanes.

Further scrutiny of the pseudo-component spectra, when compared with chloroethane, reveals spectral peaks between $500-1200\,\mathrm{cm}^{-1}$, confirming the presence of a halide functional group. This observation is supported by the identification of an alkane group, evidenced by a distinctive peak at $3000\,\mathrm{cm}^{-1}$, characteristic of C-H stretching vibrations.

In the context of acetic acid, the pseudo-component spectra exhibit peaks within the $1000 - 2000\,\mathrm{cm}^{-1}$ range, signifying the presence of C-O stretching and C=O stretching vibrations – hallmark features of carboxylic acids. A prominent peak at approximately

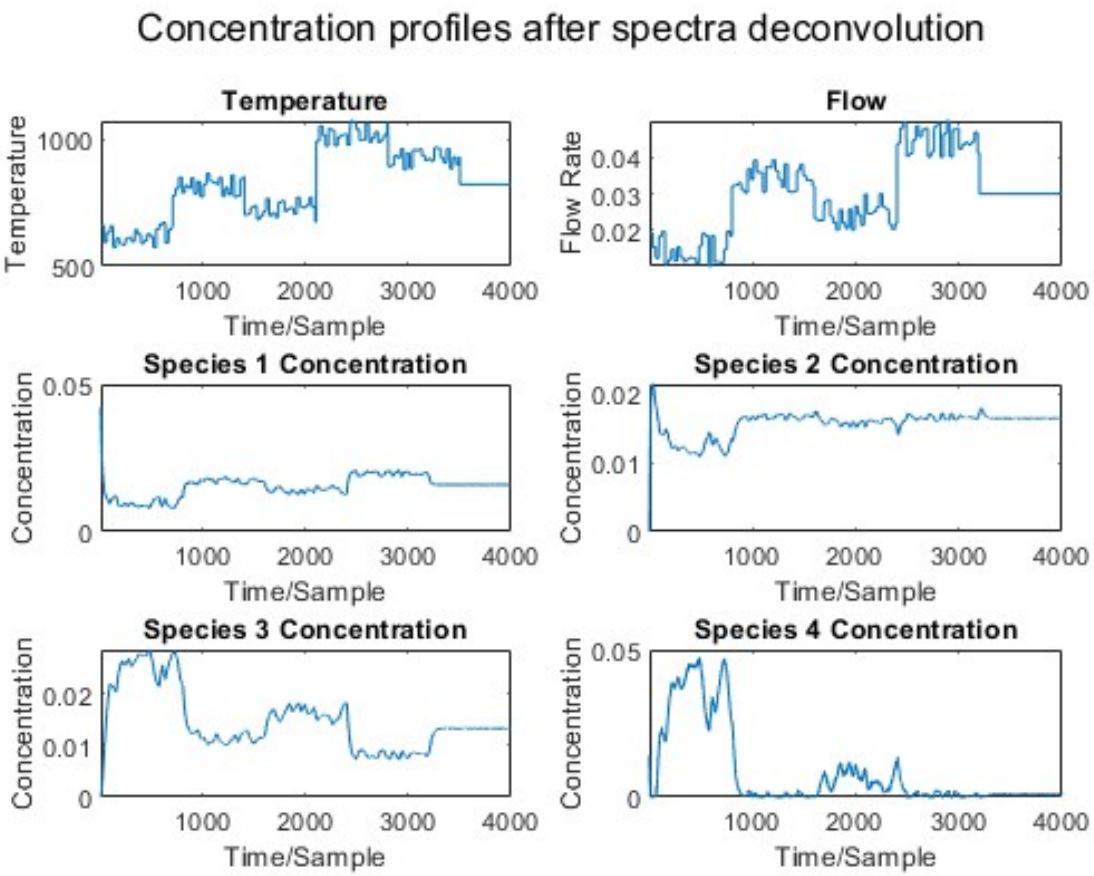

Figure 2.6: Concentration profiles after deconvolution

$3500\,\mathrm{cm}^{-1}$ further suggests the presence of an alcohol functional group, indicative of an O-H stretch. The concurrent presence of ketonic and alcoholic peaks infers the presence of carboxylic acid functional groups, underscoring the chemical complexity captured by the pseudo-component analysis.

Lastly, the pseudo-component spectra aligned with the spectra of ethanol highlight an alcohol functional group, evidenced by a peak at approximately $3800\,\mathrm{cm}^{-1}$ alongside C-O stretching vibrations discernible in the $750 - 1200\,\mathrm{cm}^{-1}$ region. However, there seems to be the presence of noise in wavenumber $460 - 660\,\mathrm{cm}^{-1}$, which is possibly because of the overlap between acetic acid and ethanol overlap during those regions. Further close comparison between spectral features of PC3 and PC4 indicates at presence of an alcohol functional group in the pseudo-component.

As visible in figure 2.8, the introduction of perturbations, in the form of temperature

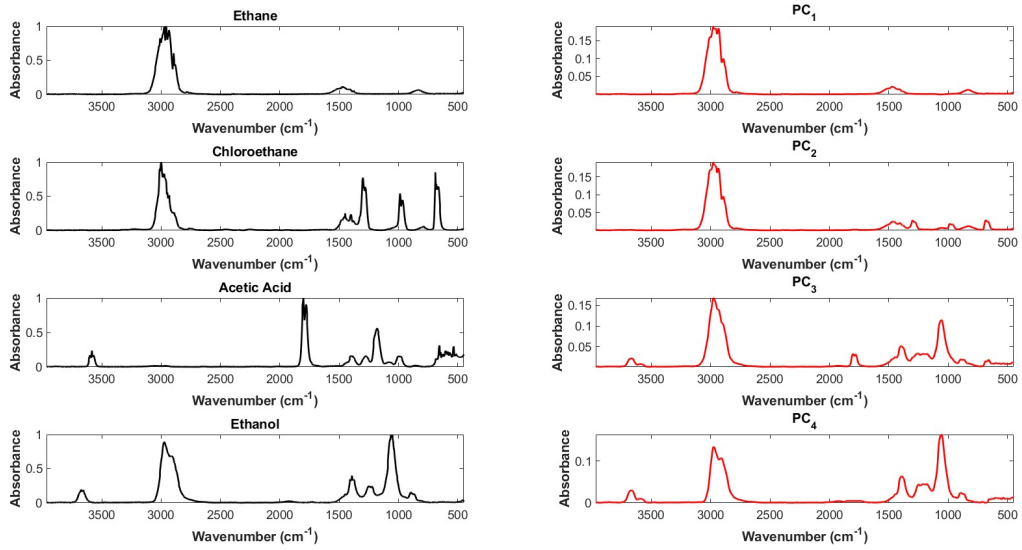

Figure 2.7: Comparison between pure and pseudo-component spectra

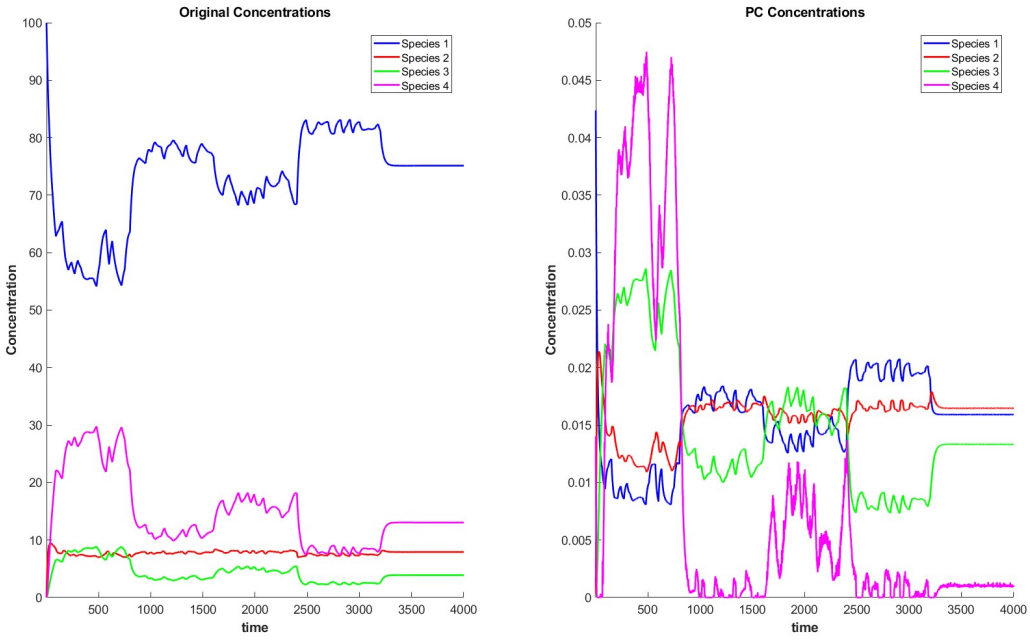

Figure 2.8: Comparison between original and PC concentration profiles

and flow variations, impacts the concentration profiles in a manner consistent with their effects on the ODE system. Consequently, the trends that capture the excitation from the input signals are captured, although the magnitude of the deconvolved concentrations,

which are rescaled, may not precisely mirror these dynamics. Specifically, Species 1 and 2 exhibit a noticeable shift and fail to attain the same magnitude observed in the original data. For Species 3 and 4, an overlap is evident as the simulations progress. Furthermore, within the deconvolved concentration profile of Species 4, it is observable that concentrations, which initially trended towards negative values at the start of the simulation period, are constrained to zero. This adjustment is due to the enforcement of non-negativity constraints in the factorization process, ensuring that the concentration values remain physically feasible throughout the analysis.

2.1.3 Reaction hypothesis generation from pseudo-component spectra using Bayesian network

Bayesian networks present a mathematical framework for encoding conditional dependencies between variables within complex systems. These probabilistic graphical models are particularly adept at modeling and reasoning about the uncertainty and causality inherent in diverse fields, ranging from systems biology to chemometrics.[61]

Bayesian networks encapsulate systems as sets of variables (nodes) and conditional dependencies (directed edges) within a Directed Acyclic Graph (DAG). Each node in the network corresponds to a random variable, which may represent a measurable property, an event, or a state. Directed edges, meanwhile, denote the causal or conditional relationships between these variables. The structure of Bayesian networks allows computation and representation of joint probabilities, facilitating inference and learning:

$$P(X_1, X_2, \ldots, X_N) = \prod_{i=1}^{N} P(X_i | Pa(X_i)) \tag{2.17}$$

where $Pa(X_i)$ denotes the parents of node X_i.

Applying Bayesian networks with FTIR spectra involves modeling the spectral features as variables within a network that reflect the dependencies among these features and the chemical constituents or reactions they indicate.

Several factors underpin the effectiveness of Bayesian networks in analyzing FTIR data. Handling of Uncertainty by Bayesian networks can model the uncertainty and incomplete

knowledge typical of complex chemical datasets. The integration of prior knowledge with these networks allows for the seamless incorporation of existing domain knowledge, enhancing interpretability and accuracy.

Constructing Bayesian networks from FTIR data entails structure learning and parameter estimation. Structure learning can be approached via:

1. Constraint-based methods, using statistical tests for conditional independence.

2. Score-based methods, employing criteria like the Bayesian Information Criterion (BIC) or Akaike Information Criterion (AIC) and search algorithms to evaluate network structures.

3. Hybrid methods, which combine the strengths of constraint-based and score-based approaches.

Parameter estimation often utilizes algorithms like expectation maximization (EM) to refine network parameters, maximizing the observed data's likelihood.

Bayesian networks are a tool for extracting causal relationships from FTIR spectroscopy data, and the causal relationships obtained can be interpreted as reactions.

$$AIC(M_k) = -2\log L(M_k) + 2k, \tag{2.18}$$

$$BIC(M_k) = -2\log L(M_k) + \log(n)k, \tag{2.19}$$

Equations 2.18 and 2.19 represent the formulas for AIC and BIC, which serve as metrics to evaluate the goodness of a fit of the statistical model. M_k represents the k^{th} model among a set of candidate models being evaluated. k denotes the number of parameters in the model M_k, reflecting the complexity of the model. A higher value of k indicates a more complex model with more parameters to estimate from the data. The term n stands for the sample size. For BIC, the term $\log(n)$ acts as a penalty factor that increases with the sample size, enforcing a more stringent penalty on the complexity of the model as the amount of data

increases. This penalty helps with guarding against the overfit tendency by laying penalties on very complex models.

To identify the best network structure for BIC maximization, commonly used algorithms include Hill Climbing (HC)[62], Tabu Search[63], and Maximum Minimum Hill Climbing (MMHC)[64]. HC incrementally finds local optima but may require random restarts to avoid settling for suboptimal solutions. Tabu Search improves upon HC by avoiding previous solutions, thus navigating more effectively towards better overall solutions. MMHC combines initial constraint-based pruning of the search area with subsequent HC optimization, creating a more directed and efficient search process. These algorithms are favoured for their ability to handle the complex search spaces found in Bayesian network structure learning, ensuring a balance between data fit and simplicity. By using these algorithms in tandem, the robustness of the resulting network structure is reinforced, particularly when multiple algorithms converge on the same solution. The pseudo-component spectra reported in figure 2.7 are input into the Bayesian Network algorithms and the probabilistic reaction networks are obtained with scores allocated to each fit. The importance of an arc, or directed edge, is determined by its impact on the overall score of the network. Specifically, the arc's strength is assessed based on the change in the network's score when the arc is removed: it is the disparity between the score with the arc absent and the score with the arc included. A negative number signifies a reduction in the network's score, while a positive number indicates an enhancement.

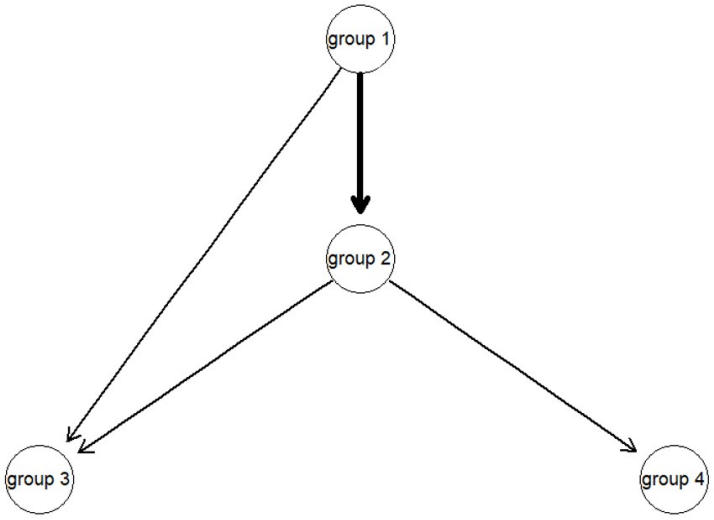

Figure 2.9: Reaction network for HC & MMHC

Table 2.2: Inter-group Strength Values for HC & MMHC

From	To	Strength
Group 1	Group 2	-9452.06
Group 2	Group 4	-8735.27
Group 2	Group 3	-8640.34
Group 1	Group 3	-30.83

HC, MMHC, and Tabu all identify the same reaction network structure. However, the Tabu method assigns a higher score to the arc between nodes 2 and 3 than to the arc between nodes 2 and 4. The arc from node 1 to node 3 has the lowest score in the network and can be considered less critical compared to the other arcs. Therefore, by omitting this least significant arc, we revert to the original reaction network depicted in Figure 2.2. It's worth mentioning that when analyzing spectral data with Bayesian networks, it is a common practice to apply preferential weighting to improve the resolution of specific wavenumbers, aiding in the recovery of the initial reaction framework. Additionally, some models are designed to establish direct connections between nodes, adjusting model parameters to fit the data and effectively capturing the underlying structure.

2.2 Data-driven modeling approach for complex reaction kinetics

The interplay of various species under different conditions lends complexity to chemical reactions, demanding advanced modeling techniques. Conventional methods, anchored in predetermined reaction kinetics and mechanisms, may not sufficiently unravel the intricacies of such systems, particularly when their fundamental processes are not thoroughly elucidated.

Kinetic modeling of reactions fundamentally involves four stages: understanding the feedstock composition, discerning the interaction among components within the reaction network, formulating rate equations and parameters, and incorporating continuity equations[65]. In the context of this study, the composition and reaction networks have been ascertained using established methods, and the continuity equations are addressed since the system is modeled in a Continuous Stirred Tank Reactor (CSTR).

When dealing with systems where only experimental data is available, empirical methods like curve fitting and utilization of auxiliary or qualitative information are typically employed to simplify parameter identification. While suitable for simpler systems, this approach hits a bottleneck as the complexity of the reaction network increases, necessitating a larger experimental dataset for an adequate kinetic model. The development of robust mechanistic models is not only time-consuming but also lacks convenience.

In contrast, data-driven models are increasingly favoured for their utility across various scientific fields. Notably, deep learning models, particularly those predicated on molecular simulations, have garnered attention. These models dissect feedstocks on a molecular level and track their behaviour during reactions[66]. Despite the reliability of robust predictive models enabled by machine learning advancements, they tend to compromise interpretability. With hundreds to thousands of parameters, there's a risk of overfitting and losing critical qualitative insights, which poses a challenge when applying the model to data outside the studied system's scope.

Hybrid models represent a fusion of physics and complex architectural modeling. These models, contrary to classic machine learning techniques, can be trained on limited data sets and do not require a fully physical process description. Capable of delivering accurate

predictions even beyond the measured data range, hybrid models overcome the constraints of data-driven methods. Physical models, which are simpler and dependent on a finite number of parameters, can be enhanced using neural networks and other machine learning strategies. In scenarios where a complete model is unattainable due to complexity, more advanced hybrid models are necessary. These approaches have been applied in various domains such as fault diagnosis, process modeling, optimization, and control.[67, 68, 69, 70, 71, 72, 73, 74]

There are several ways to meld physics with machine learning models. The development of advanced programming libraries enables the identification of process parameters using sparse regression for non-linear system dynamics[75]. While computationally efficient, these algorithms can falter with high-dimensional dynamic systems with ambiguous constraints. Deep neural network approaches like model identification, matching derivatives, or integrating residuals have been effective but often include a multitude of parameters. However, Physics-Informed Neural Networks (PINN) and Neural Ordinary Differential Equations (NODE) stand out in modeling reaction kinetics. PINNs and deep networks that match derivatives tend to depend on deep architectures loaded with parameters and may struggle to capture transient dynamics, limiting generalizability. Thus, NODEs, aligned with differential equations that describe chemical kinetics, are utilized in this study, underpinning a system-agnostic method that harnesses spectroscopic data for species identification and the hypothesizing of reaction pathways. These are further used to estimate kinetic models, constrained by the reaction network adjacency matrix informed by Bayesian learning and the fundamental principles of mass action and temperature dependencies.

2.2.1 Neural Ordinary Differential Equation

Neural Ordinary Differential Equations (ODEs) have emerged as a pioneering interface between the realms of deep learning and the theory of dynamical systems, facilitating the modeling of continuous-time phenomena across various scientific fields, as delineated by Chen et al. [76]

Central to neural ODEs is the conceptualization of a neural network's hidden layers as the trajectory of a state within a dynamical system governed by an ordinary differential equation. This paradigm shift introduces a continuous model where the evolution of the

system's state, $\mathbf{h}(t)$, is governed by:

$$\frac{d\mathbf{h}(t)}{dt} = f(\mathbf{h}(t), t, \theta), \tag{2.20}$$

where f denotes the neural network with parameters θ, and t serves as an analog to the network's depth.

This approach allows for the seamless modeling of state transitions, utilizing ODE solvers to transition from an initial state, $\mathbf{h}(0)$, to a final state, $\mathbf{h}(T)$, effectively encapsulating the model's output.

A noteworthy advantage of neural ODEs is their parsimonious approach to parameter utilization. Unlike traditional deep networks that necessitate distinct parameters for each layer, neural ODEs operate with a singular parameter set across its entirety. Dupont et al.[77] highlighted this efficiency, noting the model's capability to encapsulate complex dynamics within a compact framework.

Additionally, neural ODEs excel in processing time-series data that is non-uniformly sampled, offering the flexibility to assess the hidden state at any desired point in time, thereby enhancing their applicability to real-world data scenarios.

Optimizing the parameters of neural ODEs to align with empirical data involves leveraging the adjoint sensitivity method, enabling scalable and memory-efficient training processes. This innovative approach utilizes a differential equation solver as a "black box" to compute the output layer, eliminating the need to backpropagate and scale to problem complexity

Applications of neural ODEs span a variety of tasks, underscoring their versatility in modeling continuous-time series, implementing normalizing flows, and offering a continuous-depth alternative to Residual Networks (ResNets), which has been demonstrated by Rahman et al.[78] for nonlinear system identification.

2.2.1.1 Chemical Reaction Neural Network

CRNN serves as a useful tool for autonomous discovery of reaction models and has been demonstrated on multiple chemical and biochemical systems successfully. An updated model that accounts for uncertainties has been developed in the form of Bayesian CRNN. These

models have been applied to batch systems dealing with biomass feedstocks. The figure below shows the structure of the first-ever proposed CRNN that follows the law of mass action and temperature-dependent Arrhenius law to model the kinetics of the reaction.[79, 80] Figure 2.10 has been adapted from [81] and depicts a CRNN structured to model a batch reaction system that involves multiple series and parallel reactions.

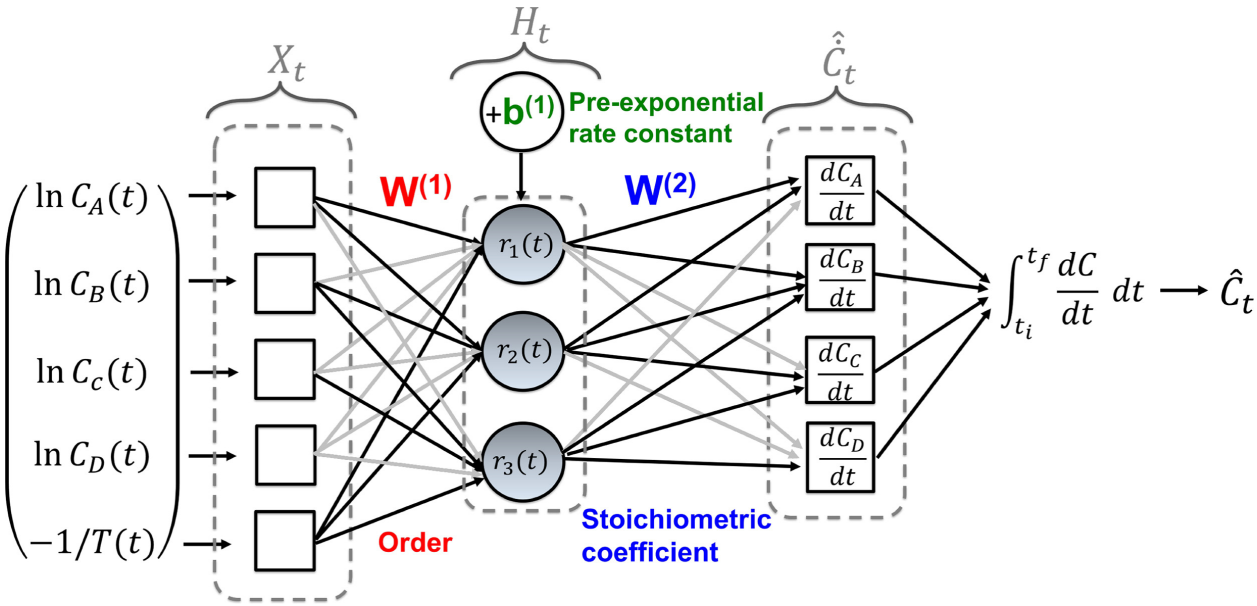

Figure 2.10: Neural ODE for chemical reactions. Adapted from [81]

When equations 2.1 and 2.2 are combined, the rate law can be expressed as:

$$r = k_0 \exp\left(-\frac{E_a}{RT}\right)[A]^a[B]^b \tag{2.21}$$

Using logarithms and exponentials on the equation results in equation 2.22 that governs the structure of CRNN. Additionally, it is constrained through an adjacency matrix obtained through a reaction template resulting from a Bayesian network.

$$r = \exp\left[\ln k_0 - \frac{E_a}{RT} + a \ln C_A + b \ln C_B\right] \tag{2.22}$$

The outputs of the CRNN are the derivatives that represent the dynamics of changing concentrations with time at each time instant, which are then integrated to obtain the concentrations. In addition to existing CRNN models, the one shown in figure 2.10 stands

out because of the adjacency constraints laid on the weights, which enhance the learning by the inclusion of additional information like reaction network, which improves learning by designing loss functions such that only the weights corresponding to specific components that are involved in that particular reaction are learned.

2.2.2 Neural ODE: Application to continuous system

CRNNs and neural ODEs have been successfully applied in kinetic modeling of batch systems. However, CRNN's application on continuous systems, such as a CSTR, has not been successful. CSTR is a complex dynamic system widely used in chemical engineering to model various chemical processes. The standard NODE approach faces several challenges when modeling CSTR due to its complexity for reasons involving:

1. **Complex Dynamics:** Combinations of dynamic behaviours such as high nonlinearity, multiple steady states, and transient responses, when combined, do not succeed in CRNN test case scenarios.

2. **Stiffness:** CSTR systems tend to be stiff, where the contained process evolves on very different timescales leading to NODE solvers struggling to handle stiff systems efficiently.

3. **Modeling Assumptions:** The dynamics of a CSTR reactor can depend on various factors such as reaction kinetics, heat and mass transfer, fluid flow, and mixing. CRNN models do not capture all these factors adequately.

Research has been conducted by Rahman et al. [78], Qian et al. [82], and Yang et al.[83] for modeling CSTR reactor dynamics by using neural ODEs. Rahman et al. [78] used a neural ODE architecture with a single hidden layer while experimenting with hidden layer size in the number of parameters, and utilizing over 10,000 data points to obtain concentration profiles. The results were compared with neural state space models and linear models. Although the neural ODE architecture provided accurate results, the number of parameters required for the system was high. Qian et al.[82] have proposed an autoencoder with NODE to reduce the order of NODE to deal with stiffness in the system, which still

lacks interpretability in the model and deals with a lot of parameters. Yang et al.[83] rely on the residence time distribution equation to model a generalizable neural ODE, which accounts for the changing dynamics of the system but relies on different methods of data generation accompanied with a black-box architecture that tries to match the derivatives accompanied by an adjoint solver to obtain the dynamic profiles.

This study, however, emphasizes interpretability as one of the key aspects and hence tries to incorporate that into the architecture of the model. Equation 2.22, which applies to a batch system, needs to account for the flow flux in a continuous setup, which is obtained by the introduction of flow term to CRNN.

$$\frac{dC}{dt} = \frac{F(t)}{V} \cdot (C_0 - C) + k_0 \exp\left(-\frac{E_a}{RT(t)}\right) C^n \tag{2.23}$$

$$\frac{dC}{dt} - \frac{F(t)}{V} \cdot (C_0 - C) = k_0 \exp\left(-\frac{E_a}{RT(t)}\right) C^n \tag{2.24}$$

Taking logarithm of both sides:

$$\ln\left(\frac{dC}{dt} - \frac{F(t)}{V} \cdot (C_0 - C)\right) = \ln(k_0) - \frac{E_a}{RT(t)} + n\ln(C) \tag{2.25}$$

Building upon the neural ODE structure previously described, we introduce a refinement to incorporate flow dynamics into the model. This enhancement is delineated through the following modified set of equations incorporating the flow term $F(t)$ in the system's temporal evolution:

$$\left(\frac{dC}{dt} - \frac{F(t)}{V}(C_0 - C)\right) = \exp\left(\ln(k_0) - \frac{E_a}{RT(t)} + n\ln(C)\right) \cdot \frac{e^{WF}}{e^{WF}} \tag{2.26}$$

To facilitate the inclusion of the flow term within the NODE framework, the differential equation is re-expressed as follows:

$$\frac{dC'}{dt} = \exp\left(\ln(k_0) - \frac{E_a}{RT(t)} + n\ln(C) + WF\right) \tag{2.27}$$

Equation (2.27) is an augmentation to the canonical rate of change represented by the CRNN, which is depicted in Figure 2.10. Here, the necessity for an additional input — the flow value $F(t)$ — becomes evident, permitting the learning of an ancillary weight that accommodates alterations in flow. The predicted output from the CRNN is the modified rate term, as shown in equation (2.27). Subsequent to this, an additional transformation, shown in equation (2.28), is imperative to derive the actual rate terms.

$$\frac{dC}{dt} = \frac{dC'}{dt}e^{-WF} + \frac{F}{V}(C_0 - C) \tag{2.28}$$

This incorporation results in an expanded architecture, as depicted in Figure 2.11. The adjusted NODE architecture encompasses three distinct layers:

Table 2.3: Dimensions of the neural network layers and parameters.

Layer	**Dimension**
1st layer	$N_s + 2$
2nd layer	N_r
3rd layer	N_s
W_1	$(N_s + 2) \times N_r$
b_1	N_r
W_2	$N_r \times N_s$

where N_s represents the number of species in the system and N_r is the number of reactions.

The first layer receives inputs represented by the equation 2.29:

$$X_t = [\ln(C_a(t)), \ln(C_b(t)), \ln(C_c(t)), \ln(C_d(t)), -1/T(t), F(t)]^T \tag{2.29}$$

The input layer interacts with the first layer's weights of the NODE, which are constrained by an adjacency matrix. This matrix ensures that only the species participating in a specific reaction have their weights adjusted during the training process. As information propagates to the next layer, the reaction terms are learned. These terms then interact with the second set of weights. The adjacency matrix, derived from a Bayesian network, guides the combinations of reactions. This, in turn, influences the computation of the rate of change

$\frac{dC}{dt}$ for each species. The weights corresponding to this study are:

$$W^{(1)} = \begin{bmatrix} a & 0 & 0 \\ 0 & b & c \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -\frac{E_{a_1}}{R} & -\frac{E_{a_2}}{R} & -\frac{E_{a_3}}{R} \\ W_1 & W_2 & W_3 \end{bmatrix} \qquad W^{(2)} = \begin{bmatrix} -e & f & 0 & 0 \\ 0 & -g & h & 0 \\ 0 & -f & 0 & i \end{bmatrix}$$

The forward pass through this NODE is:

$$H_t = \exp\left(\left[W^{(1)} * \left(1_{(\text{Adj}\neq -1)}\right)^{N_R \times 1}\right] X_t + b^{(1)}\right) \tag{2.30}$$

$$\frac{dC'}{dt} = \left(W^{(2)} * \left(1_{(\text{Adj}\neq 0)}\right)^T\right) H_t \tag{2.31}$$

$$\frac{dC}{dt} = \frac{dC'}{dt} e^{-wF} + \frac{F}{V}(C_0 - C) \tag{2.32}$$

$$\hat{C}_{t+n} = \hat{C}_t + \int_t^{t+n} \frac{dC}{dt}\, dt \tag{2.33}$$

The loss function comprises of 3 terms:

$$L_{conc}(\theta) = \sum_{t=1}^{N} \left(C_{t+n} - \hat{C}_{t+n}(\theta)\right)^2 \tag{2.34}$$

$$L_{rate}(\theta) = \left(\left.\frac{dC}{dt}\right|_t - \left.\frac{d\hat{C}}{dt}\right|_t (\theta)\right)^2 \tag{2.35}$$

$$L_{weights}(\theta) = \alpha \sum_{i,j} \left|W_{ij}(\theta) - \text{Adj}_{ij}\right| \tag{2.36}$$

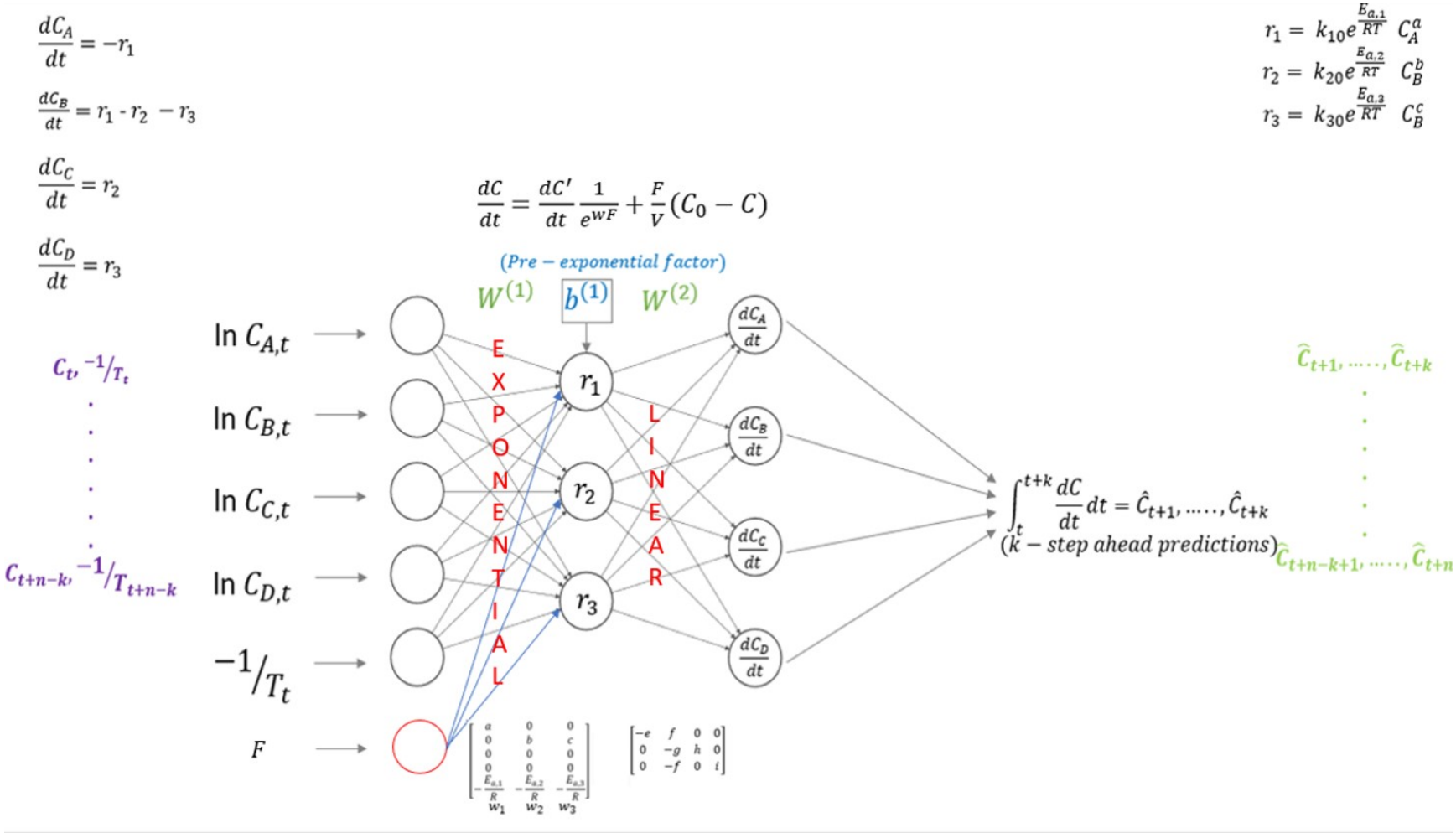

Figure 2.11: Structure abiding neural ODE for a CSTR

$$L_{total}(\theta) = \lambda_1 L_{conc}(\theta) + \lambda_2 L_{rate}(\theta) + \lambda_3 L_{weights}(\theta) \tag{2.37}$$

The CRNN, in the context of this study, is modeled as a step-ahead predictor, and the loss function depicted in equation 2.34 accounts for 3-step ahead values in loss functions. Given the relative nature of the concentration profiles derived from spectral deconvolution, it is inappropriate to deem them absolute. Consequently, the species from C_a to C_d are henceforth referred as pseudo-components, labeled PC-1 through PC-4, respectively.

While testing the scenarios, one-step-ahead predictions serve as input for the next time step, which is a common practice in time series forecasting models; the training information has been presented in table 2.4.

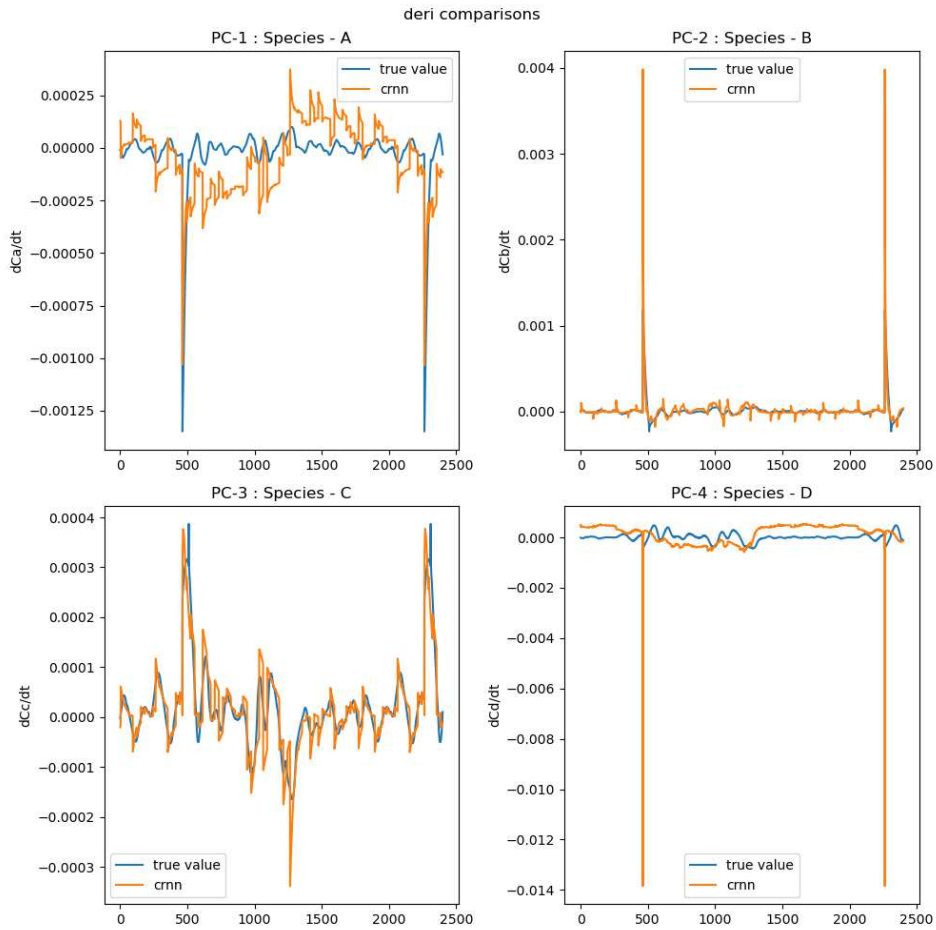

Figure 2.12: CRNN predictions of time derivatives of concentrations

In figure 2.12 the CRNN proficiently captures the dynamics of PC-2 and PC-3, as evi-

denced by the mean absolute errors of 2.79×10^{-5} and 2.42×10^{-5}, respectively, observable in their concentration profiles in Figure 2.13. In contrast, PC-1 exhibits more pronounced fluctuations, with a mean absolute error of 1.21×10^{-4}; nevertheless, the CRNN manages these variations with reasonable accuracy. However, for PC-4, several misalignments are evident, resulting in a mean absolute error of 3.70×10^{-4}, thereby slightly compromising the accuracy of the step-ahead concentration profiles for this species.

Table 2.4: Neural ODE Training details

Parameter	**Value**
Learning Rate (lr)	1e-3 to 1e-7
Optimizer	AdamW
Train Loss-MSE	4.27e-5
Validation Loss-MSE	1.25e-5
Test Loss-MSE	4.3e-5
Test Accuracy	93.52%
Trainable parameters	15
Epochs	1000
Solver	dopri8
atol & rtol	1e-8 & 1e-8

It can be noted that with the number of parameters needed for this model is much smaller than in any deep learning model; the CRNN demonstrates its capability as a reliable tool for complex reaction systems given the stiffness in the equations[84]. The plots comparing CRNN output and numerically differentiated $\frac{dC}{dt}$ are shown below in figure 2.12. The tolerance used by adjoint solvers is low because of the stiffness present in the dynamics, which helps avoid running into underflow and overflow issues. The presented architecture also relies on a non-negativity constraint on concentration values that are fed to the CRNN to avoid numerical issues because of the logarithmic function. At such low tolerance, training is computation intensive and time-consuming but yields good results on unseen data, as shown in figure 2.13 Given the temperature and flow perturbations present in figure 2.13, the predictions for

species 1, species 2 and species 3 are reasonably accurate for 1,5&10 step ahead predictions, while the predictions for species 4 are less accurate.

The prediction errors in initial step-ahead forecasts, when significant, begin to compound as successive step-ahead predictions are approached. Specifically, the observed offset in PC-4, in addition to discrepancies in time derivatives, can be attributed to the deconvolution process. In this process, non-negativity constraints occasionally force concentrations to zero at various time points. While the CRNN endeavors to align with these derivatives and project further step-ahead forecasts, the imposition of adjacency constraints aims to steer the learning process. This regulatory influence, however, can induce a divergence between the actual and predicted concentrations for PC-4, thereby contributing to the prediction mismatch.

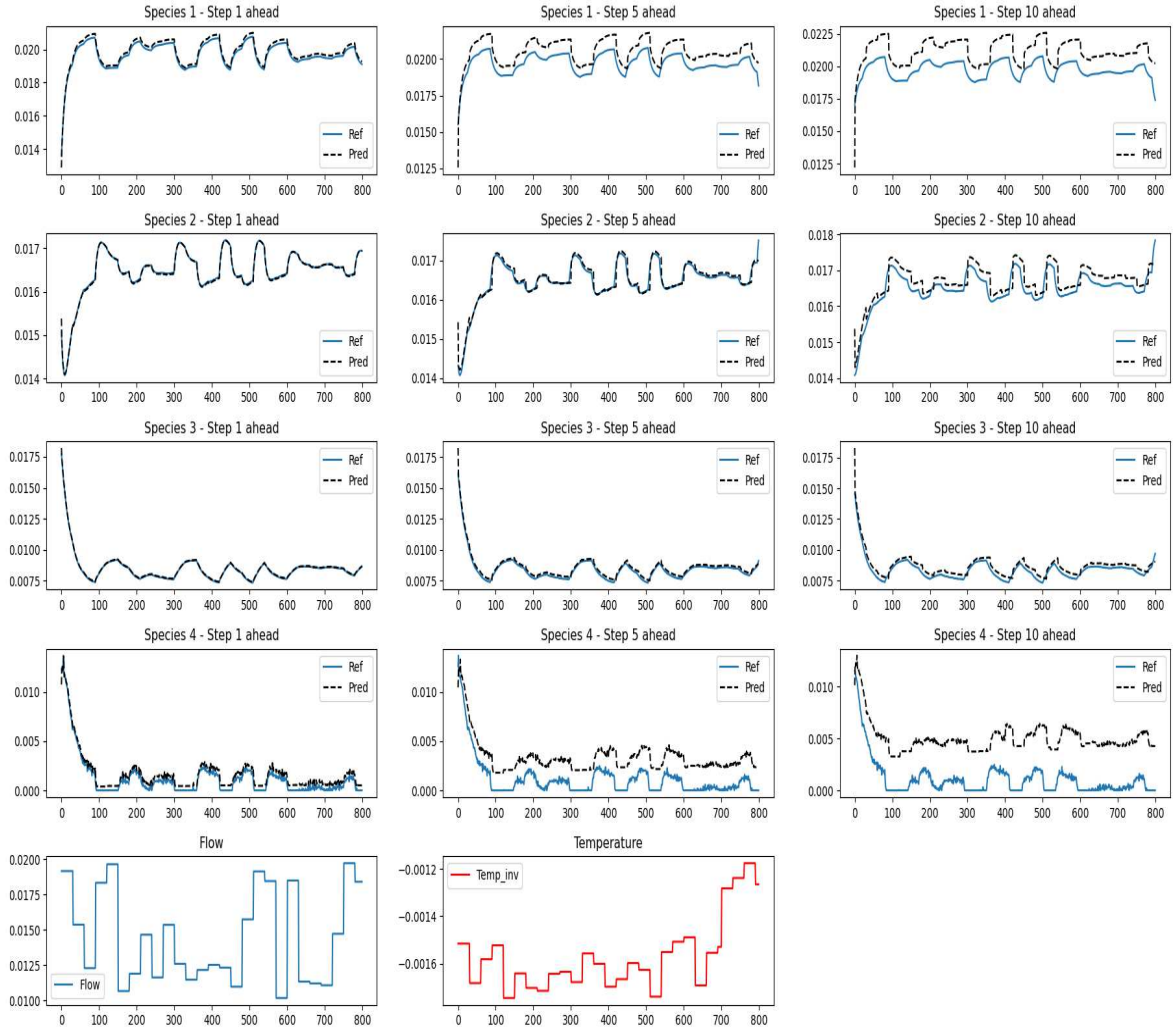

Figure 2.13: CRNN : 1, 5 &10 step ahead predictions

2.2.3 Neural ODE: Comparison to black-box approaches

The previous section discussed the neural ODE and the structure modification that adapts it to the CSTR system, still making sure that the governing equation drove the architecture. The reaction parameters embedded in the form of weights and biases provided reliable multistep-ahead prediction when tested on unseen data in an open loop. This section focuses on developing a black-box deep learning model to provide multistep-ahead predictions to compare the performance of the model.

As the data is nonlinear, choices for models include Artificial Neural Networks (ANN),

Recurrent Neural Networks (RNN), and Long Short Term Memory (LSTM). ANN has been used in a lot of applications as nonlinear autoregressive exogenous model (NARX) models; however, RNNs allow for model building on entire data sequences. The same weights are used for each time step of the input. This form of parameter sharing makes RNNs much deeper models in time without increasing the number of parameters linearly with the size of the time dimension. ANNs require fixed-length inputs, while RNNs can deal with varying inputs. RNNs suffer from gradient vanishing or gradient explosion problems which can be taken care of by advanced architectures, e.g., LSTMs retain information for longer periods which has been a limitation of RNN. The architecture involves different gates that decide the data that needs to be stored for the long term and discarded.

2.2.4 Long-Short Term Memory

Olah [85] describes the architecture and forward pass mechanics of LSTM networks. An LSTM manages memory through three distinct gates: the forget gate f_t, which evaluates portions of the cell state C_{t-1} for retention or removal; the input gate i_t, which identifies new, relevant information to be updated; and the output gate o_t, which influences the generation of the output hidden state h_t. The cell state is updated according to

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t, \tag{2.38}$$

while the hidden state is updated by

$$h_t = o_t \odot \tanh(C_t). \tag{2.39}$$

Figure 2.14 illustrates the roles of gates and activation functions within an LSTM unit, with the cell state indicated by a dashed line. Beginning with the prior hidden state $h_{(t-1)}$ and the current input x_t, the LSTM's forget gate determines which elements of the cell state are obsolete and should be eliminated. Simultaneously, the input gate selects fresh information to be incorporated into the cell state. These determinations from both gates guide the evolution of the previous cell state C_{t-1} into the new cell state C_t. Ultimately, the output gate decides the new hidden state h_t based on the freshly updated cell state C_t.

This gating mechanism enables LSTMs to capture and leverage long-term dependencies in sequential data.

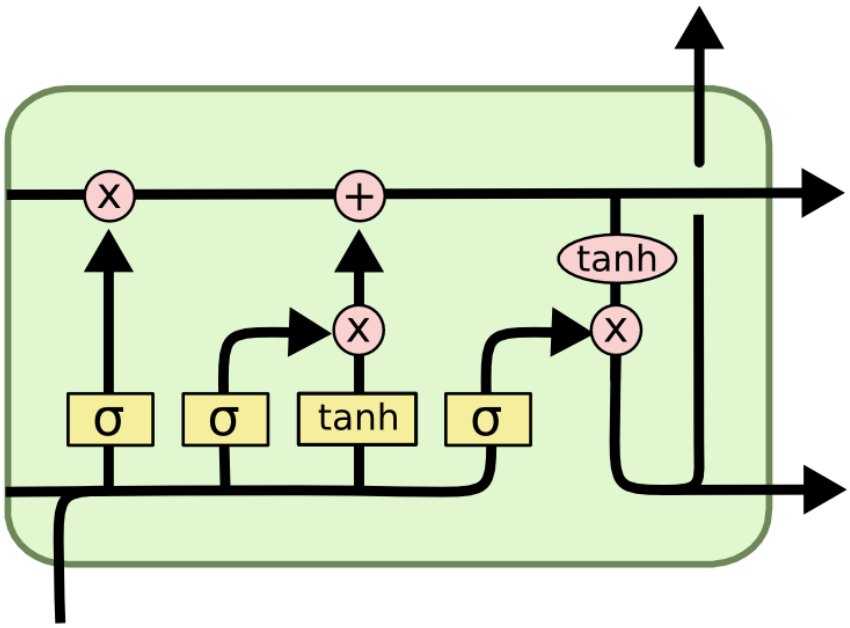

Figure 2.14: LSTM architecture. Adapted from [85]

Jung et al.[86] have presented the application of LSTM as a simulator for the ODE system that represents the reactor dynamics.

2.2.4.1 Application of LSTM for reaction kinetics modeling

The specifics of LSTM model training are delineated in table 2.5, which underscores the configuration and outcomes of the training process. LSTM has been trained on the same data that has been used to train neural ODE. The LSTM architecture was configured with 57,760 trainable parameters, and the model utilized a history of 30 past time steps to achieve the predictive performance shown in figure 2.15.

Table 2.5: LSTM training details

Parameter	**Value**
Input size	(30,6)
Learning Rate (lr)	2e-5
Optimizer	Adam
Train Loss-MSE	7.8e-7
Validation Loss-MSE	2.3e-6
Test Loss-MSE	9.2e-7
Test Accuracy	97.2%
Trainable parameters	57,760
Epochs	1500

In the evaluation over a five-step-ahead forecast horizon, PC-1 and PC-3 exhibit close alignment with empirical trends due to temperature and flow perturbations. However, slight deviations are observed towards the end of the simulation. On the other hand, PC-2 and PC-4 have low test MSE rates of 9.2×10^{-7}, but they fail to capture dynamic trends and do not respond to perturbations, resulting in flat predictive trajectories.

The CRNN demonstrates superior architectural efficiency, with a mere 15 trainable parameters compared to the LSTM's 57,760 parameters. This contrast not only underscores the CRNN's potential for enhanced generalization but also mitigates the risk of overfitting while facilitating physical laws into the modeling framework. In terms of loss metrics, the CRNN demonstrates remarkable consistency from training through to testing phases. Unlike the LSTM, which, despite its higher accuracy of 97.2%, tends to produce flat predictive trajectories that fail to respond adequately to perturbations in temperature and flow, particularly for pseudo-components PC-1 and PC-3. Although the CRNN requires substantial computational resources due to the use of an adjoint solver, its capability to utilize input from just one-time step for making predictions up to ten steps ahead significantly enhances its predictive capabilities. Given these attributes, the CRNN is affirmed as the model of choice for this study, ensuring reliable and dynamically sensitive performance.

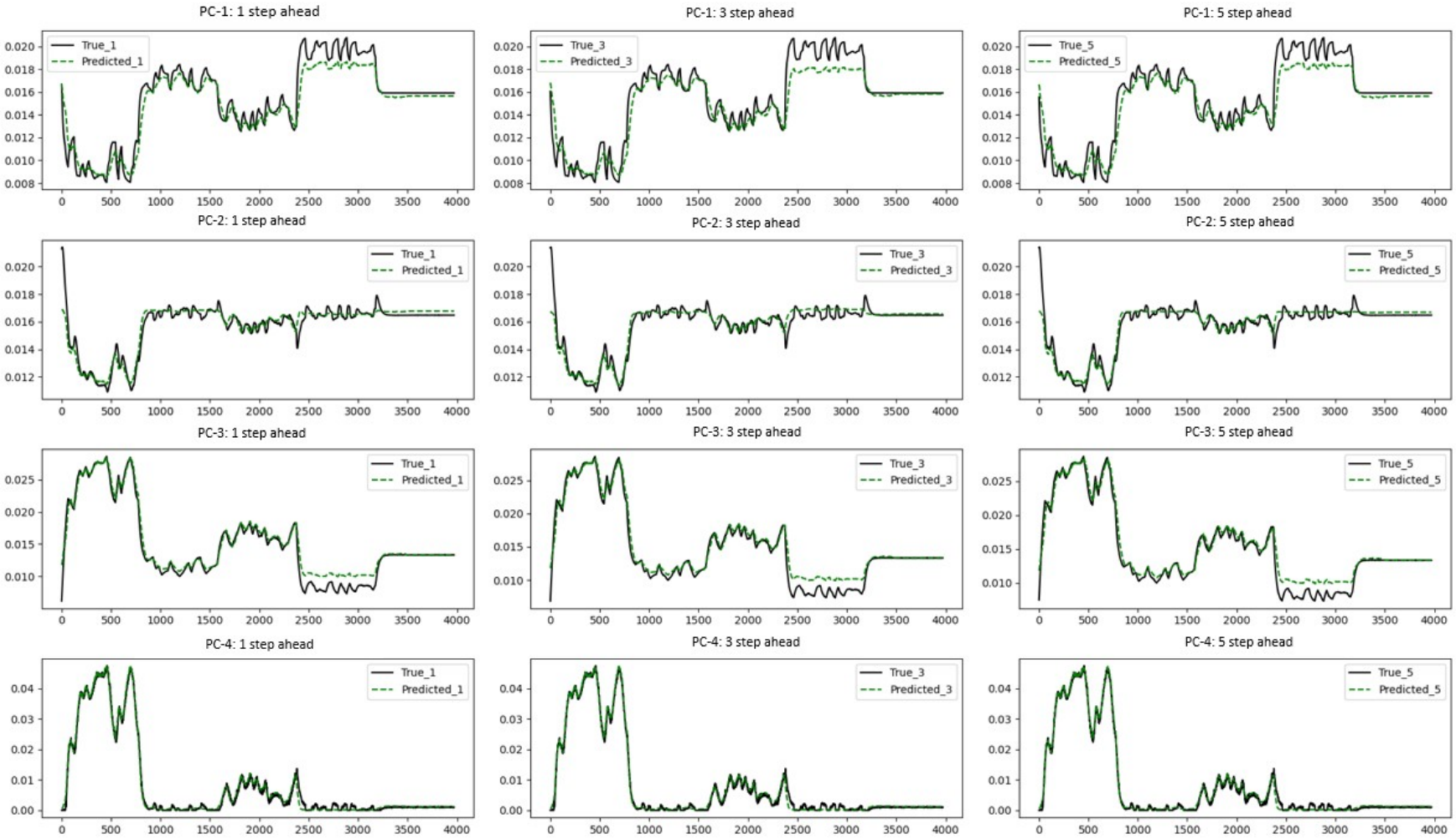

Figure 2.15: LSTM: 1,3 &5 step-ahead predictions

Chapter 3

Control of complex reaction systems

At its core, control is about influencing the behavior of systems to achieve desired outcomes. The necessity for control arises from the inherent unpredictability and complexity of physical systems, coupled with the need for efficiency, precision, and responsiveness.

Traditional control techniques such as Proportional-Integral-Derivative (PID) controllers, have been the backbone of industrial control systems due to their simplicity and effectiveness in a wide range of applications. However, as systems become more complex and interconnected, the limitations of traditional methods become apparent. Processes on an industrial scale are highly complex in nature. Their models contain many equations with coupled interactions between variables, which require decoupling to apply PIDs to control individual variables. Apart from that, the difficulties in handling nonlinearities, constraints, and multi-variable systems, as well as a lack of adaptability to changing conditions or system dynamics, is often a problem.

Advanced control strategies, including Model Predictive Control (MPC) and Reinforcement Learning (RL), offer several advantages over traditional methods. These include enhanced adaptability, the ability to handle multi-variable systems and constraints, and the potential for real-time optimization. By integrating predictive models and learning mechanisms, these strategies provide a more nuanced and responsive approach to control. Unlike traditional methods that react to deviations from a set point, optimal control strategies proactively anticipate future system states and make decisions that optimize a defined performance criterion over time. This forward-looking approach allows for considering constraints

and multiple objectives.

With its model-based predictive capabilities, MPC calculates optimal control actions over a prediction horizon in the presence of constraints. RL adapts and improves control policies based on interaction with the environment by balancing exploration and exploitation.

This chapter explores these advanced control strategies, tracing their development from theoretical foundations to practical applications. This exploration aims to demonstrate the potential of MPC and RL in what is achievable with control systems when focusing on complex reaction systems as the operational environment.

3.1 Model Predictive Control

Optimal control focuses on deriving strategies that optimize a performance metric over time. Control laws are formulated to minimize or maximize a cost function, accounting for both system dynamics and constraints. It serves as a foundational pillar for crafting methodologies that ensure optimal performance under predefined constraints.

Model Predictive Control (MPC) represents a subset of optimal control devised to overcome some of the practical hurdles encountered with traditional methods like dynamic programming (DP). While DP theoretically addresses the full scope of optimal control by solving the Bellman equation, it often proves impractical for systems characterized by complex dynamics or large state spaces due to its intensive computational demands[87]. In response, MPC adopts a receding horizon approach, continuously refreshing its control strategy based on real-time data and forward-looking predictions.

MPC is distinguished by its method of addressing control. It solves a finite horizon optimal control problem at each discrete time step, starting from the current system state and projecting forward using a model to simulate future states. After calculating the optimal control sequence for this horizon, only the immediate first control action is executed before the cycle repeats in the next time step. This rolling optimization technique, coupled with model updates, enables MPC to adapt to dynamic changes and disturbances, ensuring consistent system performance.

MPC offers a practical implementation of dynamic programming by sequentially solving

manageable, short-term optimal control problems, thus sidestepping the extensive computational requirements associated with traditional DP. This approach enables the application of optimal control principles in complex, real-world scenarios, establishing MPC as a dynamic and adaptable control tool in challenging environments.

MPC operation involves predicting future system outputs and optimizing control actions based on a model of the system. The operational procedure for MPC is presented below:

Algorithm 1 MPC Algorithm

1: Initialize system model and MPC parameters
2: Set simulation time period T and convergence threshold ϵ
3: Define the cost function J and horizons (prediction N and control M)
4: Initialize state x_0 and control input u_0
5: **while** not converged or $t < T$ **do**
6: Predict future states over the prediction horizon:

$$\hat{x}_{t+k|t} = f(\hat{x}_{t+k-1|t}, u_{t+k-1}), \quad k = 1, \ldots, N$$

7: Formulate the optimization problem to minimize the cost function:

$$\min_{\mathbf{U}_t} J = \sum_{k=0}^{N-1} \left(\left\| \hat{x}_{t+k|t} - x_{\text{ref}} \right\|_Q^2 + \sum_{j=0}^{\min(k,M-1)} \|u_{t+j}\|_R^2 \right)$$

8: subject to the state and input constraints for all k:

$$x_{\min} \leq \hat{x}_{t+k|t} \leq x_{\max}, \quad u_{\min} \leq u_{t+j} \leq u_{\max}, \quad j = 0, \ldots, M-1$$

9: Apply the first control action u_t from the optimized sequence $\mathbf{U}_t$:

$$u_t = \mathbf{U}_t[0]$$

10: Measure new state x_{t+1} and update the model with new measurement
11: Check for convergence or timeout
12: **if** the optimization problem is infeasible **then**
13: Break the loop or apply contingency measures
14: **end if**
15: $t \leftarrow t + 1$
16: **end while**

The foundational principle of MPC is consistent despite variations in its formulation, and the selection of a particular formulation will dictate adjustments to both the convergence criteria and the cost function.

3.2 Reinforcement Learning

Reinforcement Learning (RL) distinguishes itself through its reliance on interaction with a dynamic environment rather than the fixed datasets typical of supervised learning. This method enables RL algorithms to generalize behaviors and decision-making strategies to novel scenarios absent from the training data without direct supervision.[88] Unlike supervised learning, which optimizes models based on predefined labels, RL employs a trial-and-error approach, systematically exploring the state space of the environment. The learning process in RL is primarily driven by a mechanism of rewards and penalties: actions that bring the system closer to a desired outcome generate rewards, whereas less optimal actions incur penalties.[88] This framework incentivizes the algorithm to maximize cumulative rewards, thereby aligning its learned policies with the objectives defined by the reward structure. Ultimately, this approach enables the formulation of sophisticated strategies that promote optimal behaviors, operationalized through policy or value-based functions that map observed states to actions aimed at achieving specific goals.

Markov Decision Processes (MDPs) provide a formal framework for modeling the decision-making process in RL where outcomes are partially random(exploration) and partially under the control(exploitation) of a decision-maker.[89] The key elements of MDP include:

- **States (S)**: A comprehensive set of all possible situations the agent might encounter.
- **Actions (A)**: For each state, there is a set of actions available to the agent.
- **Transition Function (P)**: It defines the probability of transitioning from one state to another, given an action. It is denoted by $P(s' \mid s, a)$, indicating the probability of moving to state s' from state s under action a.
- **Reward Function (R)**: This function returns the immediate reward received after transitioning from one state to another via an action, denoted as $R(s, a, s')$.
- **Discount Factor (γ)**: A parameter that values the importance of immediate rewards versus future rewards, typically within the range $[0, 1]$.

In MDP-based RL, the goal is to find an optimal policy π^* that maximizes expected rewards. This is achieved using iterative algorithms that leverage the Bellman equation to recursively estimate state values and refine the policy[90].

$$V^{\pi}(s) = \sum_{a \in A} \pi(a \mid s) \sum_{s',r} P(s', r \mid s, a)[R(s, a, s') + \gamma V^{\pi}(s')] \tag{3.1}$$

Techniques such as Monte Carlo (MC) methods and Temporal-Difference (TD) learning are used to learn optimal policies within the MDP framework.[91] DP in RL utilizes a complete model of the environment to solve MDPs. By iteratively applying the Bellman equations, DP methods systematically evaluate and improve policies.

Monte Carlo methods in RL exploit the randomness inherent in the sampling processes to estimate the value functions and subsequently derive policies. Unlike methods that require a complete model of the environment, Monte Carlo methods operate by learning directly from episodes of experience—sequences of states, actions, and rewards. MC methods wait until the end of an episode and use the total accumulated return to update the value function estimates for the states visited. This approach, known as episode-by-episode learning, does not bootstrap (update estimates based on other estimates) but rather relies solely on empirical returns. MC methods do not require knowledge of transition probabilities and reward functions, making them ideal for environments where this information is unavailable or impractical to obtain[92]. Under conditions of constant policy and sufficient exploration, MC estimates converge to the true value functions as the number of episodes increases. The return paths can vary significantly, leading to high variance in the estimates, which may slow down the convergence.[88] MC methods require the completion of episodes, making them unsuitable for continuing tasks without clear terminal states.

TD learning, on the other hand, represents a class of model-free algorithms that learn by bootstrapping—updating estimates based on other learned estimates. TD methods combine the sampling techniques of Monte Carlo with the bootstrapping techniques of Dynamic Programming.[88] A quintessential example of TD learning is the TD(0) algorithm, where the value of the current state is updated based on the estimated value of the next state and the reward received, adjusting estimates partly towards the more certain subsequent

estimates. Unlike Monte Carlo methods, TD can learn from incomplete sequences, making updates after each step. This allows TD to be applicable in both episodic and continuous tasks. TD methods typically exhibit lower variance in updates than Monte Carlo, providing more stable learning progress. Bootstrapping methods introduce bias into the estimates, especially when the initial estimates are poor.[93] The quality of TD estimates depends significantly on the policy being followed, particularly for on-policy methods like SARSA. As both Monte Carlo and TD learning methods are tailored to learn optimal policies within the RL framework, their application can be optimized by understanding their inherent characteristics and situational advantages. These methods provide powerful tools for agents to learn from interaction with complex environments, enabling the formulation of sophisticated strategies for decision-making under uncertainty.

3.2.1 RL classification: model-based and model free

The classification of Reinforcement Learning (RL) algorithms broadly falls into two categories: model-based and model-free approaches. A crucial decision in designing RL algorithms is whether to incorporate a model of the environment. Model-based methods involve either learning or having access to a model that can predict state transitions and rewards, enabling agents to plan actions by thinking ahead.[94] This approach can greatly improve sample efficiency, as demonstrated by algorithms like AlphaZero.[95] However, the challenge with these methods is the potential bias in the learned models, which might result in less optimal performance in real-world scenarios.

Model-based RL requires the agent to build a model of the environment from its interactions. This model accurately predicts the outcomes of actions in specific states, allowing for a more strategic approach to decision-making. In this scenario, the agent learns not only to assess immediate rewards but also to use the model to anticipate future states and rewards, planning multiple steps ahead. Such a capability is especially beneficial in complex environments where long-term strategic planning is essential.

In contrast, model-free RL does not assume any knowledge of the environment's dynamics. Instead, it focuses on learning the value of actions directly from experiences without attempting to build an underlying model of the environment. This approach is split further

into two main methods: Q-learning and policy gradient methods. Model-free RL is generally simpler and more versatile, making it suitable for a variety of applications where the environment is too complex, or the agent's interaction with the environment is limited.

Conventionally, model-free RL is preferred over model-based methods, driven by several practical considerations. Model-free RL simplifies development by eliminating the need to construct a model of the environment, thus reducing complexity. It is also more robust, as it learns directly from actual experiences and is not affected by inaccuracies in a model's predictions. This approach offers greater flexibility and adaptability, making it suitable for a wide range of environments that are complex or unpredictable. Additionally, model-free RL typically requires less computational power since it focuses solely on learning from interactions rather than on modeling and planning.[96] It has proven to be effective across various applications, demonstrating its capabilities without the need for detailed environmental models. Modern enhancements like experience replay have also improved the sample efficiency of model-free methods, further boosting their attractiveness. Overall, the simplicity, robustness, and flexibility of model-free RL make it a preferred choice in dynamic or uncertain settings across various domains.

3.2.2 Model-free RL: Policy-based and value-based learning

In model-free RL, two principal approaches guide how agents derive knowledge from their interactions with the environment: policy-based and value-based learning.[88] Each strategy comes with distinct methodologies and inherent benefits.

Policy-based learning directly focuses on learning the policy, which is a mapping from states to actions, often represented as a function or a probability distribution. This approach is particularly advantageous for handling high-dimensional or continuous action spaces and for learning stochastic policies. However, policy-based methods typically suffer from high variance in their updates and can be less efficient in terms of sample usage.[97] Examples of policy-based methods include REINFORCE, where the policy gradient is used, and Proximal Policy Optimization (PPO), which moderates the extent of policy updates to avoid destabilization.[88]

Conversely, value-based learning centers on determining a value function that evaluates

the quality of states (or state-action pairs), reflecting the expected return from those states under a specific policy.[88] This approach excels in sample efficiency as it allows the reuse of past experience to update the value estimates for multiple states based on a single set of actions. It generally provides more stable and consistent updates but struggles with large state or action spaces due to the need to accurately estimate the value function across all possibilities.[97] Notable value-based methods include Deep Q-Networks (DQN), which applies deep learning to estimate the optimal action-value function, and Temporal Difference (TD) Learning, which updates the value function based on differences between estimated values over time steps.

Actor-critic methods in model-free reinforcement learning combine the direct policy optimization of policy-based learning with the stable value estimation of value-based learning. This hybrid approach employs an actor to determine actions and a critic to evaluate these actions via a value function, enhancing stability and reducing the variance associated with policy-based methods alone. This dual setup quickens convergence improves sample efficiency, and maintains a balance between exploration and exploitation, which is crucial for navigating complex environments.[88]

Examples of sophisticated Actor-Critic methods include Deep Deterministic Policy Gradient (DDPG), which pairs a deterministic policy actor with a Q-function critic for continuous action spaces; Soft Actor-Critic (SAC), which uses entropy regularization to foster exploration; and Twin Delayed DDPG (TD3), which minimizes bias and stabilizes training through delayed updates. These methods represent significant advancements in RL, effectively integrating policy and value methods to adapt dynamically to diverse applications and continuously pushing the boundaries of what autonomous agents can achieve.

RL algorithms can be distinguished by their on-policy or off-policy approach and their suitability for discrete or continuous action spaces. Among these, Deep Deterministic Policy Gradient (DDPG) stands out as a robust off-policy, actor-critic method optimized for continuous action spaces, making it ideal for complex environments like robotics and autonomous vehicles. DDPG excels over discrete space methods like Q-Learning and SARSA, which struggle with continuous domains due to the need for action space discretization that complicates implementation and hampers efficiency. Unlike on-policy methods such as A2C and

PPO, which require fresh environmental samples for each update, DDPG enhances learning efficiency by using a replay buffer to learn from past experiences. Furthermore, DDPG maintains simplicity and accessibility compared to other advanced actor-critic methods like SAC and TD3, which introduce additional complexity. It has been extensively tested and proven effective, providing a reliable option for applications requiring precision and practical usability.[98]

3.2.3 Deep Deterministic Policy Gradients

Deep Deterministic Policy Gradient (DDPG)[99] is a model-free, off-policy actor-critic algorithm designed for the continuous action domain. It merges the concepts from Deterministic Policy Gradient (DPG) and Deep Q-Networks (DQN) to operate in environments with continuous action spaces, overcoming the shortcomings of each individual approach of policy based and value based methods for model-free RL. DDPG adopts the actor-critic framework, comprising two main components: the actor function that specifies the policy by mapping states to actions, and the critic function that estimates the value of state-action pairs. Thus providing a powerful solution for complex control tasks that neither purely policy-based nor value-based methods could solve efficiently on their own. DDPG utilizes two neural networks that work in tandem to learn optimal policies and value functions.

Actor - Deterministic Policy Gradient : The actor in DDPG defines a deterministic policy, which is an explicit function mapping states to actions. This policy is deterministic in the sense that for a given state, the actor outputs the same action every time, as opposed to a stochastic policy that would output a distribution over actions.

$$a_t = \mu_\theta(s_t), \tag{3.2}$$

where a_t is the action, s_t is the current state, and μ_θ represents the deterministic policy network parameterized by θ^μ. This network is updated using a policy gradient method that aims to maximize the expected return by adjusting the parameters in the direction that increases the probability of good actions.

Critic - Deep Q-learning Network : The critic evaluates the chosen actions given the

current state. It approximates the Q-function, which estimates the expected return of taking an action a_t in state s_t and following policy μ thereafter. The Q-function is learned using temporal difference learning, with the target values provided by a separate target network to improve learning stability.

$$Q(s_t, a_t|\theta^Q) \approx r_{t+1} + \gamma Q'(s_{t+1}, \mu'(s_{t+1}|\theta^{\mu'})|\theta^{Q'}),$$

where θ^Q and $\theta^{Q'}$ are the parameters of the critic and target critic networks, respectively, and γ is the discount factor.

Exploration and Exploitation Balance : A critical aspect of reinforcement learning is the balance between the exploration of the action space and the exploitation of known information. DDPG manages this balance by incorporating a noise process, often an Ornstein-Uhlenbeck process, into the action selection policy. This allows the algorithm to explore efficiently, avoiding local optima and ensuring diverse experiences are gathered.

Continuous Action Space : One of the defining features of DDPG is its ability to handle continuous action spaces. This capability is critical for tasks that require precise control, such as robotic arm manipulation. The deterministic policy in DDPG eliminates the need for action space discretization, which is both inefficient and may discard valuable structural information.

Experience Replay : Experience replay in DDPG improves data efficiency by storing and reutilizing past experiences, thus maximizing the utility of each observation and breaking temporal correlations.

Stabilization with Soft Updates : To stabilize learning in the presence of non-stationary targets, DDPG utilizes a technique known as *soft updates*, characterized by the parameter τ. This parameter controls the extent to which the target networks are updated, providing a smoother and more stable learning signal for the algorithm. The target networks $\theta^{Q'}$ and $\theta^{\mu'}$ slowly track the learned networks θ^Q and θ^μ using the update rule:

$$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'},$$

$$\theta^{\mu'} \leftarrow \tau\theta^{\mu} + (1-\tau)\theta^{\mu'},$$

where τ is a hyperparameter determining the mix between the target and main network parameters, typically chosen to be much less than 1.

The algorithm shown below explains the workflow of DDPG as explained by Lillicrap et al. for continuous control is adapted from [100]

Algorithm 2 Deep Deterministic Policy Gradient (DDPG)

Initialize critic network $Q(s, a|\theta^Q)$ and actor $\mu(s|\theta^\mu)$ with random weights
Initialize target networks Q' and μ' with weights $\theta^{Q'} \leftarrow \theta^Q$, $\theta^{\mu'} \leftarrow \theta^\mu$
Initialize replay buffer
for episode = 1 to M **do**
    Initialize a random process $\mathcal{N}$ for action exploration
    Receive initial observation state s_1
    for $t = 1$ to T **do**
        Select action $a_t = \mu(s_t|\theta^\mu) + \mathcal{N}_t$ according to the current policy and exploration
        Execute action a_t and observe reward r_t and new state s_{t+1}
        Store transition (s_t, a_t, r_t, s_{t+1}) in replay buffer
        Sample a random minibatch of N transitions (s_i, a_i, r_i, s_{i+1}) from replay buffer
        Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$ for each in the minibatch
        Update the critic by minimizing the loss: $L = \frac{1}{N}\sum(y_i - Q(s_i, a_i|\theta^Q))^2$
        Update the actor policy using the sampled policy gradient:
            $\nabla_{\theta^\mu} J \approx \frac{1}{N}\sum \nabla_a Q(s, a|\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu}\mu(s|\theta^\mu)|_{s_i}$
        Update the target networks:
            $\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$
            $\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}$
    end for
end for

3.3 Control comparison between MPC and RL

In the field of process control, the juxtaposition of MPC and RL, notably deep RL, signifies substantial advancements in managing complex systems. MPC involves forecasting of future system states and consequent adjustments in control actions. This proficiency is crucial for handling constraints and ensuring system stability. Nevertheless, the reliance of MPC on precise system models, which are challenging to maintain, and its computational intensity

in complex systems pose significant limitations .[101]

Conversely, RL provides a model-free methodology, learning optimal strategies through direct system interactions, a boon in environments where the system model is complex or unknown. RL's adaptability is critical, as it continuously refines its control strategies based on real-time data. However, the effectiveness of RL is contingent on substantial data availability for training, and its initial performance can be suboptimal until the system sufficiently learns from its environment.[102]

Addressing the shortcomings and leveraging the strengths of both MPC and RL, recent studies have explored their integration. This hybrid approach seeks to combine MPC's predictive accuracy with RL's dynamic adaptability. By utilizing MPC to guide the RL training process, the integrated system enhances learning speed and efficiency, which is beneficial in managing systems with significant delays and nonlinear behaviors. Such systems have shown promising results in simulations and controlled environments, suggesting enhanced control performance in practical applications such as CSTRs. [102, 103]

Despite these theoretical and controlled successes, the practical deployment of combined MPC-RL systems in industrial settings is nascent. The integration faces challenges such as high computational demands and the need to manage complex, noisy data effectively. Future research is expected to focus on these challenges, aiming to improve the robustness and efficiency of these systems for broader real-world application.

As computational power increases and data availability expands, the potential for MPC and RL to revolutionize process control continues to grow. Ongoing research is likely to extend the application of RL across various types of chemical processes, aiming to validate its effectiveness and refine its implementation in industrial settings. The integration of RL with model-free control methods like Model-Free Predictive Control (MFPC) and Model-Free Learning Control (MFLC) is already improving operational stability and response times to disturbances, indicating a significant step forward in the automation of chemical engineering and other complex industries. [104]

These studies have demonstrated the deployment of MPC and RL-based controllers on a CSTR system. However, the application is currently limited to ODE systems involving simple concentration and height equations. In contrast, the complex reaction systems discussed in

the previous chapter will require the use of Neural ODE as the chosen model for implementing MPC and RL (DDPG) based controllers.

To evaluate the capabilities of the Neural ODE model in capturing the complex reaction dynamics within a Continuous Stirred Tank Reactor (CSTR), an examination of its performance under an open-loop configuration is conducted. In this setup, the system inputs are predetermined and applied without any feedback-based adjustments contingent on the system's outputs or its current state. This methodological approach is instrumental in elucidating the intrinsic response characteristics of the model to specific control inputs. Furthermore, it serves to validate the fidelity of the Neural ODE model in accurately representing the fundamental dynamics of the process, thereby providing a robust platform for assessing the model's predictive precision and stability under fixed operational conditions. Reactant concentration is depicted by PC-1 in figure 3.1. PC-2 is an intermediate, followed by PC-3 and PC-4 which are products.

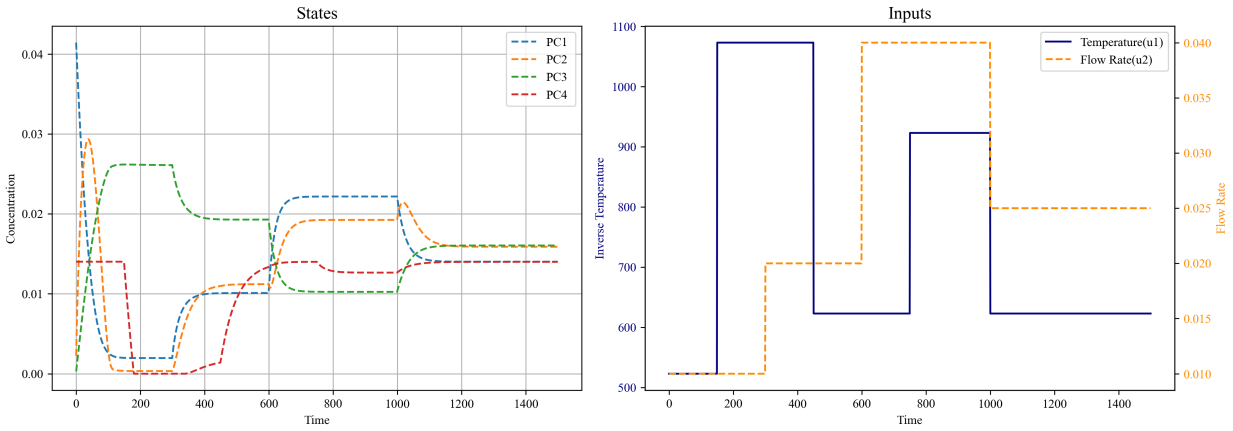

Figure 3.1: Neural ODE: Open-loop simulation (Species A-D: PC:1-4)

3.3.1 DDPG based control

The DDPG algorithm is described in this section. The reward function r, which dictates the learning mechanism within this framework, is defined by :

$$r = -\left[(state_3 - sp)^2 + (state_4 - sp)^2\right] \tag{3.3}$$

Here, $state_3$ and $state_4$ are the PC3 (species C) and PC4 (species D) concentrations,

representing products of the reaction system, respectively, and *sp* represents their targeted setpoint values. This reward function formulation is essential for the optimization process, as it aims directly at minimizing the operational costs, which, within the DDPG paradigm, translates to maximizing the cumulative reward.

The rationale behind setting specific setpoints for $state_3$ and $state_4$ is linked to the kinetics of the reaction processes being controlled. The modeling and control objectives focus on driving the reaction kinetics to a point where the production of the desired product is maximized, while simultaneously ensuring that the production of any undesired by-products is minimized. Such minimization is crucial as it prevents the undesired by-products from hindering the synthesis of the target product.

Hyper Parameter	**Value**
Soft update parameter(τ)	0.001
Discount factor(γ)	0.95
Buffer size	5,00,000
Actor architecture	[20, 75, 75, 75, 20] neurons
Critic architecture	[20, 75, 75, 75, 20] neurons
Actor learning rate	0.0003
Critic learning rate	0.0003
Total episodes	1500
Warmup episodes	500
Episode length	1000
Per episode execution(Testing)	0.43 second

Table 3.1: DDPG training details

Table 3.1 details the training hyperparameters, while the episodic rewards are illustrated in Figure 3.3. The controller rapidly achieves the setpoints, even with disturbances introduced at the 100^{th} and 200^{th} time steps, demonstrating aggressive control actions due to the absence of penalties on states or inputs. Employing a discount factor of 0.95 plays a role akin to considering predictions over a future horizon in the control strategy.

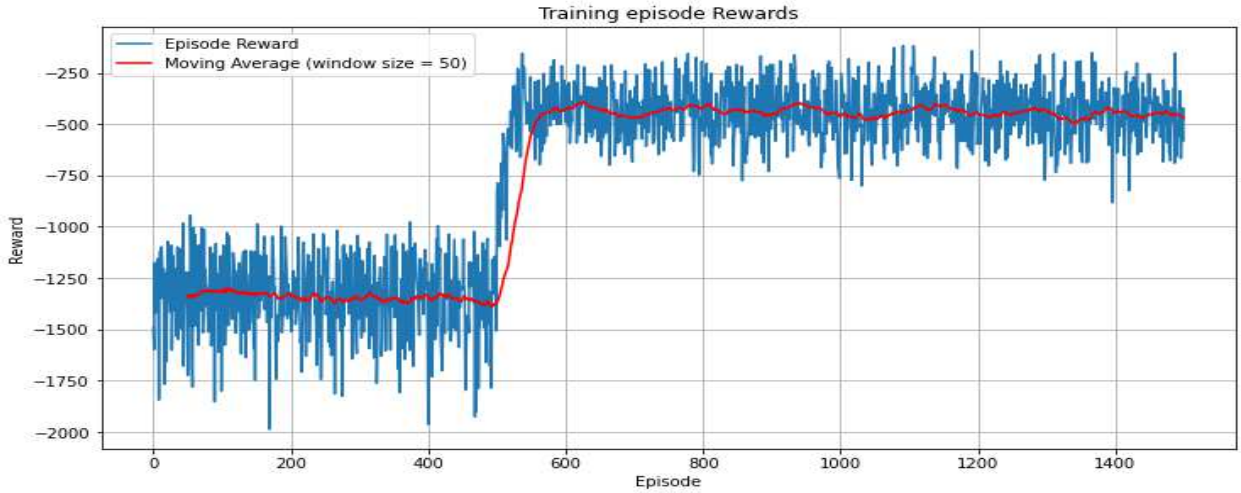

Figure 3.2: Rewards during training

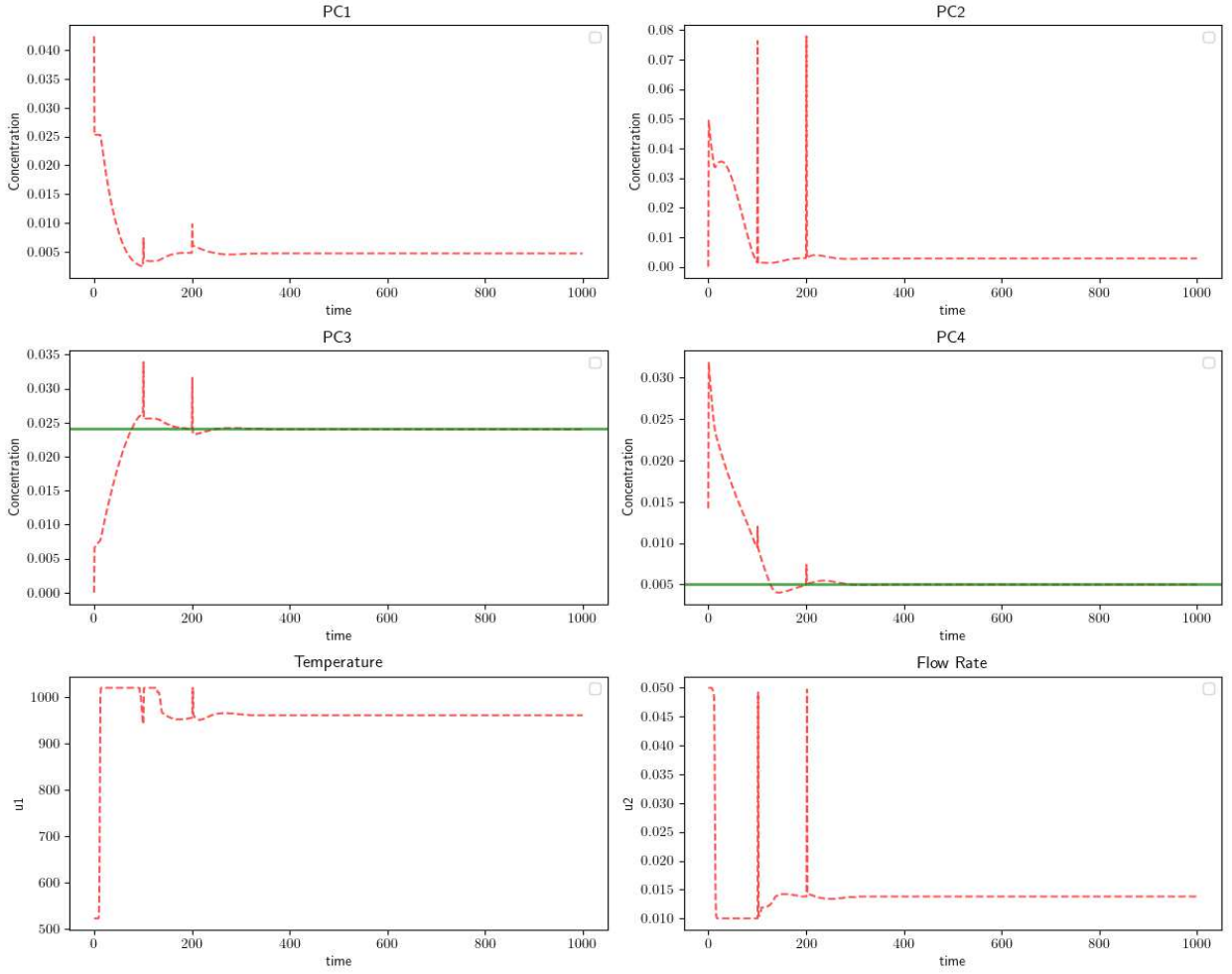

Figure 3.3: DDPG : Setpoint tracking

3.3.2 DDPG based control: MPC reward function

In this section, the reward function is adapted to mirror the objective function used in MPC, which includes penalties for deviations from the state setpoint and the magnitude of inputs. The modified reward function is expressed in the following equation:

$$r = -\left[100 \cdot (state_3 - sp)^2 + 10^{-2} \cdot (T)^2 + 1 \cdot (F)^2\right] \tag{3.4}$$

Hyper Parameter	**Value**
Soft update parameter(τ)	0.001
Discount factor(γ)	0.95
Buffer size	5,00,000
Actor architecture	[20, 75, 75, 75, 20] neurons
Critic architecture	[20, 75, 75, 75, 20] neurons
Actor learning rate	0.0003
Critic learning rate	0.0003
Total episodes	1500
Warmup episodes	500
Episode length	1000
State penalty(PC3)	100
Input penalty(T, F)	0.01,1
Per episode execution(Testing)	3.2 seconds

Table 3.2: DDPG training details

The hyperparameters remain consistent, with the addition of penalties during training. This inclusion helps refine the training process, ensuring that the model effectively minimizes deviations from setpoints and controls the magnitude of inputs.

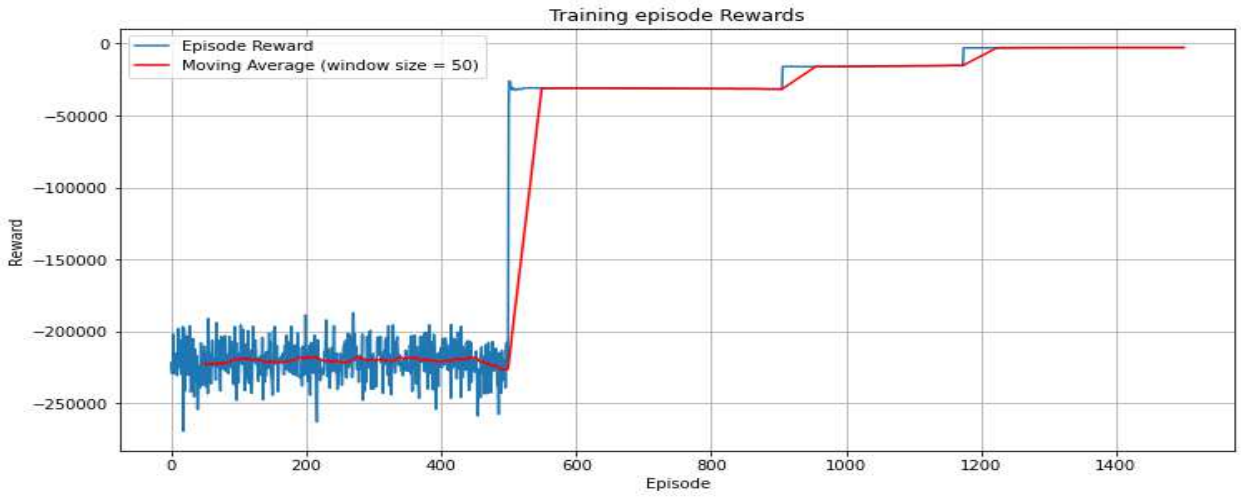

Figure 3.4: Rewards during training

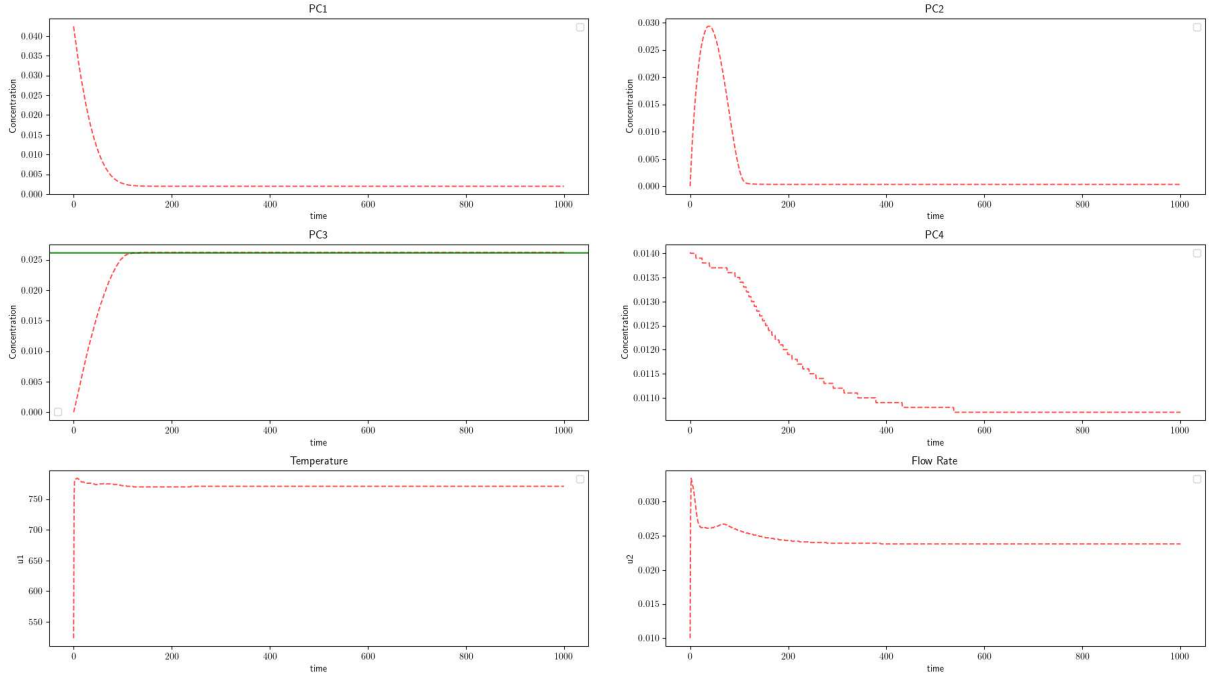

Figure 3.5: DDPG-MPC reward : Setpoint tracking

Setpoint attainment remains rapid, similar to previous cases, but the fluctuations in input values are less aggressive. However, using the discount factor as a basis for comparing the performance of an RL-based controller with MPC is challenging, as there is no established correlation between the discount factor and the prediction horizon that can be generalised.

3.3.3 MPC-RL comparison

In RL, particularly in DDPG, the discount factor plays a pivotal role in balancing the importance of short-term versus long-term rewards. Studies have tried to incorporate RL with MPC, where a discounted value function measures control performance across different prediction horizons, trying to understand the link between horizon and discount factor.[105, 106, 107] These strategies optimize the trade-off between immediate and future costs, suggesting its utility for aligning the discount factor in RL-DDPG with the predictive effectiveness of a n-step MPC horizon. This configuration effectively manages the trade-off between exploiting known rewards and exploring actions that may yield greater future rewards. The selected discount factor aids in stabilizing the learning process by mitigating the overvaluation of speculative long-term returns, which is particularly beneficial in environments with uncertainty or dynamic variations.

Aligning the discount factors in both DDPG and MPC ensures that both control strategies adhere to a similar principle of temporal valuation of costs and rewards. This alignment is particularly helpful when dealing with infinite horizon problems[108], especially in RL. This alignment is essential for applications where MPC and DDPG may be used jointly or where one is employed for real-time control and the other for simulation or long-term planning. By setting the discount factor at 0.7 through trial and error, the behaviors of both control strategies are synchronized, leading to more predictable and coherent system responses and facilitating logical performance comparisons.

The discounted cost function for MPC can be formulated as follows:

$$J(x(t), u) = \sum_{k=0}^{N-1} \gamma^k \left(x(t+k)^T Q x(t+k) + u(t+k)^T R u(t+k) \right) + \gamma^N x(t+N)^T P x(t+N) \tag{3.5}$$

Using this discounted cost function as the objective for the MPC, with a discount factor of 0.7 and the inclusion of terminal penalties aligns with the same discount factor for the DDPG controller during simulation with the training parameters shown in the table below.

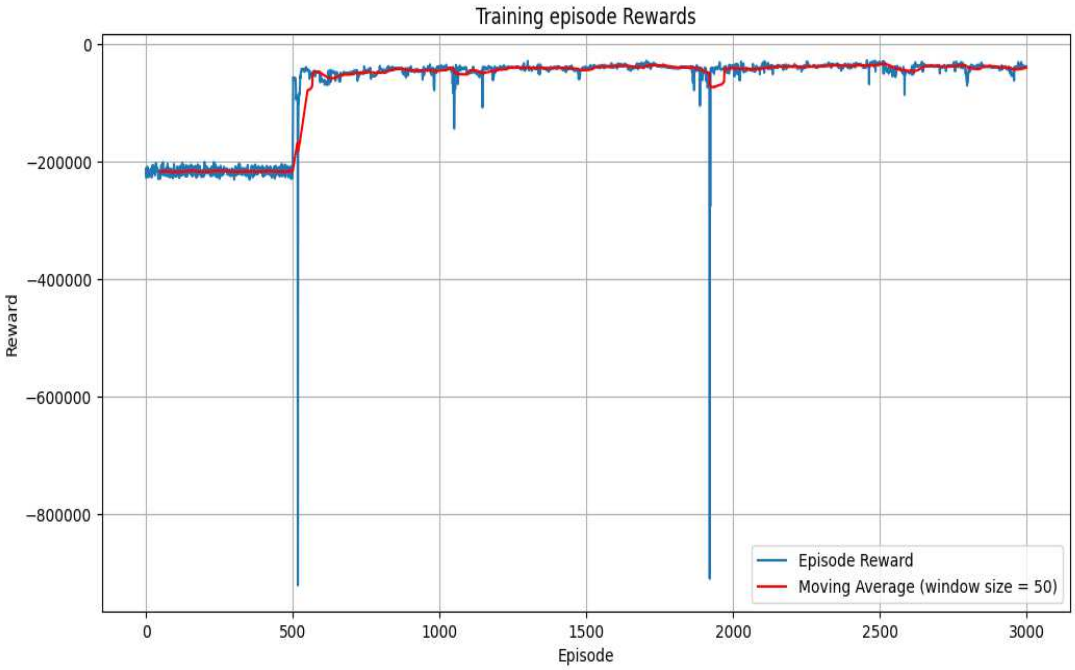

Figure 3.6: Rewards during training

Hyper Parameter	**Value**
Soft update parameter(τ)	0.001
Discount factor(γ)	0.7
Buffer size	10,00,000
Actor architecture	[20, 75, 75, 75, 20] neurons
Critic architecture	[20, 75, 75, 75, 20] neurons
Actor learning rate	0.0003
Critic learning rate	0.0003
Total episodes	3000
Warmup episodes	500
Episode length	2000
State penalty(PC3,PC4)	100,10
Input penalty(T,F)	0.01,1
Per episode execution(Testing)	3.2 seconds

Table 3.3: MPC-DDPG training details

In figure 3.7, the disturbance rejection scenario is demonstrated while accounting for changing setpoints of PC-3. Gaussian noise is added to inputs from time steps 500 to 600, and at 1500^{th} time step, states are perturbed. Both controllers handle setpoint tracking

along with disturbance rejection effectively. The optimal trajectories obtained for MPC and DDPG-based controllers are very similar, with minor differences observed at certain time points. The input penalties are more heavily weighted for temperature values leading to control actions dominant in flow rate.

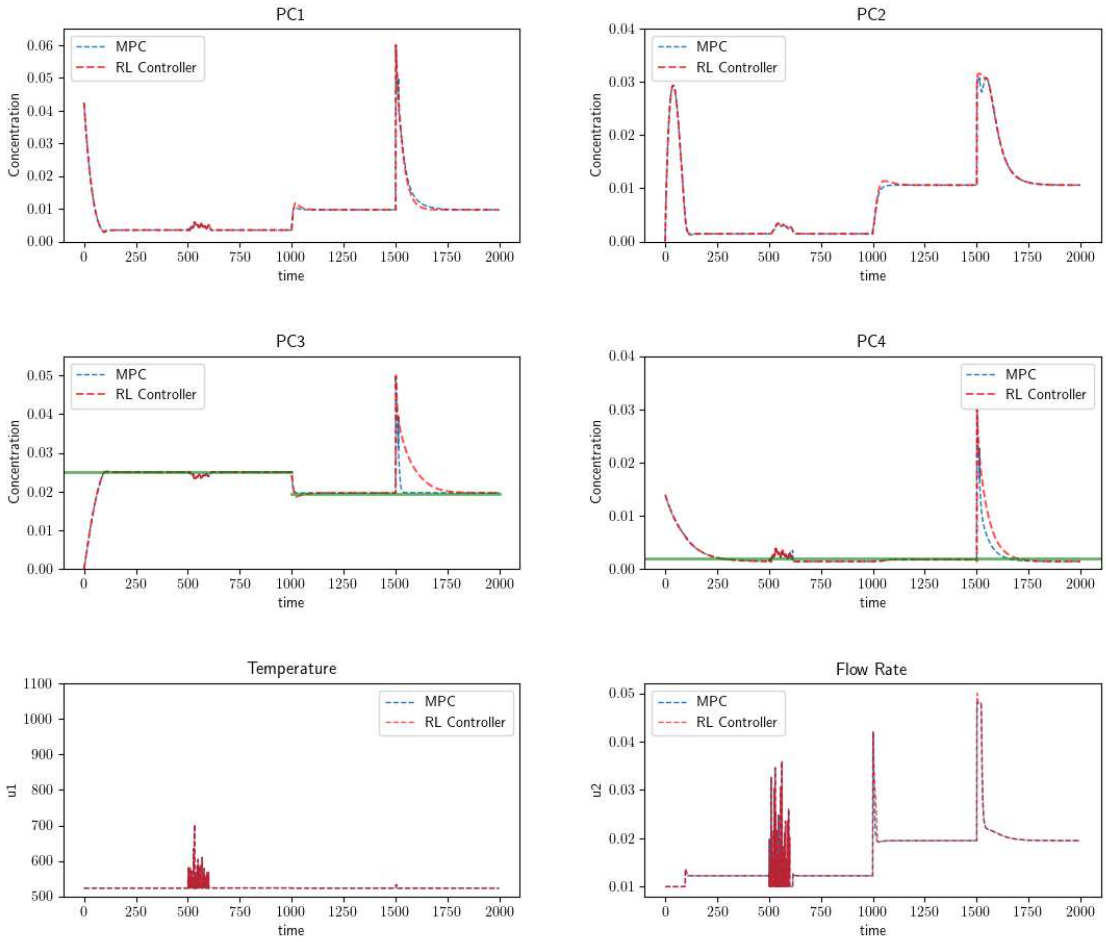

Figure 3.7: DDPG with MPC reward: Setpoint tracking and disturbance rejection scenario

An additional instance is discussed here for a scenario that involves conducting setpoint tracking for two components, PC-3 and PC-4, simultaneously as shown in figure 3.8. The objective function is to meet setpoint tracking for both components at the same time, which is later changed after the first half of the simulation. The penalties on inputs are less stringent than those in figure 3.7 to ensure that the setpoints are attained for both components. The

weights used are highlighted in the table below, and cost comparisons are made to quantify the comparison between RL and DDPG. As the input penalty on temperature has been reduced, it leads to more fluctuations than as seen in Figure 3.7. A significant drop is also noted when the latter half of the simulation aims for new setpoints for PC-3 and PC-4.

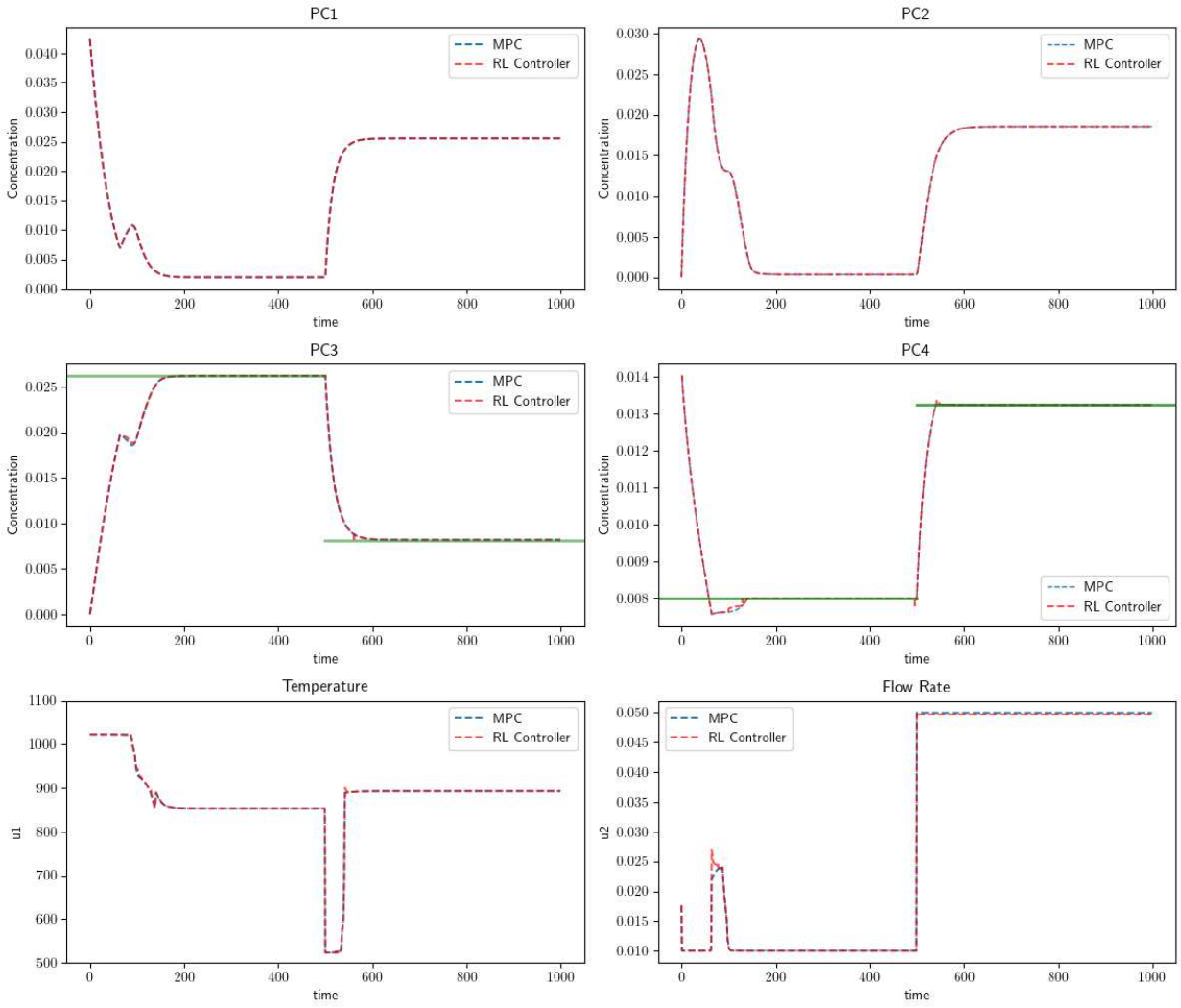

Figure 3.8: DDPG-MPC: Setpoint tracking (PC-3 &PC-4) with same cost function/reward

Parameter	**MPC**	**RL**
Temperature Penalty	1×10^{-4}	1×10^{-4}
Flow Penalty	1×10^{-2}	1×10^{-2}
PC-3 Weight	10	10
PC-4 Weight	1	1
Cost per Step	77.66758	77.6671

Table 3.4: Comparison of MPC and RL performance under basic control parameters.

Parameter	**MPC**	**RL**
Temperature Penalty	1×10^{-2}	1×10^{-2}
Flow Penalty	1	1
PC-3 Weight	100	100
PC-4 Weight	10	10
Cost per Step	2747.433	2748.611

Table 3.5: Detailed cost comparison in a scenario involving disturbance rejection and setpoint tracking.

In the basic control scenario, both MPC and RL underwent similar penalties for temperature and flow with additional weights for specific control points, as detailed in Table 3.4. The results demonstrate nearly identical performance in terms of total cost per step. This minimal difference indicates that both control strategies are effectively equivalent under standard conditions, with RL showing a slight edge in cost efficiency.

While for the scenario including both setpoint tracking and disturbance rejection, the results are closely matched, MPC exhibits a slight superiority in managing disturbances and tracking setpoints more efficiently. This suggests MPC's potential for better handling of dynamic changes and complex control tasks with current system dynamics.

Chapter 4

Conclusions and Future Work

4.1 Conclusions

This research, in its initial phase, demonstrates data acquisition from FTIR mixture spectra without prior knowledge. The adoption of Joint Non-negative Tensorial Factorization (JNTF) facilitates the extraction of temporal concentrations and spectral profiles, supplemented by auxiliary information. This acquired data, combined with Bayesian network analysis, enables the hypothesization of a reaction network. Integrating this reaction network with JNTF-derived pseudo-component concentrations provides a foundation for kinetic modeling of complex reaction systems. Furthermore, a grey box modeling approach, underpinned by first principles, i.e., Neural ODE, is employed alongside an LSTM to capture time series dynamics, showcasing the advantages of data-driven modeling approaches with a stark comparison of 15 versus 57,760 parameters against traditional black-box neural networks.

The research then utilizes Neural ODEs as a model/environment within a control framework to facilitate a comparison between Model Predictive Control (MPC) and Deep Deterministic Policy Gradient (DDPG), an RL technique particularly suited for control tasks. The control task is structured around maximizing concentration to highlight selectivity for desired components in the reaction system.

In aligning the formulations of MPC and RL, the control chapter ensures a logical basis for comparison. The optimal trajectories derived from these formulations demonstrate comparable results, positioning RL as a viable alternative for optimal control.

While MPC has been the more traditional and dependable control method since its inception, advancements in computing have paved the way for the adoption of machine learning and deep learning techniques, which have proven to be reliable and effective. The RL-based controller using DDPG, which interacts dynamically with its environment, exemplifies this trend.

MPC's dependency on model accuracy is a limitation, but it does not require retraining when model changes occur—unlike RL, which develops an optimal policy independent of the model and can utilize transfer learning to adapt and converge towards optimality when the model is altered. RL's dependence on continuous interaction with a simulator contrasts with the static nature of traditional optimal control. Both MPC and DDPG generally yield similar outcomes, although achieving optimal performance with RL involves extensive hyperparameter tuning and reward formulation adjustments.

In terms of constraint handling, RL's exploration phase can lead to constraint violations, typically mitigated by adjusting the reward structure, an aspect not inherently addressed in MPC. Despite MPC's robust performance with nonlinear models in this study, it may face challenges as problem dimensionality increases with increasing species and reactions, particularly with complex feedstocks. Conversely, RL is model-agnostic and capable of addressing highly nonlinear systems or even partial differential equations effectively during the training phase.

Although the comparison between MPC and RL is not direct, the choice between these control methods depends on several factors highlighted above. For the system and model studied, both MPC and RL perform effectively. However, considering the training time and the computationally intensive nature of RL, MPC maintains a strong position for certain applications. Given its promising outcomes, RL should not be disregarded and may be particularly advantageous as system complexity increases.

4.2 Future Work

For future development of this thesis, the performance of Reinforcement Learning (RL) could be enhanced by bypassing traditional modeling approaches and directly utilizing spec-

tral data to control reaction kinetics. This approach would involve advanced preprocessing techniques to extract key features from spectra indicative of reaction progress or specific outcomes. A significant challenge is integrating real-time spectral data into the RL framework, creating models that can correlate spectral signatures with reaction kinetics. An initial step toward addressing this challenge could be to focus on isolated spectral signatures at wavenumbers that do not overlap with other peaks, using these distinct signatures to model the RL environment. This could be framed as an area maximization problem within a specific wavenumber region to enhance concentration control.

As the research progresses, this approach could be expanded to tackle more complex scenarios where multiple pseudo-components contribute to peaks in the same spectral region. Here, the objective would shift to maximizing the production of one specific species while minimizing others, effectively controlling the reaction to favor the formation of desired products over undesired ones. This could evolve into a selectivity or yield optimization problem within the reaction system, guiding the kinetics to achieve specific objectives.

Furthermore, extending this problem to include the determination of physical properties such as density and viscosity would provide additional value. These properties are crucial for the design, optimization, and scaling of chemical processes, influencing aspects like equipment design, safety, and operational efficiency. Accurately determined physical properties, such as viscosity and density, affect critical parameters like heat transfer coefficients and flow characteristics, which are essential for process design and energy consumption during transportation.

The identification of these physical properties could result from a linear or nonlinear combination of individual component properties, weighted by their concentrations or fractions within the system. This holistic approach not only enhances the control of reaction kinetics but also contributes significantly to the broader field of chemical process design and optimization, paving the way for more efficient, safe, and sustainable chemical manufacturing processes.

Bibliography

[1] J. C. J. Bart, E. Gucciardi, and S. Cavallaro, "1 - Renewable lubricants," in *Biolubricants*, ser. Woodhead Publishing Series in Energy, J. C. J. Bart, E. Gucciardi, and S. Cavallaro, Eds. Woodhead Publishing, 2013, pp. 1–9. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B978085709263250001X

[2] S. Samih, M. Latifi, S. Farag, P. Leclerc, and J. Chaouki, "From complex feedstocks to new processes: The role of the newly developed micro-reactors," *Chemical Engineering and Processing-Process Intensification*, vol. 131, pp. 92–105, 2018.

[3] R. Kumar, "A review on the modelling of hydrothermal liquefaction of biomass and waste feedstocks," *Energy Nexus*, vol. 5, p. 100042, 2022.

[4] L. Petzold and W. Zhu, "Model reduction for chemical kinetics: an optimization approach," *Aiche Journal*, vol. 45, pp. 869–886, 1999.

[5] K. Toch, J. Thybaut, and G. Marin, "A systematic methodology for kinetic modeling of chemical reactions applied to n-hexane hydroisomerization," *Aiche Journal*, vol. 61, pp. 880–892, 2015.

[6] A. Kemmler, H. Anderson, K. Heldt, D. Haberland, and B. Hinz, "Kinetic on-line evaluation of chemical reactions," *Journal of Thermal Analysis and Calorimetry*, vol. 52, pp. 187–194, 1998.

[7] C. Westbrook and F. Dryer, "Chemical kinetics and modeling of combustion processes," vol. 18, pp. 749–767, 1981.

[8] J. Wei and C. D. Prater, "The structure and analysis of complex reaction systems," in *Advances in catalysis*. Elsevier, 1962, vol. 13, pp. 203–392.

[9] F. L. Dryer, F. M. Haas, J. Santner, T. I. Farouk, and M. Chaos, "Interpreting chemical kinetics from complex reaction–advection–diffusion systems: Modeling of flow reactors and related experiments," *Progress in energy and combustion science*, vol. 44, pp. 19–39, 2014.

[10] W. Zhang, M. Binns, C. Theodoropoulos, J.-K. Kim, and R. Smith, "Model building methodology for complex reaction systems," *Industrial & Engineering Chemistry Research*, vol. 54, no. 16, pp. 4603–4615, 2015.

[11] M. Frenklach, A. Packard, P. Seiler, and R. Feeley, "Collaborative data processing in developing predictive models of complex reaction systems," *International Journal of Chemical Kinetics*, vol. 36, no. 1, p. 57–66, Nov. 2003. [Online]. Available: http://dx.doi.org/10.1002/kin.10172

[12] M. S. Okino and M. L. Mavrovouniotis, "Simplification of mathematical models of chemical reaction systems," *Chemical Reviews*, vol. 98, no. 2, p. 391–408, Feb. 1998. [Online]. Available: http://dx.doi.org/10.1021/cr950223l

[13] S. Prickett and M. Mavrovouniotis, "Construction of complex reaction systems—ii. molecule manipulation and reaction application algorithms," *Computers amp; Chemical Engineering*, vol. 21, no. 11, p. 1237–1254, Jan. 1997. [Online]. Available: http://dx.doi.org/10.1016/S0098-1354(97)00003-3

[14] M. Neurock, C. Libanati, A. Nigam, and M. T. Klein, "Monte carlo simulation of complex reaction systems: molecular structure and reactivity in modelling heavy oils," *Chemical Engineering Science*, vol. 45, no. 8, pp. 2083–2088, 1990.

[15] C. Heath Turner, J. K. Brennan, M. Lisal, W. R. Smith, J. Karl Johnson, and K. E. Gubbins, "Simulation of chemical reaction equilibria by the reaction ensemble monte carlo method: a review," *Molecular Simulation*, vol. 34, no. 2, pp. 119–146, 2008.

[16] M. A. Kayala and P. Baldi, "Reactionpredictor: Prediction of complex chemical reactions at the mechanistic level using machine learning," *Journal of Chemical Information and Modeling*, vol. 52, no. 10, p. 2526–2540, Oct. 2012. [Online]. Available: http://dx.doi.org/10.1021/ci3003039

[17] M. Meuwly, "Machine learning for chemical reactions," *Chemical Reviews*, vol. 121, no. 16, p. 10218–10239, Jun. 2021. [Online]. Available: http://dx.doi.org/10.1021/acs.chemrev.1c00033

[18] S. Stocker, G. Csányi, K. Reuter, and J. T. Margraf, "Machine learning in chemical reaction space," *Nature Communications*, vol. 11, no. 1, Oct. 2020. [Online]. Available: http://dx.doi.org/10.1038/s41467-020-19267-x

[19] E. S. Blurock, "Characterizing complex reaction mechanisms using machine learning clustering techniques," *International Journal of Chemical Kinetics*, vol. 36, no. 2, p. 107–118, Dec. 2003. [Online]. Available: http://dx.doi.org/10.1002/kin.10179

[20] Z. W. Ulissi, A. J. Medford, T. Bligaard, and J. K. Nørskov, "To address surface reaction network complexity using scaling relations machine learning and dft calculations," *Nature Communications*, vol. 8, no. 1, Mar. 2017. [Online]. Available: http://dx.doi.org/10.1038/ncomms14621

[21] J. T. Margraf, H. Jung, C. Scheurer, and K. Reuter, "Exploring catalytic reaction networks with machine learning," *Nature Catalysis*, vol. 6, no. 2, p. 112–121, Jan. 2023. [Online]. Available: http://dx.doi.org/10.1038/s41929-022-00896-y

[22] D. Fooshee, A. Mood, E. Gutman, M. Tavakoli, G. Urban, F. Liu, N. Huynh, D. Van Vranken, and P. Baldi, "Deep learning for chemical reaction prediction," *Molecular Systems Design amp; Engineering*, vol. 3, no. 3, p. 442–452, 2018. [Online]. Available: http://dx.doi.org/10.1039/C7ME00107J

[23] Z. Zhou, X. Li, and R. N. Zare, "Optimizing Chemical Reactions with Deep Reinforcement Learning," *ACS Central Science*, vol. 3, no. 12, pp. 1337–1344, Dec. 2017. [Online]. Available: https://pubs.acs.org/doi/10.1021/acscentsci.7b00492

[24] C. Komives and R. S. Parker, "Bioreactor state estimation and control," *Current Opinion in Biotechnology*, vol. 14, no. 5, p. 468–474, Oct. 2003. [Online]. Available: http://dx.doi.org/10.1016/j.copbio.2003.09.001

[25] R. Simutis and A. Lübbert, "Bioreactor control improves bioprocess performance," *Biotechnology Journal*, vol. 10, no. 8, p. 1115–1130, Jul. 2015. [Online]. Available: http://dx.doi.org/10.1002/biot.201500016

[26] S. Mitra and G. S. Murthy, "Bioreactor control systems in the biopharmaceutical industry: a critical perspective," *Systems Microbiology and Biomanufacturing*, vol. 2, no. 1, p. 91–112, Aug. 2021. [Online]. Available: http://dx.doi.org/10.1007/s43393-021-00048-6

[27] S. Ramaswamy, T. Cutright, and H. Qammar, "Control of a continuous bioreactor using model predictive control," *Process Biochemistry*, vol. 40, no. 8, p. 2763–2770, Jul. 2005. [Online]. Available: http://dx.doi.org/10.1016/j.procbio.2004.12.019

[28] D. Dochain, M. Perrier, and M. Guay, "Extremum seeking control and its application to process and reaction systems: A survey," *Mathematics and Computers in Simulation*, vol. 82, no. 3, p. 369–380, Nov. 2011. [Online]. Available: http://dx.doi.org/10.1016/j.matcom.2010.10.022

[29] S. Vaidyanathan, "Anti-synchronization of brusselator chemical reaction systems via adaptive control," *International Journal of ChemTech Research*, vol. 8, pp. 759–768, 09 2015.

[30] T. D. Knapp, H. M. Budman, and G. Broderick, "Adaptive control of a cstr with a neural network model," *Journal of Process Control*, vol. 11, no. 1, pp. 53–68, 2001.

[31] R. K. Al Seyab and Y. Cao, "Differential recurrent neural network based predictive control," *Comput. Chem. Eng.*, vol. 32, no. 7, pp. 1533–1545, Jul. 2008.

[32] E. Pan, P. Petsagkourakis, M. Mowbray, D. Zhang, and E. A. d. Rio-Chanona, "Constrained model-free reinforcement learning for process optimization," *Comput. Chem. Eng.*, vol. 154, no. 107462, p. 107462, Nov. 2021.

[33] P. Petsagkourakis, I. Sandoval, E. Bradford, D. Zhang, and E. del Rio-Chanona, "Reinforcement learning for batch bioprocess optimization," *Computers amp; Chemical Engineering*, vol. 133, p. 106649, Feb. 2020. [Online]. Available: http://dx.doi.org/10.1016/j.compchemeng.2019.106649

[34] K. Alhazmi and S. M. Sarathy, "Continuous control of complex chemical reaction network with reinforcement learning," in *2020 European Control Conference (ECC)*. IEEE, May 2020. [Online]. Available: http://dx.doi.org/10.23919/ecc51009.2020.9143688

[35] G. Varhegyi, M. J. Antal Jr, E. Jakab, and P. Szabó, "Kinetic modeling of biomass pyrolysis," *Journal of analytical and Applied Pyrolysis*, vol. 42, no. 1, pp. 73–87, 1997.

[36] S. I. Martins, W. M. Jongen, and M. A. Van Boekel, "A review of maillard reaction in food and implications to kinetic modelling," *Trends in food science & technology*, vol. 11, no. 9-10, pp. 364–373, 2000.

[37] M. Mehl, W. J. Pitz, C. K. Westbrook, and H. J. Curran, "Kinetic modeling of gasoline surrogate components and mixtures under engine conditions," *Proceedings of the Combustion Institute*, vol. 33, no. 1, pp. 193–200, 2011.

[38] J. E. White, W. J. Catallo, and B. L. Legendre, "Biomass pyrolysis kinetics: a comparative critical review with relevant agricultural residue case studies," *Journal of analytical and applied pyrolysis*, vol. 91, no. 1, pp. 1–33, 2011.

[39] P. Picotti, O. Rinner, R. Stallmach, F. Dautel, T. Farrah, B. Domon, H. Wenschuh, and R. Aebersold, "High-throughput generation of selected reaction-monitoring assays for proteins and proteomes," *Nature Methods*, vol. 7, pp. 43–46, 2010.

[40] M. A. Bernstein, "Reaction monitoring using nmr," *Magnetic Resonance in Chemistry*, vol. 54, no. 6, p. 422–422, March 2016. [Online]. Available: http://dx.doi.org/10.1002/mrc.4436

[41] M. Khajeh, A. Botana, M. A. Bernstein, M. Nilsson, and G. A. Morris, "Reaction

kinetics studied using diffusion-ordered spectroscopy and multiway chemometrics," *Analytical chemistry*, vol. 82, no. 5, pp. 2102–2108, 2010.

[42] G. G. Allison, "Application of fourier transform mid-infrared spectroscopy (ftir) for research into biomass feed-stocks," *Fourier Transforms–New Analytical Approaches and FTIR Strategies*, pp. 71–88, 2011.

[43] K. Sivaramakrishnan, A. Puliyanda, A. de Klerk, and V. Prasad, "A data-driven approach to generate pseudo-reaction sequences for the thermal conversion of athabasca bitumen," *Reaction Chemistry & Engineering*, vol. 6, no. 3, pp. 505–537, 2021.

[44] D. T. Tefera, L. M. Yanez Jaramillo, R. Ranjan, C. Li, A. de Klerk, and V. Prasad, "A bayesian learning approach to modeling pseudoreaction networks for complex reacting systems: Application to the mild visbreaking of bitumen," *Industrial & Engineering Chemistry Research*, vol. 56, no. 8, pp. 1961–1970, 2017.

[45] A. Kramida, Y. Ralchenko, J. Reader *et al.*, "Nist atomic spectra database (ver. 5.3)," 2015.

[46] M. A. Nemeth, "Multi-and megavariate data analysis," 2003.

[47] T. Kourti, "Process analytical technology beyond real-time analyzers: The role of multivariate analysis," *Critical reviews in analytical chemistry*, vol. 36, no. 3-4, pp. 257–278, 2006.

[48] A. Puliyanda, K. Sivaramakrishnan, Z. Li, A. De Klerk, and V. Prasad, "Structure-Preserving Joint Non-negative Tensor Factorization to Identify Reaction Pathways Using Bayesian Networks," *Journal of Chemical Information and Modeling*, vol. 61, no. 12, pp. 5747–5762, Dec. 2021. [Online]. Available: https://pubs.acs.org/doi/10.1021/acs.jcim.1c00789

[49] A. Puliyanda, K. Sivaramakrishnan, Z. Li, A. de Klerk, and V. Prasad, "Data fusion by joint non-negative matrix factorization for hypothesizing pseudo-chemistry using bayesian networks," *Reaction Chemistry & Engineering*, vol. 5, no. 9, pp. 1719–1737, 2020.

[50] H. Gampp, M. Maeder, C. J. Meyer, and A. D. Zuberbühler, "Calculation of equilibrium constants from multiwavelength spectroscopic data—iiimodel-free analysis of spectrophotometric and esr titrations," *Talanta*, vol. 32, no. 12, p. 1133–1139, Dec. 1985. [Online]. Available: http://dx.doi.org/10.1016/0039-9140(85)80238-1

[51] H. Gampp, M. Maeder, C. J. Meyer, and A. D. Zuberbühler, "Calculation of equilibrium constants from multiwavelength spectroscopic data—iv: Model-free least-squares refinement by use of evolving factor analysis," *Talanta*, vol. 33, no. 12, pp. 943–951, 1986.

[52] H. Gampp, M. Maeder, C. J. Meyer, and A. D. Zuberbuehler, "Quantification of a known component in an unknown mixture," *Analytica chimica acta*, vol. 193, pp. 287–293, 1987.

[53] M. Maeder and A. Zilian, "Evolving factor analysis, a new multivariate technique in chromatography," *Chemometrics and Intelligent Laboratory Systems*, vol. 3, no. 3, pp. 205–213, 1988.

[54] H. Keller and D. Massart, "Evolving factor analysis," *Chemometrics and intelligent laboratory systems*, vol. 12, no. 3, pp. 209–224, 1991.

[55] R. Bro, "Parafac. tutorial and applications," *Chemometrics and intelligent laboratory systems*, vol. 38, no. 2, pp. 149–171, 1997.

[56] A. de Juan and R. Tauler, "Data fusion by multivariate curve resolution," in *Data handling in science and technology*. Elsevier, 2019, vol. 31, pp. 205–233.

[57] C. Ruckebusch and L. Blanchet, "Multivariate curve resolution: a review of advanced and tailored applications and challenges," *Analytica chimica acta*, vol. 765, pp. 28–36, 2013.

[58] D. T. Tefera, A. Agrawal, L. M. Yanez Jaramillo, A. de Klerk, and V. Prasad, "Self-modeling multivariate curve resolution model for online monitoring of bitumen conversion using infrared spectroscopy," *Industrial & Engineering Chemistry Research*, vol. 56, no. 38, pp. 10 756–10 769, 2017.

[59] M. Veeramani, S. S. Doss, S. Narasimhan, and N. Bhatt, "Semi-supervised machine learning approach for reaction stoichiometry and kinetic model identification using spectral data from flow reactors," *Reaction Chemistry & Engineering*, vol. 9, no. 2, pp. 355–368, 2024.

[60] R. Bro and H. A. L. Kiers, "A new efficient method for determining the number of components in parafac models," *Journal of Chemometrics*, vol. 17, no. 5, p. 274–286, May 2003. [Online]. Available: http://dx.doi.org/10.1002/cem.801

[61] H. Thodberg, "A review of bayesian neural networks with an application to near infrared spectroscopy," *IEEE Transactions on Neural Networks*, vol. 7, no. 1, pp. 56–72, 1996.

[62] B. Selman and C. P. Gomes, "Hill-climbing search," Jan. 2006. [Online]. Available: http://dx.doi.org/10.1002/0470018860.s00015

[63] X. Bai and R. Padman, *Tabu Search Enhanced Markov Blanket Classifier for High Dimensional Data Sets*. Springer US, p. 337–354. [Online]. Available: http://dx.doi.org/10.1007/0-387-23529-9_22

[64] I. Tsamardinos, L. E. Brown, and C. F. Aliferis, "The max-min hill-climbing bayesian network structure learning algorithm," *Machine Learning*, vol. 65, no. 1, p. 31–78, Mar. 2006. [Online]. Available: http://dx.doi.org/10.1007/s10994-006-6889-7

[65] de Oliveira, Luís P., Hudebine, Damien, Guillaume, Denis, and Verstraete, Jan J., "A Review of Kinetic Modeling Methodologies for Complex Processes," *Oil Gas Sci. Technol. – Rev. IFP Energies nouvelles*, vol. 71, no. 3, p. 45, 2016. [Online]. Available: https://doi.org/10.2516/ogst/2016011

[66] N. Jiscoot, E. A. Uslamin, and E. A. Pidko, "Model-based evaluation and data requirements for parallel kinetic experimentation and data-driven reaction identification and optimization," *Digital Discovery*, vol. 2, no. 4, p. 994–1005, 2023. [Online]. Available: http://dx.doi.org/10.1039/d3dd00016h

[67] S. Subramanian, F. Ghouse, and P. Natarajan, "Fault diagnosis of batch reactor using machine learning methods," *Modelling and Simulation in Engineering*, vol. 2014, pp. 15–15, 2014.

[68] E. C. Martínez, "Batch process modeling for optimization using reinforcement learning," *Computers & Chemical Engineering*, vol. 24, no. 2-7, pp. 1187–1193, 2000.

[69] Y. Zheng, X. Wang, and Z. Wu, "Machine learning modeling and predictive control of the batch crystallization process," *Industrial & Engineering Chemistry Research*, vol. 61, no. 16, pp. 5578–5592, 2022.

[70] M.-J. Mehrani, F. Bagherzadeh, M. Zheng, P. Kowal, D. Sobotka, and J. Makinia, "Application of a hybrid mechanistic/machine learning model for prediction of nitrous oxide (n2o) production in a nitrifying sequencing batch reactor," *Process Safety and Environmental Protection*, vol. 162, pp. 1015–1024, 2022.

[71] T. Jahnke, "On hybrid models in stochastic reaction kinetics," vol. 1168, pp. 1540–1543, 2009.

[72] H. Zander, R. Dittmeyer, and J. Wagenhuber, "Dynamic modeling of chemical reaction systems with neural networks and hybrid models," *Chemical Engineering Technology*, vol. 22, pp. 571–574, 1999.

[73] N. D. Otalvaro, P. G. Bilir, K. H. Delgado, S. Pitter, and J. Sauer, "Kinetics of the direct dme synthesis: State of the art and comprehensive comparison of semi-mechanistic, data-based and hybrid modeling approaches," *Catalysts*, 2022.

[74] L. Chen, O. Bernard, G. Bastin, and P. Angelov, "Hybrid modelling of biotechnological processes using neural networks," *Control Engineering Practice*, vol. 8, pp. 821–827, 1999.

[75] B. de Silva, K. Champion, M. Quade, J.-C. Loiseau, J. Kutz, and S. Brunton, "Pysindy: A python package for the sparse identification of nonlinear dynamical systems from data," *Journal of Open Source Software*, vol. 5, no. 49, p. 2104, 2020. [Online]. Available: https://doi.org/10.21105/joss.02104

[76] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural Ordinary Differential Equations," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2018/file/69386f6bb1dfed68692a24c8686939b9-Paper.pdf

[77] E. Dupont, A. Doucet, and Y. W. Teh, "Augmented neural odes," *Advances in neural information processing systems*, vol. 32, 2019.

[78] A. Rahman, J. Drgoňa, A. Tuor, and J. Strube, "Neural Ordinary Differential Equations for Nonlinear System Identification," Mar. 2022, arXiv:2203.00120 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2203.00120

[79] W. Ji, F. Richter, M. J. Gollner, and S. Deng, "Autonomous kinetic modeling of biomass pyrolysis using chemical reaction neural networks," *Combustion and Flame*, vol. 240, p. 111992, Jun. 2022. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0010218022000116

[80] Q. Li, H. Chen, B. C. Koenig, and S. Deng, "Bayesian chemical reaction neural network for autonomous kinetic uncertainty quantification," *Physical Chemistry Chemical Physics*, vol. 25, no. 5, pp. 3707–3717, 2023.

[81] A. Puliyanda, K. Srinivasan, Z. Li, and V. Prasad, "Benchmarking chemical neural ordinary differential equations to obtain reaction network-constrained kinetic models from spectroscopic data," *Engineering Applications of Artificial Intelligence*, vol. 125, p. 106690, 2023.

[82] H. E. Dikeman, H. Zhang, and S. Yang, "Stiffness-reduced neural ode models for data-driven reduced-order modeling of combustion chemical kinetics," in *AIAA SCITECH 2022 Forum*. American Institute of Aeronautics and Astronautics, Jan. 2022. [Online]. Available: http://dx.doi.org/10.2514/6.2022-0226

[83] J. Yin, J. Li, I. A. Karimi, and X. Wang, "Generalized reactor neural ode for dynamic reaction process modeling with physical interpretability," *Chemical Engineering Journal*, vol. 452, p. 139487, Jan. 2023. [Online]. Available: http://dx.doi.org/10.1016/j.cej.2022.139487

[84] W. Ji, W. Qiu, Z. Shi, S. Pan, and S. Deng, "Stiff-PINN: Physics-Informed Neural Network for Stiff Chemical Kinetics," *The Journal of Physical Chemistry A*, vol. 125, no. 36, pp. 8098–8106, Sep. 2021. [Online]. Available: https://pubs.acs.org/doi/10.1021/acs.jpca.1c05102

[85] "Understanding LSTM Networks – colah's blog — colah.github.io," https://colah.github.io/posts/2015-08-Understanding-LSTMs/, [Accessed 31-03-2024].

[86] M. Jung, P. R. da Costa Mendes, M. Önnheim, and E. Gustavsson, "Model predictive control when utilizing lstm as dynamic models," *Engineering Applications of Artificial Intelligence*, vol. 123, p. 106226, Aug. 2023. [Online]. Available: http://dx.doi.org/10.1016/j.engappai.2023.106226

[87] D. E. Kirk, *Optimal Control Theory: An Introduction*. Courier Corporation, 2004, specifically see p. 78.

[88] R. S. Sutton and A. G. Barto, *Reinforcement learning: an introduction*, second edition ed., ser. Adaptive computation and machine learning series. Cambridge, Massachusetts: The MIT Press, 2018, pg. 1-13, 91-138.

[89] A. Singh, "Introduction to Reinforcement Learning : Markov-Decision Process — towardsdatascience.com," https://towardsdatascience.com/introduction-to-reinforcement-learning-markov-decision-process-44c533ebf8da, [Accessed 20-04-2024].

[90] Z. Zhao, "Variants of bellman equation on reinforcement learning problems," in *2nd International Conference on Artificial Intelligence, Automation, and High-Performance Computing (AIAHPC 2022)*, L. Zhu, Ed. SPIE, Nov. 2022. [Online]. Available: http://dx.doi.org/10.1117/12.2641841

[91] dan lee, "Reinforcement Learning, Part 5: Monte-Carlo and Temporal-Difference Learning — medium.com," https://medium.com/ai%C2%B3-theory-practice-business/reinforcement-learning-part-5-monte-carlo-and-temporal-difference-learning-889053aba07d, [Accessed 20-04-2024].

[92] "Monte Carlo &x2014; Machine Learning and Data Science Compendium — lazyprogrammer.me," https://lazyprogrammer.me/mlcompendium/rl/mc.html.

[93] "incompleteideas.net," http://incompleteideas.net/papers/DeAsis-MSc-2018.pdf, pg.1.

[94] T. Tournaire, "Model-based reinforcement learning for dynamic resource allocation in cloud environments," Ph.D. dissertation, Institut Polytechnique de Paris, 2022, pg.28.

[95] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis, "A general reinforcement learning algorithm that masters chess, shogi, and go through self-play," *Science*, vol. 362, no. 6419, p. 1140–1144, Dec. 2018. [Online]. Available: http://dx.doi.org/10.1126/science.aar6404

[96] K. P. Kielak, "Do recent advancements in model-based deep reinforcement learning really improve data efficiency?" 2020. [Online]. Available: https://openreview.net/forum?id=Bke9u1HFwB

[97] J. Hui, "RL—Reinforcement Learning Algorithms Comparison — jonathan-hui.medium.com," https://jonathan-hui.medium.com/rl-reinforcement-learning-algorithms-comparison-76df90f180cf, [Accessed 20-04-2024].

[98] R. F. Prudencio, M. R. O. A. Maximo, and E. L. Colombini, "A survey on offline reinforcement learning: Taxonomy, review, and open problems," *IEEE Transactions on Neural Networks and Learning Systems*, p. 1–0, 2024. [Online]. Available: http://dx.doi.org/10.1109/TNNLS.2023.3250269

[99] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *International conference on machine learning*. Pmlr, 2014, pp. 387–395.

[100] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," 2015. [Online]. Available: https://arxiv.org/abs/1509.02971

[101] D. Wang, W. Zheng, Z. Wang, Y. Wang, X. Pang, and W. Wang, "Comparison of reinforcement learning and model predictive control for building energy system optimization," *Applied Thermal Engineering*, vol. 228, p. 120430, Jun. 2023. [Online]. Available: http://dx.doi.org/10.1016/j.applthermaleng.2023.120430

[102] K. Alhazmi, F. Albalawi, and S. M. Sarathy, "A reinforcement learning-based economic model predictive control framework for autonomous operation of chemical reactors," *Chemical Engineering Journal*, vol. 428, p. 130993, Jan. 2022. [Online]. Available: http://dx.doi.org/10.1016/j.cej.2021.130993

[103] H. Shah and M. Gopal, "Model-free predictive control of nonlinear processes based on reinforcement learning," *IFAC-PapersOnLine*, vol. 49, no. 1, pp. 89–94, 2016.

[104] L. Estrada-Rayme and P. Cardenas-Lizana, "Model-free learning control of a nonlinear cstr system," in *2021 IEEE XXVIII International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*. IEEE, Aug. 2021. [Online]. Available: http://dx.doi.org/10.1109/INTERCON52678.2021.9532890

[105] L. Zhang, Y. Peng, W. Sun, and J. Li, "Q-learning-based finite control set model predictive control for lcl-coupled inverters with deviated parameters," in *2023 IEEE 12th Data Driven Control and Learning Systems Conference (DDCLS)*. IEEE, 2023, pp. 949–955.

[106] J. Arroyo, C. Manna, F. Spiessens, and L. Helsen, "Reinforced model predictive control (rl-mpc) for building energy management," *Applied Energy*, vol. 309, p. 118346, Mar. 2022. [Online]. Available: http://dx.doi.org/10.1016/j.apenergy.2021.118346

[107] R. Nian, "Machine learning for industrial processes: Prediction, monitoring, and adaptive control," 2020. [Online]. Available: https://era.library.ualberta.ca/items/101390d9-2390-40aa-a83a-42546b0e4a79

[108] S. Najafi Birgani, B. Moaveni, and A. Khaki-Sedigh, "Infinite horizon linear quadratic tracking problem: A discounted cost function approach," *Optimal Control Applications and Methods*, vol. 39, no. 4, p. 1549–1572, Apr. 2018. [Online]. Available: http://dx.doi.org/10.1002/oca.2425