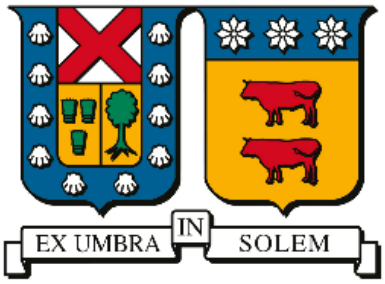
UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

DEPARTAMENTO DE INGENIERÍA QUÍMICA Y AMBIENTAL

VALPARAÍSO - CHILE



DATA-DRIVEN MODELING OF ANAEROBIC DIGESTION PROCESSES WITH A VIEW TO PROCESS CONTROL

Andrés Ignacio Pino Santana

TESIS PARA OPTAR AL GRADO DE MAGÍSTER EN CIENCIAS DE LA INGENIERÍA QUÍMICA

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN CHEMICAL ENGINEERING

Advisor: Dr. Santiago García-Gen

Co-advisor: Dr. Alain Vande Wouwer

October – 2025

CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☐ Memoria o trabajo de título ☑ Tesis de Postgrado

Título del trabajo: Data-Driven Modeling of Anaerobic Digestion Processes with a View to Process Control

Nombre del candidato(a): Andrés Ignacio Pino Santana

Carrera / Grado: Magíster en Ciencias de la Ingeniería Química

Campus: Casa Central Valparaíso **Departamento:** Departamento de Ingeniería Química y Ambiental

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, SANTIAGO GARCÍA GEN, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

☑ El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por **(marcar una opción)**:

☐ 6 meses ☐ 12 meses ☐ 2 años ☐ 3 años ☐ 5 años ☐ 10 años

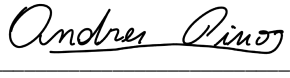
Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 20/10/2025 **Firma:** 

Estudiante o Candidato(a):

Fecha: 19/10/2025 **Firma:** 

Acknowledgments

For now we see through a glass, darkly...

—1 Corinthians 13:12

I would like to begin by expressing my sincerest gratitude to those who made the completion of this thesis possible, a work that marks the closing of a fundamental stage in my life. The path has been complex, full of lessons that transcended the purely academic.

First and foremost, I thank my advisor, Dr. Santiago García-Gen. It was his vision and trust that opened the doors for the research stay in Belgium that would define this work and my professional future. I deeply value that he was the one to offer me this opportunity and his fundamental support in making the trip a reality.

To my co-advisor, Dr. Alain Vande Wouwer, I extend my profound gratitude. His rigor and his "step-by-step" method were crucial for structuring this research. I thank him for his trust, for his decisive support in the publication of our work, and, above all, for offering me the opportunity to continue my training with a Ph.D. under his supervision. I look forward with enthusiasm to our future collaboration.

My stay in Belgium would not have been the same without the *Systems Estimation, Control and Optimization* unit at UMONS. Thank you to all its members for the warm welcome and for making me part of their team.

This work was made possible by the financial support of the FONDECYT project No. 11220818 from ANID (National Agency for Research and Development, Chile) and from Wallonie-Bruxelles International (WBI) through the VII Chile/Wallonia-Brussels Joint Commission. I thank my alma mater, Universidad Técnica Federico Santa María, for being my academic home. I am also grateful to the Université de Mons for welcoming me during my research stay and providing the institutional framework that made it possible.

On a personal level, my infinite gratitude goes to my family. To my parents, who have been an unconditional pillar of support throughout my entire life. We have faced every trial together, and it was their strength and unconditional love that gave me the peace I needed to keep going, both in Chile and during my time abroad.

A special thanks goes to the Jorquera-Briceño family in Brussels. Don Elías, "tía" Blanca, and their children welcomed me not as a guest, but as another member of the family. Their home was a refuge of warmth and normalcy that I will cherish forever.

To my friends, who were my anchor. In Belgium, to Antoine, Akmal and Hugo, for their invaluable company. In Chile, to my lifelong friends, Daniel and Thomas, and to my great university companions, Javier and Felipe, for listening to me during my worst moments and reminding me that I was not alone.

Finally, this thesis closes my chapter at UTFSM. It was a stage of highs and lows where I not only trained as an engineer but also grew as a person. I am grateful to the students I had the honor of guiding as a teaching assistant, for their respect and affection. Beyond my academic life, I am thankful for the lessons learned in resilience, especially to my boxing coach, Francisco, for teaching me to get back up after every hit.

Abstract

The operational complexity of anaerobic digestion (AD) processes often hinders their efficiency. Control strategies based on mechanistic models face significant challenges related to calibration and the formal requirement of state observers, particularly as demonstrating system observability from limited online measurements remains a non-trivial analytical task that is often considered intractable for complex, nonlinear high-dimensional systems. This thesis presents a systematic investigation into the feasibility of a purely data-driven Nonlinear Model Predictive Control (NMPC) strategy as a pragmatic alternative, framed as a proof-of-concept. The proposed framework utilizes a Long Short-Term Memory (LSTM) neural network, whose architecture was informed by preliminary studies from the author's undergraduate work, as a surrogate predictor. A key aspect of this approach is that the model is trained only on readily available online measurements, such as influent and methane flow rates, thereby testing the hypothesis that this limited information set is functionally sufficient for control.

The study methodically evaluates the LSTM-NMPC approach in a simulated environment by progressively increasing the complexity of the AD model, from the simplified AM2 to the comprehensive ADM1. In the simulated scenarios, the data-driven controller successfully achieved setpoint tracking and rejected stochastic disturbances. A comparative analysis showed that the non-linear MPC strategy resulted in a lower tracking error compared to both a linear model-based MPC and a classically tuned PI controller. The scalability of the methodology was further examined through its application to a multi-input anaerobic co-digestion (ACoD) scenario, where Bayesian Optimization was used as an exploratory tool to identify an effective control strategy. Finally, a comparison with model-free Reinforcement Learning (RL) was conducted to contextualize the implementation trade-offs.

This work concludes that a data-driven NMPC strategy is a viable approach for the control of simulated AD processes. The results suggest a pathway for achieving stability and performance without relying on complex, first-principles models, dedicated state observers, or formal observability analysis, demonstrating the potential of this framework as a computationally feasible and scalable control solution.

Keywords:

Anaerobic Co-digestion (ACoD), Anaerobic Digestion (AD), Anaerobic Digestion Model No. 1 (ADM1), Anaerobic Model No. 2 (AM2), Black-Box Modeling, Data-Driven Modeling, Long Short-Term Memory (LSTM), Model Predictive Control (MPC), Nonlinear AutoRegressive with eXogenous inputs (NARX), Nonlinear Model Predictive Control (NMPC), Process Control, Reinforcement Learning (RL), System Identification.

List of Publications

The core findings of the research presented in Chapters 4 and 5 of this thesis were published in the following article:

Pino Santana, A., Garcia-Gen, S., Dewasme, L., & Vande Wouwer, A. (2025). Model predictive control of anaerobic digestion processes using a long short-term memory network predictor. *Water Science and Technology*, 92(7), 1063–1076. https://doi.org/10.2166/wst.2025.139

Glossary

A

- **ACoD (Anaerobic Co-digestion)**: A process involving the simultaneous digestion of two or more different substrates. This practice can enhance the performance of anaerobic digestion by leveraging the complementary characteristics of different materials.
- **Actor-Critic**: A class of Reinforcement Learning algorithms where two components are learned: the Actor (the policy that selects actions) and the Critic (a value function that evaluates those actions).
- **AD (Anaerobic Digestion)**: A biological process in which a consortium of microorganisms breaks down biodegradable organic matter in the absence of oxygen.
- **ADM1 (Anaerobic Digestion Model No. 1)**: A comprehensive and high-fidelity mechanistic model developed by the International Water Association (IWA) to describe the anaerobic digestion process.
- **AM2 (Anaerobic Model No. 2)**: A simplified, control-oriented dynamic model for anaerobic wastewater treatment processes. It simplifies the process into two main stages: acidogenesis and methanization.
- **ANN (Artificial Neural Network)**: A computational model inspired by the structure of biological neural networks, capable of approximating complex nonlinear functions directly from data.
- **ARX (AutoRegressive with eXogenous inputs)**: A linear model structure used in system identification that predicts the current output based on a history of past outputs and external inputs.
- **Attention Mechanism**: A technique in neural networks that allows the model to dynamically focus on the most relevant parts of the input sequence when making a prediction, rather than relying on a fixed-size context.

B

- **Black-Box Modeling**: A system identification approach that focuses on modeling the input-output relationship of a process without detailing the internal mechanisms based on first principles.

C

- **COD (Chemical Oxygen Demand)**: A measure of the amount of oxygen required to chemically oxidize the organic matter in water. It is a key indicator of the pollutant load.

- **CSTR (Continuously Stirred Tank Reactor)**: A common type of chemical reactor in which reactants are continuously fed, and products are continuously removed, while the contents are well-mixed.

D

- **DDPG (Deep Deterministic Policy Gradient)**: A model-free, off-policy, actor-critic reinforcement learning algorithm designed for tasks with continuous action spaces.

G

- **GRU (Gated Recurrent Unit)**: A type of Recurrent Neural Network (RNN) that, like LSTM, uses gating mechanisms to manage information flow but with a simpler architecture.

L

- **LSTM (Long Short-Term Memory)**: A specialized type of Recurrent Neural Network (RNN) designed to learn long-term dependencies in sequential data through a "gating" mechanism.

M

- **MISO (Multi-Input, Single-Output)**: Refers to a system that has multiple inputs and a single output.
- **ML (Machine Learning)**: A field of artificial intelligence that uses statistical techniques to give computer systems the ability to "learn" from data.
- **MPC (Model Predictive Control)**: An advanced control method that uses a dynamic model of a process to predict and optimize its future behavior over a defined time horizon.

N

- **NARX (Nonlinear AutoRegressive with eXogenous inputs)**: A nonlinear model structure that predicts the current output using a history of its own past outputs and past external inputs.
- **NMPC (Nonlinear Model Predictive Control)**: A variant of MPC that uses a nonlinear model to predict the process behavior.

P

- **PID Controller (Proportional-Integral-Derivative)**: A classical feedback control loop mechanism widely used in industrial control systems that calculates an error value as the difference between a measured process variable and a desired setpoint.

R

- **RL (Reinforcement Learning)**: A machine learning paradigm in which an agent learns to make decisions by interacting with an environment to maximize a cumulative reward.
- **RMSE (Root Mean Squared Error)**: A primary metric for evaluating the accuracy of a model's predictions.
- **RNN (Recurrent Neural Network)**: A class of neural networks designed for sequential data, which use feedback loops to maintain an internal state or "memory".

S

- **SAC (Soft Actor-Critic)**: A model-free, off-policy, actor-critic reinforcement learning algorithm based on the maximum entropy framework.
- **SISO (Single-Input, Single-Output)**: Refers to a system that has a single input and a single output.

T

- **TD3 (Twin Delayed Deep Deterministic Policy Gradient)**: An extension of the DDPG algorithm that introduces improvements to increase stability and performance.

V

- **VFA (Volatile Fatty Acids)**: Intermediate compounds produced during the acidogenesis stage of anaerobic digestion.

Nomenclature

AM2 Model Variables

Symbol	Description	Units
X_1	Acidogenic biomass concentration	$g \cdot L^{-1}$
X_2	Methanogenic biomass concentration	$g \cdot L^{-1}$
S_1	Biodegradable organic substrate concentration	$g \cdot L^{-1}$
S_2	Volatile fatty acids (VFA) concentration	$g \cdot L^{-1}$
Z	Total alkalinity concentration	$mmol \cdot L^{-1}$
C	Total inorganic carbon concentration	$mmol \cdot L^{-1}$
D	Dilution rate	d^{-1}
μ_1, μ_2	Specific growth rate for biomass	d^{-1}
$k_1 \ldots k_6$	Yield and stoichiometric coefficients	Dimensionless
q_M	Methane flow rate	$mmol \cdot L^{-1} \cdot d^{-1}$
q_C	Carbon dioxide flow rate	$mmol \cdot L^{-1} \cdot d^{-1}$
pH	System pH	Dimensionless

Key ADM1 Emulator Variables

Symbol	Description	Units
Q_{inf}	Influent flow rate (manipulated variable)	$L \cdot h^{-1}$
Q_{CH_4}	Effluent methane flow (controlled variable)	$L \cdot h^{-1}$
$Q_{str1}, Q_{str2}, Q_{str3}$	ACoD influent flow rates (manure, glycerine, protein)	$L \cdot h^{-1}$
S_{su}	Monosaccharides concentration	$molC \cdot L^{-1}$
S_{aa}	Amino acids concentration	$molC \cdot L^{-1}$
S_{fa}	Long-chain fatty acids concentration	$molC \cdot L^{-1}$
X_{ch}, X_{pr}, X_{li}	Particulate carbohydrates, proteins, and lipids	$molC \cdot L^{-1}$
$X_{su,aa}, X_{fa}, \ldots$	Biomass concentrations (sugar degraders, etc.)	$molC \cdot L^{-1}$
S_{ac}	Acetate concentration	$molC \cdot L^{-1}$
pH	System pH	Dimensionless
ρ_j	Kinetic rate of process j	$molC \cdot L^{-1} \cdot h^{-1}$
$\nu_{i,j}$	Stoichiometric coefficient of component i in process j	Dimensionless
V_{liq}	Liquid volume of the reactor	L
$k_{dis}, k_{hyd}, k_{dec}$	Disintegration, hydrolysis, and decay rate constants	h^{-1} or d^{-1}

k_m	Maximum substrate uptake rate	molC·(molC·h)$^{-1}$
K_S	Half-saturation constant (substrate affinity)	molC/L
I	Inhibition function	Dimensionless

Data-Driven Predictor Variables

Symbol	Description
$\hat{y}_t$	Predicted model output at time t
y_t	Measured process output at time t
u_t	Manipulated variable (input) at time t
w	Window size (history of past data)
$f(\cdot)$	Nonlinear mapping function (approximated by the LSTM)

LSTM Cell Variables

Symbol	Description
c_t	Cell state (long-term memory) at time t
h_t	Hidden state (short-term memory) at time t
x_t	Input to the LSTM cell at time t
i_t	Input gate
f_t	Forget gate

o_t	Output gate
g_t	Candidate for cell state update
σ_g, σ_c	Activation functions (sigmoid and hyperbolic tangent)

MPC Controller Variables

Symbol	**Description**
J_k	Cost function at sampling instant k
N_p	Prediction horizon
N_c	Control horizon
y_{ref}	Desired reference value (setpoint)
u_k	Control action (manipulated variable)
Δu_k	Change in control action
w_y , w_u	Weighting factors for error and control effort

Reinforcement Learning (RL) Variables

Symbol	**Description**
S_t	State of the environment at time step t
A_t	Action taken by the agent at time step t
R_{t+1}	Reward received after taking action At

G_t	Expected return (cumulative discounted reward)
γ	Discount factor for future rewards
π	Policy: the agent's strategy for choosing actions
θ	Parameters of the policy network (Actor)
w_c	Parameters of the value function network (Critic)

Table of contents

List of figures

List of Tables

1 Introduction

1.1 Motivation and Context

Anaerobic digestion (AD) is a bioprocess technology applied to challenges in waste management and renewable energy production (Barahmand & Samarakoon, 2022; Piadeh et al., 2024). Amid increasing energy demands and concerns over resource depletion and climate change, there is a continued need for robust renewable energy sources (Yoshida & Shimizu, 2020). In this context, AD's ability to convert waste streams into valuable energy and nutrient-rich fertilizers positions it as a key technology for advancing the circular economy (Adekunle & Okolie, 2015; Piadeh et al., 2024)

This biological process takes place without oxygen, where a diverse consortium of microorganisms breaks down biodegradable organic matter (Barahmand & Samarakoon, 2022). It is widely used to treat various waste and wastewater streams, such as sewage sludge, municipal solid waste, and agricultural residues (Arzate, 2019; Rodgers et al., 2024). The benefits of AD include reducing the Chemical Oxygen Demand (COD) of the influent, stabilizing organic wastes like sewage sludge, and generating a valuable energy source as biogas, which consists mainly of methane and carbon dioxide (Jiménez-Ocampo et al., 2021; Lima et al., 2025).

Additionally, AD processes show lower sludge production than aerobic digestion (Méndez-Acosta et al., 2005) , and the resulting digestate can be used as a nutrient-rich fertilizer (Adekunle & Okolie, 2015). These characteristics position AD as a key component for achieving a circular economy by converting waste into valuable energy and resources (Piadeh et al., 2024).

Despite its established benefits, the operational efficiency of many AD plants is often limited by the intrinsic complexity of the biological system (Giovannini et al., 2017; Ordace et al., 2012), which is defined by slow reaction rates, strong nonlinearity, and high sensitivity to a range of internal and external disturbances (Arzate, 2019; Méndez-Acosta et al., 2005; Tawai & Sriariyanun, 2022). Such disturbances include variations in substrate composition, temperature changes, and the buildup of intermediate compounds like volatile fatty acids (VFAs) or ammonia, which can inhibit methanogenic activity and lead to process instability or failure (He et al., 2024). As a result, plant operators often face sub-optimal biogas output and effluent quality, partly due to the absence of reliable online measurement systems for key process variables (Gaida et al., 2012). This operational challenge suggests that advanced engineering and control strategies could ensure more stable and efficient AD performance.

1.2 Problem Statement

The operational difficulty of anaerobic digestion (AD) requires advanced control strategies to ensure stable and efficient performance. Model-based control is an established strategy for this, but it depends on an accurate mathematical representation of the process. One common approach is to use mechanistic models that describe the underlying biophysical phenomena. The Anaerobic Digestion Model No. 1 (ADM1) is a well-known example, offering a detailed description useful for simulation and process understanding (Batstone et al., 2002).

However, using ADM1 for real-time control is not straightforward. Its complexity, with many states and parameters, makes calibration a difficult and time-consuming task in an industrial setting (Catenacci et al., 2021; Mudzanani et al., 2023). This complexity reflects the biological reality: the AD process involves numerous interacting biochemical variables, many of which are not well understood or cannot be monitored directly. From a control theory standpoint, this lack of access to the process's internal workings makes verifying the model's observability a major challenge. Observability is the ability to infer the internal state of a system from its outputs. While studies have shown that simplified versions of ADM1 are observable, these analytical methods are too computationally demanding for more complex structures like the full ADM1 (Hellmann et al., 2023). This issue, along with the limited online sensors available (e.g., pH, gas flow), means that fully reconstructing the internal state is a difficult, analytically intractable task. The lack of guaranteed observability is a fundamental problem for control strategies based on full state estimation, like Extended Kalman Filters (EKF) or Particle Filters, because they require a known and observable state-space model (Bornhöft et al., 2013; Gaida et al., 2012).

Simplified models like the Anaerobic Model No. 2 (AM2) were created to help with these control issues. Even with their reduced complexity, however, they are often limited to specific feedstocks and still rely on state estimation for variables that can't be measured online (Bernard et al., 2001; Dewasme et al., 2019; Donoso-Bravo et al., 2011; Jamilis et al., 2018). While formal analytical studies can sometimes be used to confirm observability in these simpler structures, this property is not guaranteed and depends critically on which variables are measured. Confirming observability for a system with few instruments remains a necessary and non-trivial step before designing a state-based controller. The effort involved in this entire approach, from parameterization to observability analysis, has led to interest in other strategies. An alternative is to develop a surrogate model, a data-driven approximation of the process that shifts the engineering focus from first-principles modeling to collecting data and validating an input-output predictor.

This methodological shift becomes particularly relevant for complex models like ADM1, where analytical observability checks are often intractable, creating a practical impasse for control strategies that require a state observer. The data-driven framework offers an alternative path by reframing the problem. Instead of first needing to prove formal observability, it directly

investigates whether a model built on limited measurements is functionally sufficient for robust control. This is not the same as formal observability. It is plausible that certain internal states, particularly those related to physicochemical equilibria and charge balances that are better reflected in measurements like pH, may not be fully captured from gas production alone. Therefore, a central question in this work is whether a practical control solution can be developed without the prerequisites of a formal observability analysis and a dedicated state observer, even if the underlying system might not strictly observable from the available measurements.

Challenges with mechanistic models and the growing availability of online process data have increased interest in black-box, data-driven methods (Gupta et al., 2023; Rutland et al., 2023). Machine Learning (ML) provides a variety of useful techniques. For example, artificial neural networks (ANNs) have been effective at predicting biogas flow rate, handling the complex nonlinear dynamics of AD, especially when combined with metaheuristics for variable selection (Beltramo & Hitzmann, 2019). These techniques can also be used to create software sensors; ANNs have been used to estimate key components from low-cost online measurements to help with process monitoring (Dewasme, 2020). Among these methods, Recurrent Neural Networks (RNNs) like Long Short-Term Memory (LSTM) networks are designed for sequential data (Hochreiter & Schmidhuber, 1997). They have proven useful for predictive monitoring (McCormick & Villa, 2019), and have been considered as internal models for predictive control in similar chemical processes (Zarzycki & Ławryńczuk, 2021). While variations of this approach may have been proposed for AD, this thesis contributes by providing a comprehensive feasibility study, from data generation and model development to closed-loop controller performance analysis and benchmarking against alternative strategies.

The central hypothesis is that an input-output predictor, such as an LSTM network, can effectively model the process dynamics using only historical online measurements. This approach offers a pragmatic alternative to control strategies requiring explicit state estimation, thus bypassing the practical challenges of model observability. However, this methodological shift implies a critical trade-off: the model's reliability is fundamentally constrained to the operational domain represented in the training data, and it cannot, by design, extrapolate to unseen conditions or anticipate instabilities originating from internal dynamics not captured in the input-output record.

While accurate prediction is useful, the main goal is robust process control. Model Predictive Control (MPC) is an advanced tool that uses a dynamic model to optimize future control actions, which is well-suited for the slow dynamics and constraints of AD systems (Bolmanis et al., 2023). Studies have shown that Nonlinear MPC (NMPC) using simplified mechanistic models can be effective, with successful applications in full-scale plants for demand-driven biogas production (Cortés et al., 2022; García-Gen et al., 2022; Mauky et al., 2016). However, these methods still depend on having a reasonably accurate physics-based model. The field of ML for AD modeling is still developing (Gupta et al., 2023), so there is limited research on combining

NMPC with purely data-driven predictors like NNs for AD control. The existing work in this area often uses simulation platforms (e.g., BSM1) or has difficulty with real-world applications, highlighting a gap in the literature regarding the systematic, closed-loop assessment of such data-driven control frameworks (Liu et al., 2023).

This thesis presents a systematic study that demonstrates the principal stages of developing and assessing a data-driven MPC strategy. The work methodically evaluates the framework's performance and scalability, encompassing the sequence from predictor development using simulated data to its final assessment in a closed-loop control environment. The core contribution is the rigorous assessment of this strategy's performance and scalability across a curated progression of simulated environments, moving from a simplified model (AM2) to the comprehensive ADM1, and culminating in a more challenging multi-input co-digestion scenario. The study thereby demonstrates the feasibility of the approach and characterizes its behavior under varying levels of process complexity.

1.3 Justification for the Control Objective

Anaerobic digestion (AD) is commonly applied for the dual purpose of waste management and renewable energy production. The former focuses on stabilizing organic matter and reducing pollutant load, while the latter centers on biogas generation. This thesis concentrates on the energy aspect, specifically on regulating biogas output as the primary control objective.

The methane flow rate (Q_{CH_4}) was selected as the controlled variable for this proof-of-concept study for a few reasons. It is a key performance indicator (KPI) with direct energetic and economic value, so its regulation corresponds to managing a value-added product. Also, methane production results from a complex, nonlinear, and slow-dynamic biochemical network that is sensitive to unmeasured disturbances in the feed, which makes its regulation a non-trivial control problem. Controlling this variable provides a benchmark to test the proposed methodology's ability to handle the complexity, nonlinearity, and uncertainty inherent to the process.

This focus is also consistent with current research. For instance, the work of (Haugen, 2014) on a reactor treating animal waste sets as a primary aim the design of a control system for "controlling the methane gas production to its setpoint" to achieve "constant power production". Similarly, García-Gen et al. (2014) propose an optimization method for co-digestion with the explicit goal of "maximizing COD conversion into methane". In other studies, algorithms have been developed to "track the methane production flowrate set as reference" (García-Gen & Wouwer, 2021) or to act on the dilution rate "to track a target methane flow rate" (Azúa-Poblete et al., 2025). These examples indicate that focusing on methane flow control is aligned with current research practices.

The objective of this thesis, therefore, is not to design a comprehensive operational plan for a treatment plant, but rather to evaluate a data-driven control methodology. The work aims to show that a Model Predictive Control (MPC) strategy, using a black-box model like an LSTM network that operates without knowledge of internal process states (e.g., biomass, VFAs), can be a viable alternative to complex mechanistic models.

By successfully regulating the methane flow rate, this work serves to evaluate a control tool. In subsequent stages, this tool could be used to pursue more complex, multi-objective operational goals. In this sense, this thesis addresses a prerequisite by showing that the process can be controlled with a data-driven approach, which can support future optimization studies.

1.4 Hypothesis and Objectives

1.4.1 Hypothesis

A Nonlinear Model Predictive Control (NMPC) strategy that uses a Long Short-Term Memory (LSTM) neural network as its internal model can provide robust and efficient control for an anaerobic digestion (AD) process. By relying only on readily available online measurements like influent and methane flow rates, this data-driven approach can effectively track setpoints and reject disturbances, offering a practical alternative to control strategies based on complex mechanistic models.

1.4.2 Objectives

1.4.3 General Objective

To develop and characterize the performance and applicability of a data-driven NMPC strategy, using an LSTM network as its internal model, for controlling anaerobic digestion processes.

1.4.4 Specific Objectives

1. **Develop and validate data-driven predictors for the AD process.** This includes:
    a. Generating dynamic data for training and validation from two simulated AD environments: the simplified AM2 and the comprehensive ADM1.
    b. Designing and training LSTM networks to predict methane flow rate from historical input-output data.
    c. Evaluating the one-step and multi-step prediction accuracy of the trained LSTM models on new data.
2. **Design and implement a data-driven NMPC controller.** This involves integrating the validated LSTM models into an MPC framework to optimize control actions while respecting operational constraints.

3. **Evaluate the closed-loop performance of the LSTM-NMPC strategy.** This will be done by:
    a. Testing the controller's ability to track setpoints in both the AM2 and ADM1 simulations.
    b. Assessing the controller's robustness against unmeasured stochastic disturbances in the influent.
4. **Analyze the computational feasibility of the proposed control strategy.** This requires measuring the computation time per control step to determine its suitability for real-time implementation.

1.5 Thesis Outline

This thesis presents a proof-of-concept for a data-driven control strategy. The approach is systematically evaluated in a simulated environment by progressively increasing the complexity of the process, creating a clear narrative progression. The analysis begins with a simplified model (AM2) to establish the fundamental viability, move to the high-fidelity ADM1 to test the methodology against a more complex system, and finally, apply the strategy to a multi-input anaerobic co-digestion (ACoD) scenario to evaluate its scalability.

By using a simulated environment for this study, we create a controlled and repeatable setting. This is a deliberate choice that allows for a direct assessment of the control method's performance, separate from the inherent noise, unmodeled dynamics, and practical challenges of a physical experiment. This approach ensures that the conclusions drawn are directly attributable to the methodology itself, providing a solid foundation before considering the potential surprises of a real-world implementation.

2 Theoretical Framework and State of the Art

2.1 The Anaerobic Digestion Process

Anaerobic digestion (AD) is a complex biological process where a consortium of microorganisms deconstructs biodegradable organic matter without oxygen (Donoso-Bravo et al., 2011). This method is widely used for treating various organic waste streams, including municipal sewage sludge, industrial wastewater, solid waste, and animal manure (Piadeh et al., 2024), while also producing a valuable biogas mainly composed of methane (CH_4) and carbon dioxide (CO_2) (Haugen et al., 2013).

The conversion of organic matter in an anaerobic digester occurs through a series of biochemical reactions, typically categorized into four main stages: hydrolysis, acidogenesis (fermentation), acetogenesis, and methanogenesis (Jeong et al., 2021). In the initial stage, hydrolysis, extracellular enzymes break down complex insoluble organic polymers like carbohydrates, proteins, and fats into simpler, soluble monomers such as monosaccharides, amino acids, and long-chain fatty acids (Adekunle & Okolie, 2015). For complex organic substrates, hydrolysis is often the rate-limiting step (Adekunle & Okolie, 2015).

Next, in acidogenesis, acid-forming bacteria convert these soluble monomers into intermediate products, including volatile fatty acids (VFAs), hydrogen, and carbon dioxide (Adekunle & Okolie, 2015). An accumulation of VFAs can cause acidification, which could destabilize the entire process (Nguyen et al., 2015). Following this, acetogenesis converts VFAs into acetic acid, also producing hydrogen and carbon dioxide (Adekunle & Okolie, 2015). The final stage, methanogenesis, is performed by methanogenic archaea, which transform acetic acid and hydrogen/carbon dioxide into methane (Rodríguez-Jara et al., 2023). Methanogenesis is often the rate-limiting step when dealing with easily biodegradable substrates (Adekunle & Okolie, 2015). The overall AD process has non-linear dynamics and is sensitive to various operating parameters (e.g., temperature, pH, C/N ratio), presenting a significant challenge for ensuring stable and efficient operation (Nguyen et al., 2015).

2.1.1 Anaerobic Model No. 2 (AM2)

The AM2 model is a dynamic, mass-balance-based framework designed for anaerobic wastewater treatment processes (Bernard et al., 2001). Unlike more complex models, AM2 simplifies the process into two steps, acidogenesis and methanization, and explicitly includes physicochemical equilibria to account for alkalinity. This structure allows AM2 to capture both transient and steady-state behaviors under different dilution rates (D). Given its structure, the AM2 is often parameterized to represent the digestion of readily biodegradable substrates.

For a continuously stirred tank reactor (CSTR) at a given dilution rate D, the model tracks six state variables representing the concentrations of acidogenic bacteria, methanogenic bacteria,

organic substrate, volatile fatty acids (VFAs), total alkalinity, and total inorganic carbon. The model's key non-linear behavior is seen in the relationship between the dilution rate and steady-state methane production (Figure 1). At low dilution rates, methane flow increases to an optimal point. Past this point, higher dilution rates cause a sharp drop in performance, leading to a process failure known as washout.

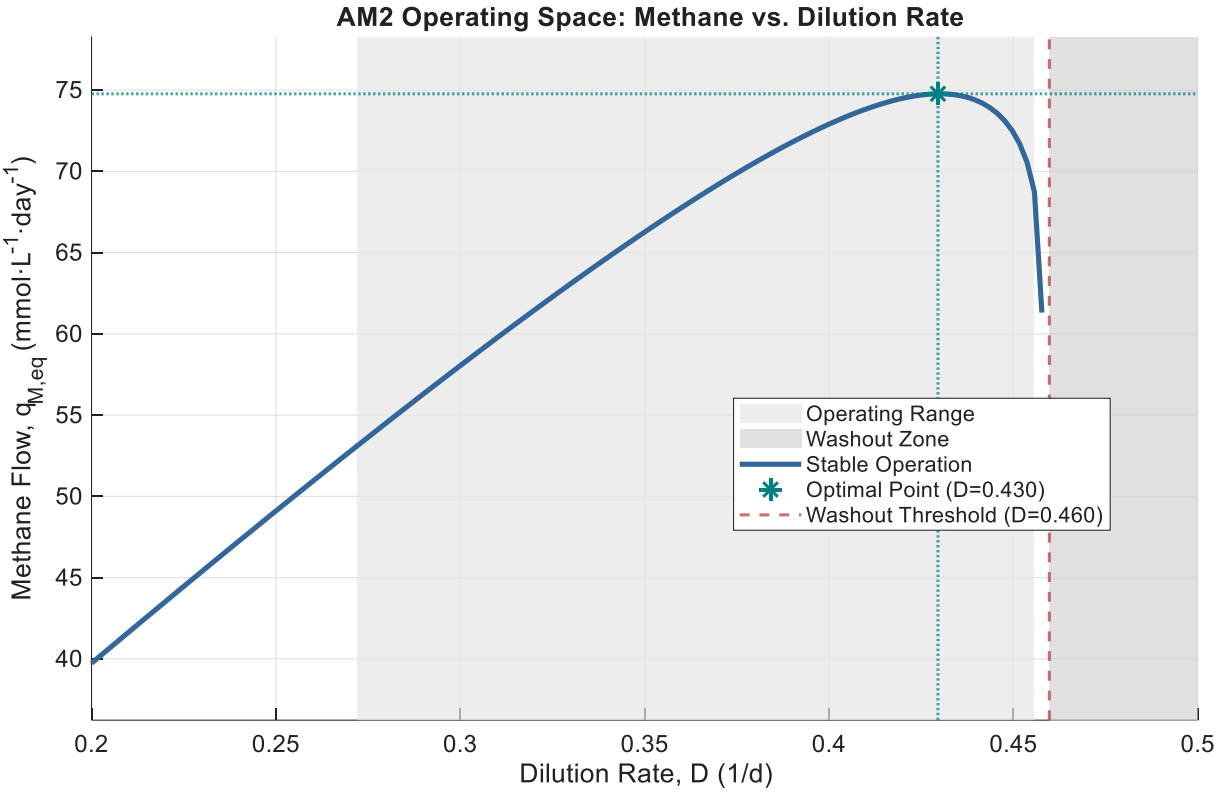


Figure 1: Steady-state operating space of the AM2 model, illustrating the stable operating range, the optimal point for methane production, and the washout threshold.

The mass balances for the state variables are described by the following system of equations:

$$\frac{dX_1}{dt} = \mu_1 X_1 - \alpha D X_1 \quad (1)$$

$$\frac{dX_2}{dt} = \mu_2 X_2 - \alpha D X_2 \quad (2)$$

$$\frac{dS_1}{dt} = D(S_{1,in} - S_1) - k_1 \mu_1 X_1 \quad (3)$$

$$\frac{dS_2}{dt} = D(S_{2,in} - S_2) + k_2 \mu_1 X_1 - k_3 \mu_2 X_2 \quad (4)$$

$$\frac{dZ}{dt} = D(Z_{in} - Z) \tag{5}$$

$$\frac{dC}{dt} = D(C_{in} - C) - q_C + k_4\mu_1 X_1 + k_5\mu_5 X_2 \tag{6}$$

The acidogenic growth rate (μ_1) is modeled by a Monod-type expression, $\mu_1 = \mu_{1,max}\, S_1/(K_{S1} + S_1)$, while the methanogenic growth rate (μ_2) follows Haldane kinetics to account for VFA inhibition: $\mu_2 = \mu_{2,max}\, S_2/\left(k_{S2} + S_2 + S_2^2/K_{I_2}\right)$. Methane production is computed from the methanogenic biomass as $q_M = k_6\mu_2 X_2$.

2.1.2 Anaerobic Digestion Model No. 1 (ADM1)

The Anaerobic Digestion Model No. 1 (ADM1), developed by the International Water Association (IWA) Task Group, represents the most comprehensive mechanistic framework for describing the AD process (Batstone et al., 2002). Its primary purpose was to provide a generalized model that could serve as a common basis for full-scale plant design, operation, and optimization, helping to transfer technology from research to industry.

Unlike simplified models, ADM1 offers a structured description that covers multiple interacting processes. The model's core is a set of mass balance equations for a Continuously Stirred Tank Reactor (CSTR), describing the dynamics of both particulate (X_i) and soluble (S_i) state variables. The general form of these differential equations is:

$$\frac{dX_i}{dt} = \frac{Q_{in}}{V_{liq}}\left(X_{in,i} - X_i\right) + \sum\nolimits_j \nu_{i,j} \cdot \rho_j \tag{7}$$

$$\frac{dS_i}{dt} = \frac{Q_{in}}{V_{liq}}\left(S_{in,i} - S_i\right) + \sum\nolimits_j \nu_{i,j} \cdot \rho_j \tag{8}$$

where Q_{in} is the influent flow rate, V_{liq} is the liquid volume, ρ_j represents the kinetic rate of the j-th process, and $\nu_{i,j}$ is the stoichiometric coefficient of component in process j.

The model details 19 biochemical processes, encompassing the conversion of substrates and the growth and decay of 7 microbial populations, and is complemented by physicochemical equilibria for acid-base reactions and gas-liquid transfer. This intricate structure results in a highly nonlinear system behavior, as illustrated by its steady-state operating space in Figure 2.

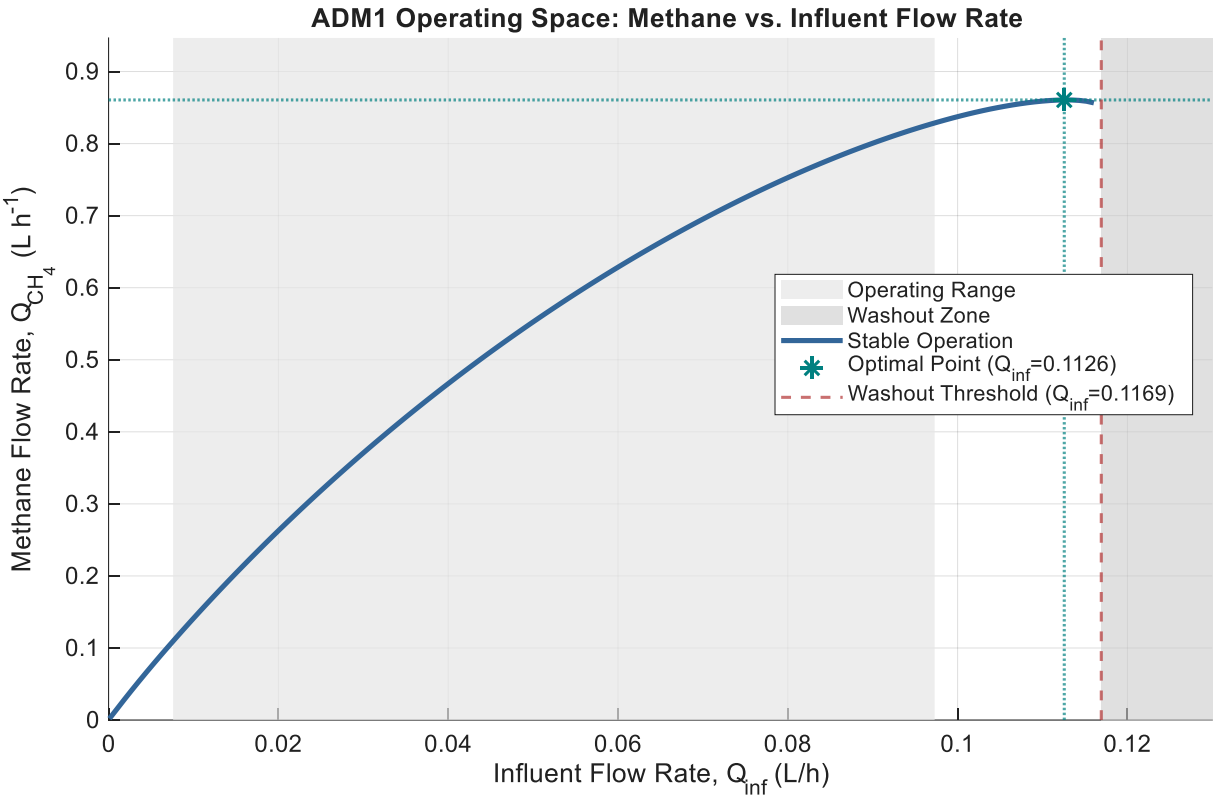


Figure 2: Steady-state operating space of the ADM1 model, showing the stable operating range and the washout threshold.

A full implementation involves a large system of coupled differential and algebraic equations, with over 30 state variables and more than 100 parameters. Presenting the complete model is impractical. Instead, to convey this complexity concisely, the model's structure is summarized in Table 1, following the approach used in recent literature (Moradvandi et al., 2025).

Table 1: Structural summary of the main process groups in ADM1, showing key variables and general rate expressions.

Process Group	Key Variables Involved	Description	General Rate Expression (ρ_j)
Disintegration	X_c	Breakdown of complex composites into inert material, proteins, lipids, and carbohydrates.	$\rho_{dis} = k_{dis} \cdot X_c$
Hydrolysis	X_{ch}, X_{pr}, X_{li}	Enzymatic breakdown of particulates into sugars, amino	$\rho_{hyd} = k_{hyd} \cdot X_i$

		acids, and fatty acids.	
Acidogenesis	S_{su}, S_{aa}	Uptake of sugars and amino acids by acidogenic bacteria ($X_{su,aa}$)	$\rho_{acid} = k_m \frac{S_i}{K_S + S_i} X_{su,aa} \cdot I(\cdot)$
Acetogenesis	$S_{fa}, S_{va}, S_{bu}, S_{pro}$	Uptake of fatty acids by acetogenic bacteria ($X_{fa,c4,pro}$).	$\rho_{acet} = k_m \frac{S_i}{K_S + S_i} X_{fa,c4,pro} \cdot I(\cdot)$
Methanogenesis	S_{ac}, S_{h2}	Conversion of acetate and hydrogen to methane by methanogens ($X_{ac,h2}$).	$\rho_{meth} = k_m \frac{S_i}{K_S + S_i} X_{ac,h2} \cdot I(\cdot)$
Biomass Decay	$S_{su,aa}$, $X_{fa,c4,pro}$, ...	Decay of all 7 microbial groups into composite particulate material.	$\rho_{decay} = k_{dec} \cdot X_{bact}$
Acid-Base	$S_{H^+}, S_{HCO_3^-}$, S_{NH_3}	Fast ion association/dissociation reactions, crucial for pH calculation.	Algebraic Equations (fast equilibria)
Gas Transfer	$S_{gas,ch4}$, $S_{gas,h2}, S_{gas,co2}$	Transfer of dissolved gases from liquid to the gas phase (headspace).	$\rho_{gas} = k_L a(S_i - K_H P_{gas,i})$

While this detailed structure is highly valuable for simulation, its complexity presents significant challenges for real-time control. The primary difficulties are the calibration of its numerous parameters and the high computational load required for its solution, which makes it generally unsuitable for direct use within computationally demanding strategies like Model Predictive Control (Batstone et al., 2006).

2.2 Data-Driven Dynamic Modelling

Given the practical challenges associated with the calibration and state estimation of mechanistic models, as outlined in the problem statement, an alternative paradigm is data-driven modeling. This approach focuses on learning the process dynamics directly from experimental data.

System identification is the process of building mathematical models for dynamic systems from observed input-output data. While linear models are often sufficient for simple systems, many processes in chemical and biological engineering exhibit significant nonlinearities. To address this, nonlinear system identification utilizes flexible model structures, among which Artificial Neural Networks (ANNs) are prominent (M. Norgaard et al., 2000). ANNs can approximate a wide range of nonlinear functions directly from data, reducing the reliance on first-principles models that can be difficult to derive.

2.2.1 Neural Network Structures

The foundational element of an ANN is the artificial neuron, or perceptron. A neuron computes a weighted sum of its inputs, adds a bias, and then passes the result through a non-linear activation function to produce an output. By arranging these neurons in layers, more complex structures can be formed. The most common architecture is the Multilayer Perceptron (MLP), which typically consists of an input layer, one or more hidden layers with non-linear activation functions (e.g., tanh), and an output layer. Figure 3 illustrates the basic structure of an artificial neuron and a multilayer network.

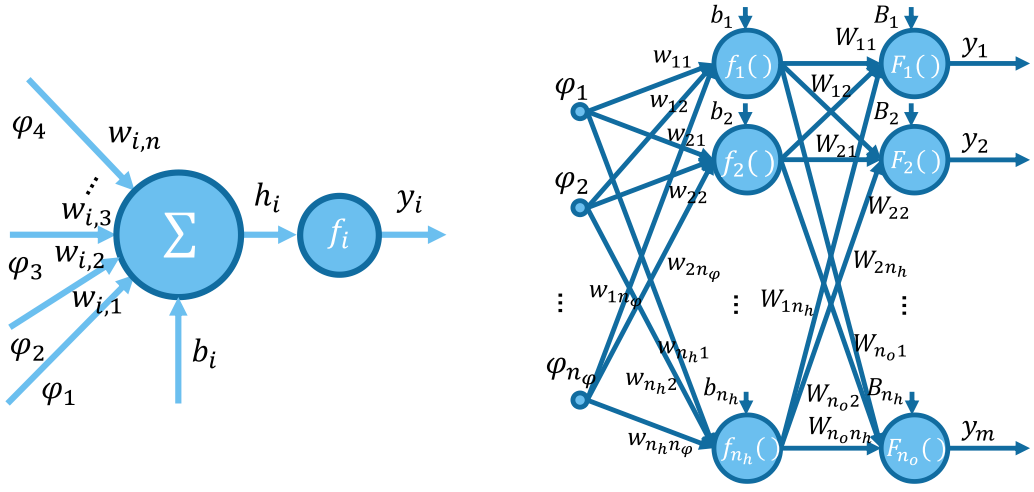


Figure 3: Structure of an artificial neuron (left) and a multilayer neural network (right).

Neural networks can be integrated into classical linear system identification frameworks to create effective non-linear models. This is done by replacing the linear mapping in structures like FIR, ARX, ARMAX, and OE with a neural network, resulting in variants such as NNFIR,

NNARX, NNARMAX, and NNOE (M. Norgaard et al., 2000). These architectures differ mainly in the inputs they use (regressors), such as past inputs, outputs, or prediction errors, which determines if they operate in a feed-forward manner or with internal feedback. An overview of these common structures is provided in Figure 4.

- **NNFIR (Neural Network Finite Impulse Response):** As a non-linear extension of the FIR model, the NNFIR architecture is the simplest structure. It predicts the current output based only on a finite history of past system inputs. Since it does not use past output data, it is a purely feed-forward model.
- **NNARX (Neural Network AutoRegressive with eXogenous inputs):** This is the non-linear version of the linear ARX model. It predicts the current output using a vector of past measurements from both the system's external inputs and its own past outputs. The structure remains feed-forward because it relies only on measured historical signals.
- **NNARMAX (Neural Network ARMAX):** The non-linear version of the ARMAX model, this structure uses past inputs, outputs, and prediction errors as regressors. Including past errors introduces an internal feedback loop into the model.
- **NNOE (Neural Network Output Error):** In the NNOE structure, the model uses its own past predictions as inputs instead of the measured past outputs. This creates a direct feedback loop, making its predictions dependent on its own previous behavior.

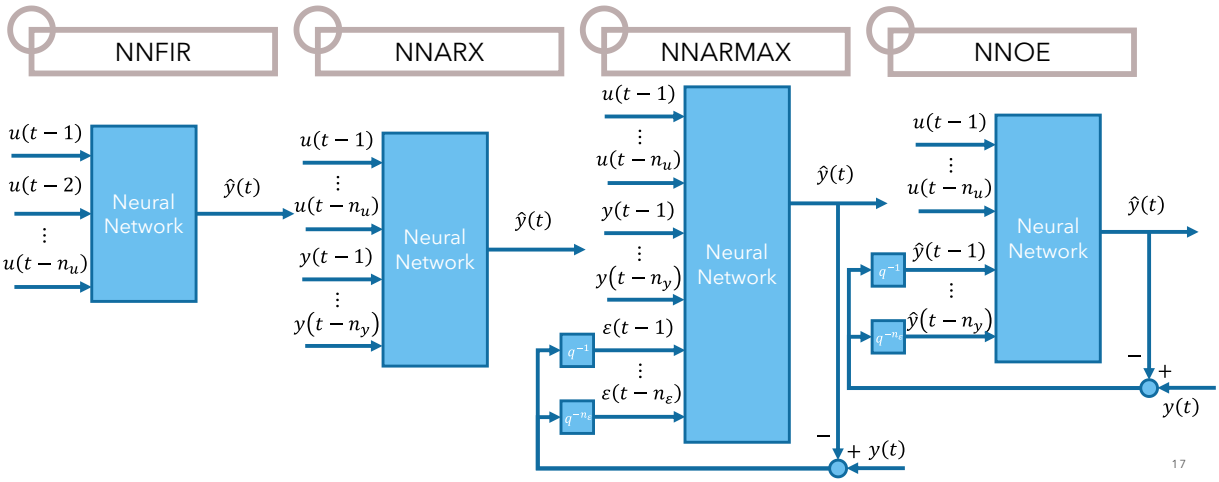


Figure 4: Overview of common neural network-based system identification structures, including NNFIR, NNARX, NNARMAX, and NNOE.

2.2.2 Recurrent Neural Networks (RNNs)

While feed-forward structures like NNARX are effective for many problems, they process each input vector independently. Recurrent Neural Networks (RNNs) are a class of architectures designed specifically for sequential data. Unlike feed-forward networks, RNNs explicitly account for the time history of a sequence, or its "memory," by using a recurrent connection or

feedback loop. This allows the network to maintain an internal state that captures information from past inputs in the sequence (Brunton & Kutz, 2022). A conceptual representation of an RNN, in both its compact (rolled) and sequential (unrolled) form, is shown in Figure 5.

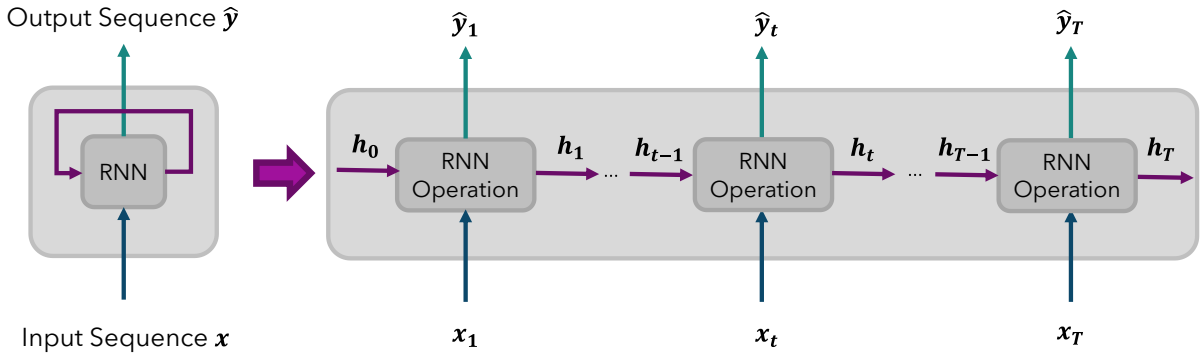


Figure 5: A rolled (left) and unrolled (right) representation of an RNN. The feedback loop allows the hidden state (h_t) to be passed from one time step to the next, enabling the network to learn temporal dependencies.

The structure of RNNs is conceptually linked to dynamical systems. A key difference from feed-forward networks is the training process: while a feed-forward network learns a mapping from a state at one time step to the next, an RNN is trained over entire trajectories or sequences. This training over sequences allows the history of the solution to shape the model, making RNNs highly suitable for time-series prediction. However, this recurrent nature introduces a significant challenge known as the vanishing gradient problem. When training with backpropagation through time, gradients can diminish exponentially as they are sent back through long sequences, making it difficult for the network to learn long-range dependencies. This limitation led to the development of more advanced architectures, such as LSTMs.

2.2.3 Long Short-Term Memory (LSTM) Networks

Long Short-Term Memory (LSTM) networks are a specialized type of RNN created to overcome the vanishing gradient problem and effectively learn long-term dependencies (Hochreiter & Schmidhuber, 1997). The key innovation in LSTMs is a cell architecture featuring a dedicated memory cell and a series of "gates" to regulate the flow of information. The structure of an LSTM operation, showing the cell state and the forget, input, and output gates, is depicted in Figure 6.

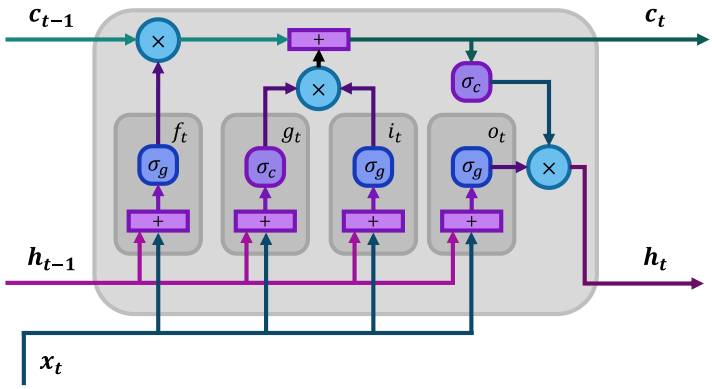


Figure 6: Detailed architecture of a Long Short-Term Memory (LSTM) unit, illustrating the flow of information through the forget (f_t), input (i_t), and output (o_t) gates.

The core components of an LSTM cell are:

1. **Cell State (c_t):** This acts as the long-term memory of the network. Information can be added to or removed from the cell state, which flows through the entire sequence with only minor linear interactions. This is essential for preserving the gradient over long time periods.

2. **Hidden State (h_t):** This acts as the **short-term memory**, carrying information from one step to the next within the input sequence. It is the **output of the LSTM cell at each time step**, representing a filtered version of the cell state relevant for that specific moment. While it is calculated at every step, its use for the final prediction depends on the network architecture. In sequence-to-one configurations, such as the one used in this work, typically only the hidden state from the **final time step** is used to generate the model's ultimate prediction for the entire sequence.

3. **Candidate for cell state update (g_t):** This component generates a vector of new candidate values based on the current input (x_t) and the previous hidden state (h_{t-1}). It represents the **potential new information** that could be added to the long-term memory.

4. **Gates:** These are neural network layers (typically with a sigmoid activation function, σ_g) that control the flow of information.

    - **Forget Gate (f_t):** Decides what information from the previous cell state (c_{t-1}) should be discarded.
    - **Input Gate (i_t):** Determines what new information from the candidate update (g_t) should be stored in the cell state.
    - **Output Gate (o_t):** Controls what information from the current cell state is passed on to the next hidden state (h_t).

The governing equations for the operations within an LSTM unit at each time step t are given by the gate activations and the state updates. First, the activations for the input gate (i_t), forget gate (f_t), candidate cell update (g_t), and output gate (o_t) are computed:

$$i_t = \sigma_g(W_i x_t + R_i h_{t-1} + b_i) \quad (9)$$

$$f_i = \sigma_g(W_f x_t + R_f h_{t-1} + b_f) \quad (10)$$

$$g_t = \sigma_c(W_g x_t + R_g h_{t-1} + b_g) \quad (11)$$

$$o_t = \sigma_g(W_o x_t + R_o h_{t-1} + b_0) \quad (12)$$

where x_t is the input, h_{t-1} is the previous hidden state, W,R,b are the respective weight matrices and bias vectors, σ_g is the sigmoid activation, and σ_c is the hyperbolic tangent (tanh) activation.

Next, the cell state (c_t) and hidden state (h_t) are updated based on these gates:

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \quad (13)$$

$$h_t = o_t \odot \sigma_c(c_t) \quad (14)$$

where $\odot$ denotes element-wise multiplication. By using this gating mechanism, LSTMs can selectively remember or forget information over long sequences, making them highly effective for modeling complex dynamic systems with long-term dependencies.

2.2.4 Theoretical Justification for the Modeling Approach

The preceding sections established the practical challenges of mechanistic modeling and introduced the principles of data-driven alternatives. This section synthesizes these concepts to provide a unified theoretical justification for the modeling approach adopted in this work, which is built on three key theoretical pillars.

The Role of System Identification

System Identification provides the overarching engineering discipline for building mathematical models from observed input-output data (Ljung, 1999). As formalized in the work of Ljung, this field validates the use of nonlinear black-box methods like neural networks, offering a rigorous methodological framework for this research. Adopting this approach aligns this work with established practice in modern control engineering.

Representational Power via the Universal Approximation Theorem

The choice of a neural network as the core modeling tool is supported by the Universal Approximation Theorem. This theorem states that a neural network with a single hidden layer can approximate any continuous nonlinear function to an arbitrary degree of accuracy (Cybenko, 1989; Hornik et al., 1989). For this thesis, the theorem provides the theoretical support that a

neural network possesses the necessary representational power to capture the complex, nonlinear input-output dynamics of the anaerobic digestion process.

State Reconstruction from Time-Series Data

While the Universal Approximation Theorem guarantees that a neural network can represent complex functions, it does not specify what the inputs to that function should be. A key challenge in modeling dynamical systems is that the complete internal state vector is often unmeasurable. Instead, we typically only have access to a time series of a limited number of scalar outputs. This raises the question of how to reconstruct the system's underlying dynamics from this partial information.

The theoretical foundation for this problem is provided by Takens' Embedding Theorem (Takens, 1981), which was later extended to non-autonomous, or 'forced,' systems (Stark et al., 1997). This extension establishes that a system's state space can, in principle, be reconstructed from a history of not only the observed outputs but also the exogenous inputs. This directly informs the functional form of a Nonlinear Autoregressive with eXogenous inputs (NARX) model (Petré, 2025).

The choice of a NARX-like structure is also consistent with key principles from system identification theory for building reliable predictors (Ljung, 1999). The structure meets several important criteria: it is causal, as it predicts the future output $y(t)$ using only past data; it uses only measured quantities (past inputs and outputs) as regressors, a standard practice that avoids the complexities of models that rely on estimated errors, such as NARMAX; and it supports the development of a stable predictor, a formal requirement for any valid model structure. These principles of causality and stability are practical prerequisites for any model intended for use within a control framework like MPC.

However, it is critical to interpret this theoretical result with caution. The theorem demonstrates the possibility of state reconstruction, but it does not guarantee that a neural network trained by simply minimizing a standard loss function will actually converge to a faithful representation of the true system dynamics. Instead, the engineering hope is that by structuring the model's input as a time-delay embedding (a history of past inputs and outputs), we are biasing the optimization process. The expectation is that this structure guides the network to learn a useful and stable approximation of the underlying dynamics, even without a formal guarantee of convergence to the true attractor.

The theoretical possibility of state reconstruction has inspired different data-driven modeling strategies. One ambitious direction seeks to make both the transformation and the underlying dynamics explicit. For instance, the work of Bakarji et al. (2023) uses a deep autoencoder not only to reconstruct the signal but also to explicitly discover a latent coordinate system where the system's dynamics can be described by a simple, sparse set of differential equations. This represents a direct attempt to find the "governing equations" from partial data. In contrast, the

approach taken in this thesis does not aim to explicitly identify such a coordinate system or its governing laws. Instead, it relies on a more implicit application of the embedding principle. The objective is to train a predictive model (an LSTM network) that learns a functional, black-box representation of the dynamics directly from the time-delay embedding of inputs and outputs. The goal is not to uncover the underlying attractor, but rather to create a sufficiently accurate input-output model that can be effectively used for the engineering task of model predictive control.

Taken together, these principles form a cohesive theoretical foundation for the modeling approach in this thesis. System Identification validates the data-driven methodology. The Universal Approximation Theorem affirms that a neural network has the necessary flexibility. Finally, the extension of Takens' Embedding Theorem provides a strong justification for the chosen NARX architecture, suggesting it is a sound approach for attempting to model the system's behavior from its input-output history for a specific engineering purpose, without claiming to uncover the fundamental governing laws of the system.

It is essential to clarify the "black-box" nature of the data-driven approach as applied in this work. This term refers specifically to the model's internal representation; the weights and biases of the neural network form a complex mathematical structure that is not directly interpretable in terms of physical or biochemical process parameters. However, this does not imply that the internal workings of the AD process should be ignored, nor does it preclude the integration of other available online measurements into the model if they were available. The central argument of this thesis is not to dismiss process knowledge, but to demonstrate that a predictive model built upon a carefully selected, minimal set of input-output variables can be sufficient for the engineering task of effective process control. The focus is on functional sufficiency for control, rather than a comprehensive mechanistic description.

2.3 Process Control Strategies

Achieving stable and efficient operation of complex processes like anaerobic digestion requires robust control strategies. These methods range from classical feedback controllers to advanced model-based techniques. This section provides an overview of the key control methodologies relevant to this work.

2.3.1 Model Predictive Control (MPC)

Model Predictive Control (MPC) is an advanced optimal control method that uses a dynamic model of a process to compute control actions by optimizing its future behavior (Camacho & Bordons, 2004). Its ability to handle multivariable systems, non-linear dynamics, and operational constraints makes it a suitable strategy for complex processes such as anaerobic digestion.

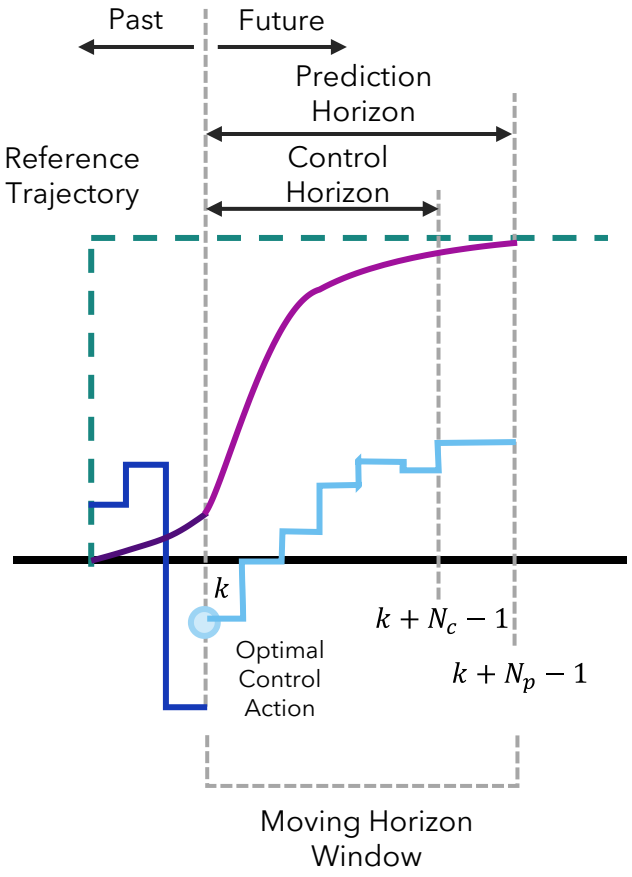


Figure 7: Conceptual illustration of the Receding Horizon Control principle. At each time step k, the controller predicts the system's future evolution (purple line) over a prediction horizon N_p by calculating an optimal sequence of future control actions (light blue steps) over a control horizon N_c to follow a reference trajectory (green dashed line).

The algorithm is based on a strategy known as Receding Horizon Control (RHC). At each sampling time k, the following sequence is executed:

1. **Prediction:** A dynamic model, in this case the LSTM network, is used to predict the future process outputs, $y_{k+i|k}$, over a defined prediction horizon of N_p steps. This prediction is based on the current state of the system and a proposed sequence of future control actions.
2. **Optimization:** An open-loop optimal control problem is solved to find the best sequence of future control actions. This sequence, denoted by the vector $U_k = \{u_{k|k}, \dots, u_{k+N_c-1|k}\}$, is the one that minimizes the following cost function, J_k, over a control horizon of N_c steps:

$$\min_{U_k}\left\{J_k = \sum_{i=1}^{N_p} w_y\left(\hat{y}_{k+i|k} - y_{ref,k+i}\right)^2 + \sum_{i=0}^{N_c-1} w_u\left(\Delta u_{k+i|k}\right)^2\right\} \quad (15)$$

The optimization is performed subject to hard constraints on both the magnitude of the manipulated variable (u) and its rate of change (Δu):

- $u_{min} \leq u_{k+i|k} \leq u_{max}$
- $\Delta u_{min} \leq \Delta u_{k+i|k} \leq \Delta u_{max}$

For the mono-digestion studies (AM2 and ADM1), the rate of change was limited to 20% per step. In the multi-input ACoD case, this constraint was tightened to 10% of the total influent flow rate to promote greater process stability.

3. **Implementation:** The open-loop solution is then implemented in a closed-loop fashion. Only the first control action of the optimal sequence, $u_{k|k}$, is applied to the plant. At the next sampling instant, $k+1$, new process measurements are taken, the prediction horizon shifts forward, and the optimization is solved again. This receding horizon strategy provides feedback, allowing the controller to correct for model-plant mismatch and unmeasured disturbances.

The cost function balances two competing objectives through its quadratic terms. The first term penalizes the predicted tracking error between the output y and the reference y_{ref}, while the second penalizes control effort, represented by the change in control action, Δu. The non-negative weighting factors, w_y and w_u, are tuned to adjust the priority between these objectives.

2.3.2 Reinforcement Learning (RL) Control

Reinforcement Learning (RL) is a distinct machine learning paradigm where an agent learns to take actions within an environment to maximize a cumulative reward (Sutton & Barto, 2018). Unlike supervised learning, the agent is not told which actions to take; instead, it must discover which actions yield the most reward through trial and error. This learning process is formalized as a Markov Decision Process (MDP).

A key distinction within RL is between model-based and model-free approaches. This study focuses specifically on the model-free paradigm, where an agent learns a control policy directly through interaction with its environment, guided by reward signals, without requiring an explicit process model. This makes it a compelling alternative to model-based strategies like MPC, as it circumvents the need for a separate system identification step.

The agent-environment interaction loop is conceptually illustrated in Figure 8. At each time step, the agent observes the environment's state, selects an action based on its policy, and receives a reward and the new state from the environment in return.

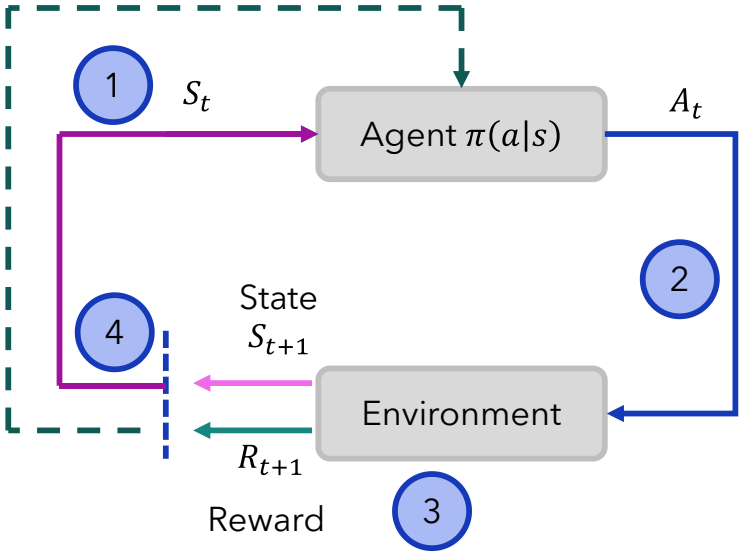


Figure 8: The agent-environment interaction loop in Reinforcement Learning. The cycle consists of four sequential steps: (1) The agent observes the current state of the environment, S_t. (2) Based on this state, the agent's policy, $\pi(a|s)$, selects an action, A_t. (3) The environment receives the action and transitions to a new state, S_{t+1}, providing a reward signal, R_{t+1}, as feedback. (4) The agent observes the new state, and the cycle repeats.

The core elements of the RL framework are:

- **The Agent:** The learner and decision-maker.
- **The Environment:** Everything outside the agent with which it interacts.
- **The State (S_t):** A representation of the environment at a given time step t.
- **The Action (A_t):** A choice made by the agent from a set of available actions.
- **The Reward (R_{t+1}):** A scalar feedback signal received from the environment after taking an action, which indicates the immediate desirability of that action.

The agent's objective is to learn a policy (π), which is a mapping from states to actions, that maximizes the expected return (G_t), defined as the cumulative sum of future discounted rewards:

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \tag{16}$$

where $\gamma \in [0,1]$ is the discount factor.

To achieve this goal, RL methods can be broadly categorized into two main families. Value-based methods, like Q-learning, focus on learning an optimal action-value function and then derive a policy from it. In contrast, policy-based methods directly learn a parameterized policy without needing a value function to select actions. This approach is particularly advantageous

in continuous action spaces, where finding the maximizing action for a value function can be computationally expensive.

The methods used in this thesis belong to the Policy Gradient family. Their core idea is to adjust policy parameters, θ, by performing stochastic gradient ascent on the expected return. Estimating the gradient of the policy's performance is a key challenge in this approach, which is a challenge that is addressed by Actor-Critic architectures.

In an actor-critic architecture, two components are learned simultaneously:

- **The Actor:** This is the policy, $\pi_\theta(a|s)$, which is responsible for selecting actions.
- **The Critic:** This is a value function, typically an action-value function $q_w(s, a)$, which estimates the value of the actions taken by the actor.

The learning process is driven by the interaction between these two components. When the actor takes an action, the critic evaluates it by computing a Temporal-Difference (TD) error. This error signal is then used to update both the critic's parameters, w_c, for better value estimates, and the actor's parameters, θ, to favor actions leading to better-than-expected outcomes. The specific algorithms evaluated in this work, such as DDPG, TD3, and SAC, are all advanced instances of this actor-critic framework.

2.4 Hyperparameter Optimization

The performance of complex models and controllers is highly sensitive to their hyperparameters. As manual tuning is often impractical for computationally intensive systems, a systematic approach is required.

For this task, Bayesian Optimization was selected. It is an efficient, model-based search algorithm designed to find the optimal set of hyperparameters with a minimal number of evaluations (Mockus J.B & Mockus L.J, 1991). The algorithm operates by building a statistical model of an objective function (e.g., controller tracking error) and uses this model to intelligently select the most promising parameters for the next iteration.

This methodology was implemented using the bayesopt function from the MATLAB Statistics and Machine Learning Toolbox™ (MathWorks, 2025) which provides a robust and straightforward workflow for defining the hyperparameter search space and running the optimization. In this thesis, Bayesian Optimization was used to systematically tune the cost function weights for the multi-input MPC controller and to find the optimal hyperparameters for the Reinforcement Learning agents This was done to ensure each strategy was configured with a high-performing set of parameters.

3 Materials and Methods

This chapter details the methodology for the foundational study presented in Chapter 4, covering the simulation environment, data generation, model and controller development, and performance evaluation metrics.

3.1 Simulation Environment

To rigorously develop and evaluate the data-driven control strategy, the anaerobic digestion (AD) process was emulated using two established mechanistic models of varying complexity. Serving as virtual "plants," these models provided a controlled and repeatable environment to generate dynamic data and test the performance of the predictive models and controllers under diverse and challenging conditions. The use of simulators is integral to this study, as it enables the exploration of operating scenarios that would be impractical or unsafe on a physical plant.

The two models chosen for this study were:

- **The Anaerobic Model No. 2 (AM2):** A simplified, control-oriented model used as the first process emulator (see Section 2.1.1). For this study, it was parameterized to represent the digestion of a readily biodegradable substrate. The simulated system was operated near its stability limit, which, for this parameterization, contains a sharp non-linearity at the washout threshold. While this region presents a modeling challenge for the data-driven predictor, the subsequent control task was designed to operate primarily within the process's more stable, quasi-linear range.
- **The Anaerobic Digestion Model No. 1 (ADM1):** A comprehensive, high-fidelity model used as the second process emulator (see Section 2.1.2). The ADM1 implementation was adapted from the generalized framework for co-digestion developed by García-Gen et al. (2013). The original model, built in MATLAB/Simulink, was modified for this work to operate purely as a MATLAB script, removing the Simulink dependency. This structure was used to simulate mono-digestion of municipal sewage sludge, using the parameters reported in García-Gen & Wouwer (2021), and multi-substrate co-digestion scenarios. Its complexity, with numerous states and interacting biochemical pathways, provided a more challenging environment to test the performance of the data-driven control strategy.

By testing the approach against both a simplified and a complex emulator, this study aims to validate the effectiveness of the data-driven strategy across different levels of process representation.

All simulations and modeling were conducted within the MATLAB environment. The development of neural network architectures (LSTM, GRU, Attention) utilized the Deep

Learning Toolbox™, while the Reinforcement Learning (RL) study used the Reinforcement Learning Toolbox™.

Computationally intensive tasks were performed on a high-performance workstation equipped with an AMD Threadripper 2950X CPU, 64 GB of RAM, and an NVIDIA RTX 2070 GPU. These tasks included the final, single-worker training of the RL agents, as well as processes accelerated by the Parallel Computing Toolbox™. The toolbox was specifically employed to:

- Run the Bayesian Optimization searches in parallel for both MPC controller tuning and RL hyperparameter discovery.
- Train replicate networks concurrently for the data quantity analysis.

Less demanding tasks, such as individual model training and the closed-loop control simulations, were conducted on a laptop with an AMD Ryzen 5600H CPU, 32 GB of RAM, and an NVIDIA RTX 3060 Laptop GPU.

3.2 Data Generation and Experimental Design

Training and validating the data-driven models required informative datasets that capture the process's dynamic behavior. Data was generated by simulating the response of each plant emulator (AM2 and ADM1) to a series of deliberate perturbations in their respective manipulated variables. The full datasets for the training (Dataset 1) and testing (Dataset 2) simulations of the AM2 model are detailed in Appendix A**.**

For the AM2 emulator, training data was generated by applying 6 step changes to the dilution rate (D) near washout. Each step lasted approximately 4.5 days, with data recorded at a sampling rate of about 1.75 hours.

For the more complex ADM1 emulator, training data was generated by applying 10 step changes to the inlet flow rate (Q_{in}) across a wide operational range. Each step was maintained for approximately 14 days, with a sampling rate of 6 hours.

In all simulation scenarios, multiple sources of noise were introduced to increase the realism of the dataset. First, to represent natural variability in the influent, an AutoRegressive process of order 1 (AR(1)) was applied to the substrate composition. This process was defined by an autocorrelation coefficient of $\phi = 0.95$ and a white noise standard deviation (σ) set to 1.0% for AM2, 2.5% for ADM1, and 1.5% for ACoD. The dynamic impact of these stochastic influent variations is illustrated in Figure 9 and Figure 10. Second, to simulate sensor measurement uncertainty, Gaussian white noise was added directly to the methane flow rate output, with a standard deviation corresponding to 5% of the variable's operational range.

It is important to clarify how the different operational ranges were defined. First, a range for the manipulated variable (e.g., dilution rate) was established, and the training data was generated

by applying random step changes within these limits. The minimum and maximum values of the methane flow rate that resulted from this simulation defined the range used for all subsequent calculations. Specifically, this range from the training data was the sole basis for the min-max normalization parameters, which scaled the data to a range of [0, 1], and for defining all relative values, such as noise levels and controller constraints, to prevent any data leakage. For the control evaluation tasks, the setpoints were selected based on the known steady-state production curve calculated from each process model. This range was typically narrower than the one observed during the dynamic training simulation, as the stochastic disturbances could occasionally push the output beyond its theoretical steady-state limits.

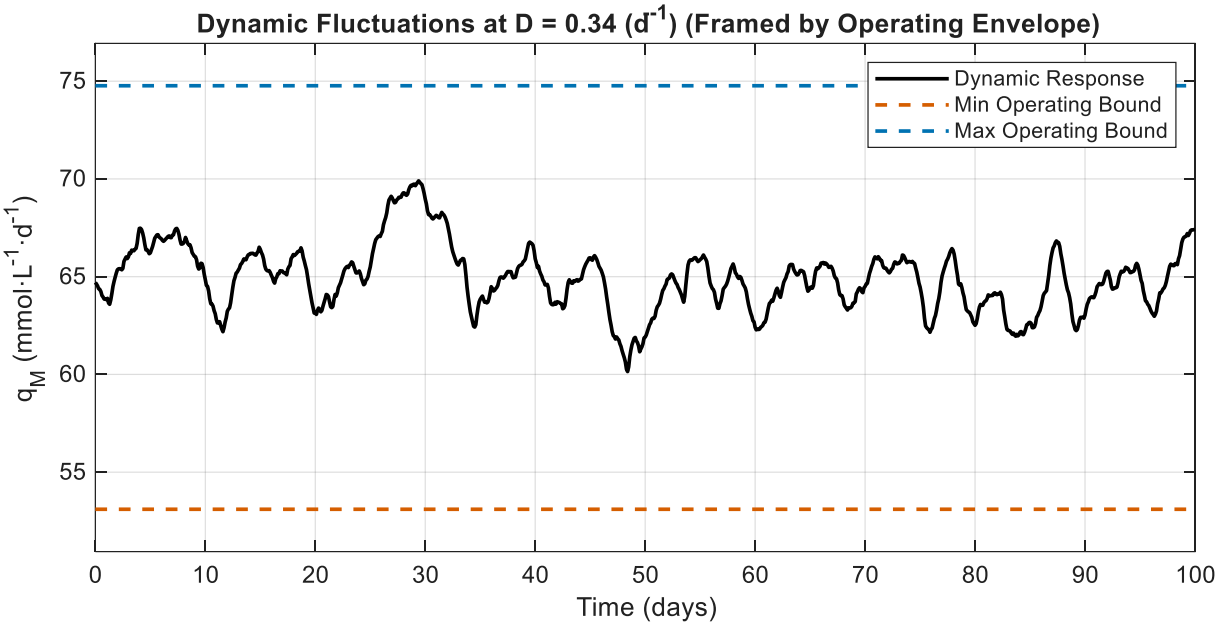


Figure 9: Example of dynamic fluctuations in methane flow rate (q_M) at a constant dilution rate ($D = 0.34\ d^{-1}$) due to stochastic variations in the influent composition.

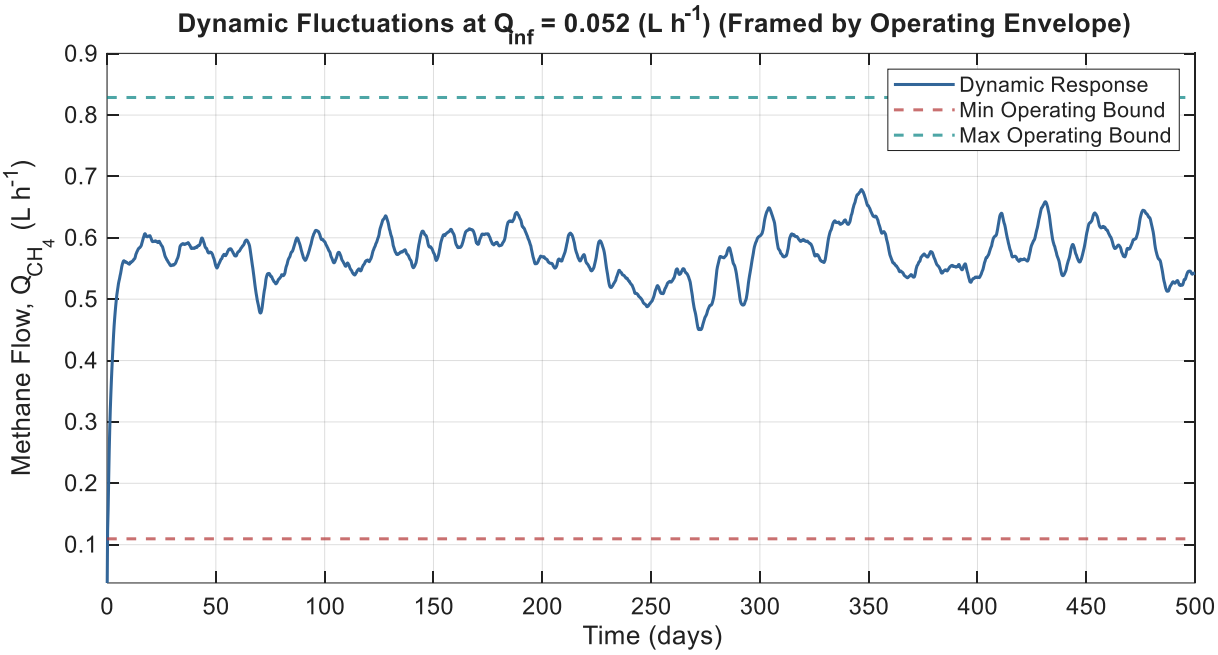


Figure 10: Example of dynamic fluctuations in methane flow rate (Q_{CH_4}) for the ADM1 model at a constant influent flow rate due to stochastic variations in the influent.

3.3 Development of the Data-Driven Control System

The control strategy integrates a data-driven LSTM predictor within an MPC framework. The specifics of each component are detailed below.

3.3.1 LSTM Predictor

The configuration of the LSTM predictor was guided by a sensitivity analysis performed in the author's undergraduate work (Pino Santana, 2025). That study systematically explored how various hyperparameters influence prediction accuracy in a simpler nonlinear system, providing the guidelines that informed the architectural decisions in this thesis. This allowed the research to focus directly on the control application.

Specifically, that undergraduate research explored the effect of data quantity, observing a pattern of diminishing returns where the model's performance stabilized after approximately 6 to 8 dynamic perturbations. This finding was key to designing an efficient data generation campaign. The analysis also highlighted the importance of the sampling rate and the size of the historical data window. It was determined that a sampling rate that was too high could degrade long-term forecasting by reducing the effective time window, while a moderate rate, combined with a sufficient window of past data (e.g., size 10), offered a better balance for capturing the system's dynamics without overfitting to noise. Finally, the study showed that high network complexity was not necessary; a single LSTM layer with a modest number of hidden units (between 15 and

20) was sufficient to model complex dynamics without introducing unnecessary computational load or instability.

Adopting these findings, the predictor model was designed to reconstruct the system's future behavior from a history of its inputs and outputs. Specifically, the model forecasts the next output, $\hat{y}_t$, using a "window" of w past values as its input. This window contains the history of both the process output (methane flow rate, y) and the manipulated variable (u).

This functional relationship is represented by the following general equation:

$$\hat{y}_t = f(\{u_{t-1}, \ldots, u_{t-w}\}, \{y_{t-1}, \ldots, y_{t-w}\}) \quad (17)$$

This structure, which uses past outputs (the Autoregressive component) and past external inputs (the eXogenous inputs component) to predict the future, is formally known in system identification literature as a NARX (Nonlinear Autoregressive with eXogenous inputs) model.

The predictor can operate in two forecasting modes. The first, one-step-ahead prediction, uses the measured process history up to time t to predict the output at $t + 1$, evaluating the model's immediate accuracy. The second, more challenging mode is multi-step-ahead prediction. Here, the model iteratively uses its own predictions as inputs to forecast further into the future. This method is crucial for testing the model's long-term stability and its ability to generate the future trajectories required by the MPC.

The nonlinear mapping function $f(\cdot)$ is approximated by a neural network constructed with a single LSTM layer. An LSTM approximates this function by processing the input sequence step-by-step. At each point t within the historical window, the network cell updates its two internal states, the cell state (c_t) and the hidden state (h_t), based on the previous states and the current inputs from the sequence:

$$[h_t, c_t] = LSTM_{Operation}(y_{t-1}, u_{t-1}, h_{t-1}, c_{t-1}) \quad (18)$$

The final prediction is then generated from the final hidden state, h_t, after the entire window has been processed. This is accomplished through an affine transformation performed by a final fully connected layer:

$$\hat{y}_t = g(h_t) \quad (19)$$

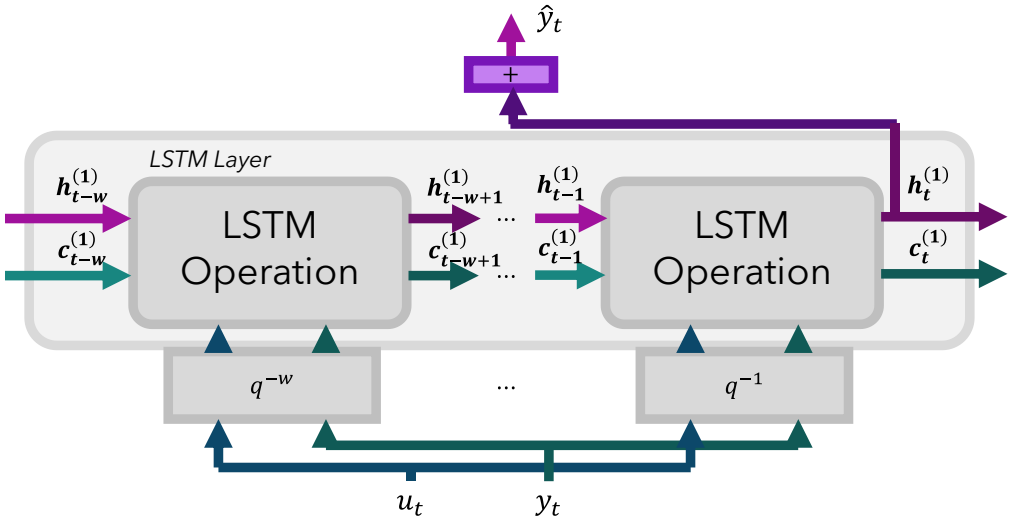


Figure 11: Unrolled representation of the LSTM layer implementing the NARX structure. The model processes a window of past manipulated variables (u) and process outputs (y) to predict the next output, $\hat{y}_t$.

An independent predictor was trained for each of the process emulators:

- For the AM2 model, the network used a window size (w) of 10 past data points and contained 15 hidden units in its LSTM layer.
- For the ADM1 model, a window size of 10 and 20 hidden units were used.

A key design choice was the model's stateless implementation, which directly aligns with the NARX framework established in the theoretical justification. In this configuration, the network's internal states (cell state c_t and hidden state h_t) are reset to zero before processing each new input window. Consequently, memory is maintained only within this fixed-size window, ensuring that each prediction is based solely on the temporal patterns derived from that specific historical sequence.

The network was trained using the L-BFGS optimizer, which operates in full-batch mode, for a maximum of 1000 epochs, with convergence defined by a gradient and step tolerance of 10^{-5}. To prevent overfitting, two strategies were employed: an L2 regularization factor of 10^{-4} was applied, and an early stopping criterion was implemented. The validation loss was monitored every 5 epochs, and training would halt if no improvement occurred over a patience of 20 checks. The final model selected was the one corresponding to the epoch with the lowest validation loss. The gradient threshold was disabled. The training data was the sole basis for determining the min-max normalization parameters used to scale all data to the range [0, 1], a practice that ensured no information from the validation or test sets could leak into the training process.

3.3.2 NMPC Controller

The control strategy implements the Model Predictive Control (MPC) scheme detailed in Section 2.3.1, utilizing the data-driven Long Short-Term Memory (LSTM) network as its internal predictive model. The overall closed-loop architecture, depicted in Figure 12, consists of three main components:

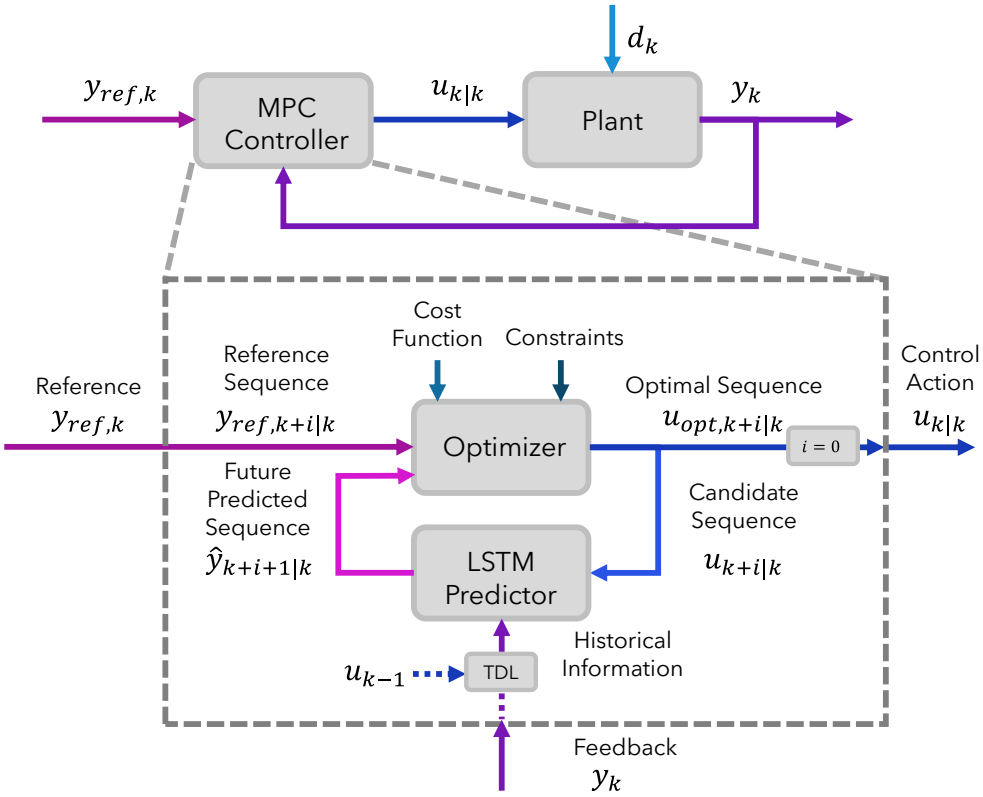


Figure 12: Schematic of the proposed LSTM-MPC strategy, illustrating the flow of information at a discrete time step k.

- **The Plant:** Represents the AD process, emulated by the AM2 and ADM1 models.
- **The LSTM Predictor:** The data-driven model described in the previous section, used to forecast the plant's future output.
- **The NMPC Controller:** The core decision-making component, which uses the LSTM's predictions to solve an online optimization problem.

The controller's practical operation hinges on a precise flow of information at each sampling instant, k. It is fundamental to clarify the temporal relationship between measurement and action: the control action calculated at step k, denoted $u_{k|k}$, does not affect the measurement y_k. Instead, y_k is the measured outcome of the previous action, u_{k-1}.

At the start of each cycle, the controller receives the measurement y_k. This signal, along with the stored history of past control actions (ending with u_{k-1}), is processed by the TDL block shown in Figure 12 to construct the full historical window that serves as context for the prediction. This real-time information anchors the predictive simulation. A key aspect of this implementation is how the LSTM predictor is utilized: to forecast the first future state, $\hat{y}_{k+1|k}$, the model uses the real measurement y_k alongside the first hypothetical action $u_{k|k}$ from the optimizer's candidate sequence. All subsequent predictions within the horizon (e.g., $\hat{y}_{k+2|k}$) are made recursively, using the model's own previous predictions as input. This "what-if" simulation allows the optimizer to evaluate the consequences of its potential decisions based on the most current and accurate information from the plant.

The controller tuning focused on the input weight (w_u), which balances the competing objectives of tracking accuracy and control effort. This parameter was selected for each emulator through a dedicated sensitivity analysis. Other controller parameters were held constant, with the prediction (N_p) and control (N_c) horizons set to 20 steps and the output weight (w_y) set to a value of 1.

The non-linear MPC schemes were implemented using the fmincon solver from MATLAB's Optimization Toolbox™, configured with the Sequential Quadratic Programming (SQP) algorithm. To ensure the optimizer could converge to a solution within a practical timeframe at each step, a maximum of 50 iterations was set to limit the search duration, while the tolerances for optimality, step size, and constraints were all defined as 10^{-6} to establish the stopping criteria. For the linear ARX-MPC, the more computationally efficient quadprog solver was used.

3.4 Evaluation Metrics

The performance of the predictive models was assessed using a set of standard statistical metrics. These metrics served both to guide the training process and to evaluate the final model accuracy on unseen data.

- **Mean Squared Error (MSE):** Used as the loss function during training. For N observations, it is defined as: $MSE = \frac{1}{N}\sum_{t=1}^{N}\left(y(t) - \hat{y}(t)\right)^2$

  Minimizing the MSE is a standard approach in regression problems, as it penalizes larger errors more heavily.

- **Root Mean Squared Error (RMSE):** The primary metric for evaluating prediction accuracy, especially for multi-step-ahead forecasts. It is the square root of the MSE. $RMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N}\left(y(t) - \hat{y}(t)\right)^2}$

- **Mean Absolute Error (MAE):** The MAE was also used as a complementary benchmark metric. It measures the average of the absolute errors and is less sensitive to large outliers compared to MSE and RMSE. $MAE = \frac{1}{N}\sum_{t=1}^{N}|y(t) - \hat{y}(t)|$
- **Mean Absolute Percentage Error (MAPE):** This metric expresses the mean absolute error as a percentage of the actual values, providing a relative measure of error that is independent of the data's scale. $MAPE = \frac{100\%}{N}\sum_{t=1}^{N}\left|\frac{y(t)-\hat{y}(t)}{y(t)}\right|$
- **Normalized Root Mean Squared Error (NRMSE):** This metric normalizes the RMSE by the mean of the observed data, which facilitates the comparison of model performance across datasets with different scales. $NRMSE = \frac{RMSE}{\bar{y}}$

4 Predictive Control of Anaerobic Mono-digestion

This chapter details the initial evaluation of the data-driven Nonlinear Model Predictive Control (NMPC) strategy, which utilizes a Long Short-Term Memory (LSTM) network as its internal predictive model. The analysis is conducted within a simulated environment where the anaerobic mono-digestion process is represented by two established mechanistic models of increasing complexity: the simplified AM2 and the comprehensive ADM1. These models serve as process emulators to test the framework's performance progressively. The evaluation begins with the AM2 emulator. In the configuration used for this study, this model is characterized by a wide, quasi-linear operating range followed by a sharp non-linearity near its washout threshold. The data-driven predictor is evaluated across this range, while the subsequent control task is designed to operate primarily within the model's more stable region. Subsequently, the framework is applied to the high-fidelity ADM1 emulator to evaluate its performance against a more complex process representation.

Part I: Analysis for the AM2 Emulator

4.1 Predictor Development and Performance

The initial evaluation of the data-driven approach consisted of developing a predictive model for the AM2 emulator. Following the methodology described in Chapter 3, an LSTM network was trained on a dataset generated from the AM2 simulation. The process was deliberately operated in its highly non-linear region to test the network's learning capabilities.

The training progress, illustrated in Figure 13, shows a rapid decrease in Mean Squared Error (MSE) for both the training and validation sets before converging to a low, stable value. The convergence of both curves to a low, stable value suggests that the network learned the process dynamics from the training data without significant overfitting.

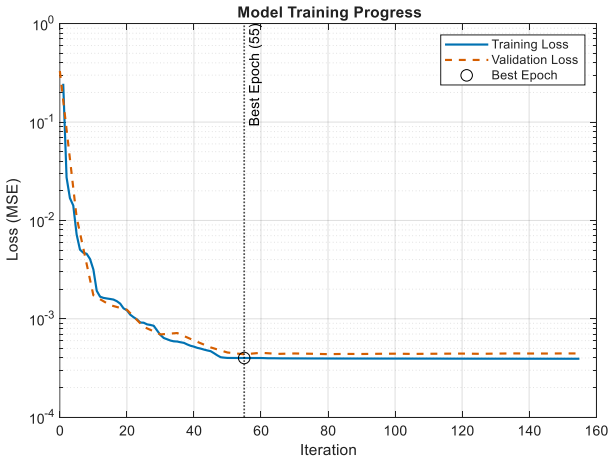


Figure 13: Training and validation loss curves for the LSTM predictor developed for the AM2 emulator.

To evaluate the predictor's long-range accuracy, the growth of the prediction error over an extended forecast horizon was analyzed (Figure 14). While the Root Mean Squared Error (RMSE) increases with the horizon's length, the error stabilizes and remains bounded, indicating that the model's predictions do not diverge over the tested horizon. This can be considered a favorable characteristic for a model intended for use in predictive control.

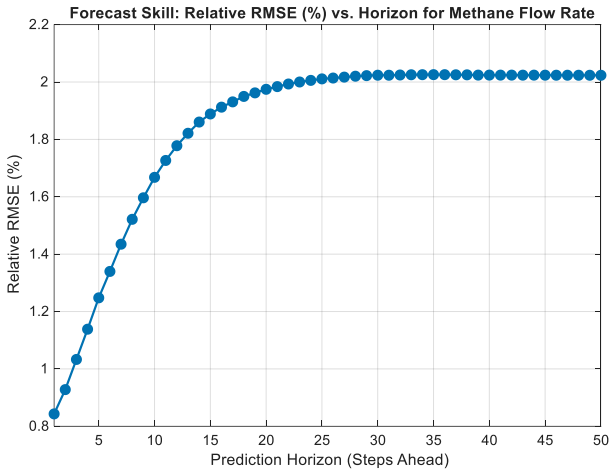


Figure 14: Multi-step prediction error (RMSE) as a function of the prediction horizon for the AM2 predictor.

Finally, the predictor's performance on unseen test data was evaluated in different forecasting modes (Figure 15). The one-step-ahead predictions closely track the ground truth, demonstrating high accuracy. The segmented multi-step prediction, which simulates the predictor's function within the MPC's receding horizon, also generates reliable short-term forecasts. Notably, the multi-step forecast provides a smoothed version of the process dynamics by effectively filtering out stochastic noise and capturing the underlying deterministic behavior. This provides a stable basis for the controller's optimization step.

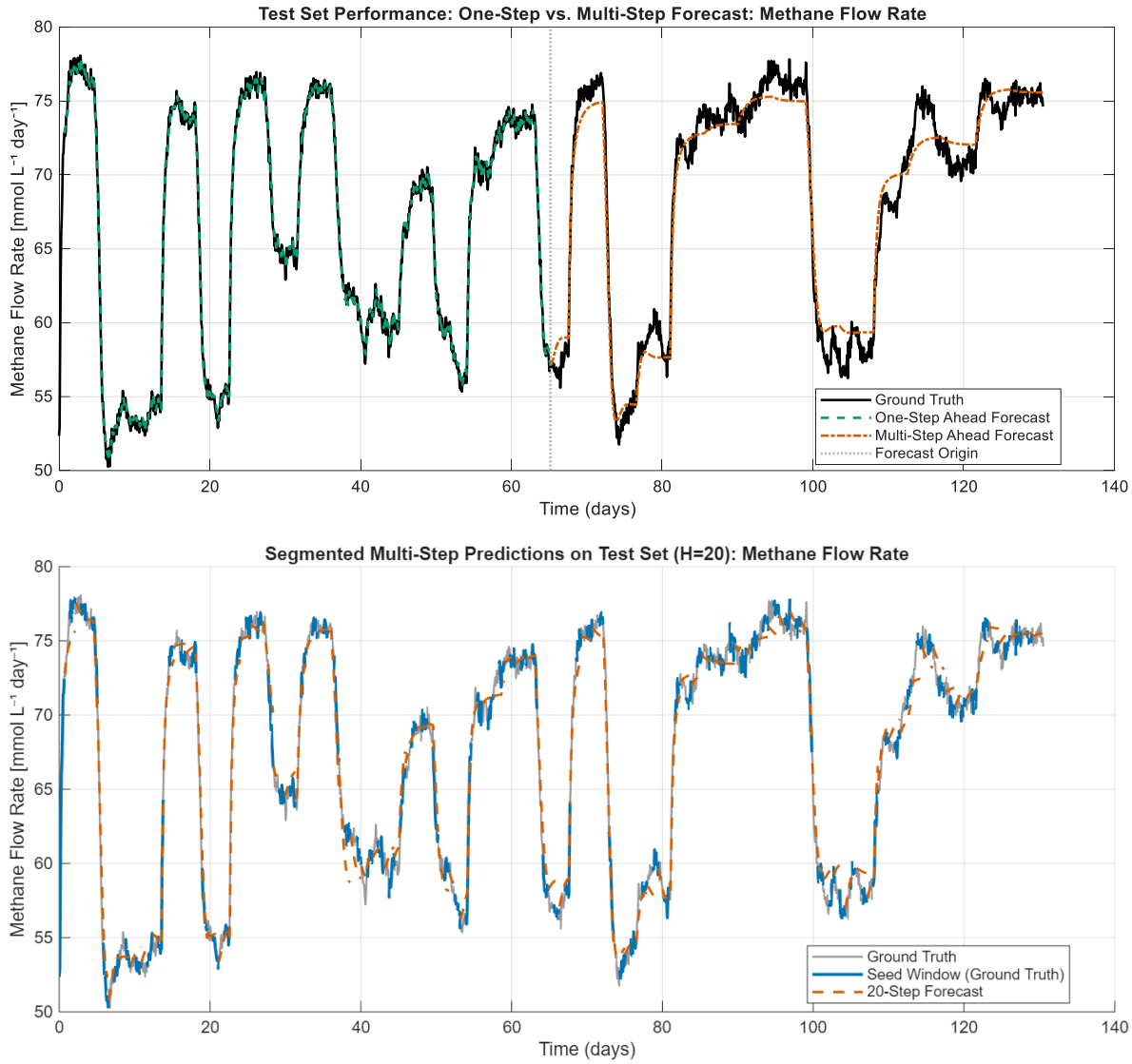


Figure 15: Test performance showing combined one-step and segmented multi-step predictions for the AM2 predictor.

Beyond its dynamic capabilities, a critical test is whether the LSTM model learned the process's fundamental steady-state behavior. Figure 16 compares the steady-state methane flow predicted by the LSTM against the true curve of the AM2 model. The analysis shows that while the model accurately captures the process's quasi-linear behavior, it fails to represent the sharp non-linearity near the washout threshold. In this critical region, the LSTM model over-predicts the achievable methane production, incorrectly suggesting process stability at dilution rates where the system would, in reality, collapse. This model-plant mismatch represents a significant limitation and a potential operational risk. For instance, if the MPC controller, relying on this inaccurate model, were tasked with reaching a setpoint beyond the stable peak (e.g., 76 mmol L^{-1} day^{-1}), its predictions would be overly optimistic, and it could inadvertently steer the real

process toward washout. This finding is fundamental for the correct interpretation of the upcoming closed-loop results.

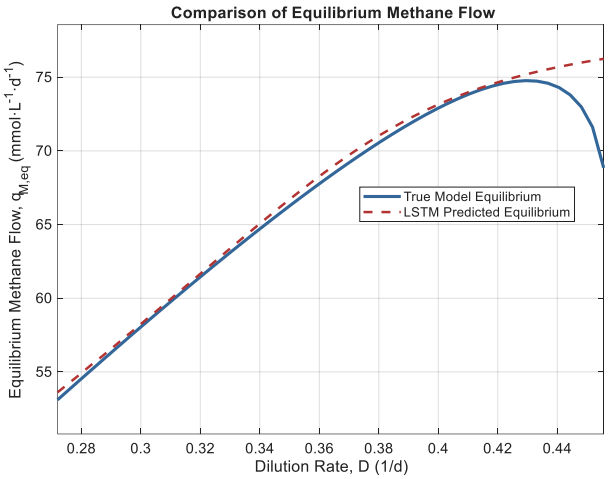


Figure 16: Comparison of the true steady-state methane flow of the AM2 model versus the equilibrium predicted by the trained LSTM network.

4.2 NMPC Controller Tuning

Following the evaluation of the predictive model, the NMPC's control effort weight (w_u) was tuned. This parameter, which penalizes changes in the manipulated variable, governs the trade-off between precise setpoint tracking and the smoothness of the control action. To determine an effective value for the AM2 emulator, a sensitivity analysis was conducted by performing a series of closed-loop simulations across a range of w_u values. As shown in Figure 17, the analysis revealed a clear minimum in the tracking RMSE at $w_u = 10^4$.. Consequently, this value was selected for the final closed-loop performance evaluation.

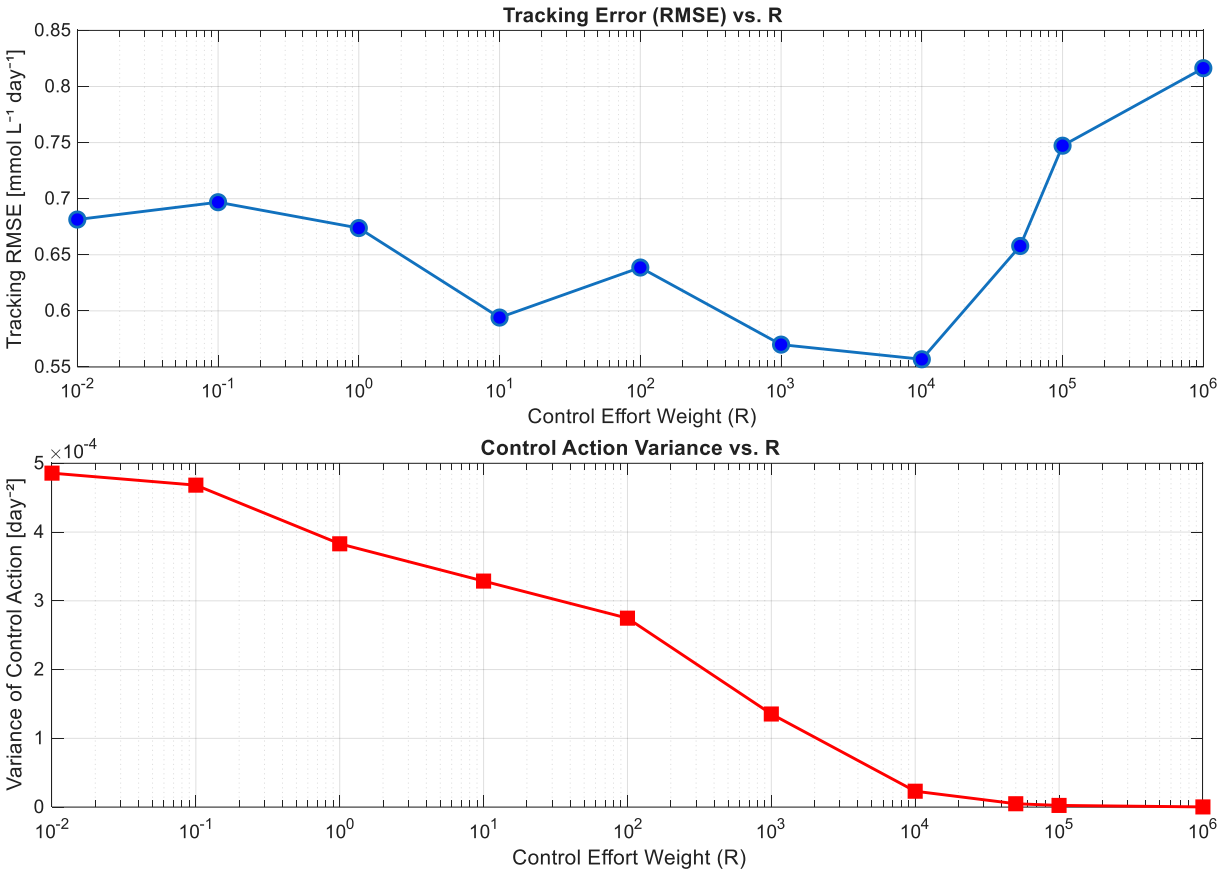


Figure 17: NMPC tuning analysis for the AM2 emulator, illustrating the trade-off between tracking error (RMSE) and control action variance as a function of the control effort weight (w_u). The analysis was performed at a fixed setpoint at 95% of the range.

4.3 Closed-Loop Control Results

With the predictor assessed and the controller tuned, the integrated LSTM-NMPC system was tested in a closed-loop simulation, tracking a series of step changes in the methane flow rate setpoint (Figure 18). The controller tracks the setpoints, reaching the new reference values with a short transient period. The control action (dilution rate, D) adjusts to meet the performance objectives while respecting operational constraints. Minor oscillations are present, particularly after setpoint changes, but remain bounded. The controller maintains stability and demonstrates robustness against the unmeasured stochastic disturbances present in the simulation.

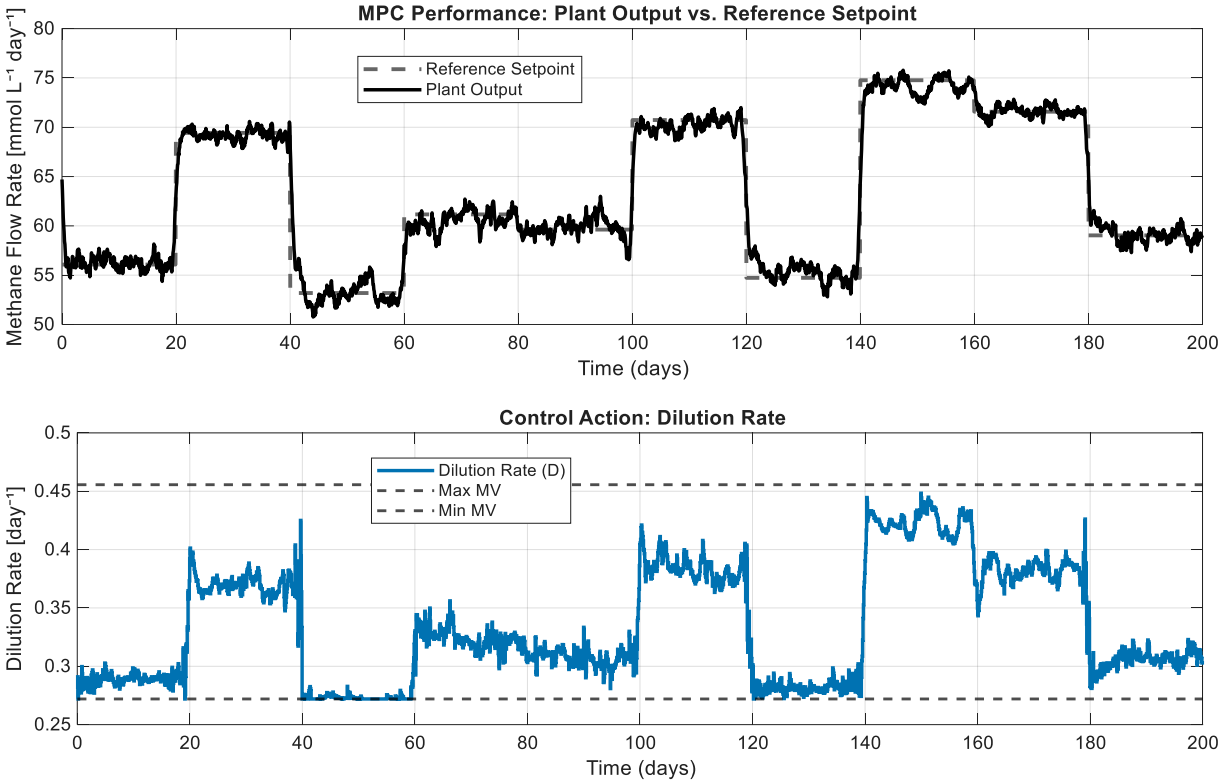


Figure 18: Closed-loop performance of the LSTM-NMPC on the AM2 emulator, showing setpoint tracking (top) and the corresponding control action (bottom).

4.4 Discussion of the AM2 Case

Having presented the open-loop performance of the predictive model and the closed-loop results of the controller, this section provides a critical discussion of the findings for the AM2 case. The objective is to interpret the controller's performance in the context of the identified model limitations and to draw initial conclusions about the framework's viability.

The closed-loop results indicate that the controller achieved accurate setpoint tracking, which supports the viability of the LSTM-NMPC strategy. However, this performance must be interpreted in the context of the model-plant mismatch identified near the washout threshold. The controller's effectiveness was contingent on the selected setpoints residing within the quasi-linear region where the LSTM predictor is accurate. Had a setpoint been chosen beyond the stable peak (e.g., 76 mmol L^{-1} day^{-1}), where the model's predictions are unreliable, the controller would likely have failed. Consequently, the control task was one of setpoint tracking within a known stable range, not washout avoidance. These results support the approach for this specific objective, while underscoring the importance of characterizing a data-driven model's domain of validity prior to its deployment.

The initial evaluation on the simplified AM2 model supported the framework's feasibility for setpoint tracking within a well-defined operating region. The next logical step, therefore, is to assess the methodology's robustness when applied to a more complex and high-fidelity process representation. This following section details the application of the same data-driven framework to the comprehensive ADM1 emulator.

Part II: Analysis for the ADM1 Emulator

4.5 Predictor Development and Performance

To assess the methodology's robustness, the same development process was applied to the high-fidelity ADM1 emulator, which involved training a separate LSTM network. The training progress for this predictor is shown in Figure 19. Similar to the AM2 case, the MSE for both training and validation sets decreases rapidly and converges, which suggests a properly converged training process without overfitting.

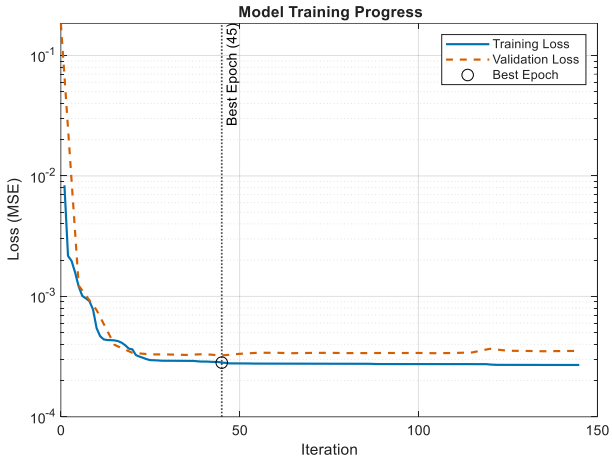


Figure 19: Training and validation loss curves for the LSTM predictor developed for the ADM1 emulator.

The long-range accuracy of the ADM1 predictor was also evaluated (Figure 20). The RMSE of the multi-step forecast increases with the prediction horizon but remains bounded. This observation of a bounded error is consistent with the results from the simpler AM2 emulator, suggesting the behavior is not exclusive to that specific model.

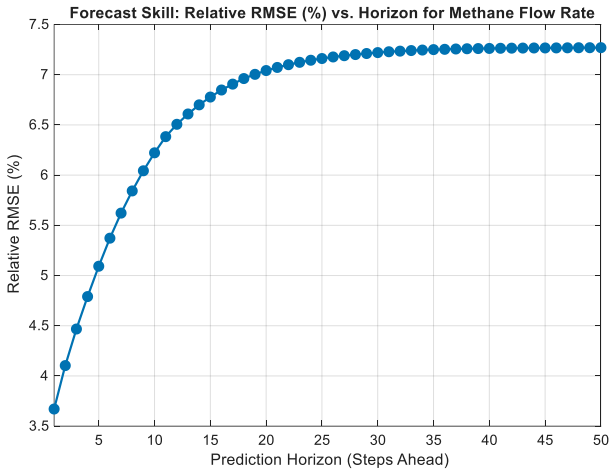
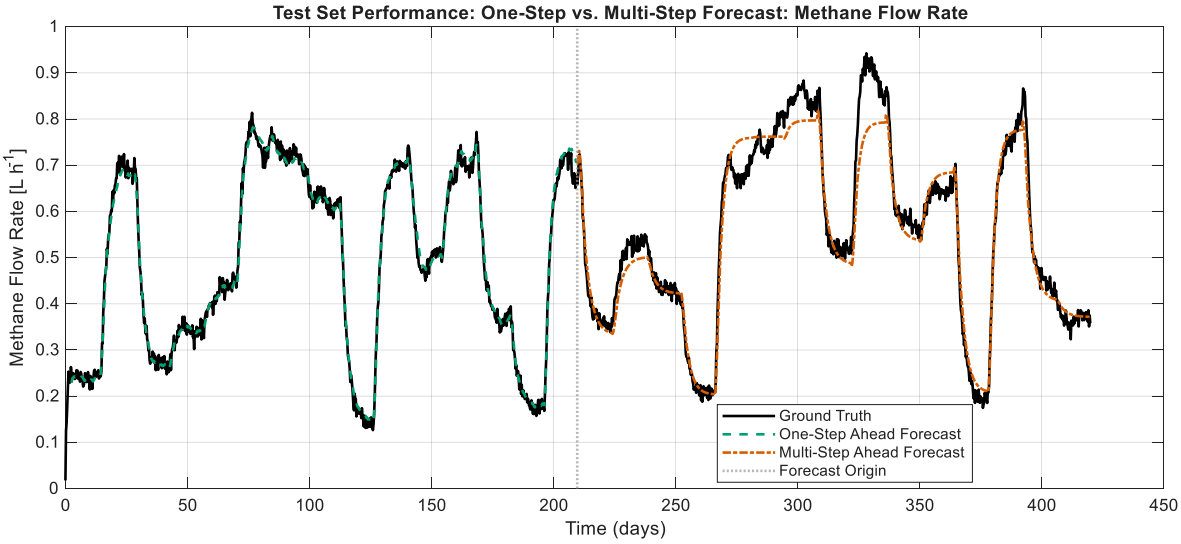


Figure 20: Multi-step prediction error (RMSE) as a function of the prediction horizon for the ADM1 predictor.

The evaluation of forecasting modes on unseen test data (Figure 21) further confirmed the predictor's reliability. Both one-step-ahead and segmented multi-step predictions align well with the ground truth. As with the AM2 case, the multi-step forecast learns a smoothed version of the process dynamics, capturing the underlying deterministic trend while filtering out stochastic noise. This result supports the model's suitability for the more complex ADM1 environment.



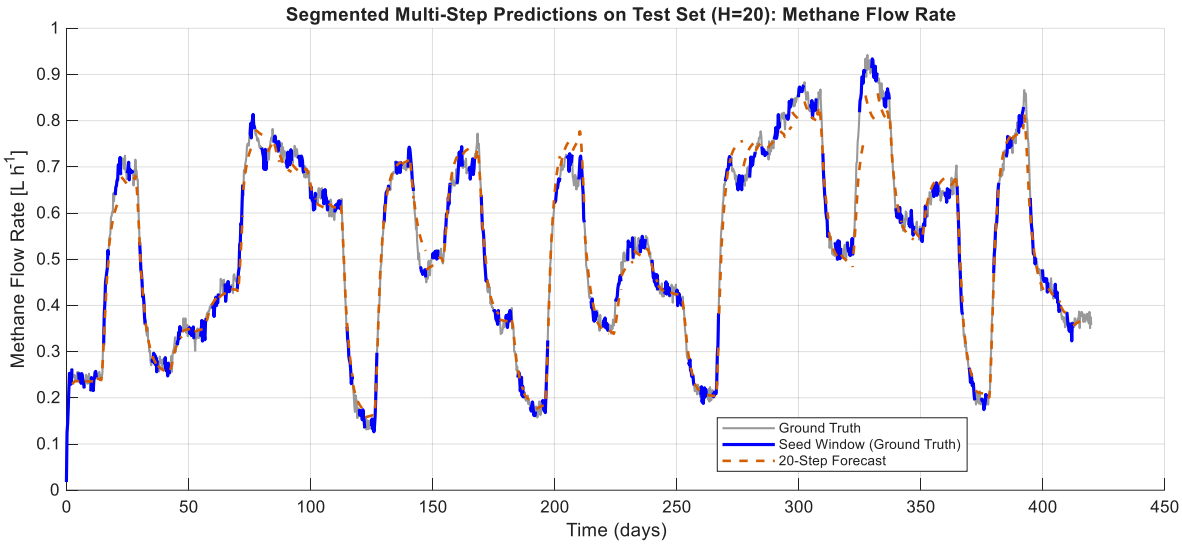


Figure 21: Test performance showing combined one-step and segmented multi-step predictions for the ADM1 predictor.

As a final assessment of the predictor's generalization capability, its representation of the ADM1's steady-state behavior was analyzed. Figure 22 compares the equilibrium methane flow predicted by the LSTM against the true equilibrium curve of the model. The data-driven model's prediction shows a close alignment with the true equilibrium across the entire operating range. This suggests that the LSTM network, trained exclusively on dynamic data, learned the complex, non-linear static mapping of the high-fidelity ADM1, which supports its suitability as a predictive model for control.

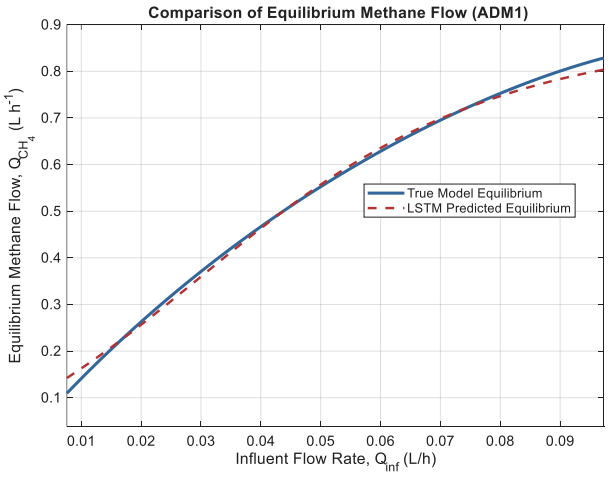


Figure 22: Comparison of the true steady-state methane flow of the ADM1 model versus the equilibrium predicted by the trained LSTM network.

4.6 NMPC Controller Tuning

The NMPC controller for the ADM1 emulator was also tuned by adjusting the control effort weight (w_u). Figure 23 illustrates the characteristic trade-off between tracking accuracy (RMSE) and control action variance. Unlike the AM2 case, there is no distinct minimum for the tracking error. A value of $w_u = 25$ was selected as a good compromise, as it significantly reduces control action variance while keeping the tracking error close to its minimum. This tuned parameter was used for the closed-loop evaluation.

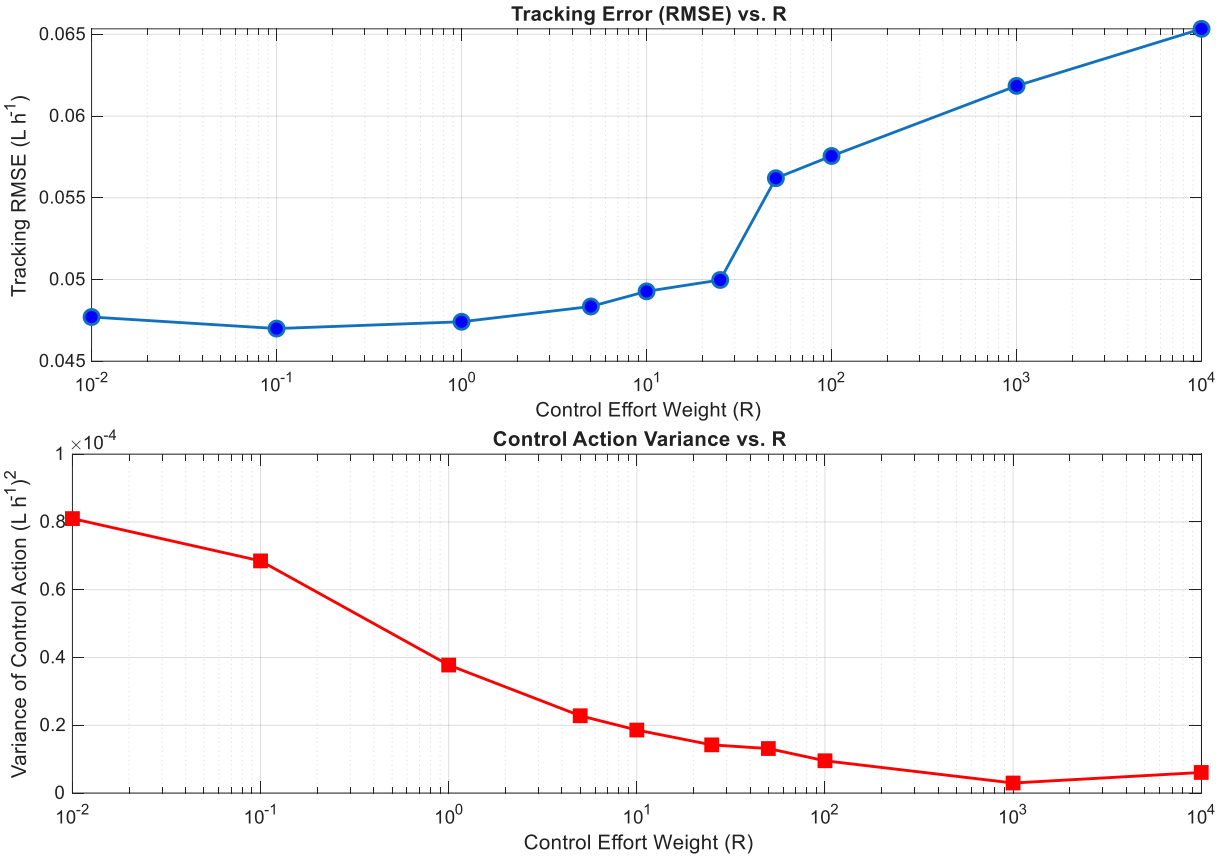


Figure 23: MPC tuning analysis for the ADM1 emulator, showing the trade-off between Tracking Error (RMSE) and Control Action Variance as a function of the Control Effort Weight (w_u). The analysis was performed at a fixed setpoint at 90% of the range.

A broader observation emerges from the tuning processes for both the AM2 (Figure 17) and ADM1 (Figure 23) emulators. The optimal control effort weight (w_u) differs significantly between the two cases (10^4 for AM2, 25 for ADM1). This discrepancy can likely be attributed to the terms in the cost function, the squared tracking error and the squared change in the control action, having different orders of magnitude, which are specific to each process's dynamics and operational range. Consequently, the tuning process becomes a non-intuitive search to find a weight that properly balances these scaled contributions.

This suggests a key methodological takeaway: for future implementations, the tuning process could be significantly streamlined by ensuring the terms in the cost function are of a comparable scale. This can be achieved through two equivalent strategies. One option is to adjust the weighting factors to be inversely proportional to the square of the typical scale of their corresponding variables, which directly balances the quadratic terms in the cost function. A second, more systematic approach is to normalize the process variables before they enter the cost function. Scaling both the output error and the control action to a common range (e.g., [0,1]) would place both terms on a comparable footing, making the selection of weighting factors more direct and less dependent on the specific scales of the process variables.

4.7 Closed-Loop Performance

The closed-loop performance of the LSTM-NMPC on the ADM1 emulator is presented in Figure 24. The controller tracks the setpoints, exhibiting a slower response that reflects the more sluggish dynamics of this high-fidelity process model. The control action (influent flow rate, Q_{in}) is smoother and less aggressive than in the AM2 case, which is consistent with controlling a slower system. As shown in the figure, some oscillations persist which the controller cannot completely eliminate. Despite this, the controller maintains process stability and keeps the output close to the desired setpoint, compensating for disturbances. These results support the applicability of the approach to a more realistic process representation.

It is noted that while the controller maintains stability and steers the process towards the setpoint, some oscillations persist. This behavior can be attributed to the nature of the data-driven predictive model. As shown in the open-loop tests (Figure 21), the LSTM network learns to predict the underlying deterministic trend of the process, effectively filtering the high-frequency stochastic noise present in the training data. Consequently, the NMPC optimizes its control actions based on these "clean," averaged predictions. However, the process emulator continues to be affected by unmeasured random disturbances. This inherent model-plant mismatch between the smoothed forecast and the noisy reality of the process requires the controller to make continuous corrections, leading to the observed oscillations around the setpoint.

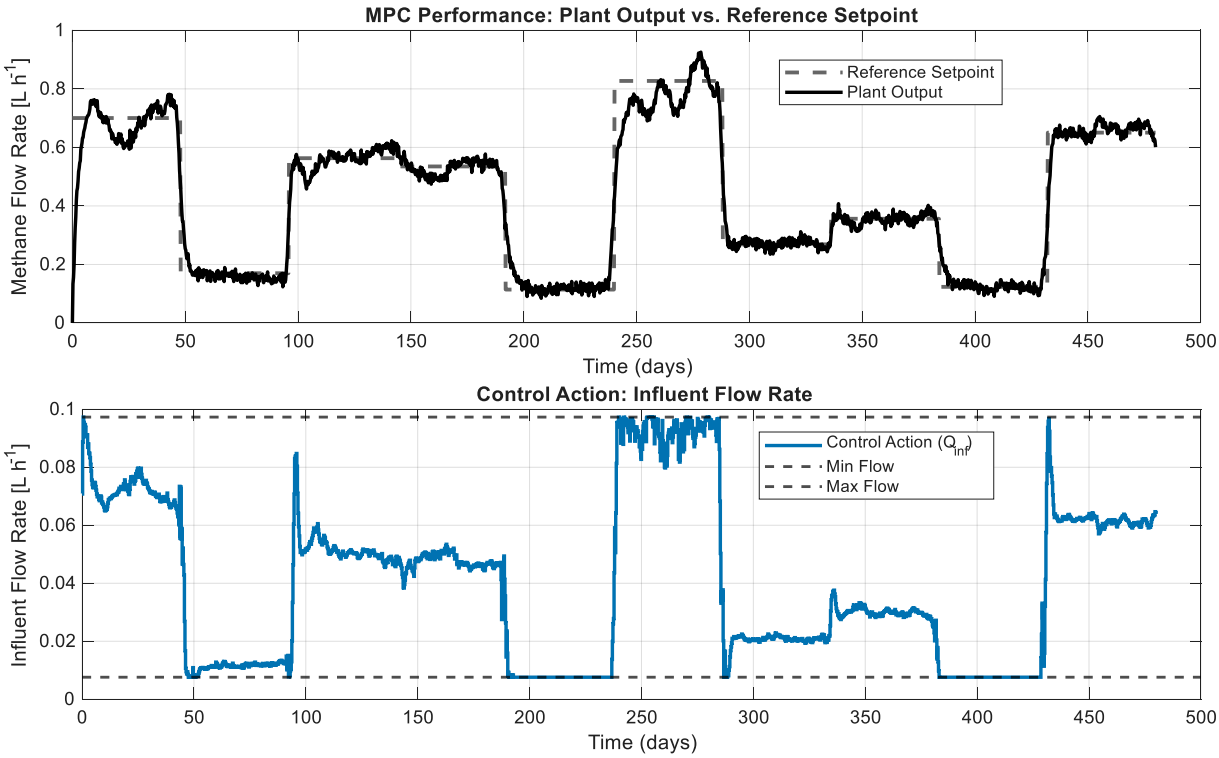


Figure 24: Closed-loop performance of the LSTM-MPC on the ADM1 emulator, showing setpoint tracking (top) and the corresponding control action (bottom).

Part III: General Discussion

4.8 Comparative Discussion and Feasibility

This final part of the chapter discusses the results obtained with the high-fidelity ADM1 emulator and synthesizes the findings from both the AM2 and ADM1 analyses. The aim is to form a general conclusion regarding the feasibility and robustness of the proposed data-driven control strategy based on the evidence presented.

The results from both emulators suggest the viability of the LSTM-NMPC framework for controlling the simulated AD processes. The performance against both a simplified (AM2) and a high-fidelity (ADM1) model indicates the robustness of this data-driven approach. The main finding is that setpoint tracking was achieved using a model trained on a minimal set of online data (methane flow rate and the manipulated variable). This suggests that while mechanistic models are important for process understanding, they may not be a prerequisite for controller design if data of sufficient quantity and quality is available.

4.8.1 Effect of Data Quantity

An analysis of the relationship between the amount of training data and predictor accuracy indicates a pattern of diminishing returns for both emulated processes. As shown in Figure 25 and Figure 26, the prediction error for both the AM2 and ADM1 predictors improves significantly as the number of perturbations in the training set increases from two to about six. Beyond this point, adding more data offers only marginal gains, as the performance curves begin to plateau. This trend, observed for both the simpler and the more complex system, suggests that the informational value of the initial, well-designed perturbations is a key factor in model performance. For the cases studied here, a training set with 6-8 dynamic events represented a reasonable compromise between data acquisition effort and model accuracy.

This observation, however, should be viewed within a practical context. While it highlights the importance of experimental design, it also brings to light a key challenge for real-world application. The "well-designed" perturbations used to generate this informative data were large step changes. Such maneuvers, while effective in simulation, might not be practical for a physical plant, where operational stability is a priority. Additionally, the process's slow dynamics mean that even an experimental campaign with a limited number of perturbations would still require a considerable amount of time to execute, as each step must be held long enough for the system to show its full dynamic response.

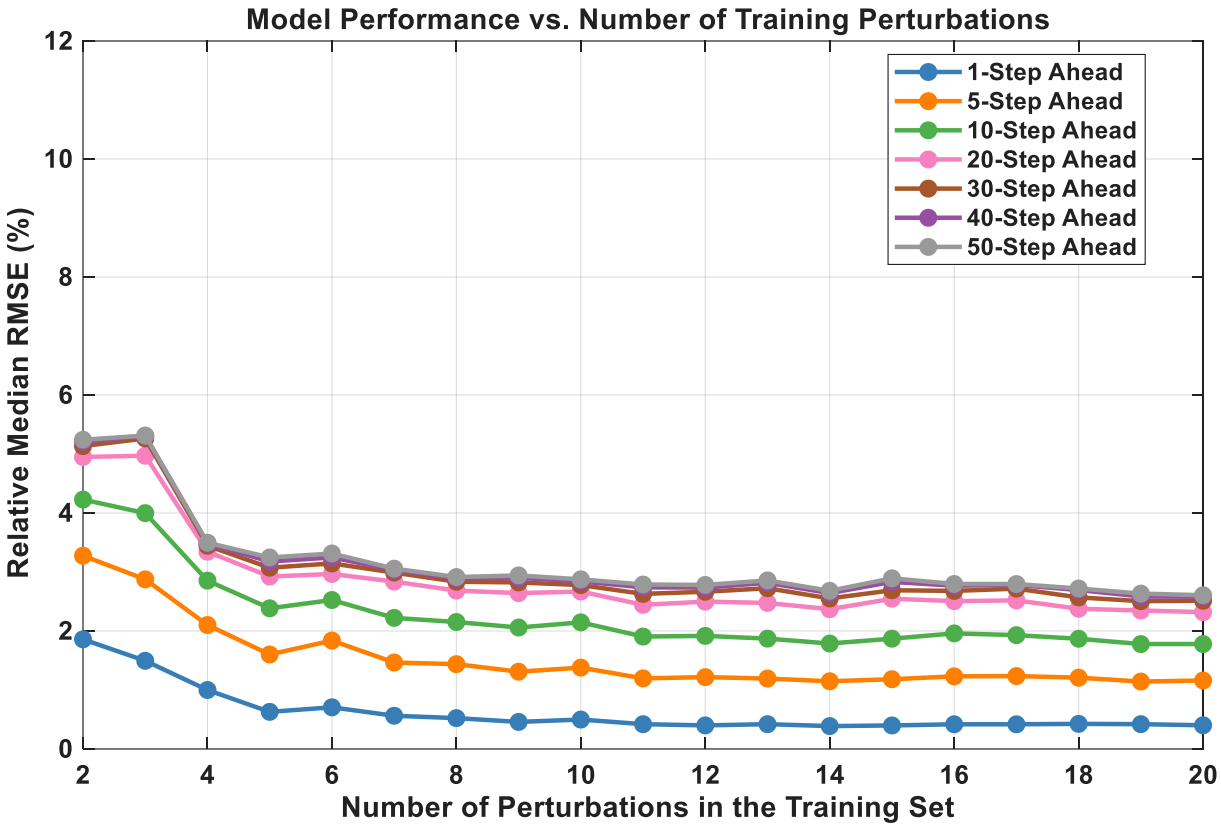


Figure 25: Effect of training data size on the n-step ahead prediction RMSE for the AM2 emulator.

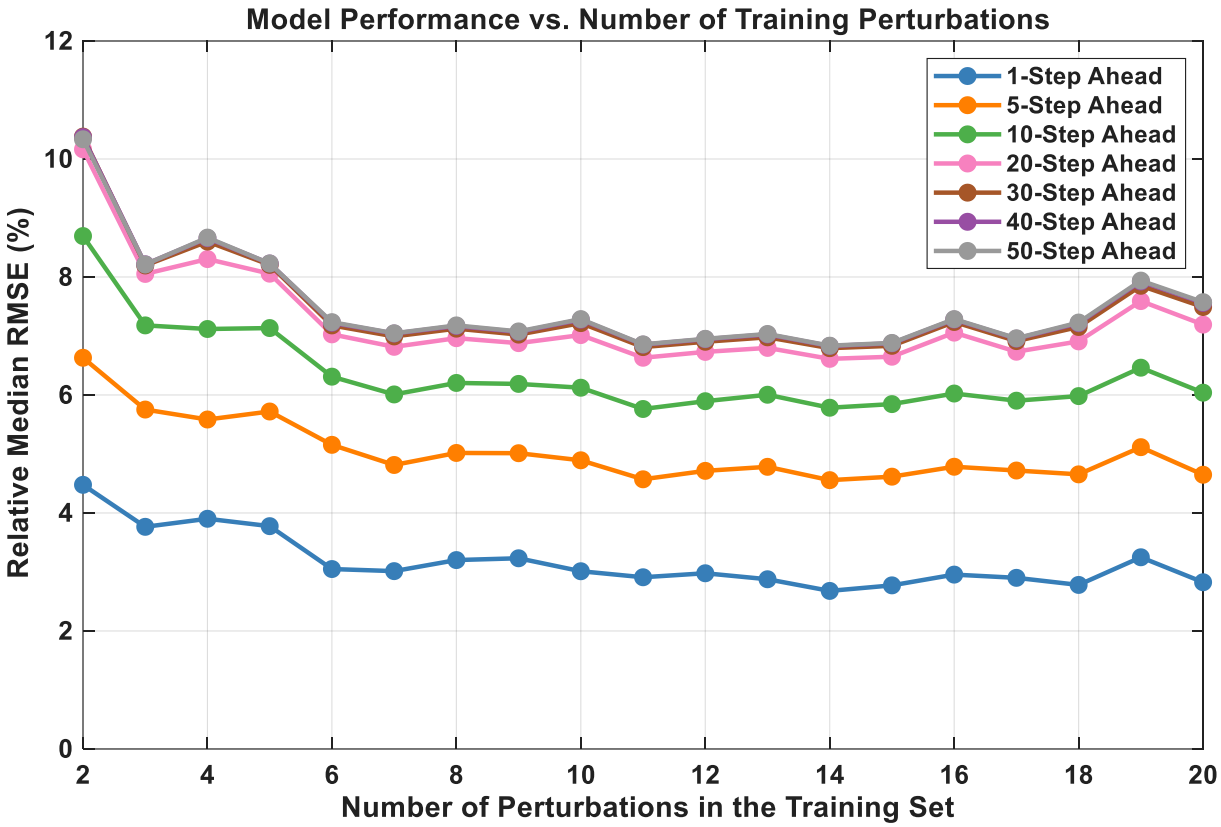


Figure 26: Effect of training data size on the n-step ahead prediction RMSE for the ADM1 emulator.

4.8.2 Computational Feasibility

A critical aspect for real-world applicability is computational demand. As shown in Figure 27, the average computation time per MPC step was on the order of seconds (1.60s for AM2 and 4.38s for ADM1). This is several orders of magnitude smaller than the process sampling interval, indicating that the proposed LSTM-NMPC approach is computationally feasible and well-suited for real-time implementation. The study thus suggests that the LSTM-NMPC strategy is a robust, efficient, and practical method for the control of the simulated anaerobic digestion processes considered here.

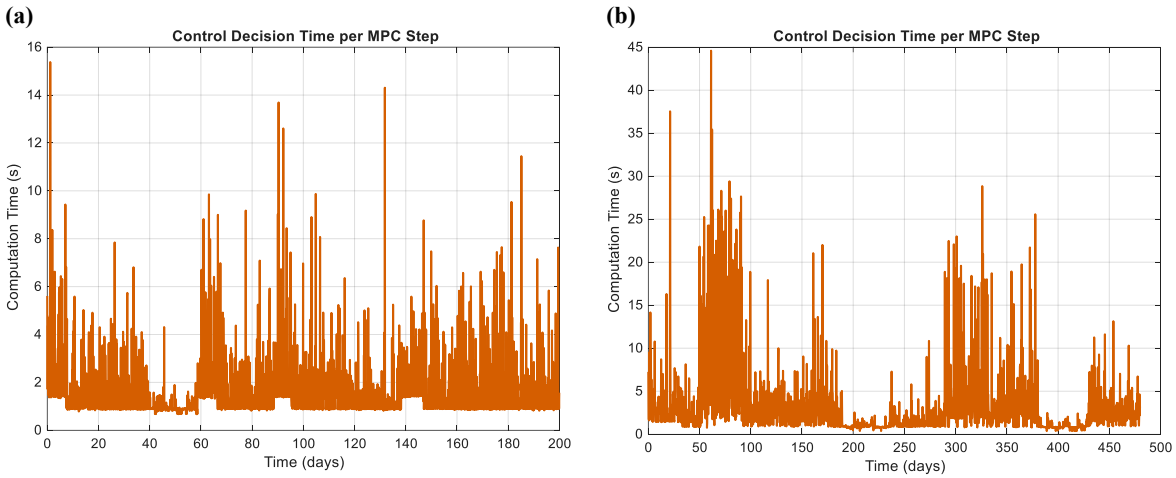


Figure 27: Computation time per control step for the LSTM-NMPC strategy applied to the (a) AM2 and (b) ADM1 emulators.

5 Comparative Analysis and Benchmarking

After establishing that the LSTM-NMPC framework is a viable approach in the previous chapter, the next step is to contextualize its performance through a benchmark analysis. This chapter evaluates the framework against five alternative control strategies. All controllers were subjected to an identical setpoint tracking task, including the same sequence of setpoint changes and unmeasured stochastic disturbances, to ensure a direct comparison.

The alternative frameworks were selected to assess the proposed strategy from two perspectives: the choice of the internal predictive model and the comparative performance against a classical control method. The evaluated strategies include:

- **Linear ARX-MPC:** To establish a linear baseline and assess the need for a non-linear approach.
- **Feed-forward NARX-NMPC:** To evaluate if a simpler, non-recurrent neural network is sufficient to capture the process dynamics.
- **GRU-NMPC:** To compare the LSTM against its main alternative within the family of recurrent neural networks.
- **Attention-based NMPC:** To benchmark against a modern, non-recurrent architecture also designed for sequential data.
- **Proportional-Integral (PI) Controller:** To quantify the performance difference relative to a widely-used standard for industrial process control.

The following sections present the results of this comparative analysis.

5.1 Modelling with alternative Architectures

The analysis begins with an open-loop evaluation of the predictive models that serve as the internal models for the MPC strategies. To this end, each of the four alternative data-driven architectures (Linear ARX, NARX, GRU, and Attention) was trained on the same dataset as the LSTM network to assess their respective predictive capabilities.

5.1.1 Dynamic Prediction Performance

The primary function of the internal model is to predict the process dynamics. As shown in Figure 28, the one-step-ahead predictions for all five models closely follow the test data, capturing the fast dynamics of the methane flow rate. In multi-step-ahead mode, the non-linear models generate a smoothed forecast that captures the underlying trend while filtering the stochastic noise.

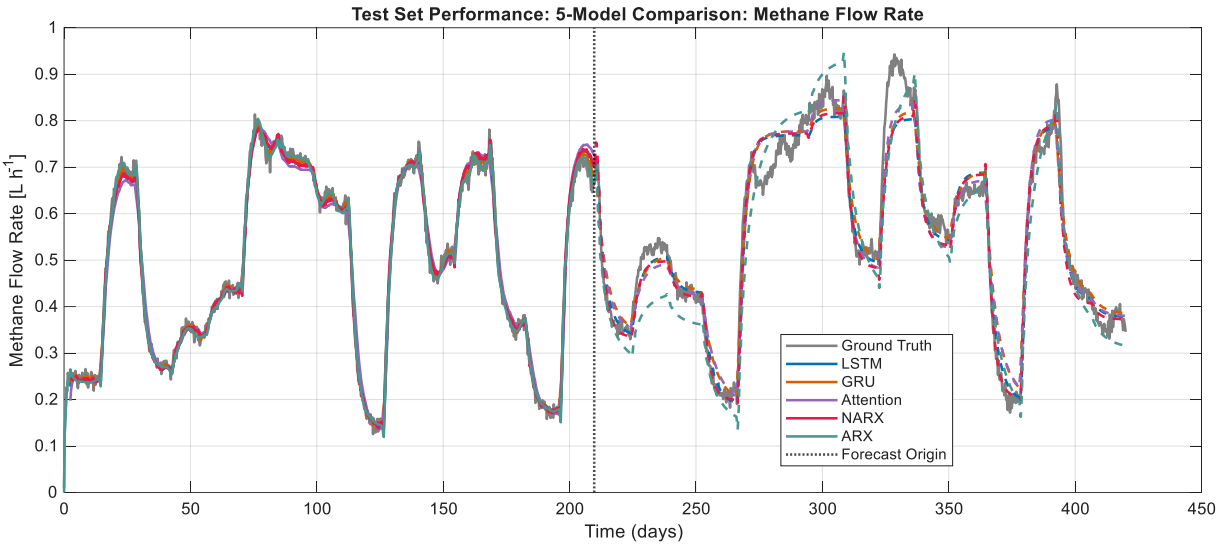


Figure 28: Comparison of one-step-ahead prediction performance for all five model architectures on the test set. All models closely track the ground truth.

While one-step-ahead prediction is a useful metric, a model's utility within an MPC framework is more dependent on its long-range forecast stability. The growth of the Root Mean Squared Error (RMSE) over a 50-step prediction horizon is illustrated for each model (Figure 29). A clear distinction emerges between the linear and nonlinear models. While the errors for all models eventually stabilize, the ARX model's error converges to a significantly higher value (around 12%) compared to all nonlinear models, which plateau at a much lower error level (between 7-8%). This suggests a diminished long-term predictive accuracy for the linear model. The fact that this bounded error was present across different nonlinear architectures is noteworthy. It suggests that the stable long-term prediction may be more attributable to the shared NARX-like input structure (i.e., using a history of past inputs and outputs) than to a specific internal processing mechanism, such as recurrence.

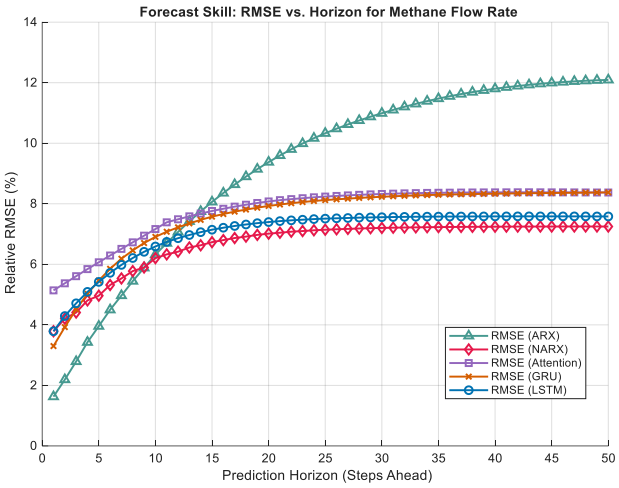


Figure 29: Multi-step prediction error (RMSE) as a function of the prediction horizon. The error of the linear ARX model diverges, while all nonlinear models remain stable.

5.1.2 Steady-State Behavior Representation

A critical assessment for a dynamic model is its ability to represent the underlying steady-state behavior of the process. Figure 30 compares the equilibrium methane flow predicted by each model against the true equilibrium curve of the ADM1 emulator. As expected, the linear ARX model approximates the nonlinear process with a straight line, resulting in significant errors across the operating range. In contrast, all four nonlinear models capture the curved shape of the true equilibrium. This indicates that these data-driven structures, trained solely on dynamic data, also learned the fundamental static characteristics of the high-fidelity process, which is an important property for precise control.

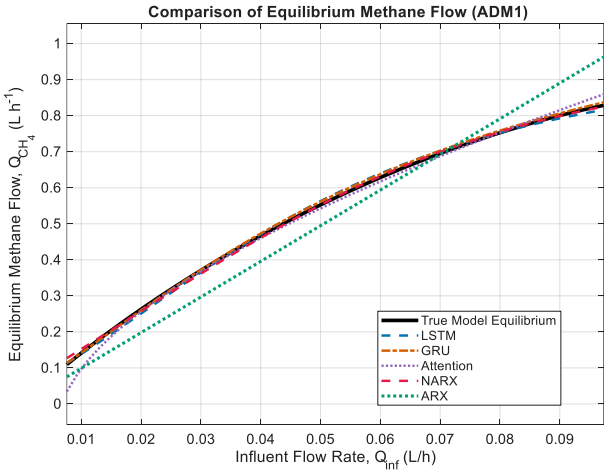


Figure 30: Comparison of the true steady-state methane flow of the ADM1 model versus the equilibrium predicted by the trained models. All nonlinear models capture the curvature, while the linear ARX model fails.

5.2 Control with Alternative Strategies

To benchmark the data-driven MPC's performance, a classical Proportional-Integral-Derivative (PID) controller was designed and tuned for the same ADM1 process.

5.2.1 PID Controller Tuning

To benchmark the MPC strategies, a classical Proportional-Integral (PI) controller was tuned using the Ziegler-Nichols step-response method. For this, a First-Order Plus Dead Time (FOPDT) model was identified for each step change in the training dataset. An automated filtering algorithm based on the Interquartile Range (IQR) was employed to exclude anomalous step responses (e.g., Step #5, Appendix C) and ensure a robust parameter estimation. The final PI parameters ($K_c = 0.24$, $T_i = 96.48$) were determined by averaging the values obtained from the valid step responses, with individual model fits detailed in Appendix C.

5.2.2 Closed-Loop Performance Comparison

All six controllers (the five MPC variants and the PI controller) were evaluated on an identical setpoint tracking task, subject to the same sequence of setpoint changes and unmeasured stochastic disturbances. The comparative closed-loop performance is shown in Figure 31.

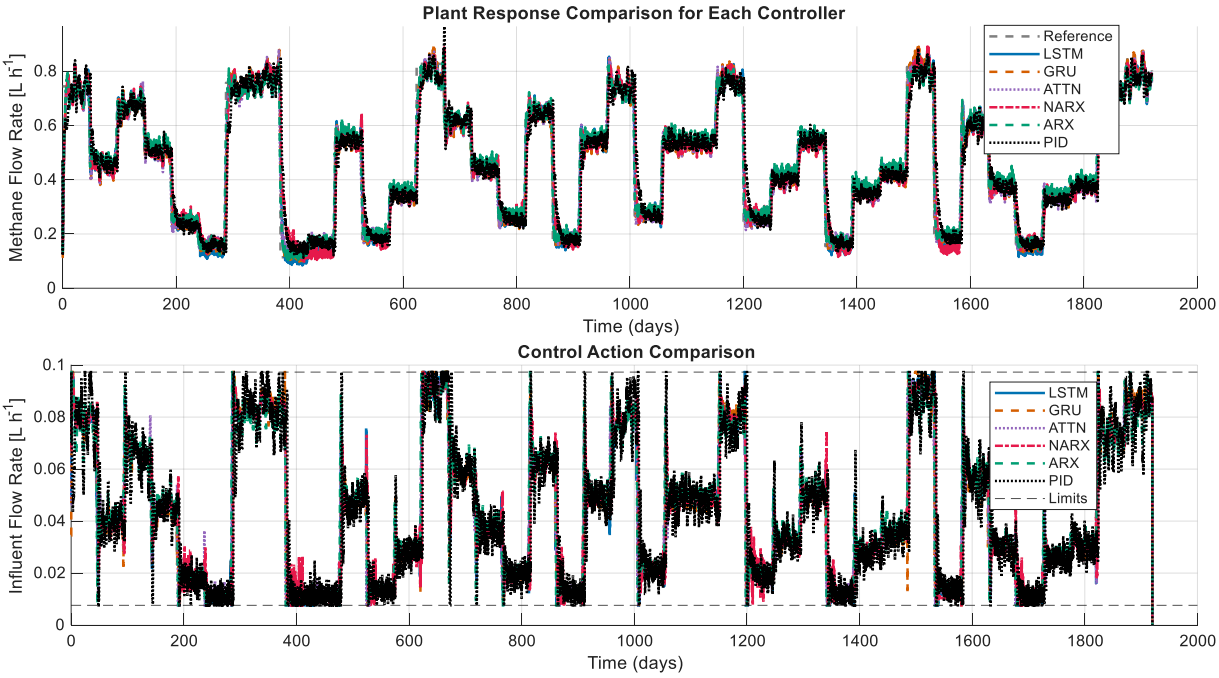


Figure 31: Closed-loop performance comparison for the MPC controllers (using LSTM, GRU, Attention, NARX, and ARX models) and the PID controller on the setpoint tracking task.

While a visual inspection confirms that all strategies are capable of tracking the setpoint to some degree, it also reveals some differences in their dynamic behavior. However, a definitive assessment based solely on visual evidence is challenging, particularly for the best-performing controllers whose trajectories overlap. This reinforces the need for a quantitative analysis to accurately compare their performance.

5.3 Comparative Discussion

A quantitative summary of each control strategy's performance and computational demands is provided in Table 2. The metrics have been normalized relative to the LSTM-NMPC controller for direct comparison. The baseline performance of the LSTM-NMPC corresponds to an RMSE of 0.040 L/h, an MAE of 0.022 L/h, a mean computation time of 9.02 seconds per control action and a median of 4.73 seconds.

Table 2: Performance and Computational Metrics for All Evaluated Controllers

Controller	RMSE	MAE	Mean Time	Median Time
LSTM-NMPC	1	1	1	1
GRU-NMPC	1.03	1.04	0.95	0.91
ATTN-NMPC	1.10	1.16	0.37	0.52
NARX-NMPC	1.54	1.43	0.41	0.53
ARX-MPC	1.18	1.36	$3.0 \cdot 10^{-4}$	$5.4 \cdot 10^{-4}$
PI	1.91	1.61	$3.8 \cdot 10^{-7}$	$5.7 \cdot 10^{-7}$

The data highlights a notable outcome in the case of the NARX-NMPC controller. It yielded a tracking error 54% higher than the baseline, despite being based on the predictive model with the lowest long-range error on the test set (Figure 29), This discrepancy suggests that a model's open-loop predictive accuracy is not always a reliable indicator of its closed-loop performance. The periods of oscillation evident in the controller's response (Figure 31) offer a potential explanation: while the NARX model was accurate for specific test trajectories, its predictions may have lacked robustness to the small deviations introduced by the feedback control loop.

In contrast, simpler strategies resulted in consistent, though less accurate, behaviors. The PI controller had the highest tracking error (RMSE 1.91), a result consistent with the high-frequency oscillations observed in its response. The linear ARX-MPC, while unable to capture the process nonlinearity, proved more robust in closed-loop than its nonlinear NARX counterpart (RMSE 1.18). Its predictable linear behavior, though leading to a persistent offset, allowed the MPC to manage it more effectively.

The lowest tracking errors were achieved by controllers based on recurrent (LSTM, GRU) and attention-based architectures. The LSTM and GRU controllers showed nearly identical tracking

accuracy. This suggests that their internal state mechanisms may provide a greater capacity for generalization under the dynamic conditions of feedback control. The Attention-based NMPC was also a robust controller that outperformed the NARX model, representing a viable, computationally faster alternative to the recurrent networks.

These findings indicate that for the process under study, a data-driven MPC with a predictive model that includes a state-aware mechanism (such as recurrent or attention structures) is an effective approach. Given that the computational cost of these models is within a feasible range for a slow process like anaerobic digestion, the improved tracking performance appears to justify the additional modeling effort.

This comparison illustrates a practical engineering trade-off. Recurrent architectures like LSTM and GRU function as general-purpose models. They represent a robust choice when the process dynamics are not known beforehand, as they can learn both linear and nonlinear behavior to achieve high tracking accuracy. In contrast, the linear ARX-MPC offers a less computationally demanding alternative. For the nonlinear process studied here, its linear model leads to a steady-state offset, a predictable consequence that could likely be corrected by incorporating integral action into the control law. This limitation makes the ARX-MPC suitable mainly for quasi-linear systems or for applications where computational speed is the overriding factor. Given the distinct nonlinearity of anaerobic digestion, the superior generalization and performance of the LSTM-based approach justify its use as the more appropriate strategy in this context.

6 Extension to a Multi-Input Co-digestion Process (ACoD)

Having established the viability of the LSTM-NMPC framework for single-input, single-output (SISO) anaerobic mono-digestion processes in the preceding chapters, a subsequent question is whether the methodology can scale to more complex, multi-input systems. This chapter addresses this by extending the framework to an industrially relevant scenario: the control of Anaerobic Co-digestion (ACoD). This process, known for its potential efficiency gains but also for its challenging, non-stationary dynamics, serves as a suitable test case for evaluating the scalability and robustness of the data-driven strategy.

6.1 Introduction to the Challenges of ACoD

Anaerobic Co-digestion (ACoD) involves the simultaneous digestion of two or more substrates. This practice can enhance the performance of anaerobic digestion processes by leveraging the complementary characteristics of different materials. The benefits of ACoD rely on potential synergies among the co-substrates, which can lead to increased biogas production compared to mono-digestion (García-Gen et al., 2014).

However, the efficiency of the ACoD process is influenced by many factors, including substrate composition, temperature, pH, the carbon-to-nitrogen (C/N) ratio, and the organic loading rate (OLR). Selecting an appropriate blend of substrates to ensure stable operation is a complex task, as the proportions must be carefully balanced to keep key process parameters within stable ranges and avoid inhibition. This makes ACoD a multi-input, single-output (MISO) control problem that is significantly more challenging to model and control than mono-digestion.

This study uses an ACoD process emulator based on the ADM1, where three distinct substrate streams are manipulated to control the methane production rate. These streams correspond to manure (Q_{str1}), glycerine (Q_{str2}), and protein (Q_{str3}).

6.2 Methodology for the ACoD study

6.2.1 Data Generation and Predictor Development

To develop a predictive model for the ACoD process, dynamic training data was generated by applying a series of step changes to the flow rates of the three input substrates. A primary goal of the experimental design was to generate data that covered a wide range of conditions within the operating space. As neural networks are known to interpolate well but extrapolate poorly, the training sequence was designed to include the eight vertices of the operational cuboid (or space) defined by the minimum and maximum ranges of the three input variables.

To determine an adequate amount of training data, an initial analysis was performed (Figure 32). Based on these results, a training set consisting of 12 perturbations was selected, as it offered a

good balance between data richness and diminishing returns in prediction accuracy. Stochastic noise was added to the influent composition ($\sigma = 1.5\%$, $\phi = 0.95$) to ensure the data was realistic. Figure 33 illustrates the open-loop effect of these disturbances, showing how the methane flow rate fluctuates naturally even when the substrate flow rates are held constant. This inherent process variability represents the baseline challenge that the predictive model must learn to capture and the controller must learn to reject.

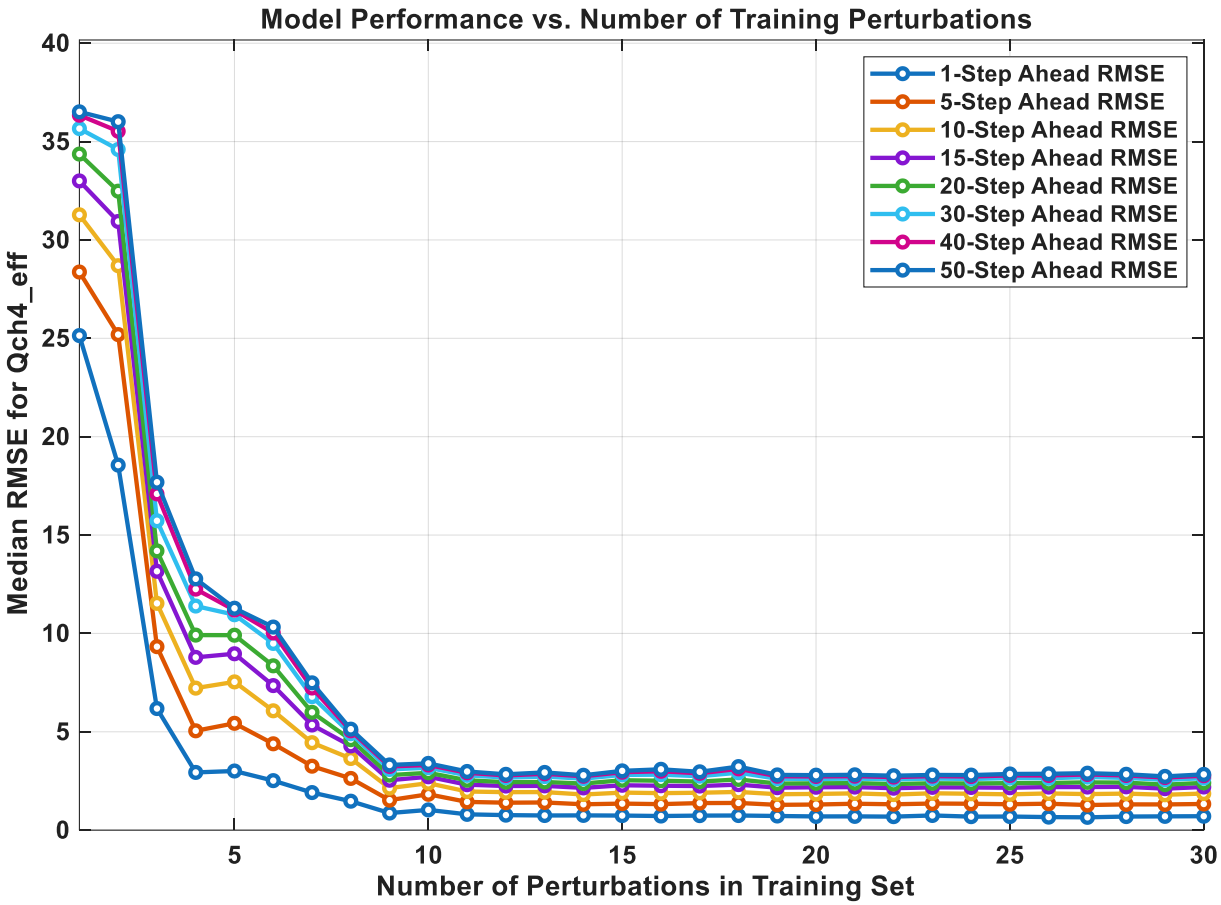


Figure 32: Analysis of LSTM predictor performance as a function of the number of perturbations in the training set for the ACoD model.

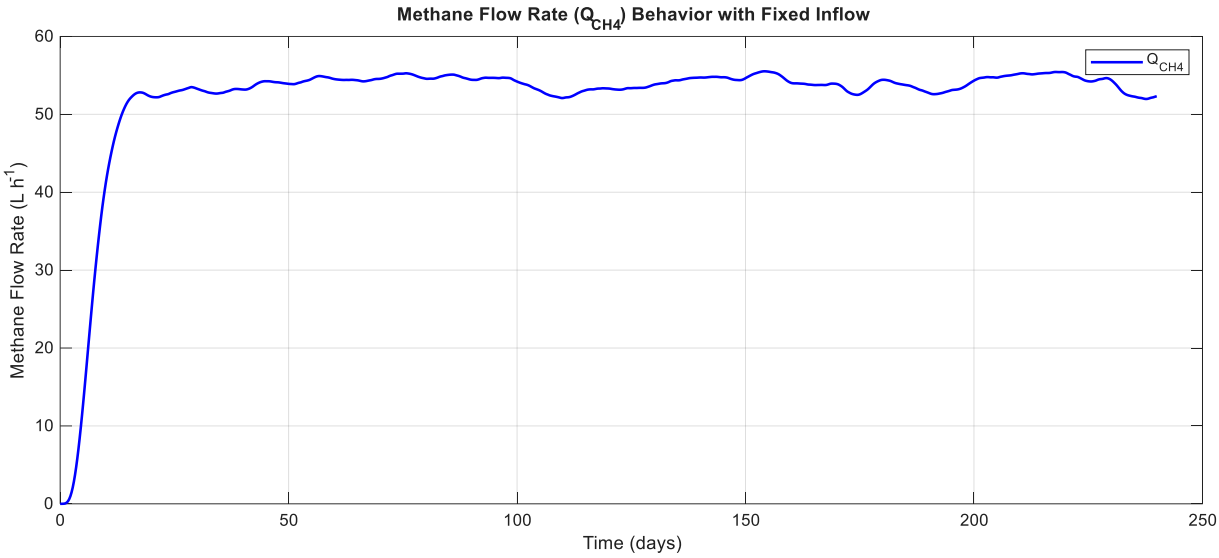


Figure 33: Dynamic fluctuations in methane flow rate at constant substrate inputs, illustrating the effect of stochastic variations in the influent composition.

An LSTM network with 5 hidden neurons was trained on this dataset to predict the methane flow rate based on the history of the three manipulated substrate flow rates and the past methane flow rate itself.

6.2.2 NMPC Controller Design and Tuning

The control objective was to track a methane flow rate setpoint by manipulating the three independent substrate feed rates. Unlike the mono-digestion case, where a one-dimensional sensitivity analysis was sufficient for tuning the single control effort weight (w_u), the multi-input co-digestion scenario presents a more complex, three-dimensional tuning problem for the weights corresponding to each substrate stream (w_{u1}, w_{u2}, w_{u3}).

Rather than attempting a manual tuning or pre-scaling the weights based on assumptions about the process, Bayesian Optimization was employed as an exploratory tool. The objective was to allow the algorithm to autonomously search the parameter space and identify a high-performing set of weights. The results of this search could then be analyzed *a posteriori* to gain insight into the control strategy that the optimizer deemed effective. The resulting weights from this search were $[w_{u1}, w_{u2}, w_{u3}] = [0.107, 3651, 9.1955]$.

This outcome provides a clear reflection of the underlying process dynamics. The algorithm assigned a very high penalty to the glycerine stream (Q_{str2}), which is known to have the most significant and rapid impact on methane production. This high weight restricts its movement, suggesting the controller uses it primarily for major setpoint adjustments rather than for continuous, fine-grained control, thus preventing potentially destabilizing oscillations. In contrast, the other two streams, manure (Q_{str1}) and protein (Q_{str3}), which have a less

pronounced effect on the output, received much lower penalties. This allows them to adjust more freely, acting as a means for fine-tuning and rejecting smaller disturbances, a behavior consistent with the steady-state behavior observed in Figure 34.

6.3 Predictive Performance of the LSTM Model in ACoD

Before evaluating the closed-loop control, the accuracy of the trained LSTM predictor was assessed. While the dynamic predictions showed good performance, a steady-state analysis revealed important insights into the model's limitations.

Figure 34 compares the true equilibrium surface of the ACoD model with the equilibrium predicted by the LSTM across different combinations of substrate inputs. While the process dynamics are largely linear across much of the operating space, there are specific regions where the non-linearity becomes more pronounced. The LSTM learns the general shape of the complex 4D surface, but a noticeable discrepancy, or model-plant mismatch, is evident in these specific regions. The mismatch is most significant at low flow rates of the glycerine (Q_{str2}) and protein (Q_{str3}) streams, where the true equilibrium surface exhibits a higher curvature. This strong non-linearity presents a challenge for the LSTM, likely because this operating region is less represented in the dynamic training data. As a result, the predictor tends to smooth out the surface, failing to capture the sharp changes in the process gain.

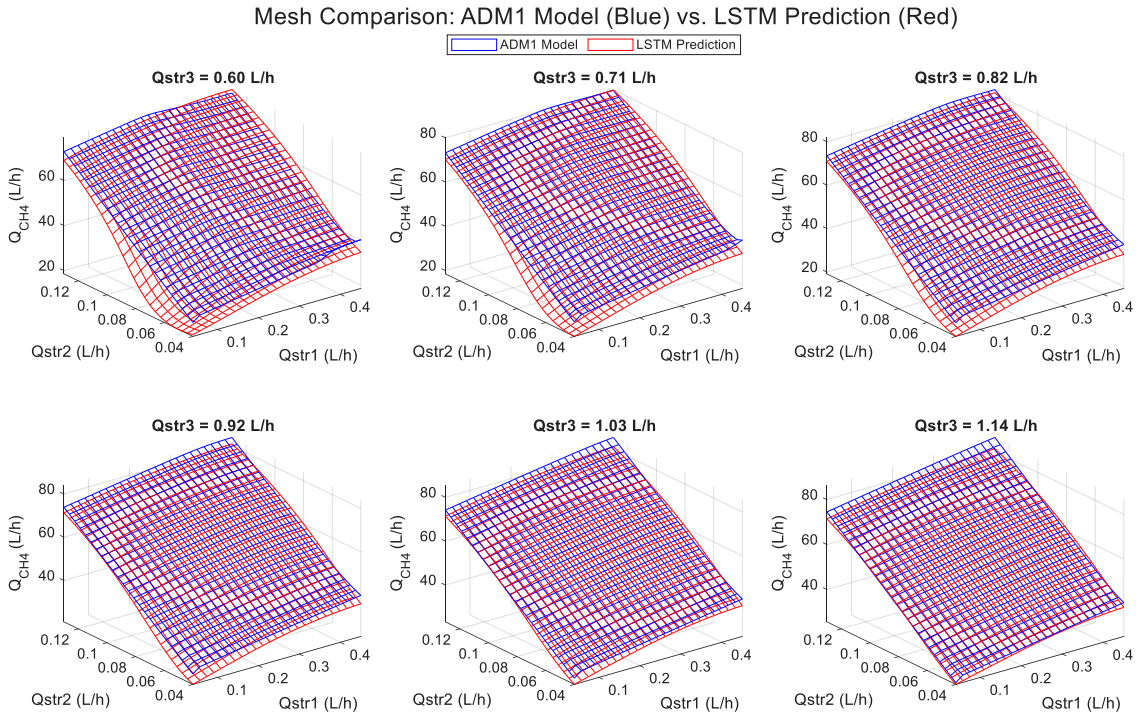


Figure 34: Comparison of the true steady-state surface of the ACoD model (blue mesh) versus the equilibrium predicted by the trained LSTM network (red mesh) for different substrate combinations.

This mismatch is likely due to a combination of two factors: the inherent difficulty for the neural network to approximate these sharp non-linearities and the fact that these operating points are less represented in the dynamic training data. This is quantified in Figure 35, which shows the prediction error surface. The largest errors (under-prediction) are concentrated in the operating region defined by low flow rates of Q_{str2} and Q_{str3}, particularly when Q_{str1} is also low. This finding is important, as it predicts that the controller's performance may be degraded when operating at low methane production setpoints, where these challenging dynamics are most relevant.

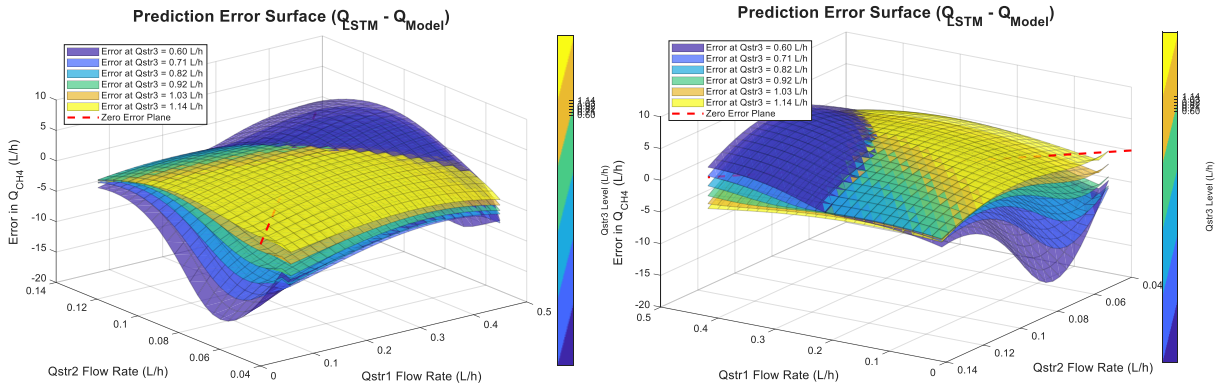


Figure 35: Prediction error surface, showing the difference between the LSTM's predicted equilibrium and the true model equilibrium.

6.4 Closed-Loop Control Results of the LSTM-NMPC in ACoD

The performance of the configured LSTM-NMPC is shown in Figure 36. The controller tracks the methane flow rate setpoints across a range of operating conditions, while the control actions remain smooth and respect the imposed rate-of-change constraints. As anticipated by the steady-state analysis, the controller's performance is less precise at lower setpoints (e.g., around 30-40 L/h). In this region, where the model-plant mismatch was identified to be larger, more pronounced oscillations around the setpoint are observed. Despite these oscillations, the controller maintains process stability and brings the output towards the desired operating point.

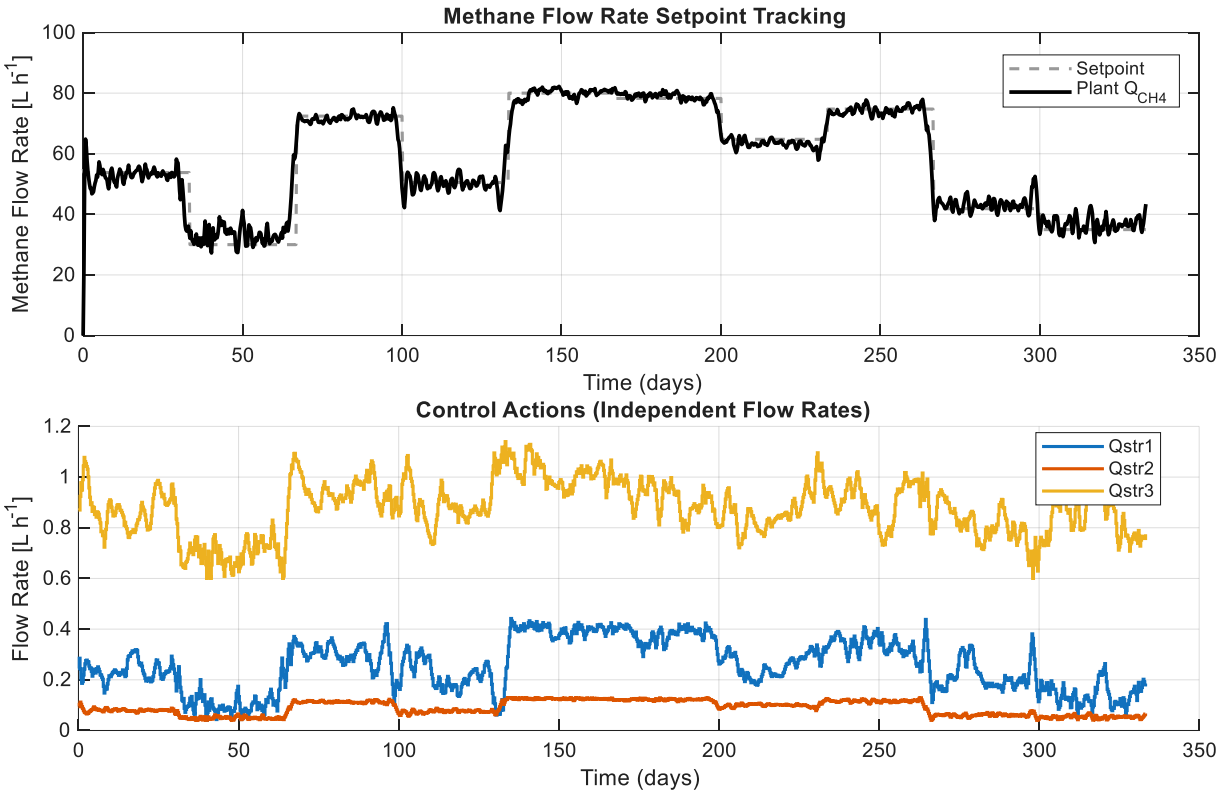


Figure 36: Closed-loop performance of the LSTM-NMPC on the ACoD emulator, showing setpoint tracking (top) and the corresponding control actions for the three substrate streams (bottom).

6.5 Discussion on the Scalability and Robustness of the Approach

The results from this chapter indicate that the LSTM-NMPC strategy can be extended to a multi-input co-digestion scenario. The methodology was applied to a three-input system, where it achieved setpoint tracking using a purely data-driven model. For the multi-variable tuning task, Bayesian Optimization was employed to identify suitable controller parameters. An important aspect of this result is that the input-output model functions without explicit knowledge of internal process dynamics, such as pH, VFA concentrations, and microbial population shifts. This suggests that a data-driven model can learn to represent the net effect of these internal interactions, capturing the mapping from manipulated inputs to the final output, which is consistent with the premise of a black-box approach for complex biological systems.

The study also included an analysis of the predictor's limitations. The identified mismatch in the steady-state predictions provided a potential explanation for the observed decrease in control performance at lower setpoints. This analysis allows for the method's limitations to be characterized, rather than being treated as unexplained deviations. In summary, the findings suggest the data-driven framework could be a scalable approach for ACoD control, and the identified areas of model mismatch point to potential targets for future improvements, such as

online model adaptation or the inclusion of an integral action term in the cost function to eliminate steady-state error.

7 Alternative Paradigm Study: Reinforcement Learning

Having examined the viability and scalability of the model-based LSTM-MPC approach in the preceding chapters, this study now turns to an alternative control paradigm: model-free Reinforcement Learning (RL). The objective of this chapter is to compare the performance, implementation challenges, and practical feasibility of several RL algorithms against the previously evaluated MPC strategy, using the AM2 emulator as a common testbed to ensure a direct comparison.

7.1 Introduction to RL as an Alternative to MPC

Reinforcement Learning (RL) encompasses a broad set of methods for learning control policies. A key distinction within RL is between model-based and model-free approaches. This study focuses specifically on the model-free paradigm, where an agent learns a control policy directly through interaction with its environment, guided by reward signals, without requiring an explicit process model. The evaluation centers on off-policy actor-critic algorithms, which are designed for continuous control tasks. Three such algorithms were selected for this work:

- **Deep Deterministic Policy Gradient (DDPG):** An algorithm that adapts the principles of Deep Q-Learning to continuous action spaces. It uses a deterministic actor policy and a critic to learn an effective control strategy (Lillicrap et al., 2019).
- **Twin Delayed Deep Deterministic Policy Gradient (TD3):** An extension of DDPG that introduces several improvements to address the overestimation bias and instability often found in actor-critic methods. It employs a pair of critics (the "twin" architecture) and delays policy updates to achieve more stable and robust learning (Fujimoto et al., 2018).
- **Soft Actor-Critic (SAC):** An algorithm based on the maximum entropy RL framework. The agent is trained not only to maximize the cumulative reward but also the entropy of its policy. This encourages more exploration and can lead to more robust and versatile policies (Haarnoja et al., 2019).

7.2 Methodology for the RL Study

7.2.1 Environment and Observation Space

The AM2 model, as described in Section 2.1.1, served as the environment for training and evaluating the RL agents. The agent's observation at each step included the current measured methane flow rate, the error relative to the setpoint, and a history of past actions and observations (defined by a WindowSize parameter) to provide dynamic context.

7.2.2 Hyperparameter Optimization

A known challenge in applying RL is its sensitivity to a large number of hyperparameters, which can make manual tuning impractical. Therefore, Bayesian Optimization was employed to systematically search for an effective set of hyperparameters for each of the three agents. The optimization aimed to minimize the Root Mean Squared Error (RMSE) on a standardized setpoint tracking simulation. The search space included parameters such as network size, learning rates for the actor and critic, the discount factor, and the observation window size. The resulting hyperparameters from this search are presented in Table 3.

Table 3: Best-Found Hyperparameters for RL Agents via Bayesian Optimization

Hyperparameter	DDPG	TD3	SAC
Network Size (L)	256	66	200
Actor Learn Rate	$3.27 \cdot 10^{-4}$	$1.27 \cdot 10^{-5}$	$2.13 \cdot 10^{-4}$
Critic Learn Rate	$9.72 \cdot 10^{-4}$	$4.14 \cdot 10^{-4}$	$8.24 \cdot 10^{-4}$
Discount Factor	0.9903	0.9736	0.9798
Target Smooth Factor	0.0069	0.0081	0.0004
Observation Window Size	10	3	2

7.3 Control Results with RL Agents on the AM2 emulator

Each agent was trained using the hyperparameters identified in the optimization search. The training progress for the DDPG, TD3, and SAC agents is shown in Figure 37, Figure 38, and Figure 39, respectively. In each case, the learning curves show the progression of the episode reward over the training episodes. All three agents show an increasing trend in the average reward, which eventually stabilizes, indicating that the training process converged.

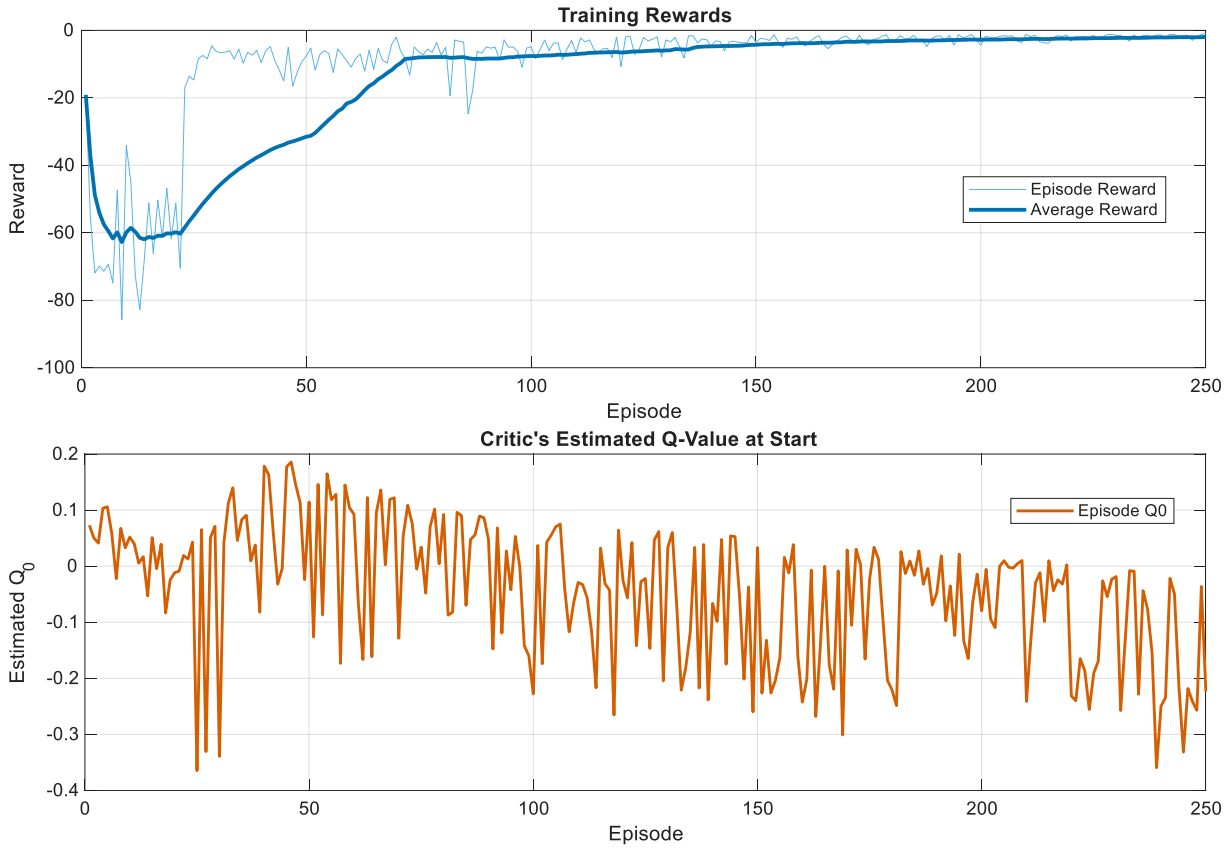


Figure 37: Training progress for the DDPG agent.

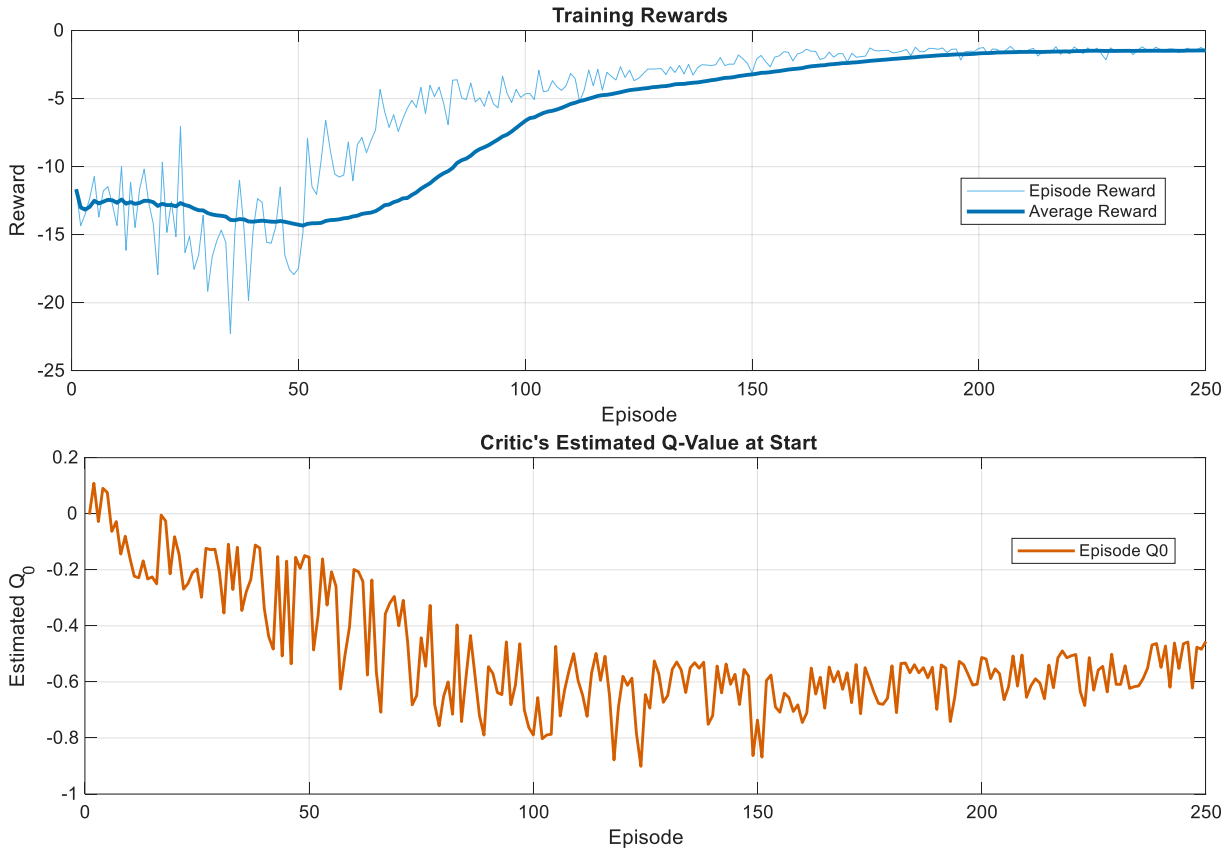


Figure 38: Training progress for the TD3 agent, showing the episode reward, average reward, and episode Q0 value over 250 training episodes.

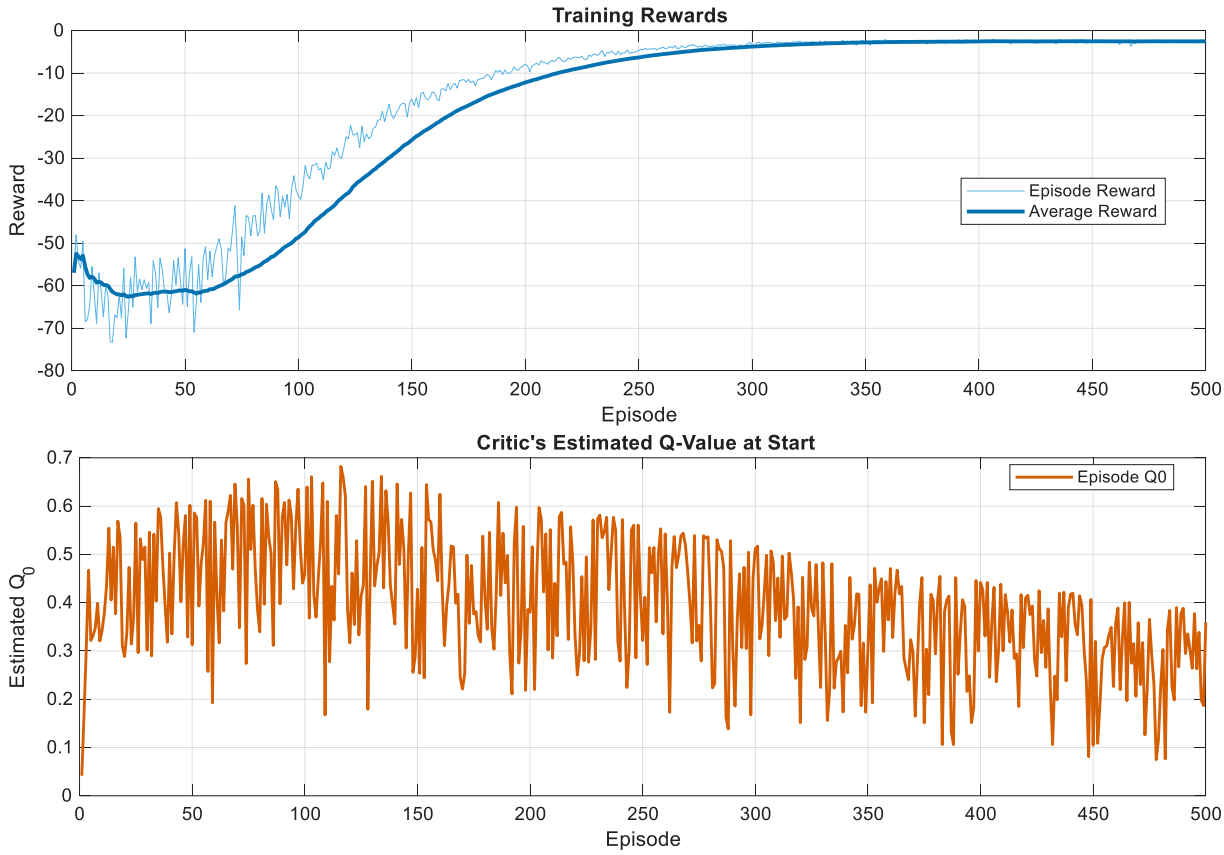


Figure 39: Training progress for the SAC agent.

The trained agents were then evaluated on a standardized setpoint tracking task. A comparative plot of the closed-loop performance for all three agents is shown in Figure 40**.** The results show that all agents track the methane flow rate setpoints, although qualitative differences in their performance, such as overshoot, offsets and settling time, are apparent and reflect the different learning strategies of each algorithm.

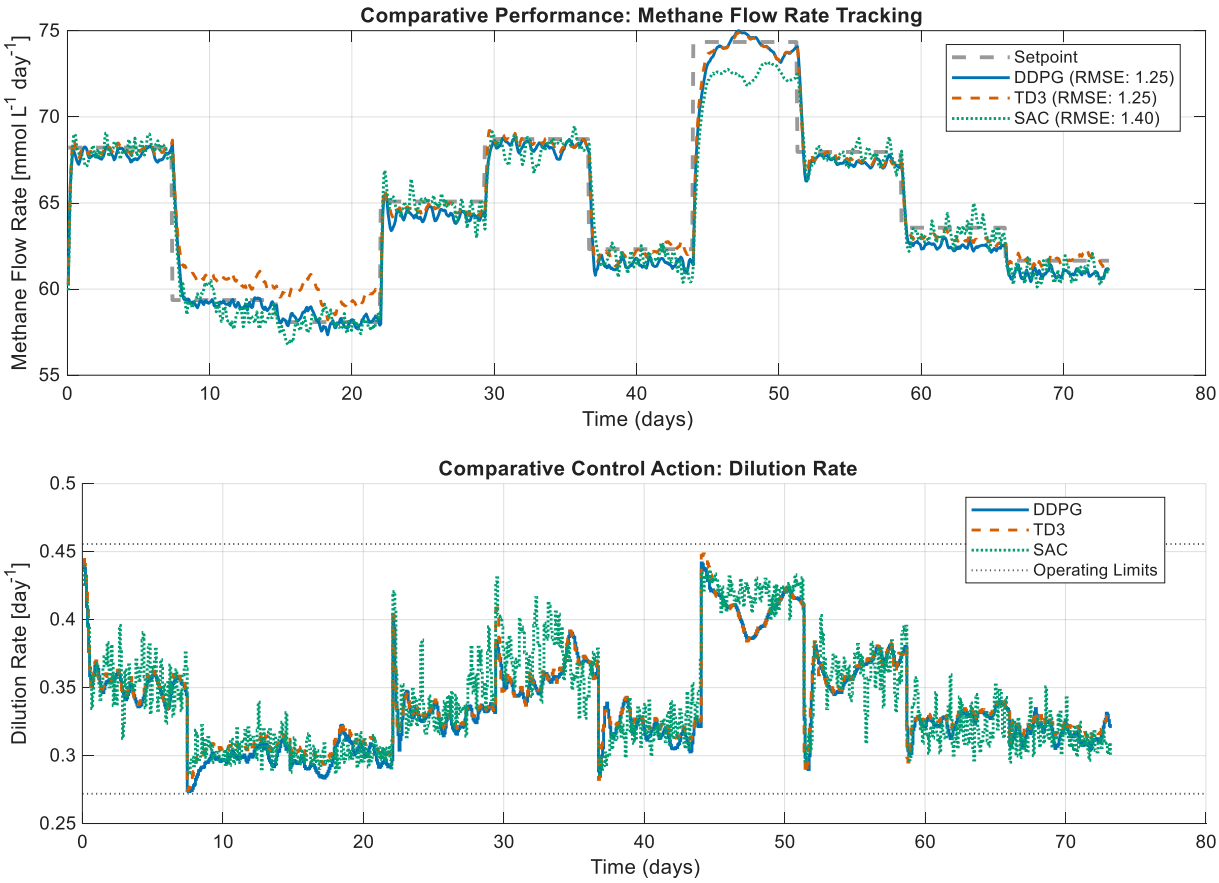


Figure 40: Comparative closed-loop performance of the DDPG, TD3, and SAC agents on the setpoint tracking task.

7.4 Critical and Comparative Discussion

The results indicate that the evaluated RL algorithms learned to control the nonlinear dynamics of the AM2 process. A comparison of the agents' performance in Figure 40 and their final RMSE suggests that while all three achieved setpoint tracking, DDPG and TD3 obtained slightly better tracking performance than SAC in this specific task. An observed distinction relates to the nature of the SAC algorithm. Unlike the deterministic policies of DDPG and TD3, SAC's policy is stochastic due to its maximum entropy framework. While this property can be beneficial for exploration during learning, it translates into a control action with higher variance, which in turn can introduce oscillations in the process output.

A comparison with the LSTM-NMPC approach highlights differences in the implementation workflow. The training process for each RL agent required hundreds of episodes, corresponding to tens of thousands of simulation steps, and the hyperparameter optimization was computationally intensive. This contrasts with the data requirements of the LSTM-NMPC

approach, where the predictor was trained on a single, relatively short dataset from a planned experiment.

Furthermore, an attempt to apply the RL methodology to the ADM1 emulator suggested the specificity required in the tuning process. The training of an agent for the ADM1 using the hyperparameters obtained for the AM2 model did not converge. This result indicates that an effective training configuration may be highly specific to the process dynamics and not directly transferable.

This lack of transferability, however, does not necessarily imply a fundamental limitation of the RL paradigm for this process. A logical next step, which fell outside the scope of this comparative study, would be to perform a dedicated Bayesian optimization search for the ADM1 model. It is plausible that such a search could identify a set of hyperparameters capable of training a functional agent, albeit at the cost of an additional, computationally intensive tuning phase.

The findings from this study therefore point to an engineering trade-off conditioned by the application context. The data-driven MPC framework, leveraging established principles of system identification, offers a direct implementation pathway when relying on limited experimental data. In contrast, the model-free RL approach, while demonstrating its capacity for effective control, presents practical challenges related to sample inefficiency and intensive, non-transferable tuning. This distinction suggests that the choice of strategy may depend on the available resources: MPC appears well-suited for scenarios based on finite experimental datasets, whereas RL emerges as a powerful alternative whose primary implementation hurdles are associated with the data generation and training phase, rather than its ultimate control potential.

8 Conclusions and Recommendations

8.1 Synthesis of Findings and Conclusions

This thesis investigated the feasibility, robustness, and scalability of a data-driven Model Predictive Control (MPC) strategy for simulated anaerobic digestion (AD) processes. The core of this work was the development of a surrogate model based on a Long Short-Term Memory (LSTM) neural network. This approach proved to be a pragmatic alternative for control design, as its main advantage lies in bypassing the primary operational hurdles of mechanistic models: their complex calibration and the need for state estimators for unmeasurable variables.

The study began by establishing the viability of the LSTM-NMPC framework on a simplified AM2 emulator. The modeling and control results in this initial phase supported the proof-of-concept. The analysis also revealed that the predictor, as trained on the generated dataset, did not fully capture the sharp non-linearity of the process near its washout threshold, a finding attributed to a lack of informative data in that specific operating region. This observation underscored that a model's domain of validity is contingent on the richness of its training data. In contrast, the robustness of the approach was then tested against the high-fidelity ADM1 emulator, where the data-driven predictor successfully captured the process's nonlinear behavior and the controller demonstrated effective performance against a more complex representation.

A comparative analysis was conducted to contextualize the performance of the chosen LSTM architecture. The results indicated that nonlinear predictive models with state-aware structures (recurrent or attention-based) resulted in a lower tracking error in the closed-loop simulations over simpler nonlinear, linear, and classical PI controllers for this specific control task.

The scalability of the framework was subsequently examined through its application to a multi-input anaerobic co-digestion (ACoD) scenario. The modeling and control methodology was successfully extended to this more challenging configuration, suggesting that the data-driven framework could be applied to more industrially relevant processes.

Finally, an investigation into the alternative paradigm of model-free Reinforcement Learning (RL) provided a basis for comparison. While the evaluated RL agents learned a control policy for the simpler AM2 process, the study highlighted practical considerations in their deployment, namely the computationally intensive and system-specific nature of the hyperparameter tuning process. This finding positioned the LSTM-NMPC framework as a more direct implementation strategy under the conditions of this study.

In synthesis, the findings of this work suggest that the proposed LSTM-NMPC strategy represents a practical alternative for the advanced control of the simulated anaerobic digestion processes considered here. The approach offers the potential for improved performance by

leveraging the representational power of neural networks while bypassing some of the traditional obstacles of mechanistic modeling.

8.2 Limitations of the Study

A primary limitation of this work is the extensive data required for training the predictive models. Although the study was conducted in a simulated environment, the experimental campaigns were lengthy: the data for the faster AM2 model corresponded to nearly 30 days of operation, while the data for the slower ADM1 and the ACoD models represented approximately 150 days each.

Executing such prolonged, deliberately perturbed experimental campaigns in a real-world industrial or laboratory setting presents significant practical challenges, primarily related to:

- **Time and Cost:** The duration of these experiments could be prohibitively long and costly.
- **Process Stability:** Applying persistent perturbations to the process runs the risk of causing significant operational instability, which is often unacceptable in a production environment.

Therefore, while this study demonstrates the feasibility of the data-driven approach, transitioning from simulation to a physical plant requires careful consideration of these data acquisition hurdles.

8.3 Recommendations for Future Work

The transition of this control framework from simulation to a physical plant requires a clear strategy for data acquisition and model development. Future work could explore three primary pathways, each presenting a different trade-off between model fidelity, cost, and operational risk.

1. The Digital Twin Approach via Mechanistic Models This approach proposes using a calibrated first-principles model to generate high-fidelity synthetic data.

- **Methodology:** The process would begin with the calibration of a detailed mechanistic model (e.g., ADM1) to represent the dynamics of a specific plant. This calibrated model would then serve as a "digital twin", from which extensive and clean dynamic data could be generated. This synthetic dataset would then be used to train a computationally efficient surrogate model, such as the LSTM network developed in this work, which would ultimately be used for the online control task.
- **Advantages:** This approach allows for the exploration of the full operating range without exposing the physical plant to risk. Furthermore, the digital twin serves as a

valuable offline tool for the entire control system design. The performance of a surrogate model-based controller can be safely tested and tuned against the digital twin before deployment, significantly reducing risks during the commissioning phase.

- **Challenges:** The main bottleneck is the initial calibration of the mechanistic model. This task can be difficult, a challenge highlighted by structural identifiability theory. This creates a methodological paradox, as this pathway reintroduces the calibration problem the data-driven approach was intended to circumvent. The key difference, however, is that this complex calibration is a one-time, offline effort. Once established, the digital twin enables the rapid development and validation of flexible surrogate models without requiring further risky experiments on the physical plant.

2. The Direct Experimental Approach This is the classical system identification method, applied directly to the physical process.

- **Methodology:** An experimental campaign would be designed and executed on the real plant, applying a sequence of deliberate perturbations to the selected input variables. The corresponding output variables of interest would be recorded to create a real-world dataset for model training.
- **Advantages:** The resulting model would possess high fidelity, as it would be trained on data originating directly from the process it will control.
- **Challenges:** Deliberately perturbing an industrial process can interfere with production, leading to economic losses and operational risks. This suggests the approach is better suited for laboratory-scale systems or for the initial commissioning of a new plant, rather than for a facility in continuous operation.

3. The Pragmatic Approach via Historical Data: This pathway would leverage existing operational data.

- **Methodology:** Historical operational data would be collected from the plant's logs. A significant effort would then be concentrated on data pre-processing. From this refined dataset, the variables that will serve as model inputs and outputs could be retrospectively selected for model training.
- **Advantages:** It is a low-cost, low-risk approach that utilizes readily available data, reflecting the actual conditions and disturbances the plant has been subjected to.
- **Challenges:** The quality and richness of historical data are the main uncertainty. The primary limitation is that operational data often lacks sufficient "excitation", that is, the dynamic variation in the inputs needed to accurately model the system's response to future control actions.

A potential path forward could be an adaptive hybrid approach that combines the strengths of these methods. This process could begin by leveraging historical data to develop an initial predictive model and, concurrently, to perform a preliminary calibration of a mechanistic model. This calibrated "digital twin" could then be used to generate high-quality synthetic data, augmenting the historical dataset to enrich dynamics in poorly represented regions. This improved dataset would serve as a more robust foundation for training the surrogate model.

Furthermore, this work suggests a logical extension to multi-objective control. The current study demonstrated the framework's applicability to single-output problems, but the underlying architecture is flexible enough to model other process variables, such as effluent COD, VFA concentrations, or biogas quality. This could be achieved either by training a single multi-output neural network or by developing a bank of independent, specialized predictors for each variable. These models could then be integrated into a single MPC framework, where the controller would optimize a cost function that balances competing operational goals (for example, maximizing methane production while ensuring VFA concentrations remain below an inhibitory threshold, a strategy also proposed for model-based approaches (Azúa-Poblete et al., 2025)) all while respecting the operational constraints for each of the controlled variables. This modular approach could offer a effective strategy for holistic plant optimization.

This modular concept, if enabled by a reliable digital twin, also makes model-free RL a more practical alternative. Just as the digital twin can provide data to train a surrogate model for MPC, it can also serve as a training ground where an RL agent can interact safely and rapidly. This addresses the sample inefficiency that can hinder its direct application on a physical plant. Future work could then directly compare the performance of the model-based MPC against a model-free RL controller, both developed from the same digital twin, to determine which paradigm offers better performance when data generation is no longer a constraint. Finally, the computational demands of these methods do not appear to present a practical barrier. The slow dynamics of anaerobic digestion provide sufficient time for the NMPC optimization to run between control actions, suggesting these advanced techniques are suitable for real-time application.

9 References

Adekunle, K. F., & Okolie, J. A. (2015). A Review of Biochemical Process of Anaerobic Digestion. *Advances in Bioscience and Biotechnology*, *6*(3), 205–212. https://doi.org/10.4236/abb.2015.63020

Arzate, J. (2019). *Modeling and simulation of biogas production based on anaerobic digestion of energy crops and manure*. Technische Universität Berlin.

Azúa-Poblete, M., Cedeño, A. L., Agüero, J. C., Santos, L. O., Dewasme, L., Vande Wouwer, A., & García-Gen, S. (2025). Enhancing anaerobic digestion performance with offset-free model predictive control. *Journal of Water Process Engineering*, *78*, 108785. https://doi.org/10.1016/j.jwpe.2025.108785

Bakarji, J., Champion, K., Nathan Kutz, J., & Brunton, S. L. (2023). Discovering governing equations from partial measurements with deep delay autoencoders. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *479*(2276). https://doi.org/10.1098/rspa.2023.0422

Barahmand, Z., & Samarakoon, G. (2022). Sensitivity Analysis and Anaerobic Digestion Modeling: A Scoping Review. *Fermentation*, *8*(11). https://doi.org/10.3390/fermentation8110624

Batstone, D. J., Keller, J., Angelidaki, I., Kalyuzhnyi, S. V, Pavlostathis, S. G., Rozzi, A., Sanders, W. T. M., Siegrist, H., & Vavilin, V. A. (2002). The IWA Anaerobic Digestion Model No 1 (ADM1). *Water Science and Technology*, *45*(10), 65–73. https://iwaponline.com/wst/article-pdf/45/10/65/30636/65.pdf

Batstone, D. J., Keller, J., & Steyer, J. P. (2006). A review of ADM1 extensions, applications, and analysis: 2002-2005. *Water Science and Technology*, *54*(4), 1–10. https://doi.org/10.2166/wst.2006.520

Beltramo, T., & Hitzmann, B. (2019). Evaluation of the linear and non-linear prediction models optimized with metaheuristics: Application to anaerobic digestion processes. *Engineering in Agriculture, Environment and Food*, *12*(4), 397–403. https://doi.org/10.1016/j.eaef.2019.06.001

Bernard, O., Hadj-Sadok, Z., Dochain, D., Genovesi, A., & Steyer, J.-P. (2001). Dynamical Model Development and Parameter Identification for an Anaerobic Wastewater Treatment Process. *Biotechnology and Bioengineering*, *75*, 424–438.

Bolmanis, E., Dubencovs, K., Suleiko, A., & Vanags, J. (2023). Model Predictive Control—A Stand Out among Competitors for Fed-Batch Fermentation Improvement. *Fermentation*, *9*(3). https://doi.org/10.3390/fermentation9030206

Bornhöft, A., Hanke-Rauschenbach, R., & Sundmacher, K. (2013). Steady-state analysis of the Anaerobic Digestion Model No. 1 (ADM1). *Nonlinear Dynamics*, *73*(1–2), 535–549. https://doi.org/10.1007/s11071-013-0807-x

Brunton, S. L., & Kutz, J. N. (2022). *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press. faculty.washington.edu/sbrunton/DataBookV2.pdf

Camacho, E. F., & Bordons, C. (2004). *Model Predictive Control* (2nd ed.). Springer.

Catenacci, A., Grana, M., Malpei, F., & Ficara, E. (2021). Optimizing ADM1 calibration and input characterization for effective co-digestion modelling. *Water (Switzerland)*, *13*(21). https://doi.org/10.3390/w13213100

Cortés, L. G., Barbancho, J., Larios, D. F., Marin-Batista, J. D., Mohedano, A. F., Portilla, C., & de la Rubia, M. A. (2022). Full-Scale Digesters: Model Predictive Control with Online Kinetic Parameter Identification Strategy. *Energies*, *15*(22). https://doi.org/10.3390/en15228594

Cybenko, G. (1989). Approximation by Superpositions of a Sigmoidal Function. *Mathematics of Control, Signals, and Systems*, *2*(4), 303–314.

Dewasme, L. (2020). Neural network-based software sensors for the estimation of key components in brewery wastewater anaerobic digester: An experimental validation. *Water Science and Technology*, *80*(10), 1975–1985. https://doi.org/10.2166/wst.2020.019

Dewasme, L., Sbarciog, M., Rocha-Cózatl, E., Haugen, F., & Vande Wouwer, A. (2019). State and unknown input estimation of an anaerobic digestion reactor with experimental validation. *Control Engineering Practice*, *85*, 280–289. https://doi.org/10.1016/j.conengprac.2019.02.003

Donoso-Bravo, A., Mailier, J., Martin, C., Rodríguez, J., Aceves-Lara, C. A., & Vande Wouwer, A. (2011). Model selection, identification and validation in anaerobic digestion: A review. *Water Research*, *45*(17), 5347–5364. https://doi.org/10.1016/j.watres.2011.08.059

Fujimoto, S., van Hoof, H., & Meger, D. (2018). *Addressing Function Approximation Error in Actor-Critic Methods*. http://arxiv.org/abs/1802.09477

Gaida, D., Wolf, C., Meyer, C., Stuhlsatz, A., Lippel, J., Bäck, T., Bongards, M., & McLoone, S. (2012). State estimation for anaerobic digesters using the ADM1. *Water Science and Technology*, *66*(5), 1088–1095. https://doi.org/10.2166/wst.2012.286

García-Gen, S., Lema, J. M., & Rodríguez, J. (2013). Generalised modelling approach for anaerobic co-digestion of fermentable substrates. *Bioresource Technology*, *147*, 525–533. https://doi.org/10.1016/j.biortech.2013.08.063

García-Gen, S., Rodríguez, J., & Lema, J. M. (2014). Optimisation of substrate blends in anaerobic co-digestion using adaptive linear programming. *Bioresource Technology*, *173*, 159–167. https://doi.org/10.1016/j.biortech.2014.09.089

García-Gen, S., Santos, L. O., & Vande Wouwer, A. (2022). Application of a Nonlinear Model Predictive Controller to the Anaerobic Digestion of Readily Biodegradable Wastes. *IFAC-PapersOnLine*, *55*(7), 909–914. https://doi.org/10.1016/j.ifacol.2022.07.560

García-Gen, S., & Wouwer, A. Vande. (2021). A model-based optimisation strategy for the start-up of anaerobic co-digestion processes. *Renewable Energy*, *170*, 693–702. https://doi.org/10.1016/j.renene.2021.02.007

Giovannini, G., Sbarciog, M., Chamy, R., & Vande Wouwer, A. (2017). Derivation of a reduced-order model for anaerobic digestion control. *IWA Winery 2017*.

Gupta, R., Zhang, L., Hou, J., Zhang, Z., Liu, H., You, S., Sik Ok, Y., & Li, W. (2023). Review of explainable machine learning for anaerobic digestion. *Bioresource Technology*, *369*. https://doi.org/10.1016/j.biortech.2022.128468

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., & Levine, S. (2019). *Soft Actor-Critic Algorithms and Applications*. http://arxiv.org/abs/1812.05905

Haugen, F. (2014). *Optimal Design, Operation and Control of an Anaerobic Digestion Reactor* [PhD Dissertation]. Telemark University College.

Haugen, F., Bakke, R., & Lie, B. (2013). Adapting dynamic mathematical models to a pilot anaerobic digestion reactor. *Modeling, Identification and Control*, *34*(2), 35–54. https://doi.org/10.4173/mic.2013.2.1

He, K., Liu, Y., Tian, L., He, W., & Cheng, Q. (2024). Review in anaerobic digestion of food waste. *Heliyon*, *10*(7). https://doi.org/10.1016/j.heliyon.2024.e28200

Hellmann, S., Hempel, A.-J., Streif, S., & Weinrich, S. (2023). *Observability and Identifiability Analyses of Process Models for Agricultural Anaerobic Digestion Plants*. http://arxiv.org/abs/2301.05068

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer Feedforward Networks are Universal Approximators. *Neural Networks*, *2*(5), 359–366.

Jamilis, M., Nuñez, S., Garelli, F., & De Battista, H. (2018). Detectability and observability analysis of the anaerobic digestion process. *Congreso Argentino de Control Automático (AADECA) 2018*.

Jeong, K., Abbas, A., Shin, J., Son, M., Kim, Y. M., & Cho, K. H. (2021). Prediction of biogas production in anaerobic co-digestion of organic wastes using deep learning models. *Water Research*, *205*, 117697. https://doi.org/10.1016/j.watres.2021.117697

Jiménez-Ocampo, U. E., Vargas, A., & Moreno-Andrade, I. (2021). Methane production from food waste using a feedback control strategy in a sequencing batch reactor. *Water Science and Technology*, *84*(8), 1969–1980. https://doi.org/10.2166/wst.2021.370

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2019). *Continuous control with deep reinforcement learning*. http://arxiv.org/abs/1509.02971

Lima, D., Li, L., & Appleby, G. (2025). Biogas Production Modelling Based on a Semi-Continuous Feeding Operation in a Municipal Wastewater Treatment Plant. *Energies*, *18*(5). https://doi.org/10.3390/en18051065

Liu, Y., Tian, W., Xie, J., Huang, W., & Xin, K. (2023). LSTM-Based Model-Predictive Control with Rationality Verification for Bioreactors in Wastewater Treatment. *Water (Switzerland)*, *15*(9). https://doi.org/10.3390/w15091779

Ljung, L. (1999). *System Identification: Theory for the User* (2nd ed.). Prentice Hall.

M. Norgaard, O. Ravn, N.K. Poulsen, & L.K. Hansen. (2000). *Neural Networks for Modelling and Control of Dynamic Systems*. Springer London.

MathWorks. (2025). *Bayesian Optimization Workflow*. https://la.mathworks.com/help/stats/bayesian-optimization-workflow.html

Mauky, E., Weinrich, S., Nägele, H. J., Jacobi, H. F., Liebetrau, J., & Nelles, M. (2016). Model Predictive Control for Demand-Driven Biogas Production in Full Scale. *Chemical Engineering and Technology*, *39*(4), 652–664. https://doi.org/10.1002/ceat.201500412

McCormick, M., & Villa, A. E. P. (2019). LSTM and 1-D Convolutional Neural Networks for Predictive Monitoring of the Anaerobic Digestion Process. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11731 LNCS*, 725–736. https://doi.org/10.1007/978-3-030-30493-5_65

Méndez-Acosta, H. O., Campos-Delgado, D. U., Femat, R., & González-Alvarez, V. (2005). A robust feedforward/feedback control for an anaerobic digester. *Computers and Chemical Engineering*, *29*(7), 1613–1623. https://doi.org/10.1016/j.compchemeng.2005.01.005

Mockus J.B, & Mockus L.J. (1991). Bayesian Approach to Global Optimization and Application to Multiobjective and Constrained Problems. *Journal of Optimization Theory and Applications*, *70*(1), 157–172.

Moradvandi, A., Heegstra, S., Ceron-Chafla, P., Schutter, B. De, Abraham, E., & Lindeboom, R. E. F. (2025). Model predictive control of feed rate for stabilizing and enhancing biogas production in anaerobic digestion under meteorological fluctuations. *Journal of Process Control*, *147*. https://doi.org/10.1016/j.jprocont.2025.103375

Mudzanani, K. E., Phadi, T. T., Iyuke, S. E., & Daramola, M. O. (2023). Enhancing Methane Production through Anaerobic Co-Digestion of Sewage Sludge: A Modified ADM1 Model Approach. *Fermentation*, *9*(9). https://doi.org/10.3390/fermentation9090833

Nguyen, D., Gadhamshetty, V., Nitayavardhana, S., & Khanal, S. K. (2015). Automatic process control in anaerobic digestion technology: A critical review. *Bioresource Technology*, *193*, 513–522. https://doi.org/10.1016/j.biortech.2015.06.080

Ordace, A., Ionescu, C. M., Vannecke, T. P. W., Volcke, E. I. P., Nascu, I., & De Keyser, R. (2012). Predictive Control of Anaerobic Digestion of Wastewater Sludge. A Feasibility Study. *2012 16th International Conference on System Theory, Control and Computing (ICSTCC)*. https://www.researchgate.net/publication/261226016

Petré, M. (2025). *Data-driven Nonlinear Model Predictive Control Using NARX Models: An Application of Takens' Embedding Theorem: Vol. TRITA – EECS-EX* [Master's Thesis]. KTH Royal Institute of Technology.

Piadeh, F., Offie, I., Behzadian, K., Rizzuto, J. P., Bywater, A., Córdoba-Pachón, J. R., & Walker, M. (2024). A critical review for the impact of anaerobic digestion on the sustainable development goals. *Journal of Environmental Management*, *349*. https://doi.org/10.1016/j.jenvman.2023.119458

Pino Santana, A. (2025). *Data-Driven Modeling of Anaerobic Digestion Processes with a View to Process Control: A First Approach*. Universidad Técnica Federico Santa María (UTFSM).

Rodgers, S., Bowler, A., Wells, L., Lee, C. S., Hayes, M., Poulston, S., Lester, E., Meng, F., McKechnie, J., & Conradie, A. (2024). A surrogate model for the economic evaluation of renewable hydrogen production from biomass feedstocks via supercritical water gasification. *International Journal of Hydrogen Energy*, *49*, 277–294. https://doi.org/10.1016/j.ijhydene.2023.08.016

Rodríguez-Jara, M., Velasco-Pérez, A., Vian, J., Vigueras-Carmona, S. E., & Puebla, H. (2023). Robust Control Based on Modeling Error Compensation of Microalgae Anaerobic Digestion. *Fermentation*, *9*(1). https://doi.org/10.3390/fermentation9010034

Rutland, H., You, J., Liu, H., Bull, L., & Reynolds, D. (2023). A Systematic Review of Machine-Learning Solutions in Anaerobic Digestion. *Bioengineering*, *10*(12). https://doi.org/10.3390/bioengineering10121410

Stark, J., Broomheadt, D. S., Davies, M. E., & Huket, J. (1997). Takens Embedding Theorems for Forced and Stochastic Systems. *Nonlinear Analysis, Theory, Methods & Applications*, *30*(8), 5303–5314. https://doi.org/10.1016/S0362-546X(96)00149-6

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction* (2nd ed.). The MIT Press.

Takens, F. (1981). Detecting strange attractors in turbulence. In D. A. Rand & L.-S. Young (Eds.), *Dynamical Systems and Turbulence, Warwick 1980* (Vol. 898, pp. 366–381). Springer-Verlag.

Tawai, A., & Sriariyanun, M. (2022). Nonlinear Optimization-Based Robust Control Approach for a Two-Stage Anaerobic Digestion Process. *Journal of Chemistry*, *2022*. https://doi.org/10.1155/2022/8966350

Yoshida, K., & Shimizu, N. (2020). Biogas production management systems with model predictive control of anaerobic digestion processes. *Bioprocess and Biosystems Engineering*, *43*(12), 2189–2200. https://doi.org/10.1007/s00449-020-02404-7

Zarzycki, K., & Ławryńczuk, M. (2021). LSTM and GRU neural networks as models of dynamical processes used in predictive control: A comparison of models developed for two chemical reactors. *Sensors*, *21*(16). https://doi.org/10.3390/s21165625

Appendix A: AM2 Simulation Datasets

A.1: Training Dataset (Dataset 1)

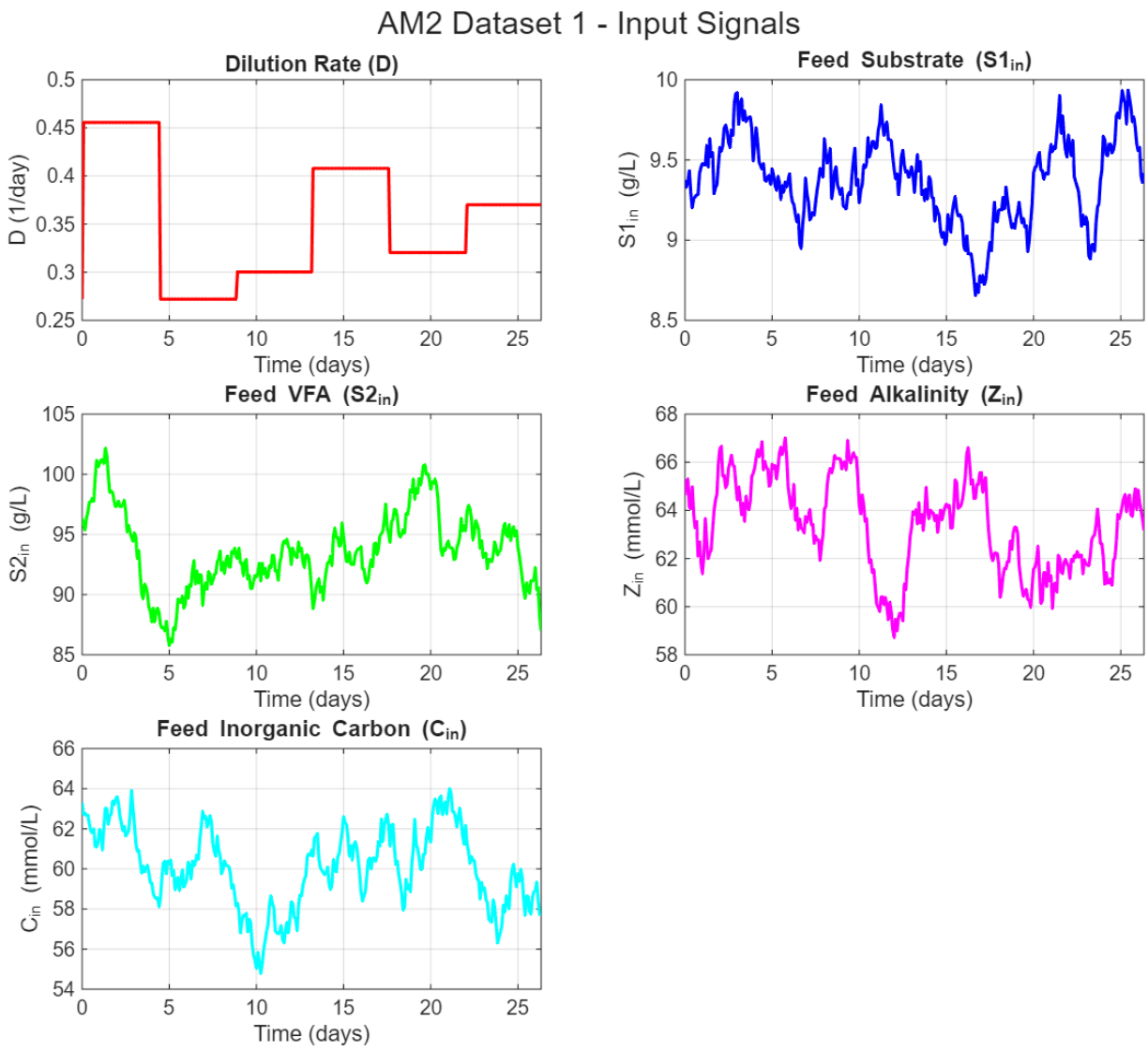


Figure A. 1: Input signals for the AM2 training dataset, including the manipulated dilution rate (D) and stochastic variations in the feed composition.

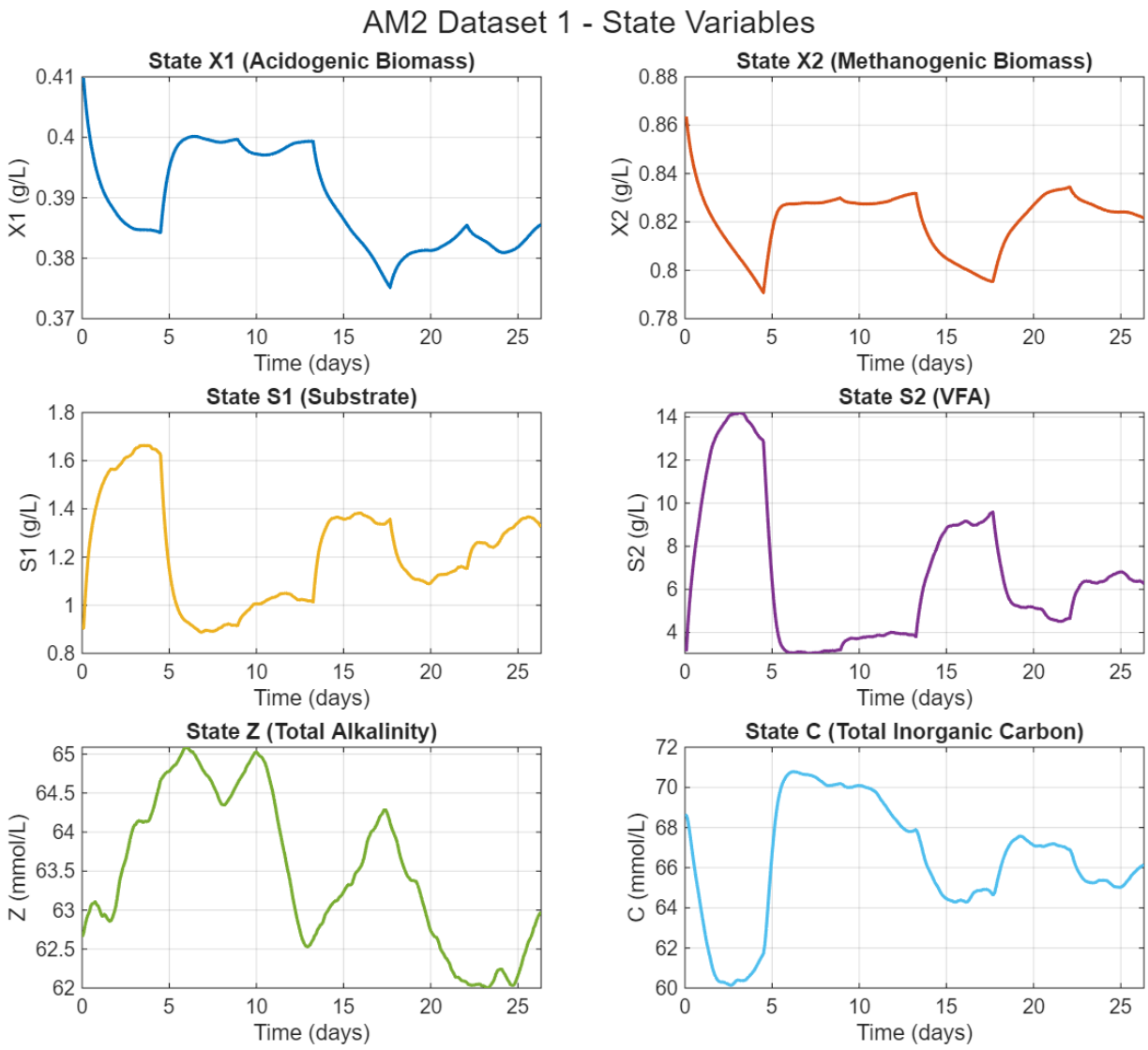


Figure A. 2: Evolution of the six state variables of the AM2 model during the training simulation.

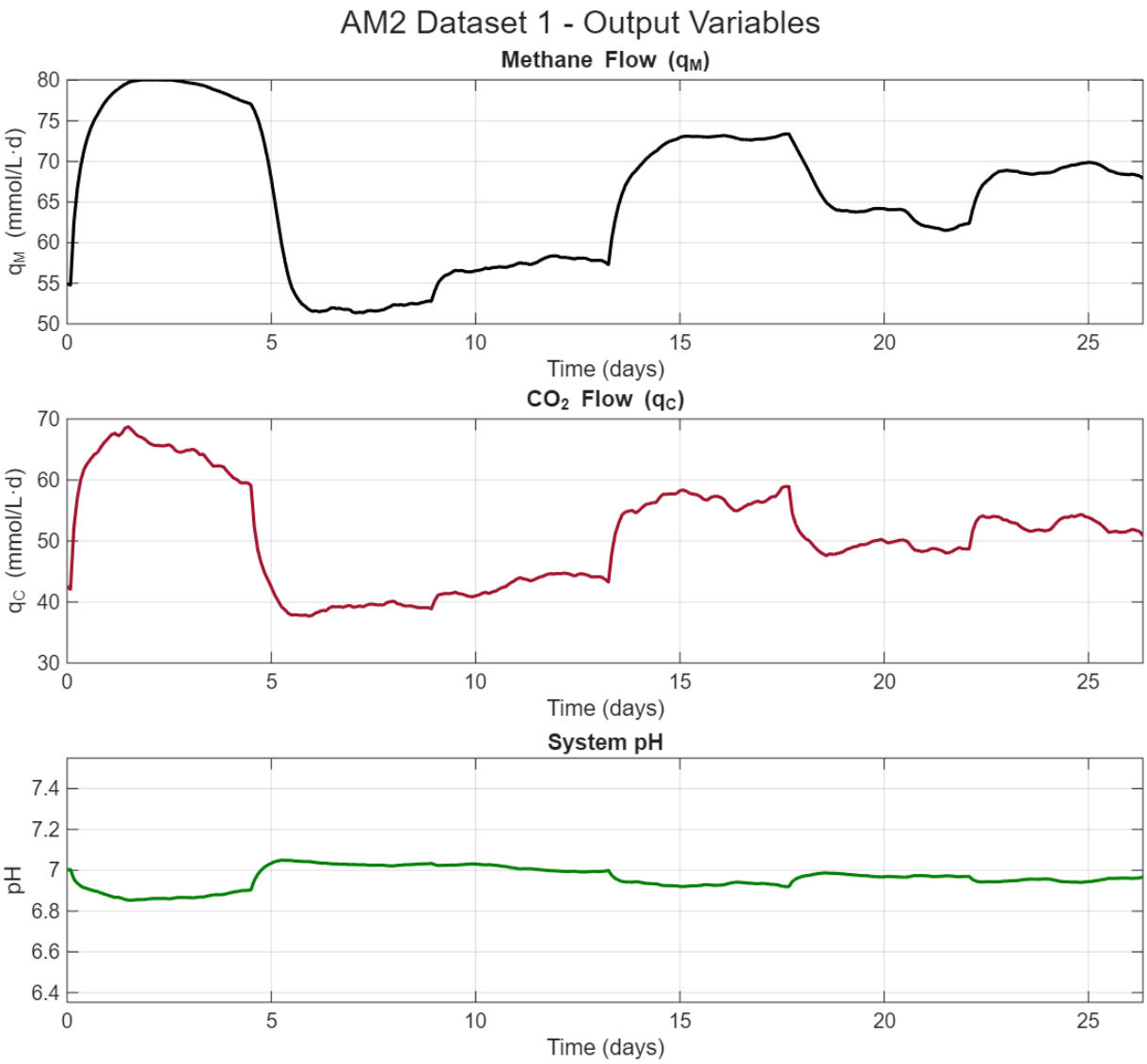


Figure A. 3: Output variables from the AM2 training simulation, including the target variable, methane flow rate (q_M).

A.2: Testing Dataset (Dataset 2)

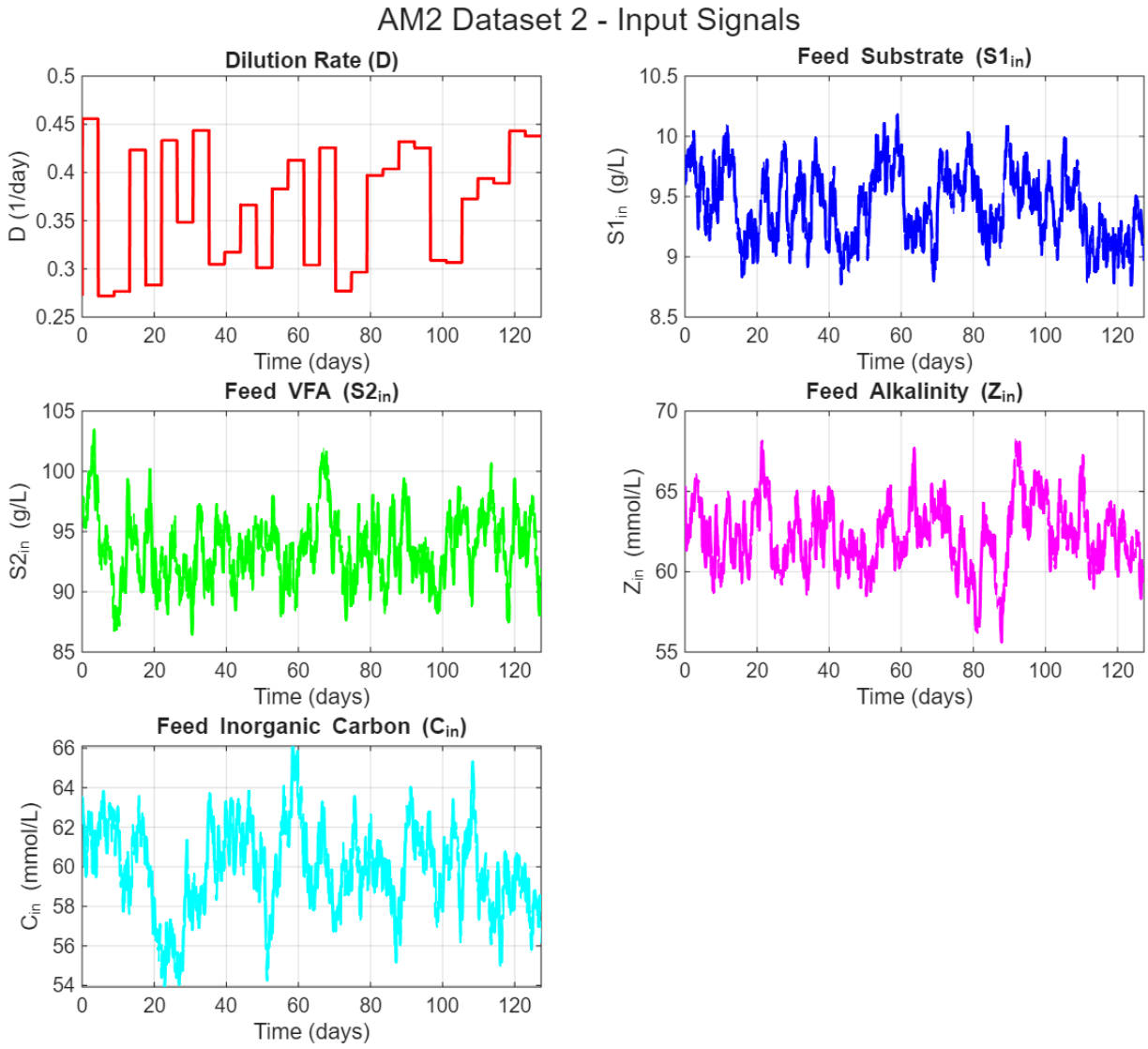


Figure A. 4: Input signals for the AM2 testing dataset, featuring a different sequence of perturbations.

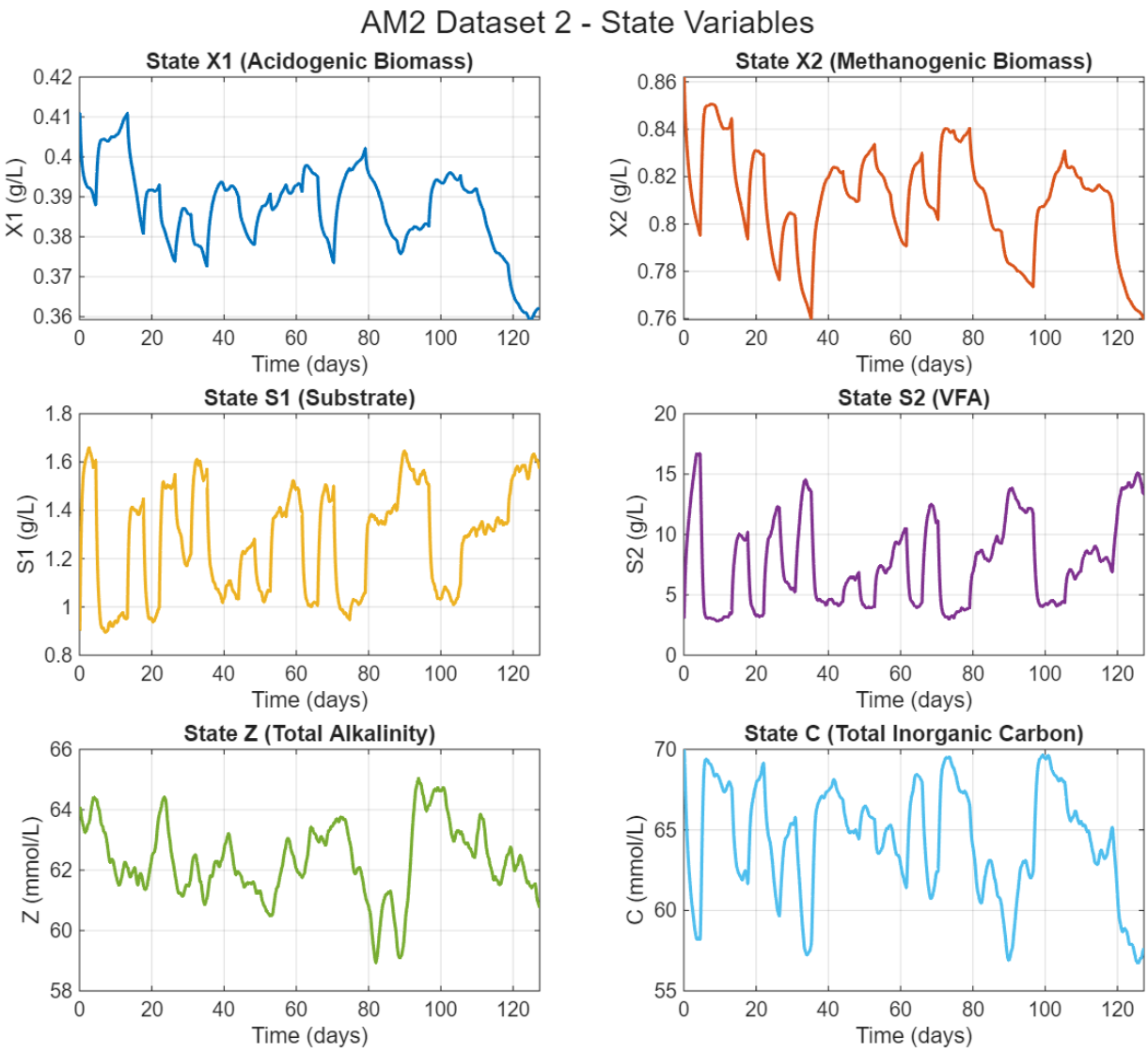


Figure A. 5: Evolution of the six state variables of the AM2 model during the testing simulation.

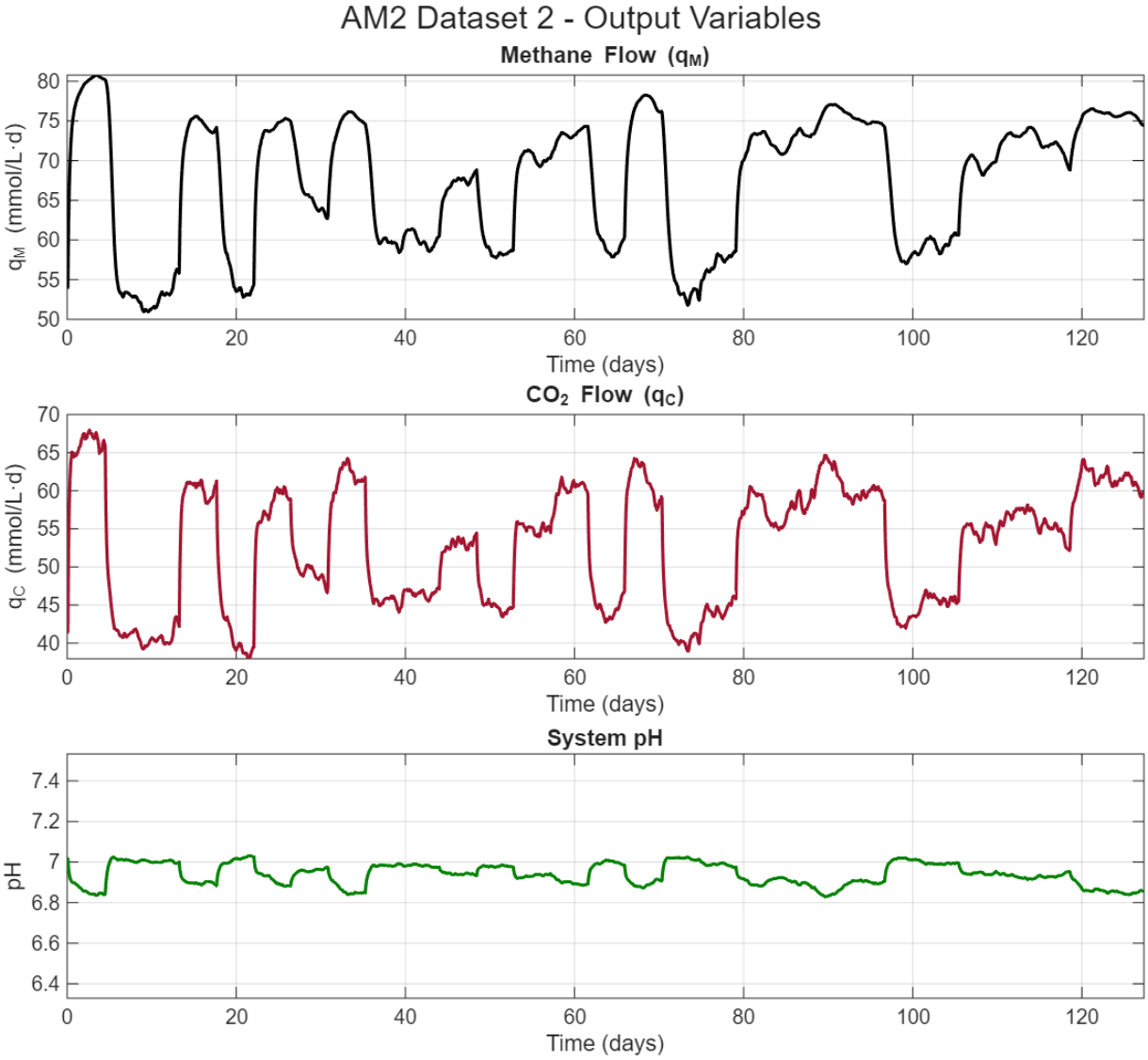


Figure A. 6: Output variables from the AM2 testing simulation.

Appendix B: ADM1 Simulation Datasets

B.1: Training Dataset (Dataset 1)

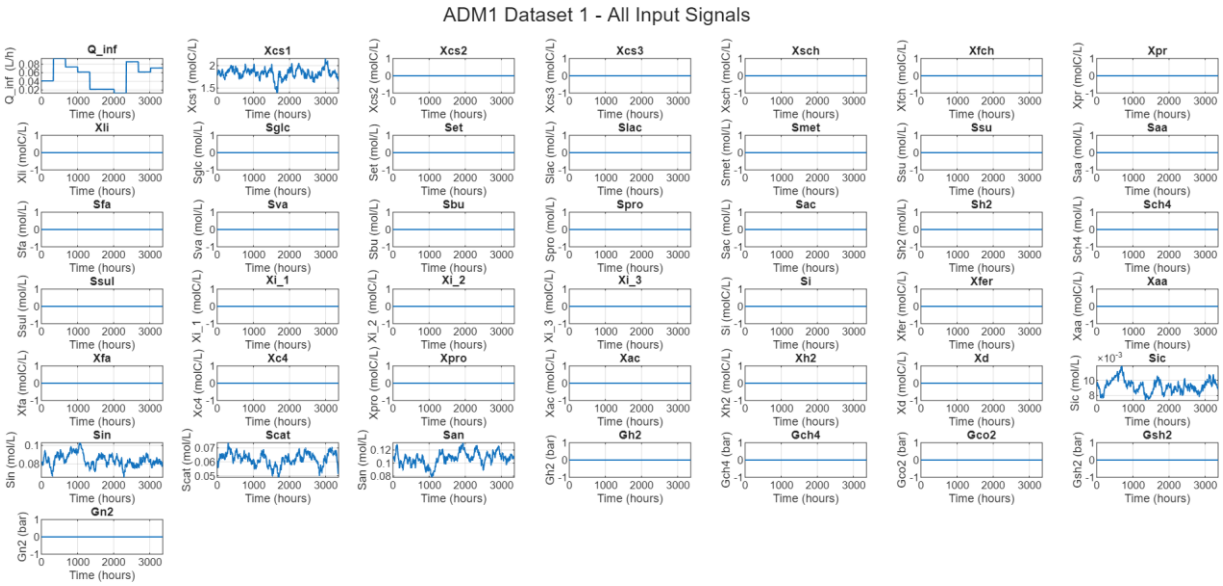


Figure B. 1: Input signals for the ADM1 training dataset, including the manipulated influent flow rate (Q_{in}) and stochastic variations in the feed composition.

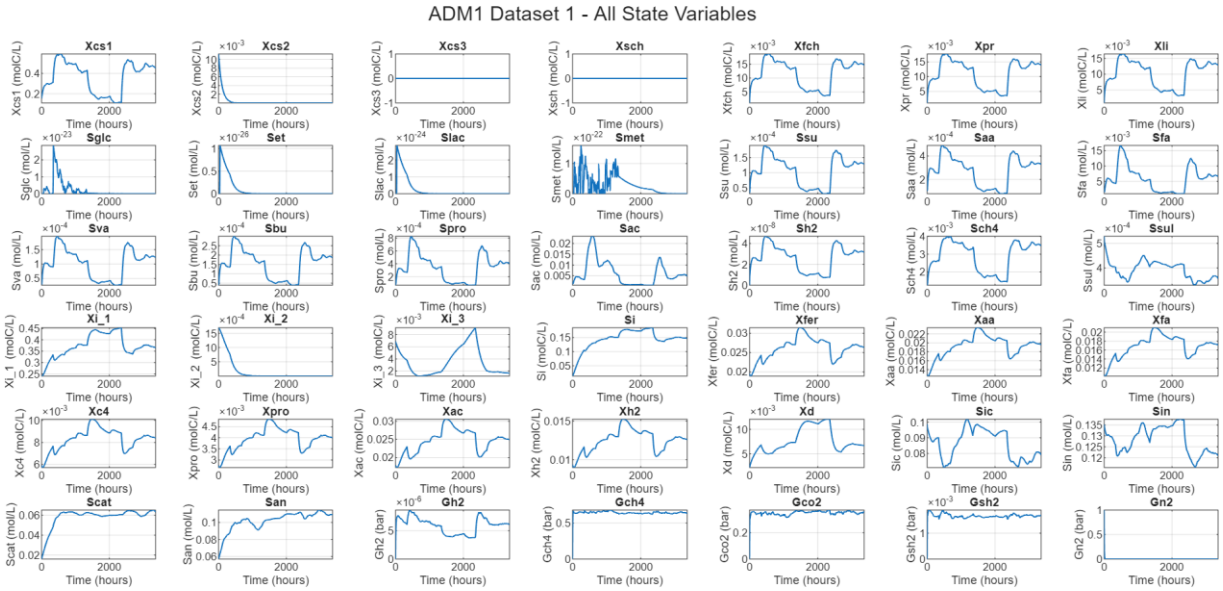


Figure B. 2: Evolution of all state variables of the ADM1 model during the training simulation.

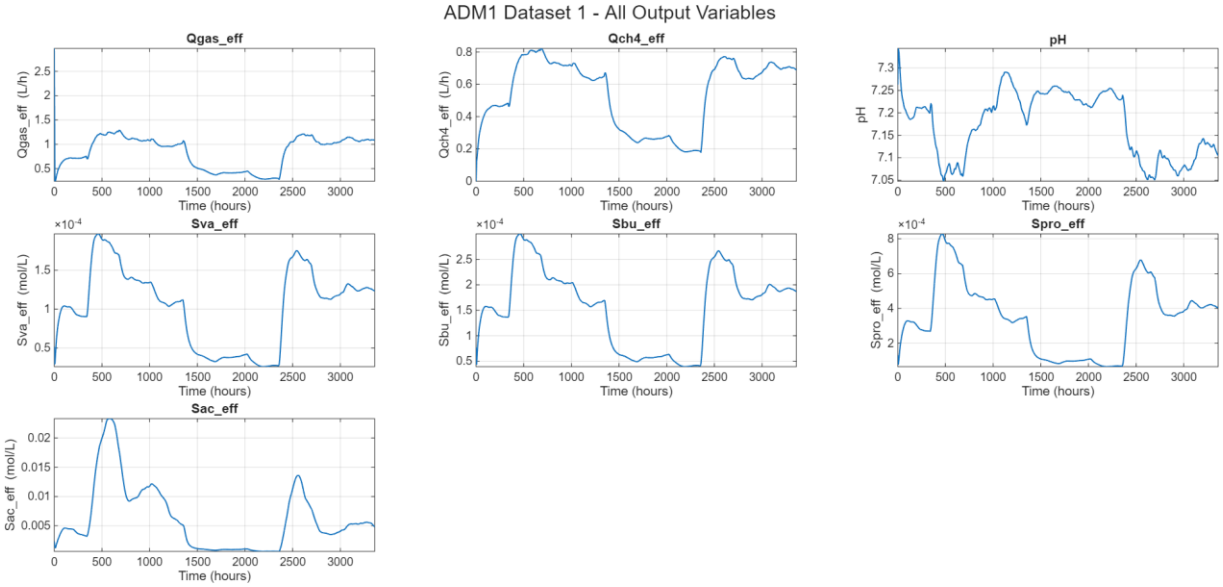


Figure B. 3: Process output variables from the ADM1 training simulation.

B.2: Testing Dataset (Dataset 2)

Figure B. 4: Input signals for the ADM1 testing dataset, featuring a different sequence of perturbations.

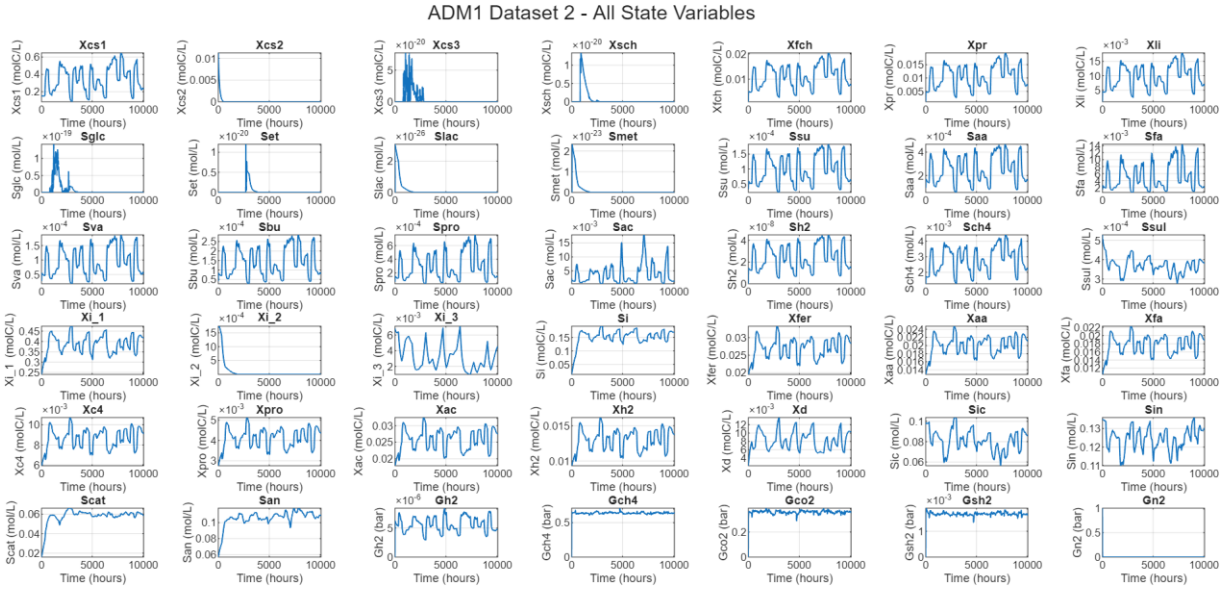


Figure B. 5: Evolution of all state variables of the ADM1 model during the testing simulation.

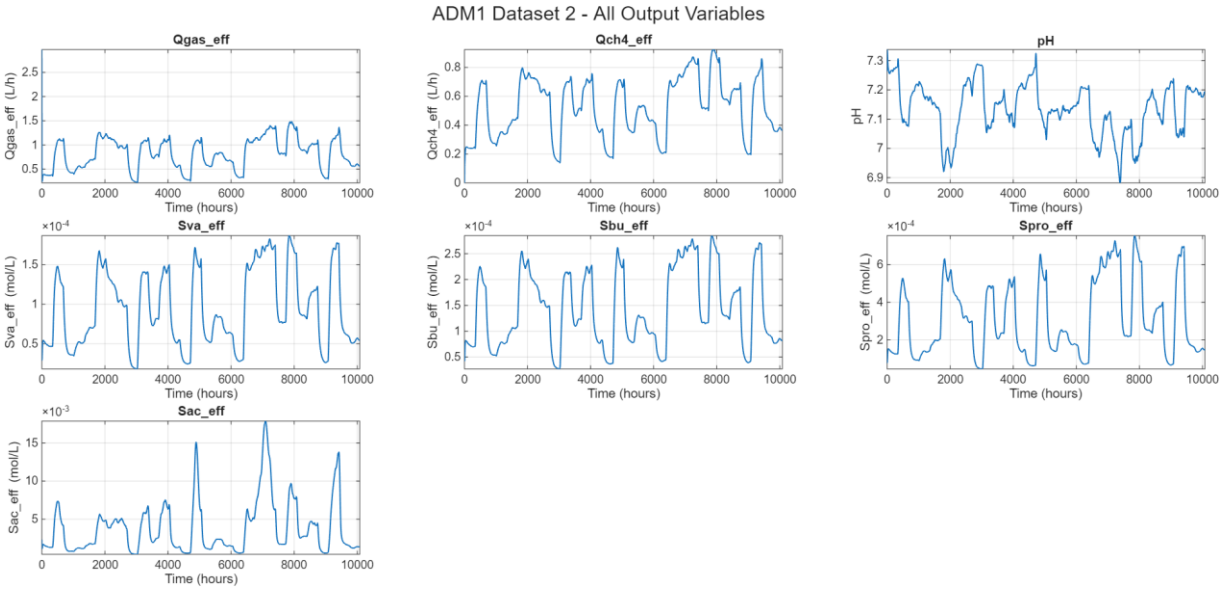


Figure B. 6: Process output variables from the ADM1 testing simulation.

Appendix C: PID Tuning

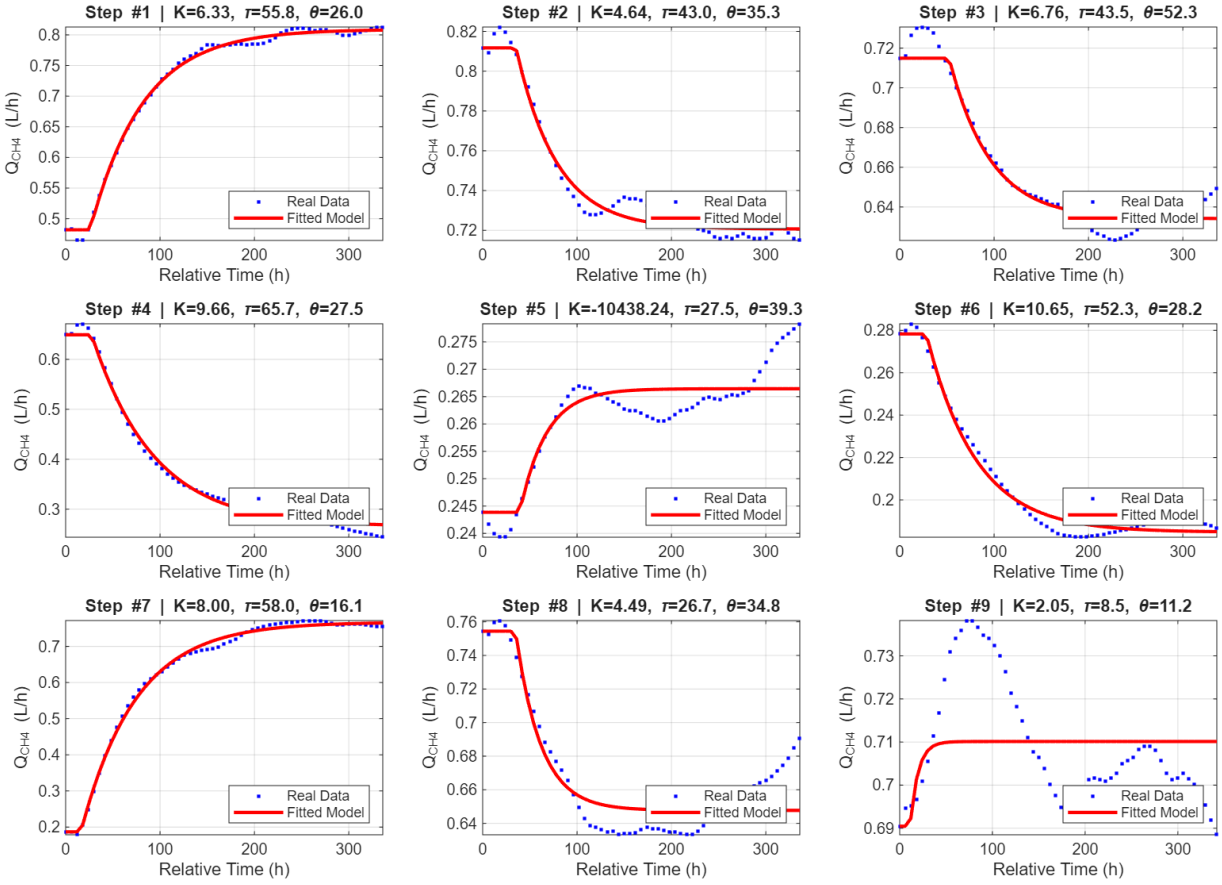


Figure C. 1: Example of FOPDT model fits for each step change in the training data, used to derive PID tuning parameters.