SLOVAK UNIVERSITY OF TECHNOLOGY
IN BRATISLAVA
FACULTY OF CHEMICAL AND FOOD TECHNOLOGY

Reg. No.: FCHPT-4622-91815

Safety-First Control and Monitoring for Industrial Systems

DISSERTATION THESIS

2026 **Ing. Marek Wadinger**

SLOVAK UNIVERSITY OF TECHNOLOGY
IN BRATISLAVA
FACULTY OF CHEMICAL AND FOOD TECHNOLOGY

Reg. No.: FCHPT-4622-91815

Safety-First Control and Monitoring for Industrial Systems

DISSERTATION THESIS

Study programme:	Process Control
Study field:	Cybernetics
Training workspace:	Institute of Information Engineering, Automation, and Mathematics
Thesis supervisor:	prof. Ing. Michal Kvasnica, PhD.

2026

Ing. Marek Wadinger

Slovak University of Technology in Bratislava
Department of Information Engineering and Process Control

Faculty of Chemical and Food Technology
Academic year: 2025/2026
Reg. No.: FCHPT-4622-91815

STU
FCHPT

DISSERTATION THESIS TOPIC

Student:	**Ing. Marek Wadinger**
Student's ID:	91815
Study programme:	Process Control
Study field:	Cybernetics
Thesis supervisor:	prof. Ing. Michal Kvasnica, PhD.
Head of department:	prof. Ing. Miroslav Fikar, DrSc.

Topic: **Safety-First Control and Monitoring for Industrial Systems**

Language of thesis: English

Specification of Assignment:

The transition of industrial and energy systems toward sustainable, deployable automation is constrained by a persistent gap between theoretical advances in control and monitoring and their reliable operation in real plants. Conservative robust controllers leave economic value on the table, aggressive economic controllers expose operators to risk, and the data-driven models that promise to close this gap frequently lack the interpretability, physical consistency, and lifecycle adaptivity required for safety-critical deployment. The dissertation shall address this gap by developing methods that pursue economic performance without compromising operational safety.

The aims of the dissertation are:

1. Develop a unified methodological framework that bridges physics-based modeling, interpretable machine learning, risk-aware control, and online monitoring.
2. Investigate economic and risk-aware control architectures that explicitly trade off economic objectives against operational risk.
3. Design data-driven modeling techniques that augment physics-informed models with interpretable machine learning.
4. Propose adaptive anomaly and change-point detection mechanisms that operate on streaming SCADA data.

Deadline for submission of Dissertation thesis:	31. 05. 2026
Approval of assignment of Dissertation thesis:	30. 05. 2026
Assignment of Dissertation thesis approved by:	prof. Ing. Miroslav Fikar, DrSc. – Chairperson of Field of Study Board

Acknowledgments

I want to express my gratitude to my thesis supervisor, prof. Ing. Michal Kvasnica, PhD., who not only sparked my interest in cutting-edge automation and information solutions and gave me numerous opportunities to challenge my critical thinking and broaden my horizons but also guided me through the research process with his practical approach, which helped me to find motivation for my research activities. I would also like to thank my parents, who warmly approached my endless question in childhood and supported me in achieving my academic goals.

Abstract

Industrial and energy systems underpin the goods, power, and food on which modern society depends, and their transition toward sustainability is making them simultaneously more complex, more interconnected, and more sensitive to disturbance. Reliable operation of such systems rests on three theoretical pillars: data-driven dynamical models that capture nonlinear behavior, model predictive control that turns those models into economically optimal actions, and real-time monitoring that safeguards operation against drift and faults. These pillars have, however, evolved as largely separate research communities, leaving industrial deployments with interpretability gaps, brittle assumptions, and limited adaptivity over the system lifecycle. The general problem addressed in this dissertation is how to couple interpretable modeling, risk-aware economic control, and adaptive monitoring into a single methodological framework deployable on real industrial infrastructure. Here we show, through seven interconnected contributions validated on industrial-scale case studies, that these pillars can be unified into a practical toolkit for sustainable industrial automation. For control, a nonlinear economic predictive controller for greenhouses jointly optimizes crop yield, energy use, and the social cost of CO_2, while a layered risk-aware energy management system for microgrids delivers more than 3000 EUR/month of profit and 159 EUR/kW annualized yield on commercial battery storage. For monitoring, an inverse-cumulative-distribution monitor, the Adaptable and Interpretable framework for anomaly Detection (AID), and a truncated online Dynamic Mode Decomposition with Control (toDMDc) provide plug-and-play dynamic limits, root-cause isolation, and interpretable change-point detection compatible with existing SCADA infrastructure. For modeling, an LSTM-enhanced Deep Koopman model achieves a threefold reduction in mean absolute error over extended Dynamic Mode Decomposition for nonlinear dynamics with input delay, and a physics-informed Dynamic Mode Decomposition with Control (piDMDc) reduces resistance prediction error by 61% on a 3 MW industrial chlor-alkali electrolyzer while tracking impurity-driven degradation. These results demonstrate that the three layers reinforce one another: monitoring detects when controller assumptions are violated, interpretable models explain what and why, and risk-aware control converts this lifecycle awareness into economic and environmental value. The framework thereby reframes sustainable industrial automation as a lifecycle reliability problem, and charts a route by which advanced control, machine learning, and process safety can jointly underwrite the energy transition.

Keywords: Safety-critical and resilient systems, Industrial artificial intelligence, Physics informed and grey box model identification, Learning methods for control, Incremental Learning

Abstrakt

Priemyselné a energetické systémy tvoria základ pre tovary, energetiku a potraviny, na ktorých závisí moderná spoločnosť, a ich prechod k udržateľnosti ich zároveň robí zložitejšími, prepojenejšími a citlivejšími na narušenia. Spoľahlivá prevádzka takýchto systémov spočíva na troch teoretických pilieroch: dátovo riadených dynamických modeloch, ktoré zachytávajú nelineárne správanie, modelovom prediktívnom riadení, ktoré tieto modely premieňa na ekonomicky optimálne akcie, a monitorovaní v reálnom čase, ktoré chráni prevádzku pred odchýlkami a poruchami. Tieto piliere sa však vyvíjali ako do veľkej miery oddelené výskumné komunity, čo v priemyselných nasadeniach zanechalo medzery v interpretovateľnosti, krehké predpoklady a obmedzenú adaptabilitu počas životného cyklu systému. Všeobecným problémom, ktorým sa zaoberá táto dizertačná práca, je, ako spojiť interpretovateľné modelovanie, ekonomické riadenie s vedomím rizika a adaptívne monitorovanie do jedného metodického rámca, ktorý je možné nasadiť v reálnej priemyselnej infraštruktúre. V tejto práci prostredníctvom siedmich prepojených príspevkov overených na prípadových štúdiách v priemyselnom meradle ukazujeme, že tieto piliere je možné zjednotiť do praktickej sady nástrojov pre udržateľnú priemyselnú automatizáciu. V oblasti riadenia nelineárny ekonomický prediktívny regulátor pre skleníky spoločne optimalizuje úrodu, spotrebu energie a sociálne náklady CO_2, zatiaľ čo vrstvený systém riadenia energie s vedomím rizika pre mikrosiete prináša zisk viac ako 3000 EUR mesačne a ročný výnos 159 EUR/kW na komerčných batériových úložiskách. V oblasti monitorovania poskytujú monitor inverznej kumulatívnej distribúcie, rámec AID (Adaptable and Interpretable framework for anomaly Detection) a skrátená online metóda toDMDc (Dynamic Mode Decomposition with Control) dynamické limity typu „plug-and-play“, identifikáciu základných príčin a interpretovateľnú detekciu bodov zmeny, ktoré sú kompatibilné so súčasnou infraštruktúrou SCADA. V oblasti modelovania dosahuje model Deep Koopman vylepšený o LSTM trojnásobné zníženie strednej absolútnej chyby v porovnaní s rozšírenou dynamickou dekompozíciou režimov pre nelineárnu dynamiku so vstupným oneskorením, a fyzikálne informovaná dynamická dekompozícia režimov s riadením (piDMDc) znižuje chybu predikcie odporu o 61 % na priemyselnom chlóralkalickom elektrolyzéri s výkonom 3 MW, pričom sleduje degradáciu spôsobenú nečistotami. Tieto výsledky dokazujú, že tri vrstvy sa navzájom posilňujú: monitorovanie detekuje, kedy

sú porušené predpoklady regulátora, interpretovateľné modely vysvetľujú čo a prečo, a riadenie s vedomím rizika premieňa toto povedomie o životnom cykle na ekonomickú a environmentálnu hodnotu. Rámec týmto preformuluje udržateľnú priemyselnú automatizáciu ako problém spoľahlivosti životného cyklu a načrtáva cestu, ktorou pokročilé riadenie, strojové učenie a bezpečnosť procesov môžu spoločne podporiť energetickú transformáciu.

Kľúčové slová: systémy kritické na bezpečnosť, priemyselná umelá inteligencia, fyzikálne informovaná identifikácia modelov, metódy riadenie na báze učenia, inkrementálne učenie

Biographical Sketch

Marek Wadinger was born in Bratislava, Slovakia on July 23, 1998. He received his Bc. degree in Cybernetics from Slovak University of Technology in Bratislava in 2020, and then joined the Department of Information Engineering and Process Control (DIEPC) at the Slovak University of Technology in Bratislava as a master student. He received his Ing. degree in Cybernetics from the Slovak University of Technology in Bratislava in 2022. He is currently a PhD student at the Department of Information Engineering and Process Control at the Slovak University of Technology in Bratislava.

Marek Wadinger is a student of Professor Michal Kvasnica. His research interests include control theory, machine learning, and their application to industrial systems. He is particularly interested in the application of control theory to industrial systems, and the use of machine learning to improve the safety of control systems. He is also interested in the application of large language models to control systems.

Contents

List of Figures

List of Tables

Notation

Symbol	Description
General Conventions	
x, T, λ	Scalars are denoted by italic letters (Latin or Greek)
$\boldsymbol{x}$, $\boldsymbol{\theta}$	Vectors are denoted by bold lowercase letters
$\boldsymbol{A}$, $\boldsymbol{\Phi}$	Matrices are denoted by bold uppercase letters
$\mathcal{X}$, $\mathcal{S}$	Sets and spaces are denoted by calligraphic letters
$\mathbb{R}$, $\mathbb{C}$, $\mathbb{N}$	Number sets are denoted by blackboard bold
$\boldsymbol{x}_k$	Bold variable with italic subscript denotes a time-indexed quantity
T_{ext}	Upright subscript denotes a descriptive label (e.g., "external")
$P_{\mathrm{ch},k}$	Combined upright label and italic index (e.g., charging power at step k)
$\tilde{}$	Tilde denotes a projected or estimated quantity
$\bar{}$	Bar denotes an augmented quantity
$\hat{}$	Hat denotes a measured or estimated value
$'$	Prime denotes the time-shifted (successor) value
Operators and Notation	
$(\cdot)^\top$	transpose
$(\cdot)^+$	Moore–Penrose pseudoinverse
$(\cdot)^{-1}$	inverse
$\\|\cdot\\|_F$	Frobenius norm
$\\|\cdot\\|_2$	Euclidean (ℓ_2) norm

continued on next page

Notation (continued)

Symbol	Description
$\circ$	Function composition
$\triangleq$	Defined as
$\frac{\partial\,\cdot}{\partial\,t}$	Ordinary derivative with respect to t
$\mathbb{E}[\cdot]$	Expected value
$\Pr(\cdot)$	Probability
arg min	Argument of the minimum
Indices and Dimensions	
k	Discrete time index
i, j, l, u	General indices (signal, cell, summation, lower, upper)
n	State dimension or number of snapshots
m	Measurement dimension or number of states
l	Number of control inputs
r	Rank of SVD truncation or subspace dimensionality
h	Number of time delays (embedding lag)
c	Number of snapshot columns or batch size
N	Prediction horizon
States and Inputs	
$\boldsymbol{x}_k \in \mathbb{R}^n$	State vector at time k
$\boldsymbol{u}_k \in \mathbb{R}^l$	Control input vector at time k
$\hat{\boldsymbol{x}}_k$	Measured or estimated state at time k
$F(\cdot)$	Nonlinear state transition map
$f(\cdot)$	System dynamics function
Koopman Operator Theory	
$\mathcal{K}$	Koopman operator (infinite-dimensional, linear)
$g : \mathcal{M} \to \mathbb{C}$	Scalar-valued observable
$\boldsymbol{g}_{\boldsymbol{x}}$	Vector of observables
φ_j	Koopman eigenfunction
λ_j	Koopman eigenvalue

continued on next page

Notation (continued)

Symbol	Description
v_j	Koopman mode
$\boldsymbol{K} \in \mathbb{R}^{p \times p}$	Finite-dimensional Koopman matrix
Dynamic Mode Decomposition	
$\boldsymbol{A}_k$	DMD state transition operator at time k
$\boldsymbol{B}_k$	Control input matrix at time k
$\boldsymbol{X}, \boldsymbol{X}'$	Snapshot matrices (current and time-shifted)
$\boldsymbol{\Theta}$	Control input snapshot matrix
$\tilde{\boldsymbol{U}}, \tilde{\boldsymbol{\Sigma}}, \tilde{\boldsymbol{V}}$	Truncated SVD components (left singular vectors, singular values, right singular vectors)
$\tilde{\boldsymbol{A}}$	DMD operator projected onto POD modes
$\boldsymbol{W}$	DMD eigenvector matrix
$\boldsymbol{\Lambda}$	Diagonal matrix of DMD eigenvalues
$\boldsymbol{\Phi}$	Full-dimensional DMD modes
$\bar{\boldsymbol{A}}$	Augmented system matrix $[\boldsymbol{A}\ \boldsymbol{B}]$
$\bar{\boldsymbol{X}}$	Augmented snapshot matrix (states and controls)
$\boldsymbol{P}_k$	Precision matrix (online DMD)
$\boldsymbol{Q}_k$	Lag covariance matrix (online DMD)
Control and Optimization	
$\ell(\cdot, \cdot)$	Stage cost function
$V_{\mathrm{f}}(\cdot)$	Terminal cost function
J	Objective function (profit or cost)
VaR_α	Value at Risk at confidence level α
CVaR_α	Conditional Value at Risk at confidence level α
α	Risk aversion parameter or confidence level
$s \in \mathcal{S}$	Scenario index and set of scenarios
$\pi(s)$	Probability of scenario s
N_S	Number of scenarios
Energy System Variables	

continued on next page

Notation (continued)

Symbol	Description
$E(k)$	Battery state of charge at time k
$P_{\mathrm{ch}}, P_{\mathrm{dch}}$	Charging and discharging power
$P_{\mathrm{imp}}, P_{\mathrm{exp}}$	Grid import and export power
P_{RES}	Renewable generation power
P_{load}	Load demand power
P_{DA}	Day-ahead scheduled power
$\eta_{\mathrm{ch}}, \eta_{\mathrm{dch}}$	Charging and discharging efficiency
E_{avail}	Available battery energy capacity
$\underline{\beta}, \overline{\beta}$	Lower and upper state-of-charge bounds
ΔT	Time step duration
Monitoring and Detection	
μ_i, s_i^2	Online mean and variance (Welford's algorithm)
q	Anomaly score threshold
$\mathcal{N}, \mathcal{N}^{-1}$	Gaussian CDF and inverse CDF
t_{e}	Expiration period (rolling window size)
t_{c}	Time constant (adaptation speed)
t_{a}	Adaptation period (change-point duration threshold)
t_{g}	Grace period (initial calibration)
T	Probabilistic detection threshold
$\mathcal{E}(\cdot, \cdot)$	Reconstruction error function
Q_k	Detection statistic (reconstruction error ratio)
Degradation Modeling	
$\boldsymbol{SoH}_k$	State of Health vector at time k
RUL	Remaining Useful Life
τ_{f}	Predefined failure threshold for SoH
$\boldsymbol{\lambda}$	Cell-specific degradation rate parameter
$\Delta \boldsymbol{x}_k$	Resistance deviation (measured minus predicted)
k_{f}	Failure time
$k_{\max}$	Maximum cell lifetime

Chapter 1

Introduction

1.1 Motivation: The Sustainability Imperative in Industrial Automation

The transformation of industrial systems toward sustainability represents one of the defining challenges of our era. Increasingly complex control systems offer enormous opportunities for resource utilization and interoperation of spatially distributed assets. Individual units may exhibit various levels of dependence on each other, enabling both completely independent operation and full cooperation, thereby democratizing the exchange of resources across energy networks, manufacturing facilities, and infrastructure systems.

Indeed, modern control systems in real-world applications deliver substantial economic, environmental, and social benefits, creating venues for more fair and efficient distribution of resources. Microgrids integrating battery energy storage systems and renewable energy sources are critical to the sustainable transformation of the energy sector. As the expansion of renewables increases variability in power networks, effective microgrid control becomes essential for maintaining supply-demand balance while supporting broader grid stability. Economic aspects are equally crucial, ensuring that microgrid operations are not only environmentally sustainable but also financially viable for operators and stakeholders in increasingly competitive energy markets. This necessitates evaluating and navigating the risk associated with the decisions made at every time instant.

However, behind the scenes, these systems are often influenced by external uncertainties and internal faults, hardly observable and controllable by conventional means. Growing security concerns emerge as systems become more interconnected and interdependent, making real-time assessment of system behavior increasingly challenging. This interconnectedness necessitates broad range of monitoring capabilities—from detecting faults before they escalate to failures, from identifying cyberattacks before they cause shutdowns, and from recognizing minor inconsistencies before they develop into major disasters.

Machine learning has been heavily relied upon in this context, as it may augment classical approaches to account for the hardly modelable dynamics, often employed as a replacement for traditional methods. The era of Industry 4.0 is ruled by data, where effective data-based decision-making is driven by the quantity of collected data. Internet of Things devices have made data acquisition seamless and positively influenced a wide range of industries.

Nevertheless, the lack of interpretability, deviation from physics principles, and significant nonlinearity make the employment of these systems difficult in industrial settings. The service lifecycle is often not considered, as systems are not designed to be self-monitoring and self-correcting over their operational lifetime. Moreover, emergence of agentic systems based on large language models has expressed significant misalignment and revealed fundamental misunderstanding of the predictions peculiarities.

1.2 Problem Statement: The Gap Between Theory and Industrial Reality

A fundamental disconnect persists between theoretical advances in control and monitoring and their deployability in industrial environments. Traditional control methodologies exhibit conservative robustness, separated from economic implications such as resource utilization and revenue generation. Meanwhile, undertaking certain risk, while allowing to achieve higher profit, necessitate stricter monitor the system to ensure the safety. Therefore, monitoring systems, evolving together with control systems, are crucial to achieve a cost-efficient service lifecycle, yet this integration remains elusive in practice.

There is an urgent need to develop frameworks that bridge this gap between theory and industry—frameworks that are interpretable, risk-aware, safe, and adaptable even in real deployments. Thus far, a strong separation exists between physical and data-driven modeling, as well as between control and monitoring layers. The blind usage of heavily parametrized models for data-driven modeling, without consideration of the underlying physics and dynamics, leads to solutions that lack the transparency. These collectively block addoption in safety-critical applications.

Nevertheless, we cannot neglect new findings as immature. As industrial processes become increasingly complex, nonlinear, and decentralized, traditional rigid automation structures fail to adapt to evolving dynamics, inherent uncertainties, and component degradation. Existing approaches often face limitations in modularity, scalability, and practical implementation under real-world conditions. Results are demonstrated in simulations only or with experiments in laboratories, thus not facing problems with deployment and long-term operation. Many frameworks are designed for particular applications with single revenue streams and pre-defined compositions, lacking generality and versatility.

Additionally, the computational complexity of some stochastic and robust control methods can impede their real-time execution, especially in systems with limited processing capabilities and fast dynamics. The practical aspect of large-scale deployment—creating cost-efficient architectures that provide long-term value across various systems—remains an open challenge.

1.3 Thesis Statement: Synergistic Frameworks for Modeling, Control, and Monitoring

The central thesis of this dissertation is that the reliable operation of modern sustainable industrial and energy systems requires a unified methodological framework that bridges the gap between advanced data-driven dynamics modeling, optimal control strategies, and real-time operational monitoring. This work argues that optimal performance and safety can only be achieved by augmenting physics informed models with interpretable machine learning models, further integrated in risk-aware control architectures and adaptive anomaly detection mechanisms.

This dissertation addresses three fundamental research questions:

1. **How to achieve economic feasibility under risk-awareness at scale?** Control systems must not only optimize physical parameters but also manage economic risks and revenue streams under uncertainty, effectively translating control theory into profitable, deployable software solutions.

2. **How to ensure safety and reliability in long-term operation?** Sustainable industrial automation is not a singular control problem but a lifecycle challenge. The long-term autonomy of monitoring systems requires continuous adaptation to changing dynamics, ensuring resilience against influence of aging, non-stationarity

and cyclic effects.

3. **How to achieve credibility and industrial adoption?** The progression begins with the fundamental challenge of representing complex nonlinear behaviors in a form suitable for control. By discovering linear representations of nonlinear dynamics we enable the application of current linear control techniques to previously intractable nonlinear problems. This approach should be integrated with existing infrastructure, utilizing IoT data streams on top of well-established SCADA systems while expecting minimal intervention to existing deployments.

1.4 Contributions

This dissertation establishes that reliable industrial control systems require a comprehensive toolkit spanning economic control, adaptive monitoring, and interpretable modeling. The main contributions are organized into three interconnected pillars, each responding directly to the fundamental research questions posed above.

1.4.1 Economic Feasibility Under Risk-Awareness at Scale

This dissertation addresses the first research question by developing risk-aware control architectures that translate control-theoretic advances into economically viable, deployable solutions. The contributions span methodological innovations in efficient uncertainty handling, practical contributions from business-oriented economic and carbon-friendly cost functions, modular system design, and empirical validation through real-world deployment.

First, we present a greenhouse climate control that optimizes all three; crop yield, energy efficiency, and CO_2 emissions using single economic objective. By integrating thermodynamic models with real-time weather and carbon intensity forecasts, this platform enables exploration of economic and sustainability trade-offs under uncertain time-varying conditions [119].

Second, a novel uncertainty approximation method enhances computational efficiency, ensuring that risk-aware decision-making remains tractable for real-time execution. This risk-aware controller is embedded within a modular framework for versatile microgrid compositions that supports diverse energy assets, alongside multiple revenue streams. To bridge theoretical model predictive control (MPC) advances with deployment challenges, a software architecture is proposed to address operational cost scaling across multiple microgrids. The complete energy management system has been deployed and validated in commercial and industrial microgrid installations, with real-world operation and financial analysis demonstrating sustained economic value [117].

1.4.2 Safety and Reliability in Long-Term Operation

The second research question is addressed through a development of adaptive monitoring methods that ensure system reliability across the operational lifecycle. These contributions establish theoretical foundations for interpretable detection and provide practical mechanisms for continuous adaptation to changing dynamics [123, 120].

Self-supervised anomaly detection algorithm is developed to track the system's state and establish dynamic operating limits, providing modern alternative to static operation thresholds [121]. We further extend this work to enable monitoring the whole system, while preserving interpretability and signal-level root cause isolation capabilities, distinguishing this approach from black-box methods that may trigger false alarms requiring extensive analysis. [122].

To address detection of fundamental shifts in system dynamics, we propose an interpretable change-point detection method based on online Dynamic Mode Decomposition with control (oDMDc) with theoretical connections between subspace reconstruction error and system dynamics changes, and a self-supervised learning mechanism that enables continuous adaptation without external supervision or labeled data [123]. The reconstruction error of physics-informed model of healthy device is further utilized to provide remaining useful life estimation for utility-scale electrolyzer, enabling condition-based maintenance strategies that extend equipment lifetime while avoiding unexpected failures [120].

These monitoring methods integrate seamlessly with existing industrial infrastructure, enabling faster incident response. By leveraging IoT data streams on top of well-established SCADA systems, the proposed approaches require minimal intervention to existing deployments—preserving infrastructure investments, avoiding operator retraining, and automating alerting for high numbers of signals without a priori knowledge of process limits. Deployments confirm the methods' ability to adapt to environmental changes and device aging—critical capabilities for long-term autonomous operation.

1.4.3 Credibility and Industrial Adoption

The third research question is addressed by developing interpretable, physics-informed modeling approaches that bridge the gap between data-driven flexibility and the transparency required for safety-critical applications. These contributions span theoretical advances in system identification, practical employment of physics-informed learning, and empirical validation on industrial-scale systems.

We introduce optimal truncation into oDMDc which provably improves convergence properties and numerical stability in real deployments while adapting to time-varying dynamics [123]. We further ease the requirement for estimating the governing dynamics by developing a dictionary-free method for learning linear representations of nonlinear systems with input delays using deep neural networks [118].This approach overcomes the fundamental limitation of traditional Koopman operator methods that rely on pre-defined dictionaries—which require expert knowledge for selection and may inadequately span the relevant function space.

Unlike black-box models that may achieve high accuracy but offer no insight into failure modes or operational limits, our physics-informed model produce outputs that can be validated against engineering intuition and regulatory requirements. Validation on industrial-scale systems—including chlor-alkali electrolyzers representing significant capital investments—demonstrates that the proposed methods deliver actionable insights for maintenance planning and operational optimization [120].

1.5 Dissertation Structure

The remainder of this dissertation is organized as follows:

Chapter 2: Background and State of the Art establishes the theoretical foundations and prior state-of-the-art for the subsequent contributions. It reviews Dynamic Mode Decomposition and its variants, Model Predictive Control under uncertainty, and automated detection paradigms.

Chapter 3: Risk-Aware and Economic Control addresses the first research question on economic feasibility. It presents Economic MPC for greenhouse climate optimization and develops a modular, risk-aware architecture for microgrid energy management with real-world deployment.

Chapter 4: Real-Time Monitoring and Adaptive Safety addresses the second research question on long-term reliability. It develops self-supervised anomaly detection with dynamic operating limits, interpretable change-point detection, and remaining useful life estimation for industrial equipment.

Chapter 5: Interpretable Data-Driven Modeling addresses the third research question on industrial adoption. It introduces advancements in Koopman operator learning, optimal truncation for oDMDc, dictionary-free Koopman learning with deep neural networks, and physics-informed DMDc which aid their industrial deployment.

Chapter 6: Conclusions summarizes the contributions, discusses their implications for sustainable industrial automation, and outlines directions for future research.

1.6 Reading Guide

The chapter ordering of this dissertation is deliberate rather than chronological. Control (Chapter 3) is presented first because the risk-awareness of the external factors influencing both environmental and economic aspects of the operation, as well as the operational costs are crucial for fulfilling the business objectives. With proper controller in place, the intermediate monitoring layer (Chapter 4) ensures that the control system of Chapter 3 itself remains safe and aligned with digital twin which must stay valid as the underlying systems evolve. Chapter 5 provides the linear interpretable surrogates upon which both controllers and monitors ultimately rest to ensure business adoption and trustworthiness. Readers preferring a methods-first progression may read Chapter 5 immediately after Chapter 2 without loss of coherence, returning to Chapter 3 and Chapter 4 once the modeling foundations are in hand. Table 1.1 summarizes the mapping from research questions to chapters, sections, primary publications, and validation domains.

Table 1.1: Navigation map: research questions, chapters, primary publications, and validation domains.

RQ	Chapter	Sections	Publication	Validation
RQ1	Chapter 3	Section 3.2 (greenhouse EMPC)	[119]	greenhouse
RQ1	Chapter 3	Section 3.3 (risk-aware microgrid EMS)	[117]	commercial microgrid
RQ2	Chapter 4	Section 4.2 (ICDF dynamic process limits)	[121]	HVAC, building sensors
RQ2	Chapter 4	Section 4.3 (AID framework for SCADA)	[122]	battery storage SCADA
RQ2	Chapter 4	Section 4.4 (CPD via toDMDc)	[123]	industrial benchmarks
RQ3	Chapter 5	Section 5.2 (LSTM-enhanced Deep Koopman)	[118]	two-tank system
RQ3	Chapter 5	Section 5.3 (physics-informed DMDc, RUL)	[120]	chlor-alkali electrolyzer

Chapter 2

Background and State of the Art

This chapter establishes the theoretical foundations and state-of-the-art for the contributions presented in subsequent chapters. The discussion progresses from the practical realities of industrial deployment toward the theoretical methods that address these challenges. Section 2.1 reviews the architecture and limitations of SCADA systems alongside deployment challenges in brownfield environments—the pre-existing infrastructure that motivates and constrains subsequent methodological choices. Section 2.2 presents data-driven dynamics modeling, from Dynamic Mode Decomposition to physics-informed approaches and Koopman operator theory, establishing the mathematical toolkit for system identification. Section 2.3 discusses advanced control strategies under uncertainty, including risk-aware formulations that build upon these modeling foundations. Finally, Section 2.4 examines operational safety and monitoring paradigms, completing the lifecycle perspective by addressing how systems maintain reliability over extended deployments.

Each subsection closes with a *Literature gap* block that states what existing work does not solve, why it matters, and which dissertation section addresses it. Table 2.1 in Section 2.5 aggregates these per-subsection gaps into a single mapping from gap to contribution.

2.1 The Industrial Context: SCADA and Deployment

Before examining advanced modeling and control techniques, it is essential to understand the industrial infrastructure within which these methods must operate. Modern industrial facilities rely on established automation architectures that have evolved over decades. Any proposed enhancement must integrate with—rather than replace—this existing infrastructure, respecting both technical constraints and organizational practices.

2.1.1 SCADA Systems Architecture

Supervisory Control and Data Acquisition (SCADA) systems form the backbone of industrial process monitoring and control. As a concept, SCADA embodies hardware components in the field, communication infrastructure, controllers, automation protocols and human-machine interfaces. Nowadays, this concept is often used interchangeably with distributed control systems (DCS), as both can control spatially and temporally distributed systems across multiple sites. The key distinction lies in their control philosophy: SCADA emphasizes central supervisory control, while DCS advocates distributed control; both with specific strengths and weaknesses.

These systems continuously monitor process data in real-time, embodying alarm handling mechanisms designed to notify operators of abnormal behavior and direct attention to the root of problems. By comparing current values to upper and lower operating limits, SCADA systems take action when variables exceed or fall below predefined thresholds [84]. This threshold-based paradigm, while simple and reliable, represents both the strength and limitation of classical automation, as illustrated in Figure 2.1.

2.1.1.1 Hierarchical Control Layers

Industrial control architectures typically follow a hierarchical structure with multiple layers. This hierarchy can be understood through two complementary perspectives: the functional roles of each layer and the temporal scales at which they operate.

From a functional perspective, each layer specializes in specific aspects of system operation. At the field level, sensors and actuators interface directly with physical processes, converting physical quantities into electrical signals and vice versa. Programmable Logic Controllers (PLCs) provide local control and data aggregation at the control layer, executing deterministic control logic with guaranteed timing. The supervisory layer, implemented through SCADA software, enables human operators to monitor system state, acknowledge alarms, and override automatic controls when necessary. Above this, Manufacturing Execution Systems (MES) and Enterprise Resource Planning (ERP) systems integrate production data with business processes, connecting the physical plant to organizational decision-making.

The temporal dimension of this hierarchy is equally important, as illustrated in

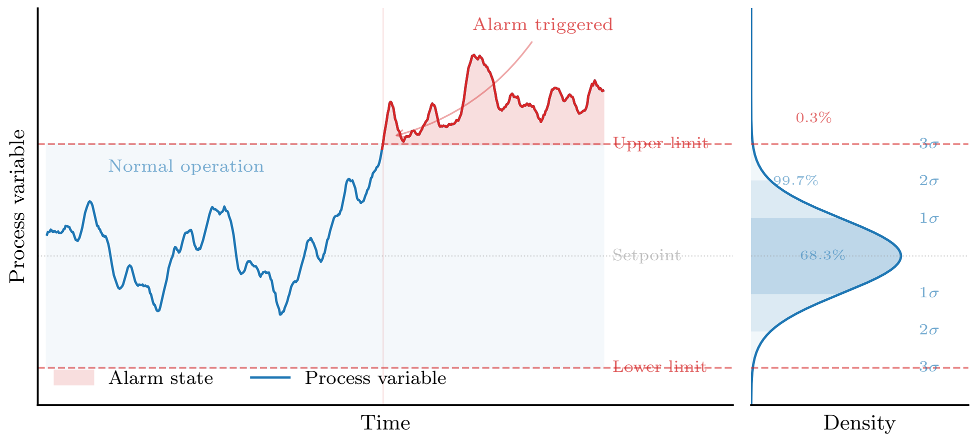

Figure 2.1: Threshold-based alarm paradigm in SCADA systems. A process variable (left) oscillates within fixed upper and lower operating limits until a drift causes it to breach the upper threshold, triggering an alarm. The companion density (right) shows the assumed Gaussian distribution underlying threshold placement: limits at $\pm 3\sigma$ from the setpoint enclose 99.7% of normal variability, leaving only 0.3% probability in the alarm tails.

Figure 2.2. Real-time control loops at the field and PLC levels operate at millisecond timescales to maintain process variables within setpoints, requiring deterministic execution guarantees. Supervisory control operates at longer timescales—typically seconds to minutes—coordinating multiple control loops and optimizing overall system performance. Strategic planning layers may operate on hourly to daily timescales, incorporating market information, demand forecasts, and maintenance schedules into operational decisions. This temporal separation enables each layer to focus on decisions appropriate to its timescale while maintaining clear interfaces with adjacent layers.

Energy management systems (EMS) exemplify this hierarchical paradigm in the context of microgrids, serving as the interface between process control and information engineering. The design of an EMS encompasses two key aspects: the development of control algorithms to optimize energy flows and meet operational objectives, and the practical implementation of software architecture to ensure reliable and scalable deployment [45]. Control algorithms such as Model Predictive Control (MPC) are critical for addressing challenges including profit maximization, renewable energy integration, and uncertainty management while adhering to system constraints. However, the choice of algorithm cannot be divorced from deployment considerations—an optimal algorithm that cannot execute in real-time provides no practical benefit.

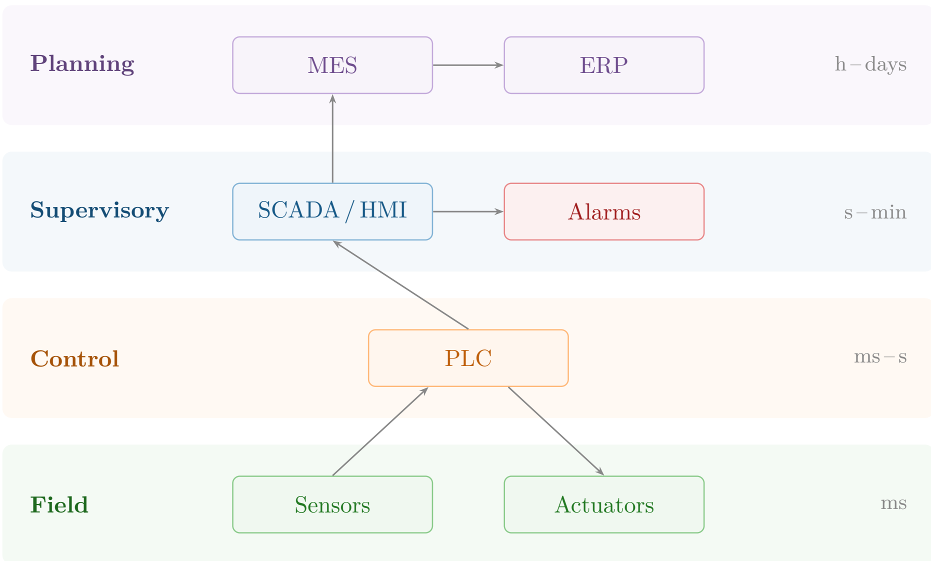

Figure 2.2: Hierarchical architecture of industrial SCADA systems. Field-level sensors and actuators interface directly with physical processes; PLCs at the control level execute deterministic logic; the supervisory level provides the SCADA/HMI interface and alarm handling; MES and ERP systems at the planning level connect plant operations to business processes. Each layer operates at progressively longer timescales.

2.1.1.2 Limitations of Classical Automation

Despite their widespread adoption and proven reliability, classical automation systems exhibit fundamental limitations when confronting modern operational challenges. Safe operating limits are often established based on a combination of equipment design limits and process dynamics, remaining indifferent to the actual state of the system and environmental conditions. This static approach fails to account for component degradation, varying operational contexts, or novel fault modes that may emerge over the system's lifetime.

Furthermore, traditional SCADA alarm handling suffers from alarm flooding during abnormal situations, potentially overwhelming operators with hundreds of alarms that obscure root causes. The lack of intelligent filtering and prioritization mechanisms makes rapid diagnosis challenging precisely when it is most critical. Human error is a major factor in industrial accidents, as fatigue, stress, and distraction can lead to missed or misinterpreted alarms.

These limitations motivate the development of automated adaptive methods that can operate within existing SCADA frameworks while providing enhanced detection and diagnostic capabilities. The challenge is to augment rather than replace established infrastructure, preserving the reliability and familiarity that operators depend upon while addressing gaps in adaptability and autonomy.

2.1.2 Deployment Challenges

Understanding the limitations of classical automation naturally leads to questions of how new solutions can be practically deployed. The deployment of modern control and monitoring systems in industrial environments faces substantial practical challenges, particularly in brownfield installations where existing infrastructure constrains available options. Three interrelated concerns dominate deployment considerations: speed, scalability, and feasibility.

2.1.2.1 Speed

Real-time control and monitoring impose strict timing requirements that fundamentally constrain algorithmic choices. Industrial process data are streamed and arrive at non-uniform rates, challenging methods dependent on uniform sampling assumptions [57, 28]. Control and monitoring in microgrids calls for low-latency detection, implying real-time prediction that respect strict boundaries on computational time [96, 20]. Meanwhile, available computational resources are limited, requiring methods that are efficient at scale.

The adaptation of batch models at scale introduces significant latency in detector adaptation [103], creating a problematic gap between when system changes occur and when models can respond. This delay can prove costly or dangerous in safety-critical applications. Consequently, incremental learning methods become essential for online operation, enabling models to update continuously without the delayed adaptation of batch trained models. The contributions presented in Chapter 4 explicitly address this requirement through streaming algorithms that maintain bounded computational complexity per update.

2.1.2.2 Scalability

Beyond raw speed, computational efficiency directly impacts operational costs of deployment at scale. The computational complexity of stochastic MPC methods, further discussed in Chapter 3, can impede real-time execution, especially in systems with limited processing capabilities [115]. Studies estimate that computational time for stochastic models increases by a factor of ten compared to deterministic formulations, growing exponentially with the number of scenarios considered. This scaling behavior fundamentally limits the sophistication of uncertainty handling that is practically achievable.

Cloud computing enables scalable monitoring, remote control, and flexible management of distributed assets while minimizing capital expenditures for operators [98]. However, the operational expenditures of EMS architectures—an aspect often neglected despite its significant impact on economic viability—must also be considered in large-scale and multi-grid deployments [5]. The true cost of advanced control includes not only algorithm development but ongoing computational resources, communication infrastructure, and maintenance overhead. Methods that appear optimal in simulation may prove economically unviable when deployment costs are fully accounted.

2.1.2.3 Feasibility

Practical deployment must ultimately accommodate existing infrastructure constraints and organizational realities. Many frameworks are finetuned for particular applications with pre-defined compositions, lacking the generality and versatility required for diverse deployments across different facilities and configurations. The business perspective of such solutions is important to consider, as EMS providers must ensure profitability while keeping computing costs in check to maintain sustainable operations.

Results demonstrated only in simulations or laboratory experiments do not face the problems encountered in deployment and long-term operation [39]. Real industrial environments present challenges including sensor failures, communication dropouts, parameter drift, and evolving operational requirements that cannot be fully anticipated in simulated or controlled laboratory settings.

2.2 Data-Driven Dynamics and Physics-Informed Modeling

The limitations of classical automation described in the previous Section 2.1.1.2 motivate the development of data-driven approaches that can approximate system behavior from operational data. However, purely data-driven methods face their own challenges: they may lack interpretability, violate physical constraints, or require prohibitive amounts of training data or require costly labeling. This section examines the learning paradigms from the perspective of the learning objectives and the transition from pure data-driven methods to physics-informed formulations, establishing the mathematical foundations for the contributions presented in Chapter 5.

2.2.1 Paradigms

The growing availability of operational data from industrial systems has given rise to several distinct paradigms for incorporating machine learning into dynamical systems analysis and control [23]. While these paradigms share common mathematical tools, they differ fundamentally in their learning objectives and in the role that data plays relative to domain knowledge. We distinguish three broad paradigms that frame the contributions of this thesis.

Learning to model (L2M) uses observational data to identify the governing dynamics of a system, producing explicit state-space or operator representations that can be interrogated, validated against physics, and reused across downstream tasks such as prediction, control design, and monitoring. Classical system identification falls within this paradigm, as do modern data-driven methods including Dynamic Mode Decomposition, Sparse Identification of Nonlinear Dynamics (SINDy), neural ordinary differential equations, and Koopman operator approximations [110, 37]. The distinguishing feature is that the learned artifact is a model with interpretable structure, not merely a predictive mapping.

Learning to control (L2C) bypasses explicit model construction and instead uses data or interaction with the environment to directly synthesize control policies [76]. Reinforcement learning is the most prominent example, but the paradigm also encompasses differentiable model predictive control and end-to-end learned policies that map sensor observations to actuator commands without an intermediate dynamical model [77]. While this paradigm can achieve high performance in simulation, its reliance on extensive exploration and the difficulty of providing safety guarantees during learning limit its adoption in safety-critical industrial settings [86].

Learning to simulate (L2S) aims to construct surrogate models that reproduce the forward evolution of a system with high fidelity and low computational cost, often replacing expensive numerical solvers [80]. Physics-informed neural networks, neural operators, and digital twin architectures are representative methods [50, 65]. Unlike the modeling paradigm, the primary objective is simulation speed and accuracy rather than the recovery of interpretable dynamical structure; the learned surrogate is typically treated as a black box during inference.

These paradigms are not mutually exclusive—a physics-informed surrogate may serve as the plant model inside a MPC, for instance—but distinguishing them clarifies the design choices made in this thesis (Figure 2.3). The contributions presented here operate primarily within the L2M paradigm: Chapter 5 develops interpretable operator-theoretic and physics-informed identification methods, while Chapter 4 exploits the structure of learned models for real-time change detection. The control strategies in Chapter 3 then leverage these models within a model-based predictive control framework rather than learning control policies directly from data.

Literature gap *Not solved* — L2C and L2S deliver predictive or actuation performance but produce artifacts that resist physical interrogation, while pure L2M methods typically discard the priors that make industrial models trustworthy. *Why it matters* — Safety-critical deployment requires artifacts that operators can validate against first-principles knowledge, embed in existing model-based controllers, and reuse across prediction, control, and monitoring tasks without retraining a black box for each. *Addressed in* — Section 5.3 and Section 5.2, which develop L2M methods that retain operator-theoretic and physics-based structure while learning from data.

2.2.2 Classical Forecasting

Classical time-series forecasting methods such as autoregressive integrated moving average (ARIMA) models and exponential smoothing remain standard baselines in industrial practice, owing to their low computational cost and well-understood statisti-

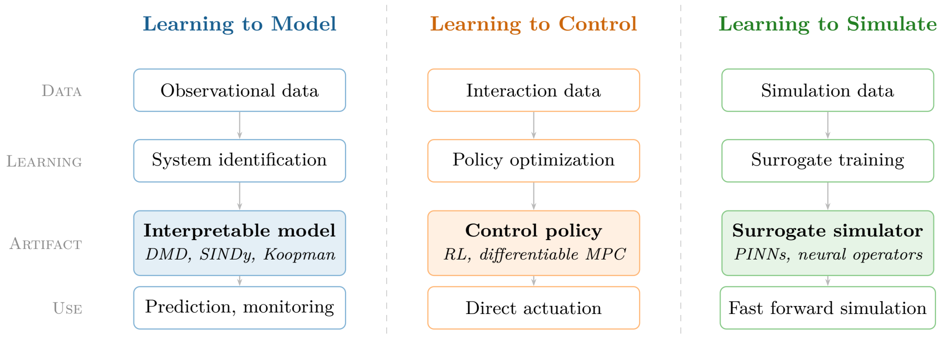

Figure 2.3: The three paradigms share a common data-to-artifact pipeline but diverge in what the artifact is and how it is used. The artifact type—interpretable model, control policy, or surrogate simulator—determines what guarantees, transparency, and reusability downstream tasks can expect. This thesis positions itself at the L2M artifact: structure that can be interrogated, validated, and embedded in a model-based controller or monitor.

cal properties. However, these methods assume stationarity and linearity—conditions that industrial systems routinely violate due to nonlinear dynamics, time-varying operating regimes, and exogenous disturbances. Their limited expressiveness motivates the transition to data-driven and physics-informed modeling frameworks discussed in the following subsection.

2.2.3 From Data-Driven to Physics-Informed Modeling

Understanding and controlling nonlinear dynamical systems is a fundamental challenge across scientific and engineering disciplines. Traditional linear control techniques often fall short when applied to such systems due to their inherent nonlinearity. This challenge becomes even more pronounced in time-delayed systems, where delays introduce additional complexity by coupling past states with current dynamics, potentially degrading control performance and stability if not adequately accounted for.

A promising approach involves identifying coordinate transformations that render nonlinear dynamics approximately linear, enabling the application of well-developed linear theory for prediction, estimation, and control. This idea—that complex nonlinear behavior might be represented linearly in an appropriately chosen coordinate system—underlies much of modern data-driven dynamics modeling.

The trade-off between model accuracy and interpretability represents a central tension

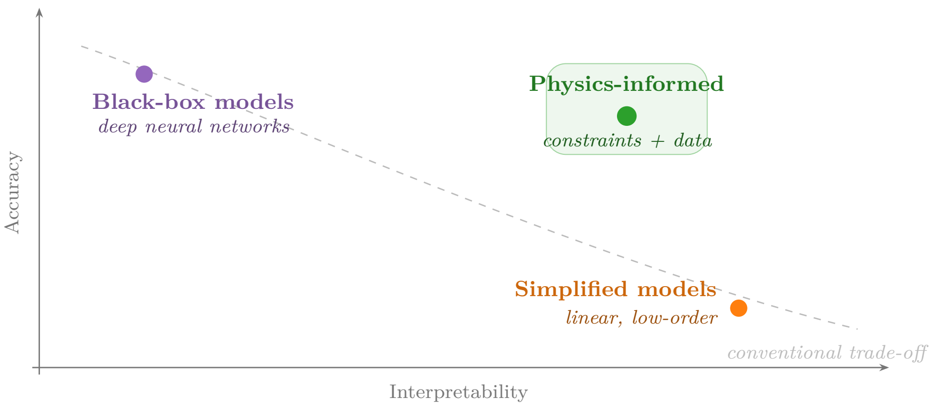

Figure 2.4: The accuracy–interpretability trade-off in data-driven modeling. Pure black-box models such as deep neural networks achieve high predictive accuracy but lack transparency, while overly simplified linear models are interpretable but miss essential dynamics. Physics-informed approaches occupy a favorable position above the conventional trade-off frontier by embedding physical constraints into data-driven models, achieving both accuracy and interpretability.

in data-driven modeling. Pure black-box models such as deep neural networks may achieve high predictive accuracy but lack the transparency required for safety-critical applications where operators must understand and trust model predictions. Conversely, overly simplified models may be interpretable but fail to capture essential dynamics, leading to poor control performance or missed anomalies.

Physics-informed approaches offer a middle ground by embedding physical constraints into data-driven models. Constraining data-driven models with domain physics is essential for reliable predictive maintenance and long-term degradation tracking, bridging the critical gap between sparse laboratory data and continuous industrial operations. By respecting known conservation laws, symmetries, and physical bounds, these hybrid approaches achieve better generalization with less data while maintaining the interpretability that industrial applications demand (Figure 2.4).

Literature gap *Not solved* — Off-the-shelf physics-informed methods typically bolt soft physical penalties onto neural surrogates, leaving conservation, stability, and symmetry properties as approximate post hoc regularizers rather than hard structural guarantees. *Why it matters* — For degradation tracking and predictive maintenance, models that occasionally violate conservation laws or produce unstable extrapolations cannot underpin long-horizon decisions or pass regulatory acceptance in industrial settings. *Addressed in* — Section 5.3, which embeds physical structure directly into the DMDc identification problem so that conservation and stability properties are enforced by construction rather than encouraged by loss design.

2.2.4 Koopman Operator Theory

The theoretical foundation for linearizing nonlinear dynamics lies in Koopman operator theory. Koopman analysis facilitates the linearization of nonlinear systems through the Koopman operator, which has gained considerable traction in recent years [102, 63]. Unlike traditional linearization approaches that approximate dynamics locally around equilibrium points, the Koopman operator offers a global linearization by mapping the original nonlinear system into a higher-dimensional space where dynamics can be represented linearly [67, 66].

Consider a discrete-time dynamical system defined by the state transition map $\mathcal{F} : \mathcal{M} \to \mathcal{M}$ on a manifold $\mathcal{M}$:

$$\boldsymbol{x}_{k+1} = \mathcal{F}(\boldsymbol{x}_k), \tag{2.1}$$

where $\boldsymbol{x}_k \in \mathcal{M} \subseteq \mathbb{R}^m$ is the state at time k. For nonlinear $\mathcal{F}$, the evolution of states is inherently nonlinear. However, consider scalar-valued observables $g : \mathcal{M} \to \mathbb{C}$ that measure properties of the state. The Koopman operator $\mathcal{K}$ is defined by its action on such observables:

$$\mathcal{K}g = g \circ \mathcal{F}, \tag{2.2}$$

where $\circ$ denotes function composition. This means $(\mathcal{K}g)(\boldsymbol{x}) = g(\mathcal{F}(\boldsymbol{x}))$—the Koopman operator advances observables forward in time by composing them with the dynamics.

The key insight is that while states evolve nonlinearly, the Koopman operator $\mathcal{K}$ is *linear*, even when $\mathcal{F}$ is nonlinear. For any observables g_1, g_2 and scalars α, β:

$$\mathcal{K}(\alpha g_1 + \beta g_2) = \alpha \mathcal{K} g_1 + \beta \mathcal{K} g_2. \tag{2.3}$$

This linearity comes at the cost of infinite dimensionality—$\mathcal{K}$ acts on the infinite-dimensional space of all observables.

The spectral decomposition of the Koopman operator provides insight into the underlying dynamics. Koopman eigenfunctions φ_j satisfy

$$\mathcal{K}\varphi_j = \lambda_j \varphi_j, \tag{2.4}$$

where $\lambda_j \in \mathbb{C}$ are the corresponding eigenvalues. Each eigenvalue encodes temporal behavior: its magnitude $|\lambda_j|$ indicates growth (> 1) or decay (< 1), while its argument

$\arg(\lambda_j)$ determines oscillation frequency. If an observable g can be expressed in terms of eigenfunctions as $g(\boldsymbol{x}) = \sum_j \varphi_j(\boldsymbol{x}) v_j$, where v_j are Koopman modes, then its time evolution becomes

$$g(\boldsymbol{x}_k) = \sum_j \lambda_j^k \varphi_j(\boldsymbol{x}_0) v_j. \tag{2.5}$$

This representation transforms nonlinear dynamics into a linear superposition of exponentially evolving modes.

The challenge lies in finding suitable finite-dimensional approximations of this infinite-dimensional operator. If a finite set of observables $\boldsymbol{g}_{\boldsymbol{x}} = [g_1(\boldsymbol{x}), \ldots, g_m(\boldsymbol{x})]^\top$ spans a Koopman-invariant subspace, then there exists a finite matrix $\boldsymbol{K} \in \mathbb{R}^{m \times m}$ such that

$$\boldsymbol{g}_{\boldsymbol{x}_{k+1}} = \boldsymbol{K} \boldsymbol{g}_{\boldsymbol{x}_k}. \tag{2.6}$$

Finding such invariant subspaces—or good approximations thereof—leads naturally to Dynamic Mode Decomposition and its extensions.

Literature gap *Not solved* — Koopman theory promises exact global linearization, yet practical implementations hinge on hand-crafted observable dictionaries that rarely span an invariant subspace for systems with unknown nonlinearities or time delays. *Why it matters* — Industrial assets are routinely commissioned with imperfect dynamics knowledge, so dictionary mis-specification yields models that pass training-error checks but extrapolate poorly under control, undermining both prediction and downstream optimization. *Addressed in* — Section 5.2, where an encoder–decoder architecture learns Koopman-invariant coordinates directly from data, removing the need for manual dictionary design while preserving the linear operator structure required for control and monitoring.

The LSTM-enhanced Deep Koopman framework developed in Section 5.2 builds directly on the dictionary-learning view of this operator.

2.2.5 Dynamic Mode Decomposition and its Variants

Finite-dimensional approximations of the Koopman operator are often achieved using Dynamic Mode Decomposition (DMD) [81]. DMD emerged from the fluid dynamics community as a method for extracting spatiotemporal coherent structures from high-dimensional measurement data. Given snapshot pairs of system state, DMD identifies a best-fit linear operator that maps one snapshot to the next, decomposing dynamics into modes characterized by oscillation frequencies and growth/decay rates.

2.2.5.1 Mathematical Formulation of DMD

The DMD algorithm aims to find the optimal linear operator $\boldsymbol{A}$ that advances the snapshot matrix in time. Given n consecutive snapshot pairs $\{\boldsymbol{x}_k, \boldsymbol{x}_{k+1}\}_{k=0}^{n-1}$, organized into matrices $\boldsymbol{X}, \boldsymbol{X}' \in \mathbb{C}^{m \times n}$, the optimal linear operator is defined as

$$\boldsymbol{A} = \arg\min_{\boldsymbol{A}} \|\boldsymbol{X}' - \boldsymbol{A}\boldsymbol{X}\|_F = \boldsymbol{X}'\boldsymbol{X}^+, \tag{2.7}$$

where $\boldsymbol{X}^+$ denotes the Moore-Penrose pseudoinverse of $\boldsymbol{X}$. The exact DMD algorithm of Tu et al. [89] does not rely on uniform sampling assumptions, enabling application to industrial data streams where timing may be irregular.

DMD utilizes the computationally efficient singular value decomposition (SVD) of $\boldsymbol{X}$ to provide a low-rank representation:

$$\boldsymbol{X} = \tilde{\boldsymbol{U}}\tilde{\boldsymbol{\Sigma}}\tilde{\boldsymbol{V}}^\top, \tag{2.8}$$

where $\tilde{\boldsymbol{U}} \in \mathbb{C}^{m \times r}$ are proper orthogonal decomposition (POD) modes, $\tilde{\boldsymbol{\Sigma}} \in \mathbb{C}^{r \times r}$ contains the singular values, and $\tilde{\boldsymbol{V}} \in \mathbb{C}^{n \times r}$ are right singular vectors. The rank $r \leq m$ denotes either the full or approximate rank of the data matrix.

Employing this decomposition, the optimal operator can be rewritten as $\boldsymbol{A} = \boldsymbol{X}'\tilde{\boldsymbol{V}}\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{U}}^\top$, which is then projected onto the POD modes to obtain a low-rank representation:

$$\tilde{\boldsymbol{A}} = \tilde{\boldsymbol{U}}^\top\boldsymbol{A}\tilde{\boldsymbol{U}} = \tilde{\boldsymbol{U}}^\top\boldsymbol{X}'\tilde{\boldsymbol{V}}\tilde{\boldsymbol{\Sigma}}^{-1}. \tag{2.9}$$

Unlike SVD, which captures spatial correlation and energy content, DMD incorporates temporal information via spectral decomposition of $\tilde{\boldsymbol{A}}$:

$$\tilde{\boldsymbol{A}}\boldsymbol{W} = \boldsymbol{W}\boldsymbol{\Lambda}, \tag{2.10}$$

where the diagonal elements of $\boldsymbol{\Lambda}$, denoted $\{\lambda_0, \lambda_1, \ldots, \lambda_r\}$, are the DMD eigenvalues, and columns of $\boldsymbol{W}$ are the DMD modes. Each eigenvalue $\lambda = a + ib$ indicates the growth or decay rate a and oscillation frequency b of its corresponding mode.

The full-dimensional DMD modes $\boldsymbol{\Phi}$ are reconstructed from the reduced modes as

$$\boldsymbol{\Phi} = \boldsymbol{X}'\tilde{\boldsymbol{V}}\tilde{\boldsymbol{\Sigma}}^{-1}\boldsymbol{W}. \tag{2.11}$$

These modes combine the spatial structure of POD with temporal evolution characterized by the eigenvalues, enabling both dimensionality reduction and dynamic modeling for prediction, as illustrated in Figure 2.5.

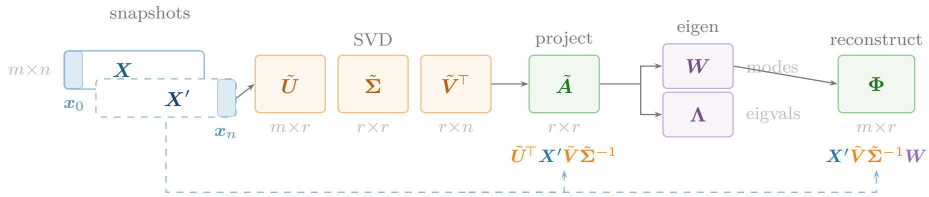

Figure 2.5: DMD computational pipeline. Snapshot matrices $\boldsymbol{X}$, $\boldsymbol{X}'$ are decomposed via truncated SVD, projected to the reduced operator $\tilde{\boldsymbol{A}}$, and spectrally decomposed to yield DMD modes $\boldsymbol{\Phi}$ and eigenvalues $\boldsymbol{\Lambda}$.

While DMD successfully identifies spatiotemporal coherent structures from high-dimensional systems, it typically fails to capture nonlinear transients due to its reliance on linear measurements of the original state space. When dynamics are inherently nonlinear, the linear operator identified by DMD provides only a local approximation that may degrade rapidly away from training conditions. This limitation motivates the development of extended methods that operate in lifted coordinate spaces.

2.2.5.2 Extended DMD

Extended DMD (eDMD) [101] addresses the limitations of standard DMD by incorporating a dictionary of basis functions that lift the state into a higher-dimensional feature space. By applying DMD to these lifted coordinates rather than raw states, eDMD can capture nonlinear relationships that appear linear in the transformed space. Common dictionary choices include polynomial bases, radial basis functions, and Fourier modes.

However, eDMD faces significant practical challenges including high dimensionality and closure issues arising because there is no guarantee that nonlinear measurements form a Koopman invariant subspace [59]. The dictionary must be chosen carefully: too restrictive a dictionary fails to capture relevant dynamics, while too rich a dictionary leads to overfitting and computational intractability. The limitations of dictionary-based approaches also include their inability to automatically discover appropriate features for systems with complex dynamics, time delays, or multiple operating regimes. This trade-off is illustrated in Figure 2.6.

2.2.5.3 Deep Learning Approaches

To overcome the dictionary selection problem, deep neural networks have been employed to learn Koopman operator representations directly from data [109]. Deep neural networks remove the bottleneck of predefined dictionaries by creating linear

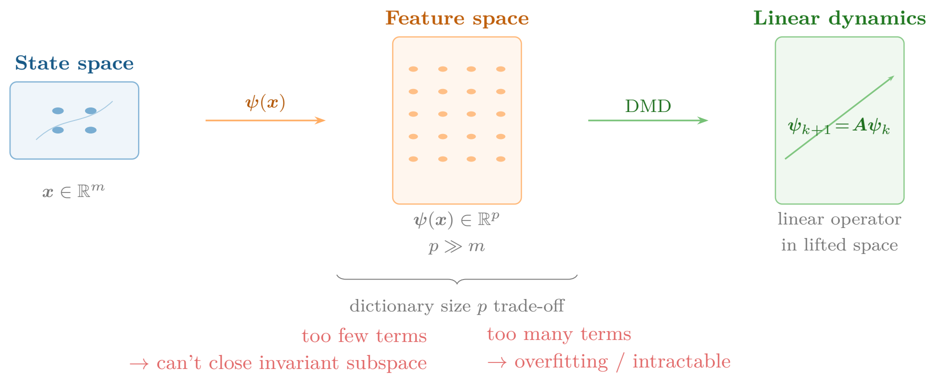

Figure 2.6: Extended DMD lifts the state $\boldsymbol{x} \in \mathbb{R}^m$ into a higher-dimensional feature space $\boldsymbol{\psi}(\boldsymbol{x}) \in \mathbb{R}^p$ via a dictionary of basis functions, enabling DMD to capture nonlinear dynamics as linear evolution in the lifted space. The dictionary must balance expressiveness against overfitting.

representations through nonlinear transformations of individual neurons. These neurons combine to form complex functions parametrized by tunable weights and biases, allowing the network to adaptively learn optimal transformations during training rather than relying on manual feature engineering.

This approach significantly enhances model fidelity, particularly in multi-step prediction tasks where error accumulation can rapidly degrade predictions from simpler models. The encoder-decoder architecture common in deep Koopman methods, shown in Figure 2.7, learns both the lifting to a linear latent space and the projection back to physical coordinates, ensuring that the model produces interpretable outputs even while exploiting high-dimensional intermediate representations. Chapter 5 presents contributions that extend this approach to handle time-delayed systems without explicit delay identification.

2.2.5.4 DMD with Control

Many industrial systems of interest are not autonomous but respond to exogenous control inputs. Dynamic Mode Decomposition with Control (DMDc) extends the framework to explicitly model input effects [73]. The discrete-time linear time-varying system can be written as

$$\boldsymbol{X}_{k+1} = \boldsymbol{A}_k \boldsymbol{X}_k + \boldsymbol{B}_k \boldsymbol{\Theta}_k, \tag{2.12}$$

where $\boldsymbol{X}_k \in \mathbb{R}^{m \times c}$ and $\boldsymbol{\Theta}_k \in \mathbb{R}^{l \times c}$ are the states and control inputs, $\boldsymbol{A}_k \in \mathbb{R}^{m \times m}$ is

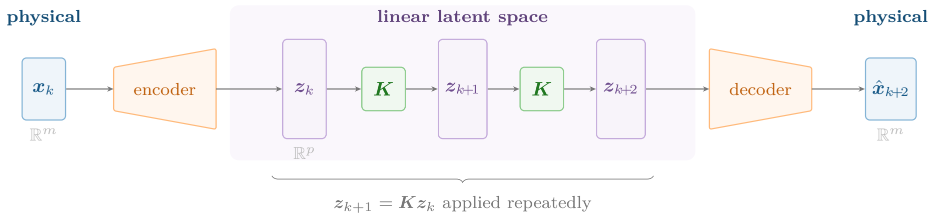

Figure 2.7: Deep Koopman architecture. An encoder network maps the physical state $\boldsymbol{x}_k$ to a latent representation $\boldsymbol{z}_k$, where a linear Koopman operator $\boldsymbol{K}$ advances the state. Multi-step predictions are obtained by repeated linear evolution in the latent space, and a decoder network projects back to physical coordinates.

the state matrix, and $\boldsymbol{B}_k \in \mathbb{R}^{m\times l}$ is the control matrix.

When the control matrix $\boldsymbol{B}$ is known, control effects can be compensated by adjusting the output snapshots:

$$\bar{\boldsymbol{X}}_k' = \boldsymbol{X}_k' - \boldsymbol{B}\boldsymbol{\Theta}_k, \tag{2.13}$$

and applying standard DMD to identify the autonomous dynamics $\boldsymbol{A}$. When neither $\boldsymbol{A}$ nor $\boldsymbol{B}$ is known, the system identification problem is solved by augmenting the state matrix:

$$\bar{\boldsymbol{A}}_k = \begin{bmatrix} \boldsymbol{A}_k & \boldsymbol{B}_k \end{bmatrix}, \quad \bar{\boldsymbol{X}}_k = \begin{bmatrix} \boldsymbol{X}_k \\ \boldsymbol{\Theta}_k \end{bmatrix}, \tag{2.14}$$

where $\bar{\boldsymbol{A}}_k \in \mathbb{R}^{m\times(m+l)}$ and $\bar{\boldsymbol{X}}_k \in \mathbb{R}^{(m+l)\times c}$. The system dynamics become $\boldsymbol{X}_k' = \bar{\boldsymbol{A}}_k\bar{\boldsymbol{X}}_k$, and both $\boldsymbol{A}_k$ and $\boldsymbol{B}_k$ are jointly identified by minimizing $\|\boldsymbol{X}_k' - \bar{\boldsymbol{A}}_k\bar{\boldsymbol{X}}_k\|_F$.

Physics-informed constraints can be incorporated into DMDc to maintain interpretability—essential qualities for reliable degradation tracking in industrial applications [7]. For example, constraints can enforce energy conservation, ensure stability of identified dynamics, or impose known structural relationships between variables. These constraints serve dual purposes: they improve generalization by reducing the hypothesis space, and they produce models whose structure is meaningful to domain experts. Chapter 5 develops physics-informed DMDc methods for electrolyzer degradation modeling. The two identification cases are summarised in Figure 2.8.

2.2.5.5 Online DMD

Deployment in industrial streaming environments requires models that can update incrementally as new data arrive. Online variants of DMD enable sequential updates

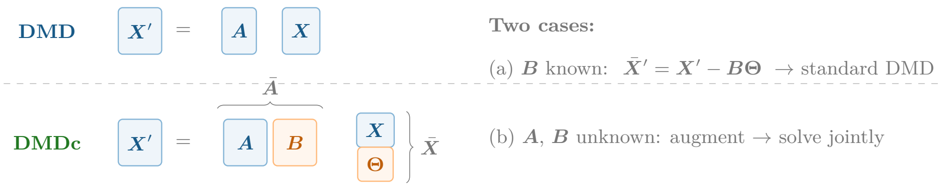

Figure 2.8: DMD with control (DMDc) extends the standard DMD formulation to account for exogenous inputs $\boldsymbol{\Theta}$ by augmenting the state and operator matrices. When $\boldsymbol{B}$ is known, control effects are compensated directly; otherwise, both $\boldsymbol{A}$ and $\boldsymbol{B}$ are identified jointly from the augmented system.

without storing or reprocessing historical data, supporting deployment in memory-constrained real-time systems [111].

As new snapshot pairs $\{\boldsymbol{X}_{k:k+c}, \boldsymbol{X}'_{k:k+c}\}$ become available, the DMD operator must be updated from $\boldsymbol{A}_k$ to $\boldsymbol{A}_{k+c}$. Under the assumption that the history of snapshots $\boldsymbol{X}_k$ has full rank, the optimal operator from (2.7) can be rewritten as

$$\boldsymbol{A}_k = \boldsymbol{X}'_k \boldsymbol{X}_k^+ = \boldsymbol{X}'_k \boldsymbol{X}_k^\top \left(\boldsymbol{X}_k \boldsymbol{X}_k^\top\right)^{-1} = \boldsymbol{Q}_k \boldsymbol{P}_k, \tag{2.15}$$

where $\boldsymbol{Q}_k = \boldsymbol{X}'_k \boldsymbol{X}_k^\top$ is the lag covariance matrix and $\boldsymbol{P}_k = \left(\boldsymbol{X}_k \boldsymbol{X}_k^\top\right)^{-1}$ is the precision matrix.

The precision matrix update employs the Woodbury formula to avoid explicit matrix inversion:

$$\boldsymbol{P}_{k+c} = \boldsymbol{P}_k - \boldsymbol{P}_k \boldsymbol{X}_{k:k+c} \boldsymbol{\Gamma}_{k+c} \boldsymbol{X}_{k:k+c}^\top \boldsymbol{P}_k, \tag{2.16}$$

where $\boldsymbol{\Gamma}_{k:k+c} = \left(\boldsymbol{C}_{k:k+c}^{-1} + \boldsymbol{X}_{k:k+c}^\top \boldsymbol{P}_k \boldsymbol{X}_{k:k+c}\right)^{-1}$ and $\boldsymbol{C}_{k:k+c}$ is a diagonal weight matrix for sample importance. The DMD operator update follows as

$$\boldsymbol{A}_{k+c} = \boldsymbol{A}_k + (\boldsymbol{X}'_{k:k+c} - \boldsymbol{A}_k \boldsymbol{X}_{k:k+c}) \boldsymbol{\Gamma}_{k+c} \boldsymbol{X}_{k:k+c}^\top \boldsymbol{P}_k, \tag{2.17}$$

where $(\boldsymbol{X}'_{k:k+c} - \boldsymbol{A}_k \boldsymbol{X}_{k:k+c})$ represents the prediction error. The operator is updated by adding a term proportional to this error, reflecting the data's covariance structure.

For time-varying systems, windowed online DMD reverts the contribution of old snapshots by applying the same update formulae with negative weights $-\boldsymbol{C}_c$, effectively forgetting observations beyond a sliding window. This enables tracking of evolving dynamics while maintaining bounded memory requirements. The full update mechanism is depicted in Figure 2.9.

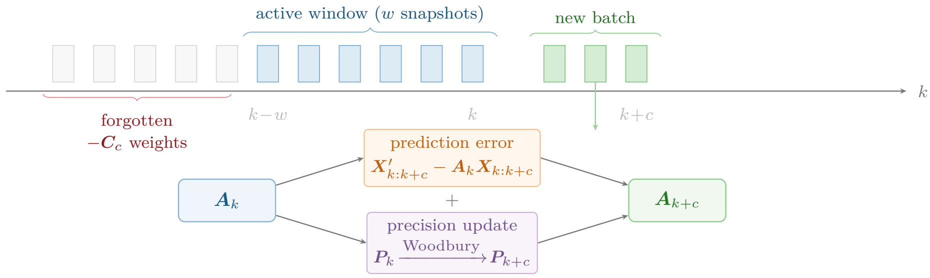

Figure 2.9: Online DMD incrementally updates the operator $\boldsymbol{A}_k$ as new snapshot pairs arrive. The update combines the prediction error with a precision matrix $\boldsymbol{P}_k$ updated via the Woodbury formula. Windowed variants forget old snapshots using negative weights, enabling tracking of time-varying dynamics.

2.2.5.6 Time-Delay Embedding

Many systems exhibit dynamics influenced by time delays that couple past states with current behavior. Hankel DMD addresses this by constructing a time-delay embedding matrix from the data [14]. Given snapshots $\{\boldsymbol{x}_k\}_{k=0}^{c}$ as row vectors, an h-times delayed embedding is formed as

$$\bar{\boldsymbol{X}}_{h,c} = \begin{bmatrix} \boldsymbol{x}_0 & \boldsymbol{x}_1 & \cdots & \boldsymbol{x}_{c-h} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{x}_h & \boldsymbol{x}_{h+1} & \cdots & \boldsymbol{x}_c \end{bmatrix}, \tag{2.18}$$

where the embedded matrix $\bar{\boldsymbol{X}}_{h,c} \in \mathbb{R}^{(m \cdot h) \times (c-h+1)}$. This augmentation enriches the state representation by including historical information, enabling DMD to capture delayed dependencies without explicit delay identification. The online update formulae (2.16) and (2.17) apply directly to time-delay embeddings by substituting $\bar{\boldsymbol{X}}_{h,c}$ for $\boldsymbol{X}$.

This capability proves essential for real-time monitoring applications where batch retraining is infeasible due to latency or computational constraints. By maintaining a compact representation of accumulated data, online DMD can detect when incoming observations deviate from learned dynamics, providing a foundation for the change-point detection methods developed in Chapter 4. The Hankel embedding construction is illustrated in Figure 2.10.

Literature gap *Not solved* — Standard DMD and its extensions assume stationary dynamics estimated from a fixed snapshot window; online variants accelerate identifica-

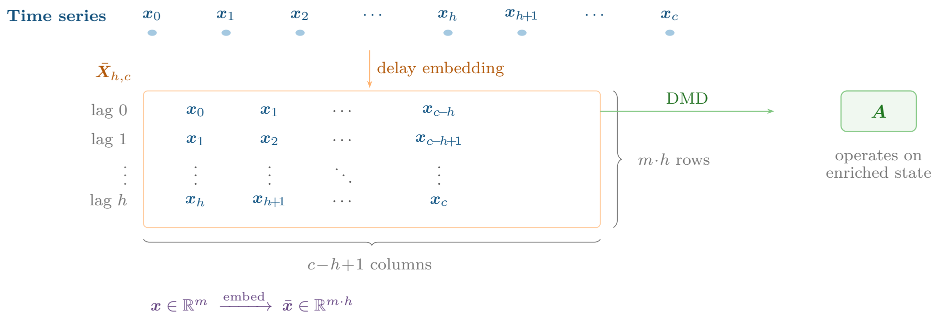

Figure 2.10: Time-delay embedding constructs a Hankel matrix $\bar{\boldsymbol{X}}_{h,c}$ by stacking h delayed copies of the snapshot sequence, enriching the state representation from $\mathbb{R}^m$ to $\mathbb{R}^{m \cdot h}$. Standard DMD or online DMD is then applied to the embedded matrix to capture delayed dependencies.

tion but rarely couple the operator update with formal change-point detection or with physics-consistent constraints. *Why it matters* — Industrial streams are non-stationary, and operators need both fast adaptation to evolving dynamics and confidence that the resulting model still respects conservation and stability properties relied upon by downstream controllers and maintenance schedulers. *Addressed in* — Section 4.4, which fuses time-delay online DMDc with reconstruction-error change-point statistics, and Section 5.3, which constrains DMDc identification to physically admissible operators.

This thesis extends DMD in two complementary directions: the time-varying toDMDc estimator for online change detection in Section 4.4 and the physics-informed piDMDc variant in Section 5.3.

The modeling foundations established in this section enable the advanced control strategies discussed next. Data-driven models, particularly those informed by physics and capable of online updates, provide the predictive capability that Model Predictive Control requires to optimize system operation under uncertainty.

2.3 Advanced Control Strategies

With appropriate system models in hand, the focus shifts to how these models enable advanced control strategies. The industrial deployment challenges outlined in Section 2.1 and the modeling approaches of Section 2.2 converge in Model Predictive Control, which leverages predictive models to optimize control actions while respecting constraints. However, the presence of uncertainty—in predictions, parameters, and operating conditions—demands control formulations that explicitly account for risk.

2.3.1 Control Under Uncertainty

The integration of renewable energy sources and battery storage systems into microgrids has been extensively studied to enhance sustainability and operational efficiency. Model Predictive Control has emerged as a prominent strategy for managing energy flows, particularly under constrained environments with multiple control objectives and inherent uncertainties in renewable outputs and load demands.

The fundamental challenge lies in making decisions now based on predictions of an uncertain future. Control actions taken at the current timestep have consequences that unfold over subsequent timesteps, during which disturbances, forecast errors, and unmodeled dynamics can cause actual outcomes to diverge from predictions. Different philosophical approaches to handling this uncertainty lead to distinct control formulations with different performance characteristics and computational requirements.

The standard MPC formulation captures this challenge mathematically [75]. At each time step, MPC solves a finite-horizon optimal control problem:

$$
\begin{aligned}
\min_{\{\boldsymbol{u}_k\}_{k=0}^{N-1}} \quad & \sum_{k=0}^{N-1} \ell(\boldsymbol{x}_k, \boldsymbol{u}_k) + V_{\mathrm{f},\boldsymbol{x}_N} && \text{(2.19a)}\\
\text{s.t.} \quad & \boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{u}_k), \quad k = 0, \dots, N-1, && \text{(2.19b)}\\
& \boldsymbol{x}_k \in \mathcal{X}, \quad \boldsymbol{u}_k \in \mathcal{U}, \quad k = 0, \dots, N-1, && \text{(2.19c)}\\
& \boldsymbol{x}_0 = \hat{\boldsymbol{x}}_k, && \text{(2.19d)}
\end{aligned}
$$

where $\boldsymbol{x}_k \in \mathbb{R}^n$ is the predicted state, $\boldsymbol{u}_k \in \mathbb{R}^m$ the control input, N the prediction horizon, $\ell(\cdot, \cdot)$ the stage cost, and $V_{\mathrm{f},\cdot}$ a terminal cost. The sets $\mathcal{X}$ and $\mathcal{U}$ encode state and input constraints, and $\hat{\boldsymbol{x}}_k$ is the current measured state. Only the first control action $\boldsymbol{u}_0^*$ is applied; the optimization then repeats at the next time step with updated measurements—the receding-horizon principle. This formulation assumes that the dynamics f and cost ℓ are known with certainty. When this assumption is relaxed, the approaches discussed below arise.

2.3.1.1 Robust Approaches

Robust optimization techniques aim to handle uncertainties conservatively by optimizing against worst-case realizations [82]. Multi-stage robust optimization ensures operational reliability while balancing cost efficiency, providing guarantees that constraints will be satisfied regardless of how uncertainty unfolds within specified bounds (Figure 2.11). This worst-case perspective is appropriate when constraint violations are catastrophic and must be avoided at all costs.

However, robustness can lead to overly conservative decisions, potentially reducing economic benefits by preparing for worst cases that rarely materialize. The system may maintain excessive reserves, curtail profitable opportunities, or operate inefficiently to guard against scenarios that have low probability. Unlike conventional robust optimization, risk-averse formulations such as those employing Conditional Value at Risk (CVaR) provide adjustable conservatism levels [49]. By tuning the risk level, operators can balance protection against adverse outcomes with expected performance under typical conditions.

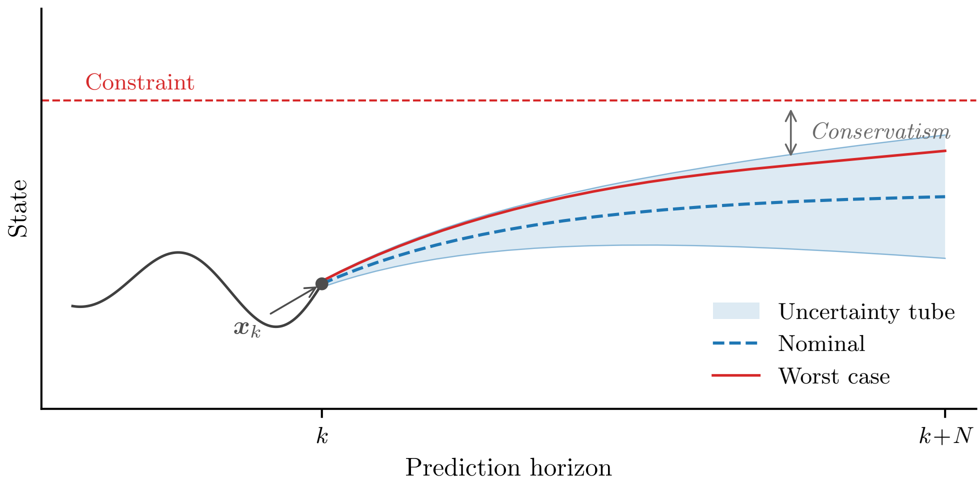

Figure 2.11: Robust MPC guarantees feasibility by optimizing against the worst-case trajectory (red) within an uncertainty tube (blue shading). The gap between the worst-case trajectory and the constraint boundary represents the conservatism cost of the robust approach.

2.3.1.2 Stochastic Approaches

Stochastic MPC methods explicitly incorporate forecast uncertainty through scenario-based optimization [115]. Rather than optimizing against worst cases, stochastic approaches consider a distribution of possible futures, optimizing expected performance while potentially allowing constraint violations with bounded probability. This probabilistic treatment can yield less conservative solutions that capture more economic value when outcomes are favorable.

Studies demonstrate that computational time increases significantly compared to deterministic models, growing exponentially with the number of scenarios considered. This computational scaling limits the number of scenarios that can be practically included, potentially missing important tail events. Two-layer approaches address this by deriving day-ahead plans using stochastic optimization while making intraday adjustments using rolling optimization to account for forecast inaccuracies [55], as illustrated in Figure 2.12. The hierarchical decomposition manages computational complexity while capturing uncertainty at appropriate timescales.

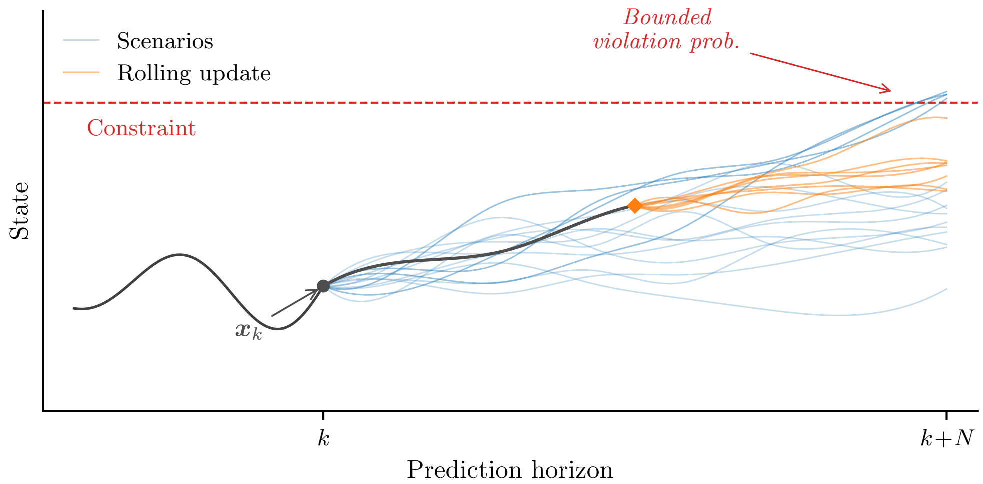

Figure 2.12: Stochastic MPC with scenario-based two-layer optimization. Scenario trajectories (blue) fan out from the current state $\boldsymbol{x}_k$; a bounded fraction may violate the constraint (dashed red). At the rolling-update point (orange diamond), a narrower intraday scenario fan (orange) re-optimizes from the realized state with reduced uncertainty.

2.3.1.3 Risk Quantification

Between pure robust and stochastic extremes, risk-aware formulations offer nuanced handling of uncertainty. CVaR-based optimization minimizes exposure to extreme financial losses rather than purely optimizing for worst cases, allowing balanced trade-offs between risk and profitability [83].

Formally, for a random cost variable J and a confidence level $\alpha \in (0, 1)$, the Value at

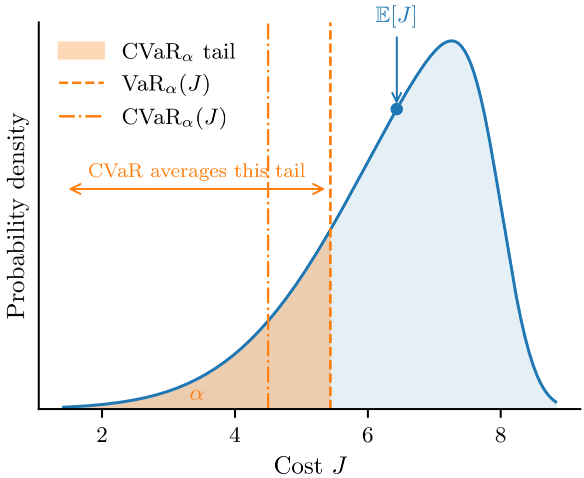

Figure 2.13: CVaR risk quantification. The shaded region shows the worst α-fraction of cost outcomes. $\mathrm{VaR}_\alpha(J)$ (dashed) marks the quantile threshold; $\mathrm{CVaR}_\alpha(J)$ (dash-dot) is the conditional mean over that tail, capturing the severity of extreme losses that VaR alone ignores.

Risk is the threshold below which costs fall with probability α:

$$\mathrm{VaR}_\alpha(J) \triangleq \inf\{\gamma \in \mathbb{R} \mid \Pr(J \leq \gamma) \geq \alpha\}. \tag{2.20}$$

VaR identifies the boundary of the worst α-fraction of outcomes but reveals nothing about the severity of losses beyond that threshold. The Conditional Value at Risk addresses this limitation by averaging over the tail:

$$\mathrm{CVaR}_\alpha(J) \triangleq \mathbb{E}\big[J \mid J \leq \mathrm{VaR}_\alpha(J)\big]. \tag{2.21}$$

CVaR is a coherent risk measure—it satisfies subadditivity, monotonicity, positive homogeneity, and translation invariance—making it well-suited for optimization [78]. By adjusting α, decision-makers control the degree of conservatism: lower α focuses on more extreme tail events, while higher α approaches the expected value. Figure 2.13 illustrates this geometry: $\mathrm{VaR}_\alpha(J)$ marks the α-quantile of the cost distribution, and $\mathrm{CVaR}_\alpha(J)$ is the mean of the shaded tail to the left of that threshold.

This approach is particularly relevant for battery storage participation in electricity markets while balancing profitability, financial risks, and degradation costs. Rather than guaranteeing constraint satisfaction, CVaR-based control accepts that adverse outcomes may occasionally occur while ensuring that such outcomes do not become catastrophic. Chapter 3 develops risk-aware MPC formulations that balance expected profit against CVaR-measured downside risk for microgrid energy management.

2.3.1.4 Plant Model Mismatch

Beyond external uncertainties in prices and weather, model mismatch represents a fundamental source of uncertainty that originates within the control system itself. Model mismatch arises from unmodeled dynamics, parameter uncertainty, and structural simplifications made during model development. Time delays can significantly affect system behavior and control performance if not adequately accounted for, as the controller responds to outdated information about system state.

Various methods exist to identify and model time delays, including state-space realization approaches extracting delay information from impulse responses [56] and correlation analysis improving control accuracy [53]. However, delays may vary with operating conditions, and other forms of mismatch—unmodeled nonlinearities, parameter drift, and structural changes—compound the challenge. The modeling methods of Chapter 5 address delay handling, while the monitoring methods of Chapter 4 detect when accumulated mismatch invalidates control models.

Literature gap *Not solved* — Robust and scenario-based stochastic MPC formulations either pay an unbounded conservatism cost or scale exponentially with the number of disturbance scenarios, leaving operators without a tractable mechanism to express asymmetric upside–downside preferences over economic outcomes. *Why it matters* — Microgrid operators and greenhouse growers face markets and weather where average performance and tail risk must be traded off explicitly; one-size-fits-all worst-case or expected-value objectives forfeit revenue without commensurate safety gains. *Addressed in* — Section 3.3, which develops a CVaR-based risk-aware MPC that exposes a tunable risk level to operators, and Section 3.2, which integrates economic objectives into a tractable predictive control formulation suitable for low-cost industrial controllers.

2.3.2 Sources of Uncertainty

Multiple uncertainty sources affect industrial control systems, each requiring appropriate modeling and mitigation strategies. Understanding these sources is essential for designing control systems that remain effective across operating conditions.

Market prices: Energy markets exhibit significant price volatility driven by supply-demand imbalances, weather events, generator outages, and market participant behavior. A market-oriented perspective considers risk-averse aggregators utilizing CVaR-based optimization to provide multiple services including reserve markets and imbalance settlements [41, 40]. Price uncertainty propagates through control decisions: profitable arbitrage opportunities may vanish if prices move adversely, while conservative hedging may forfeit value when prices remain stable.

Weather and renewable generation: Renewable power production exhibits inherent variability that cannot be eliminated through better forecasting. Solar generation depends on cloud cover, atmospheric conditions, and panel degradation; wind generation varies with weather patterns at multiple timescales. Multi-time scale approaches address forecast inaccuracies through rolling optimization, updating plans as more accurate short-term forecasts become available [55]. Hybrid architectures combine predictive optimization with rule-based logic for reliability, providing fallback behavior when predictions prove unreliable [58].

Model mismatch: Neural network-based frameworks have emerged for real-time scheduling across multiple timescales, effectively managing operational complexity when system models are imperfect [104]. Recent co-optimization frameworks demonstrate how sizing, scheduling, and EMS logic can be aligned across physical and virtual layers, reducing mismatch by jointly optimizing hardware and control [35]. Nevertheless, persistent mismatch between models and physical systems remains a fundamental challenge that motivates the monitoring approaches of Section 2.4.

Literature gap *Not solved* — Most uncertainty taxonomies in the EMS literature treat price, weather, and model-mismatch disturbances in isolation, producing controllers that hedge well against one source while leaving the others uncompensated. *Why it matters* — Real microgrids are exposed to all three uncertainty sources simultaneously, and ignoring their correlation can turn a profitable price-arbitrage strategy into a loss-making one when renewable forecasts and market prices co-vary adversely. *Addressed in* — Section 3.3, where price, renewable, and load uncertainties are jointly propagated through a scenario-based CVaR-MPC that quantifies their compounded effect on profit and reserve obligations.

2.3.3 Risk-Aware Control

Risk-aware formulations integrate uncertainty handling with economic objectives, balancing profit maximization with reliability requirements. The objective of microgrid control is most commonly formulated as cost reduction, including minimizing energy procurement costs complemented with battery degradation expenses [105, 55, 82, 3]. Other objectives include reducing peak demands and flattening load profiles [49, 31, 74], minimizing bidirectional grid exchange regardless of price [62], or minimizing imbalance deviations from scheduled positions.

When addressing imbalance, many studies overlook that not all energy deviations incur penalties—and some may even generate revenue depending on market conditions. Few approaches actively exploit imbalance pricing by strategically positioning systems out of balance when doing so is profitable [88, 83, 41]. This sophisticated market participation requires accurate forecasting, risk management, and rapid response capabilities. Additionally, some works incorporate environmental impact by integrating pollutant treatment cost models into economic optimization [27], recognizing that sustainability objectives extend beyond pure financial metrics.

Even the most sophisticated risk-aware controllers operate under assumptions about system behavior that may be violated. Equipment degrades, operating conditions change, and unforeseen events occur. This reality motivates the monitoring and safety systems examined in the following section, which detect when control assumptions no longer hold and provide the adaptive capability to maintain safe operation.

Literature gap *Not solved* — Existing risk-aware microgrid formulations seldom couple market participation strategies—imbalance settlement, reserve provision, intraday arbitrage—with battery degradation costs and operator-tunable risk preferences in a single tractable optimization. *Why it matters* — Aggregators leave revenue on the table when imbalance pricing is treated as a penalty rather than a signal, and they accelerate asset wear when degradation is not co-optimized with profit, eroding the business case for storage investment. *Addressed in* — Section 3.3, which formulates a CVaR-aware EMS that strategically exploits imbalance pricing while accounting for battery degradation, providing operators with a single dial to balance expected profit against tail risk.

2.4 Operational Safety and Monitoring

Advanced control strategies, no matter how sophisticated, ultimately depend on the validity of their underlying models and assumptions. The preceding sections established methods for data-driven modeling and uncertainty-aware control; this section addresses the complementary challenge of ensuring that these methods remain valid over extended deployments. Monitoring and safety systems close the loop, detecting when conditions have changed sufficiently that models require updating or control actions require human review. These methods play a supporting role, while uncertainty can be quantified and accounted for at the control system level, unexpected behavior cannot be predicted and must be detected and diagnosed.

2.4.1 Safety-Critical Systems

Many industrial systems are safety-critical, where process monitoring is essential to protect both profit and life. These systems operate at various operating points, often driven by control to meet production goals under varying demand and supply conditions. However, environmental fluctuations and varying component quality may lead to unexpected and persistent changes in system behavior that were not anticipated during design or commissioning.

These events can jeopardize optimal operations, accelerate wear and degradation, and occasionally result in catastrophic consequences including equipment damage, production loss, or human casualties. The financial and human stakes motivate investment in monitoring systems that can detect problems early, before minor anomalies develop into major failures. Effective monitoring requires understanding what constitutes normal behavior, how to detect deviations, and how to adapt as the definition of normal evolves.

Classical reliability engineering has developed well-established frameworks for quantifying and managing these risks. Safety Integrity Levels (SIL), defined by IEC 61508, provide a standardized metric for the required risk reduction of safety-instrumented functions, grading systems from SIL 1 to SIL 4 based on the tolerable frequency of dangerous failures. Systematic failure analysis methods such as Failure Mode and Effects Analysis (FMEA) and Fault Tree Analysis (FTA) enable structured identification of potential failure pathways and their consequences, supporting both design-time risk assessment and operational decision-making. Mean Time Between Failures (MTBF) remains a widely used reliability metric; however, MTBF alone is insufficient because it reduces the full failure-time distribution to a single scalar, masking the probabilistic uncertainty inherent in component degradation and wear-out processes. The data-driven monitoring, change-point detection, and predictive maintenance contributions developed in subsequent chapters complement these classical frameworks rather than replacing them. By providing real-time, context-aware assessments of system health, these methods supply the continuous diagnostic information that static reliability metrics and periodic inspections cannot capture, thereby strengthening the overall safety case.

Literature gap *Not solved* — Classical reliability frameworks such as SIL, FMEA, and MTBF are inherently static and design-time, providing no mechanism for context-aware updating of safety limits as operating conditions and equipment health drift over the lifecycle. *Why it matters* — Fixed thresholds set at commissioning either trip nuisance alarms once operating regimes shift or, worse, miss genuine hazards that emerge between periodic inspections, eroding operator trust in the safety case. *Addressed in* — Section 4.2, which derives data-driven operating limits that adapt to the current process distribution, and Section 4.3, which provides continuous anomaly assessment that complements rather than replaces SIL-graded protection layers.

2.4.2 Role of Machine Learning in Control

Machine learning increasingly augments and sometimes replaces classical control and monitoring approaches. Anomaly detection systems play a critical role in risk-averse systems by identifying abnormal patterns and, crucially, adapting to novel expected patterns as operating conditions evolve [16]. Recent studies underscore the need for holistic and tunable detection methods accessible to operators [51, 48, 18], recognizing that monitoring systems must be configurable to specific operational contexts rather than applied as black boxes.

Substantial aspects pose challenges to anomaly detection in IoT environments, including the temporal, spatial, and external context of measurements, multivariate characteristics, noise, and non-stationarity [18]. Deep learning methods have proven effective for complex nonlinear systems where hand-crafted features fail to capture relevant patterns [112]. Feature engineering techniques play a crucial role in capturing contextual properties that raw measurements obscure [26], though they may introduce categorical variables and significantly increase data dimensionality, requiring careful consideration of computational and statistical implications [85].

The progression from pure classification toward detection, diagnosis, and adaptation reflects growing appreciation for the lifecycle challenges of industrial monitoring. The following subsections trace this progression, from identifying known faults through detecting unknown anomalies to recognizing fundamental changes in system dynamics.

Effective monitoring system design should satisfy established design desiderata for novelty detection [60]. Key principles include robustness, defined as maximizing the exclusion of novel samples while minimizing false exclusion of known patterns; parameter minimization, reducing the number of user-set parameters to improve deployability; adaptability, leveraging detected novelties to retrain and refine the underlying model; and computational efficiency, ensuring that the method supports online operation under real-time constraints. The monitoring contributions in subsequent chapters explicitly target these desiderata: the dynamic process limits and anomaly detection framework of Chapter 4 minimize user-defined parameters and adapt continuously, while the toDMDc-based change detection maintains bounded computational complexity per update.

Literature gap *Not solved* — Most machine-learning-augmented control and monitoring methods are validated on offline benchmarks and rely on extensive hyperparameter tuning, neglecting the parameter-minimization and interpretability desiderata that determine whether a method can actually be commissioned by a plant engineer. *Why it matters* — Industrial adoption is gated not by peak benchmark accuracy but by deployability, transparency, and the ability of operators to trust and audit model outputs in safety cases. *Addressed in* — Section 5.2 and Section 5.3, which deliver learned models with auditable operator-theoretic and physics-consistent structure suitable for embedding within model-based controllers and monitors.

2.4.3 From Classification to Detection

Traditional approaches to process monitoring often frame the problem as classification: given current observations, determine whether the system is operating normally or exhibiting a known fault condition. While classification provides clear actionable outputs, it assumes a closed world where all possible conditions have been enumerated in advance—an assumption rarely satisfied in practice.

2.4.3.1 Fault Detection

Fault detection focuses on identifying known failure modes for which historical examples exist. This supervised learning perspective leverages labeled data from previous incidents to train classifiers that recognize similar conditions in future operation. However, several challenges complicate this task even when labeled data are available:

1. Faults may not have been observed historically, leaving no training examples for conditions that have not yet occurred
2. Unprecedented and unexpected behavior may occur outside known fault taxonomies, falling into gaps between defined categories
3. Known fault signatures may manifest differently in new operational contexts, causing classifiers trained on historical data to miss or misclassify current incidents

Data-driven approaches for investigating fault effects face strong dependency on operating conditions [32, 13, 57]. A classifier trained under one set of conditions may fail when conditions change, even if the underlying fault mechanism is identical. Multi-expert systems for operational state detection offer robust solutions against industrial data challenges like noise and missing values but require annotated historical samples that may be expensive or impossible to obtain [11].

2.4.3.2 Minority Class Handling

The fundamental challenge of fault detection is data imbalance: normal operation dominates the historical record, while fault events are rare. Given the infrequent occurrence of anomalies and their potential absence in training data, incorporating synthetic data or feature extraction for various detected events emerges to assist diagnosis [13]. Generating realistic synthetic faults requires domain expertise to ensure that synthetic examples reflect physically plausible failure modes.

Attribute-weighted outlier detection algorithms designed for high-dimensional datasets with mixed categorical and numerical data assign different weights based on attribute importance [54]. This weighting helps focus detection on the most informative features while downweighting noise. Deep learning methods with synthetic normal data enhance detection of outliers with subtle deviations [24], learning representations that separate normal from abnormal even when abnormal examples are scarce.

Even with corrective class weights, a fundamental limitation persists: gradients from the overwhelming majority of easy negative examples accumulate to dominate parameter updates, effectively drowning out the learning signal from rare but critical positive cases—a phenomenon known as gradient drowning. This effect renders supervised classification fundamentally unsuitable at the extreme imbalance ratios common in industrial fault detection, where the ratio of normal to faulty observations routinely exceeds 1:1000. Consequently, the field has shifted toward unsupervised anomaly detection, which sidesteps the imbalance problem entirely by modeling only normal behavior and flagging deviations from the learned representation.

2.4.3.3 Anomaly Detection

Moving beyond known faults, anomaly detection identifies unknown deviations from normal behavior without requiring examples of specific fault types. This unsupervised perspective defines normal based on historical operation and flags observations that deviate significantly. Several phenomena challenge anomaly detectors:

1. Systems slowly drifting away from initial operating conditions due to wear, degradation, or environmental changes

2. Changes in the definition of normal behavior due to process modifications, new products, or updated operating procedures

3. New operation modes that represent legitimate but previously unseen states, which should not trigger alarms despite being novel

Anomaly detection was reborn in the era of big data to face these new challenges. Former studies were mainly concerned with domain-specific detection while trained offline on static datasets [16]. However, anomalies from diverse sources mutate over time, requiring model updates that track evolving definitions of normal. Companies expanded research on generic frameworks combining prediction, detection, and alert mechanisms [51, 48], enabling automated anomaly detection pipelines that reduce dependence on domain expertise while remaining configurable for specific applications.

2.4.3.4 Change Detection

At the most fundamental level, change-point detection (CPD) recognizes shifts in the underlying dynamics governing system behavior—changes that affect not just individual observations but the statistical relationships between variables. Monitoring abrupt and gradual changes is crucial for ensuring system reliability and safety, as such changes may indicate equipment degradation, process upsets, or environmental shifts that invalidate operating assumptions.

Traditional monitoring methods like Statistical Process Control rely on independent and identically distributed assumptions often violated in industrial systems [69, 47]. Industrial process data exhibit autocorrelation, seasonality, and non-stationarity that render classical control charts unreliable. Key questions arise for any change detection method:

1. How was the detection decision made, and can operators understand and trust the reasoning?

2. Is the detection model stable under varying circumstances, or does it produce spurious detections under certain conditions?

3. How sensitive is detection to initial conditions and tuning parameters?

Change points may be characterized by shifts in dynamics more effectively captured in the frequency domain than the time domain [21, 38]. A change in oscillation frequency or damping ratio indicates fundamental alteration of system dynamics even if mean values remain constant. Subspace-based methods address distributional assumptions and efficiently extract complex dynamical features [42]. The main advantages include absence of distributional assumptions and ability to handle high-dimensional data efficiently, making them suitable for multivariate industrial systems.

To ground the contributions of Chapter 4, we introduce formal definitions of the core concepts. Given a sequence of observations $\{y_k\}_{k=1}^N$ generated by a process with distribution P_k, a *change point* τ is the time index at which the generating distribution shifts, i.e., $P_k \neq P_{k+1}$ for $k = \tau$. The goal of change-point detection is to identify τ as quickly as possible after it occurs while controlling the rate of false alarms. A widely used approach is the cumulative sum (CUSUM) detection statistic, defined recursively as

$$S_k = \max(0,\ S_{k-1} + s_k - \nu),$$

where s_k is a log-likelihood ratio comparing the post-change and pre-change hypotheses and ν is a drift parameter that controls sensitivity. A change is declared when S_k exceeds a predefined threshold. In subspace-based methods, the *reconstruction error* provides an alternative detection statistic by measuring the projection residual:

$$e_k = \|y_k - \hat{y}_k\|,$$

where $\hat{y}_k$ denotes the projection of the observation y_k onto the learned subspace. A sustained increase in e_k signals that the current subspace no longer adequately represents the observed dynamics. Optimal detection under both minimax and Bayesian formulations has well-established theoretical bounds [87] that characterize the fundamental trade-off between detection delay and false alarm rate. These bounds inform the design of the detection methods in subsequent chapters, where reconstruction error ratios serve as the primary detection statistic.

Literature gap *Not solved* — Supervised fault classifiers presume labeled fault libraries that industrial operators rarely possess, while unsupervised anomaly detectors typically require offline retraining whenever the definition of normal evolves, breaking continuous deployment. *Why it matters* — Long-running industrial systems experience legitimate regime changes, slow drift, and previously unseen operating modes; a detector that cannot distinguish these from genuine faults floods operators with false alarms and ultimately gets disabled. *Addressed in* — Section 4.3, which develops an adaptive anomaly detection framework that incrementally updates its model of normal behavior under bounded compute and avoids the labeled-fault assumption altogether.

The detection methodology in Section 4.2 and the adaptive framework of Section 4.3 operationalize this shift from supervised classification to unsupervised, distribution-aware detection.

2.4.4 From Detection to Diagnostics

Detecting that something has changed is only the first step; understanding what changed and why is essential for appropriate response. The progression from detection to diagnostics enables operators to take targeted corrective action rather than simply knowing that intervention is needed.

2.4.4.1 Measuring Detection Success

Evaluating detection performance requires metrics beyond simple accuracy, which can be misleading when classes are imbalanced or detection timing matters. Time-to-detect is critical in safety applications where delayed detection can have severe consequences—a detector that eventually identifies all anomalies but with significant delay may be useless for preventing damage.

Average run length (ARL) and Type-II error probability can be characterized using asymptotic random matrix theory, enabling principled optimization of metrics such as expected detection delay [108]. These theoretical results provide guidance for parameter selection that balances false alarm rate against detection latency. Kernel change-point algorithms can correctly identify the number and location of change points under well-chosen penalties [33], providing theoretical guarantees that support deployment in safety-critical applications.

The choice of evaluation methodology itself remains an active research question, as standard metrics may not adequately capture the temporal aspects of detection quality. Recent work has formally explored the limitations and relationships among time-series anomaly detection evaluation metrics, revealing that commonly used measures can produce misleading assessments when detection timing and event duration are not properly accounted for [95].

2.4.4.2 Explainability, Interpretability, and Observability

Recent advances have broadened the scope of monitoring to include root cause identification through explanatory methods capable of diagnosing faults across systems. Two distinct approaches exist for making detection methods interpretable.

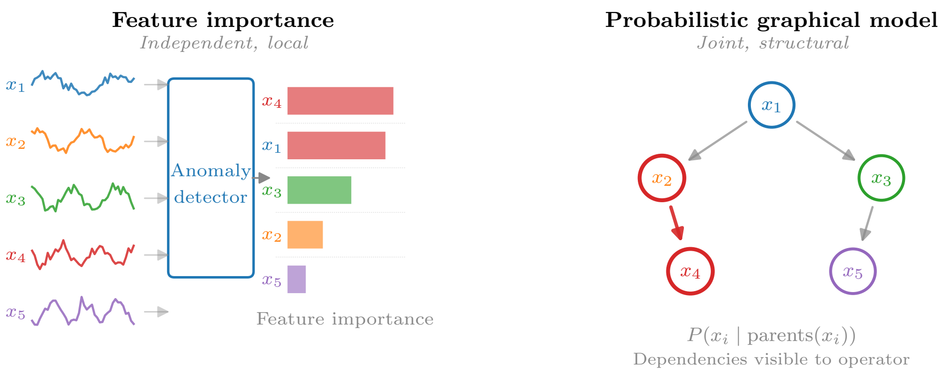

Figure 2.14: Two approaches to interpretable anomaly detection. **Left:** Feature importance treats each input independently to rank which measurements most influenced the detection decision. **Right:** Probabilistic graphical models encode joint dependency structure—nodes represent sensor variables and edges encode conditional relationships $P(x_i \mid \text{parents}(x_i))$—providing operators with an inspectable causal structure.

The first approach treats explainability as feature importance [15, 70, 4]. By examining which input features most strongly influence detection decisions, operators can identify the measurements most associated with detected anomalies. This approach allows explanation of novelty by considering features independently, though it may miss interactions between variables.

The second approach uses statistical learning to create models explainable via probability [114, 107]. Variational Bayesian inference with probabilistic graph neural networks can model posterior distributions of sensor dependency for fault localization on unlabeled data [114]. The probabilistic framework provides natural measures of confidence and uncertainty. Bayesian networks for fault detection and diagnosis, where nodes represent normally distributed variables and multiple regression defines relationships, enable offline training with online detection [107]. The graphical structure directly encodes causal assumptions that operators can inspect and validate. The comparison of the approaches can be seen in Figure 2.14.

Literature gap *Not solved* — Feature-importance explanations treat inputs independently and miss the joint sensor structure that drives faults, while Bayesian graphical diagnostics typically require offline training and curated dependency graphs that do not evolve with the system. *Why it matters* — Operators investigating an alarm need a root-cause explanation that respects multivariate sensor coupling and remains valid after process modifications, otherwise diagnoses are either misleading or stale within months of commissioning. *Addressed in* — Section 4.3, which couples detection with multivariate conditional-distribution diagnostics that adapt online, and Section 4.4, which attributes detected changes to specific modes of the identified operator.

The AID framework presented in Section 4.3 concretely closes this detection-to-diagnostics gap by coupling anomaly scores with feature-level root-cause attribution.

2.4.5 From Static to Evolving Models

The presence of non-stationarity, stemming from concept drift and change points, presents substantial challenges for long-term deployment [79]. Models trained on historical data inevitably become stale as systems evolve. In practical scenarios, changes tend to be unpredictable in both spatial and temporal aspects, requiring systems with solid outlier rejection capabilities that can distinguish genuine anomalies from normal evolution [10]. This underscores the critical importance of adaptation to evolving data structures in long-term deployments where manual retraining is infeasible.

2.4.5.1 Long-term Deployment

Systems with time-varying characteristics require models that evolve over operational lifetime rather than remaining fixed at initial training values. Offline training is inherently inadequate for adapting to anticipated novel patterns, rendering it unsuitable for sustained operation on IoT devices that must run continuously for months or years. The computational and communication constraints of edge deployment preclude frequent retraining against centralized datasets.

Self-supervised learning is a paradigm in which the system learns to predict part of its input from other parts, using the inherent structure of the data as a supervisory signal rather than requiring external labels. While both self-supervised and unsupervised learning avoid manual annotation, self-supervised learning generates explicit supervision from the data itself, producing richer learning signals that capture temporal and structural relationships—a distinction that proves critical for monitoring systems operating over extended deployments.

Self-supervised approaches facilitate intelligent adaptation concerning detected change points, increasing adaptation speed where the probability of concept drift is high. By leveraging the structure of the detection problem itself—distinguishing past from future observations—these methods can update continuously without requiring external labels or human intervention.

2.4.5.2 Sequential Detection and Updates

Online learning models relax the need for data availability during model training [32], processing data from bounded buffers sequentially [12, 2]. This streaming paradigm aligns naturally with industrial data collection, where measurements arrive continuously and storing complete histories is impractical.

Incremental learning methods allow adaptation while constraining dataset storage to bounded memory. Supervised operator-in-the-loop solutions show detector adaptation to data labeled on the fly [72], enabling human feedback to guide model updates when automatic adaptation is insufficient. The monitoring contributions of Chapter 4 develop incremental methods that update in constant time per observation.

2.4.5.3 Adaptation to Slow Changes

Self-supervised methods using contrastive learning evaluate cosine similarity between predicted future representations and anticipated representations, detecting evolution with high accuracy [22]. By learning to distinguish recent past from near future, these methods become sensitive to gradual shifts that might escape threshold-based detection.

The balance between early transition detection and low false alarm rate represents a key challenge [87]. Detecting changes quickly requires sensitivity to small deviations, but high sensitivity produces false alarms during normal variability. Practical systems must navigate this trade-off based on the costs of missed detections versus false alarms in specific applications.

2.4.5.4 Accommodation of New Operations

New operational modes may persist for undefined durations, requiring methods that distinguish legitimate mode changes from anomalies warranting intervention. A production facility may introduce new products, adjust schedules, or modify processes in ways that represent intentional changes rather than faults.

Online vectorized forecasting methods based on autoregression have shown capability to adapt to non-stationarity in multivariate systems without supervision [64, 113]. Feature decomposition and contrastive learning detect both abrupt and subtle changepoints by isolating predictable components from residual terms [9]. This decomposition helps distinguish changes in predictable dynamics from changes in noise characteristics, providing richer diagnostic information than scalar change scores alone.

Literature gap *Not solved* — Incremental and self-supervised methods adapt quickly but typically lack a principled mechanism to separate abrupt change points from slow drift, and they rarely expose the identified dynamics in an interpretable form that can be reused by downstream controllers. *Why it matters* — In long-running industrial deployments, conflating drift with change forces a choice between premature model resets that discard valid history and delayed adaptation that perpetuates stale predictions, both of which degrade control performance over months of operation. *Addressed in* — Section 4.4, which combines time-delay online DMDc with reconstruction-error change statistics under bounded per-step compute, and Section 4.3, which adapts its normal-behavior model continuously while flagging genuine regime shifts.

The toDMDc estimator developed in Section 4.4 embodies this transition by maintaining a continually refreshed linear model whose residual signals dynamic change.

This chapter has established the theoretical and practical foundations for the contributions presented in subsequent chapters. The industrial context of Section 2.1 defines the constraints and requirements that practical solutions must satisfy. The data-driven modeling approaches of Section 2.2 provide the mathematical tools for system identification. The control strategies of Section 2.3 leverage these models to optimize operation under uncertainty. Finally, the monitoring paradigms of Section 2.4 ensure that systems remain reliable as conditions evolve. The following chapters develop specific contributions within this integrated framework, demonstrating how modeling, control, and monitoring work together to enable sustainable industrial automation.

2.5 Synthesis: From Gaps to Opportunities

The per-section discussions throughout this chapter have repeatedly surfaced a recurring pattern: established methods provide strong foundations under stationary, well-instrumented, or fully labeled conditions, yet leave systematic gaps once industrial deployments confront unknown nonlinearities, evolving operating regimes, asymmetric risk, and constrained operator attention. Consolidating these gaps clarifies why the dissertation pursues a tightly coupled triad of modeling, control, and monitoring rather than improvements to any single component in isolation. Each gap identified in the preceding subsections opens a concrete opportunity that the dissertation exploits in a designated section of Chapter 3, Chapter 4, or Chapter 5. Table 2.1 aggregates this mapping, providing an at-a-glance bridge from background to contributions.

These opportunities motivate the three contribution chapters that follow: Chapter 3 pursues operator-tunable risk-aware and carbon-aware economic control by embedding CVaR and carbon-intensity signals directly into the MPC cost; Chapter 4 pursues adaptive monitoring through dynamic process limits, online conditional-distribution diagnostics, and reconstruction-error change detection that all update in bounded per-step compute; and Chapter 5 pursues interpretable data-driven identification through dictionary-free Deep Koopman embeddings and physics-informed DMDc, delivering models that controllers and monitors of the preceding chapters can audit and reuse.

Table 2.1: Synthesis of literature gaps, the opportunities they create, and the dissertation sections that exploit them.

Gap	Opportunity	Addressed in
Hand-crafted dictionaries fail for systems with unknown nonlinearities or time delays (Koopman/eDMD)	Learn lifting functions end-to-end so dictionary design becomes a training problem	Section 5.2
Pure data-driven models violate conservation/stability when extrapolating	Constrain DMDc identification to physically admissible matrix manifolds	Section 5.3
Online DMD adapts but cannot signal when a fundamental dynamics shift has occurred	Couple operator updates with reconstruction-error change-point statistics	Section 4.4
Stochastic MPC scales poorly with scenario count and offers no operator-tunable risk dial	Encode tail risk explicitly via CVaR and expose a single risk-preference parameter	Section 3.3
Economic MPC for greenhouses neglects CO_2 cost and carbon-aware grid signals	Integrate carbon intensity forecasts into a multi-objective economic cost	Section 3.2
Static SCADA alarm thresholds neither adapt to regime change nor accommodate sensor drift	Compute dynamic process limits via inverse CDF of an online Gaussian model	Section 4.2
Black-box anomaly detection fails to provide root-cause attribution under non-stationarity	Combine conditional-distribution diagnostics with continuous self-supervised adaptation	Section 4.3
Lifecycle-aware degradation tracking requires both sparse high-fidelity and dense low-fidelity data	Use physics-informed DMDc for resistance modeling and deviation-based RUL estimation	Section 5.3

Chapter 3

Risk-Aware and Economic Control

The preceding chapter established the theoretical foundations of data-driven modeling, advanced control strategies, and the industrial deployment context within which these methods must operate. This chapter applies these foundations to two complementary control problems: sustainability-driven greenhouse climate management and profit-maximizing microgrid energy management. Both applications demonstrate how Model Predictive Control frameworks can balance competing objectives—economic performance, environmental sustainability, and operational safety—while explicitly accounting for the uncertainties inherent in real-world systems.

Section 3.1 establishes the conceptual bridge from theoretical control to practical decision-making under uncertainty. Section 3.2 presents an Economic MPC framework for greenhouse climate control, demonstrating how economic and environmental trade-offs can be optimized within an educational context. Section 3.3 develops a comprehensive risk-aware energy management system for microgrids, extending the concepts to commercial-scale deployment with multiple revenue streams and explicit risk quantification. Finally, Section 3.4 reflects on the inherent limitations of even optimal controllers, motivating the monitoring approaches developed in subsequent chapters.

This chapter develops risk-aware control architectures using primarily mechanistic, first-principles models—greenhouse thermodynamics coupled with a crop growth model, and a simplified linear microgrid model with parametric uncertainty. Deeper data-driven system identification, including dictionary-free Koopman methods and physics-informed DMDc, is deferred to Chapter 5, which provides the linear surrogates that future deployments of the controllers presented here could substitute for the mechanistic models used in this chapter.

3.1 Optimal Decision Making under Uncertainty

The transition from theoretical control frameworks to practical industrial applications requires confronting a fundamental reality: uncertainty cannot be ignored. Weather forecasts deviate from actual conditions, market prices fluctuate unpredictably, and system models inevitably simplify complex physical phenomena. Rather than treating uncertainty as an obstacle to be minimized, effective control frameworks must embed uncertainty directly into the decision-making process.

This embedding takes different forms depending on the application context. In agricultural systems, uncertainty manifests through weather variability, crop response variation, and energy price fluctuations. In energy systems, uncertainty appears in renewable generation forecasts, load demand predictions, and market price volatility. Both domains share common characteristics: multiple competing objectives that cannot be simultaneously optimized, constraints that must be satisfied despite uncertainty, and decisions that have consequences extending beyond the immediate time step.

Economic Model Predictive Control provides a natural framework for addressing these challenges. Unlike tracking MPC, which minimizes deviation from predetermined setpoints, EMPC directly optimizes economic objectives—profit, cost, yield, or environmental impact—subject to system dynamics and constraints. This economic orientation aligns controller objectives with operational goals, eliminating the gap between what operators want to achieve and what controllers optimize, creating venue for embedding business related risks into the control framework.

However, purely deterministic EMPC fails to account for uncertainty in predictions. The optimal solution for expected conditions may perform poorly when actual conditions deviate from forecasts. Robust approaches that optimize against worst-case scenarios provide guarantees but often prove overly conservative, forfeiting economic value to protect against scenarios that rarely materialize. Stochastic approaches that optimize expected performance across scenario distributions offer balance but introduce computational complexity that may preclude real-time execution.

The following sections present two complementary approaches to this challenge. The greenhouse application demonstrates EMPC principles in an educational context, showing how competing objectives can be balanced through appropriate cost function design. The microgrid application extends these principles to commercial-scale deployment, introducing risk-aware formulations based on Conditional Value at Risk that provide tunable conservatism while maintaining computational tractability.

3.2 Economic MPC for Sustainability Education

Greenhouse climate control presents an ideal context for demonstrating Economic MPC principles. The system exhibits rich dynamics coupling temperature, humidity, and carbon dioxide concentration with crop growth. Multiple actuators—heating, ventilation, humidification, and CO_2 enrichment—provide control authority, while external weather conditions introduce time-varying disturbances. Economic and environmental objectives compete: maximizing crop yield requires energy-intensive climate control, yet sustainability demands minimizing both energy consumption and carbon emissions.

This section presents a Nonlinear Economic MPC framework for greenhouse climate control, demonstrating how these competing objectives can be balanced through appropriate formulation. Beyond its technical contributions, the framework serves as an educational platform, enabling exploration of economic and sustainability trade-offs in agricultural systems.

3.2.1 Greenhouse Climate Model

The greenhouse climate model captures the essential dynamics of temperature, humidity, and CO_2 concentration within the controlled environment, as illustrated in Figure 3.1. The model integrates well-established principles from thermodynamics, fluid dynamics, and mass transfer, building upon the Gembloux Dynamic Greenhouse Climate Model [61] and subsequent refinements [91, 99].

3.2.1.1 Temperature Dynamics

Temperature evolution is governed by convective, radiative, and conductive heat transfer between different compartments—cover, internal air, vegetation, growing medium, tray, floor, and soil layers. Convective heat transfer between surfaces follows standard correlations involving Nusselt, Reynolds, and Grashof numbers. The convective heat flux between two surfaces is expressed as:

$$Q_{\mathrm{conv}} = A_{\mathrm{c}}\,\mathrm{Nu}\,\lambda_{\mathrm{air}}\frac{T_1 - T_2}{d_{\mathrm{c}}}, \tag{3.1}$$

where A_{c} and d_{c} represent the characteristic area and length of a compartment, respectively, λ_{air} is the air thermal conductivity, and Nu is the Nusselt number determined by the dominant convection regime.

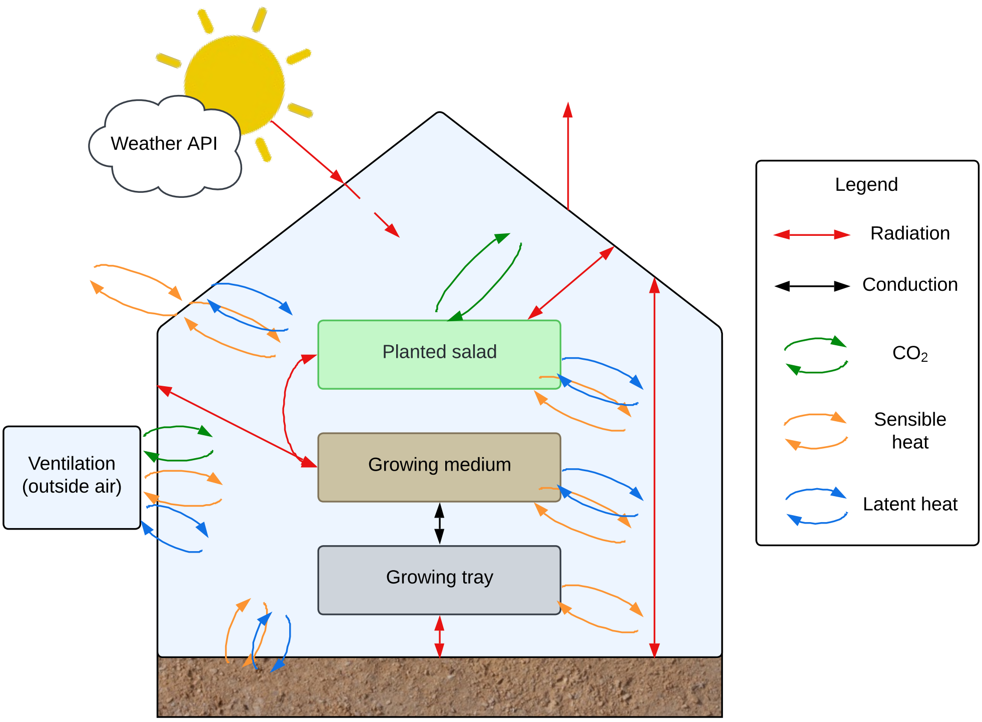

Figure 3.1: Simplified diagram of heat, mass, and CO_2 flows in the greenhouse model. Adapted from [99].

Radiative heat transfer between surfaces follows the Stefan-Boltzmann law:

$$Q_{\mathrm{rad}} = \frac{\varepsilon_1 \varepsilon_2}{1 - \rho_1 \rho_2 F_{12} F_{21}} \sigma A_1 F_{12} \left(T_1^4 - T_2^4 \right), \tag{3.2}$$

where σ is the Stefan-Boltzmann constant, F_{12} and F_{21} are view factors quantifying radiation exchange, and ε and ρ represent surface emissivity and reflectivity, respectively.

3.2.1.2 Humidity Dynamics

Humidity dynamics model vapor mass transfer, with air moisture content depending on temperature and pressure. The density of water vapor follows an empirical exponential relation:

$$C_{\mathrm{w}} = \rho_{\mathrm{air}} \cdot e^{11.56 - \frac{4030}{T + 235}}, \tag{3.3}$$

where ρ_{air} is air density and T is temperature. Mass transfer of water vapor due to convection involves the Sherwood and Lewis numbers, coupling humidity dynamics to temperature through latent heat effects.

3.2.1.3 Carbon Dioxide Concentration Dynamics

The CO_2 concentration is affected by photosynthesis, respiration, and exchange with external conditions. The external CO_2 concentration is computed from atmospheric conditions:

$$C_{\mathrm{CO_2,ext}} = \frac{4 \times 10^{-4} M_{\mathrm{CO_2}} P_{\mathrm{atm}}}{R\, T_{\mathrm{ext}}}, \tag{3.4}$$

where $M_{\mathrm{CO_2}}$ is the molar mass of CO_2, P_{atm} is atmospheric pressure, R is the gas constant, and T_{ext} is external air temperature in Kelvin. Internal concentration dynamics couple to plant growth through photosynthetic uptake and respiratory release.

3.2.1.4 State Space Representation

The complete greenhouse model integrates temperature, humidity, and CO_2 dynamics into a state vector $\boldsymbol{x} \in \mathbb{R}^{11}$ comprising cover temperature T_{c}, internal air temperature T_{i}, plant temperature T_{v}, growing medium temperature T_{m}, tray temperature T_{p}, floor temperature T_{f}, soil layer temperature T_{s}, water vapor concentration C_{w}, CO_2 concentration $C_{\mathrm{CO_2}}$, structural dry weight x_{sdw}, and non-structural dry weight x_{nsdw}.

For the length of Section 3.2, control input vector $\boldsymbol{u} \in \mathbb{R}^4$ comprises heating power Q_{heater}, ventilation rate R_{fan}, humidification rate V_{humid}, and CO_2 enrichment rate

u_{CO_2}. External conditions enter through a parameter vector $\boldsymbol{p} \in \mathbb{R}^{20}$ including external temperature, apparent temperature, wind speed, relative humidity, and plane-of-array solar radiation components.

3.2.2 Plant Growth Model

The growth of the crop—lettuce in the presented implementation—is modeled using a dynamic system capturing both structural (SDW) and non-structural dry weight (NSDW) dynamics [90]. Environmental factors including temperature, photosynthetically active radiation (PAR), and CO_2 concentration drive growth through interconnected mechanisms.

3.2.2.1 Growth Dynamics

The rate of change of structural dry weight is governed by a temperature-modulated specific growth rate:

$$\frac{\partial x_{\mathrm{sdw}}}{\partial t} = r_{\mathrm{gr},T_{\mathrm{i}}} x_{\mathrm{sdw}}, \tag{3.5}$$

where the specific growth rate r_{gr} represents accumulation of SDW based on available NSDW:

$$r_{\mathrm{gr}} = r_{\mathrm{gr,max}} \cdot \frac{x_{\mathrm{nsdw}}}{\theta\, x_{\mathrm{sdw}} + x_{\mathrm{nsdw}}} \cdot Q_{10}^{\left(\frac{T_{\mathrm{i}}-20}{10}\right)}. \tag{3.6}$$

Here $r_{\mathrm{gr,max}}$ is the maximum growth rate at 20°C, θ is a growth coefficient, and Q_{10} captures temperature sensitivity.

Non-structural dry weight evolves according to the balance between photosynthetic production, conversion to structural weight, and maintenance respiration:

$$\frac{\partial x_{\mathrm{nsdw}}}{\partial t} = c_\eta f_{\mathrm{phot},C_{\mathrm{CO_2}},T_{\mathrm{i}}} - \gamma r_{\mathrm{gr},T_{\mathrm{i}}} x_{\mathrm{sdw}} - f_{\mathrm{resp},T_{\mathrm{i}}}, \tag{3.7}$$

where c_η is a conversion efficiency, f_{phot} is the gross photosynthesis rate, f_{resp} is maintenance respiration, and γ accounts for biomass partitioning and respiratory losses.

3.2.2.2 Photosynthesis and Respiration

Gross canopy photosynthesis depends on incident PAR, ambient CO_2 concentration, and canopy characteristics. The maximum CO_2 assimilation rate follows a saturating response:

$$f_{\mathrm{phot,max}} = \frac{\epsilon\, u_{\mathrm{PAR}}\, g_{\mathrm{CO_2}}\, c_\omega (u_{\mathrm{CO_2}} - c^*)}{\epsilon\, u_{\mathrm{PAR}} + g_{\mathrm{CO_2}}\, c_\omega (u_{\mathrm{CO_2}} - c^*)}, \tag{3.8}$$

where ϵ is light-use efficiency, $g_{\mathrm{CO_2}}$ is canopy conductance to CO_2 diffusion, and c^* is the CO_2 compensation point. Actual photosynthesis is modulated by light absorption following Beer-Lambert attenuation through the canopy.

Maintenance respiration quantifies energy required for basic metabolic functions, split between shoot and root compartments with temperature-dependent scaling. As temperature increases, respiration demands rise, consuming energy that would otherwise support growth.

3.2.3 Economic Objective Function

The goal of NEMPC is to maximize revenue from crop production while minimizing costs associated with actuator use and environmental impact. The objective function comprises revenue from plant growth and costs from energy consumption and carbon emissions.

Revenue from lettuce production is proportional to biomass accumulation:

$$r_k = \frac{P_{\mathrm{L}} A_{\mathrm{c}}}{\rho_{\mathrm{dw}}} \sum_{i \in \{\mathrm{sdw},\mathrm{nsdw}\}} (x_{i,k} - x_{i,0}), \tag{3.9}$$

where P_{L} is the market price of lettuce per gram, A_{c} is the cultivated area, and ρ_{dw} is the dry-to-wet ratio.

Actuation costs at each time step include both energy costs and carbon costs:

$$c_k = \sum c_{\mathrm{energy},\boldsymbol{u},k} + c_{\mathrm{CO_2},\boldsymbol{u},k}, \tag{3.10}$$

where energy costs depend on electricity prices and actuator efficiencies, while carbon costs reflect the social cost of carbon applied to emissions associated with electricity consumption. The carbon intensity of electricity—varying with the generation mix—enables the controller to prefer actuation during periods of cleaner electricity supply.

3.2.4 NEMPC Formulation

The NEMPC problem minimizes cumulative stage cost over a finite prediction horizon N, subject to system dynamics and constraints:

$$\min_{\boldsymbol{u}} \sum_{k=0}^{N-1} (-r_k + c_k) \tag{3.11a}$$

$$\text{s.t. } \boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{p}_k), \tag{3.11b}$$

$$\boldsymbol{u}_{\min} \leq \boldsymbol{u}_k \leq \boldsymbol{u}_{\max}, \quad \forall k = 0, \ldots, N-1, \tag{3.11c}$$

$$\boldsymbol{x}_{\min} \leq \boldsymbol{x}_k \leq \boldsymbol{x}_{\max}, \quad \forall k = 0, \ldots, N, \tag{3.11d}$$

$$\boldsymbol{x}_0 = x_{\text{meas}}. \tag{3.11e}$$

The nonlinear dynamics f encode the greenhouse climate and plant growth models, while bounds on states and inputs enforce operational constraints.

External climate conditions enter as time-varying parameters $\boldsymbol{p}_k$, obtained from real-time weather data and forecasts. The optimization uses CasADi [6] for modeling and discretization, with IPOPT [92] as the nonlinear solver.

3.2.5 Educational Platform and Validation

The framework is implemented as an interactive web-based educational tool, enabling users to explore greenhouse control through multiple customization layers: greenhouse structure affecting energy exchange, orientation and location enabling connection to weather and carbon intensity forecast APIs, actuator selection and scaling, and control strategy modification including objective function weighting and constraint adjustment.

3.2.5.1 Actuator Influence Analysis

Simulation studies reveal the distinct influence of each actuator on plant growth, as shown in Figure 3.2. Ventilation and humidification have minimal direct impact on biomass accumulation, though they positively affect convection and energy transfer. Heating facilitates conversion of NSDW to SDW by maintaining optimal temperatures. Most significantly, CO_2 enrichment substantially enhances non-structural plant growth by elevating photosynthetic rates above ambient CO_2 limitations.

3.2.5.2 Performance Comparison

Comparative simulations against a no-control baseline demonstrate the effectiveness of the NEMPC approach. Over a 15-day growth period, NEMPC increased crop yield by

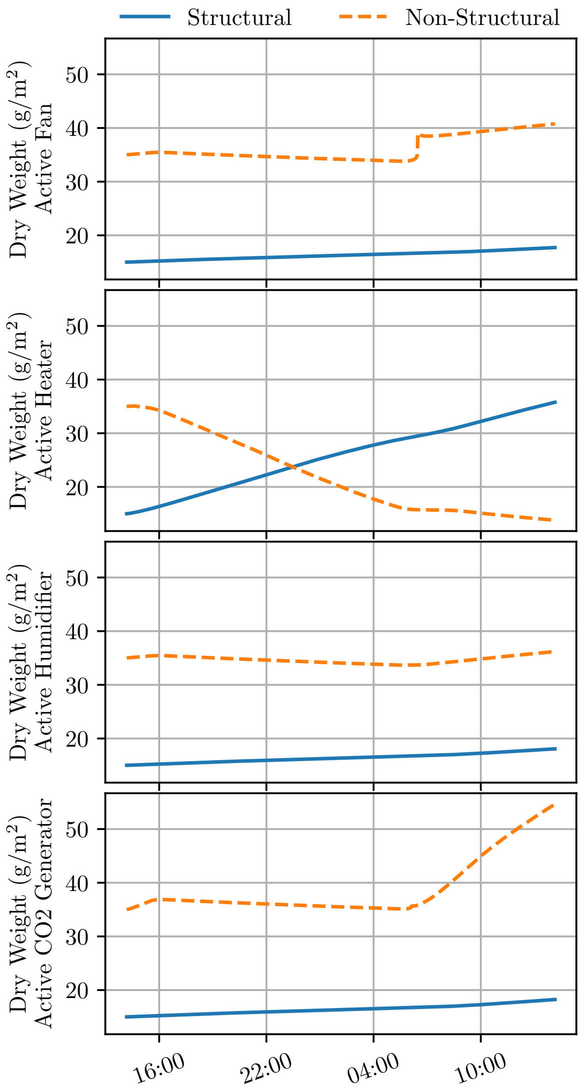

Figure 3.2: Responses of structural and non-structural dry weight to step changes from off to maximum actuation. From top to bottom: ventilation, heating, humidification, CO_2 enrichment, and no actuation (baseline).

435% and improved profit by 13.9% in the case of the CO_2 cost-aware scenario, and 17.7% in the case of the economic scenario, as summarized in Table 3.1. The control strategy heavily utilizes CO_2 enrichment while minimizing heater usage, reflecting the relative costs and growth impacts of each actuation mode (Figure 3.3).

Table 3.1: Performance comparison: NEMPC vs. no control over 15 days of simulated growth.

Parameter	No control	NEMPC (CO_2)	NEMPC ($)
Lettuce profit	834.01	4459.57	4979.23
Energy (Heater)	0.00	−59.30	−109.66
Energy (CO_2 Gen.)	0.00	−763.22	−794.75
CO_2 (Heater)	0.00	−158.51	−378.51
CO_2 (CO_2 Gen.)	0.00	−2521.36	−2705.12
Total	834.01	950.36	981.99

Incorporating the social cost of carbon intensity into the objective function reveals the trade-off between economic output and environmental impact. Carbon-aware NEMPC reduces CO_2 consumption by 13.1% compared to purely economic optimization, at the cost of 10% decreased plant growth and 3% decreased profit. This quantification enables informed decision-making about sustainability trade-offs.

3.2.5.3 Educational Impact

Survey results from students interacting with the platform indicate educational benefit across experience levels. Participants with minimal prior knowledge reported moderate improvements in understanding mathematical modeling, optimal control, and economic control, with strong positive impact noted for MPC concepts. More experienced participants rated the application as highly beneficial across all categories. The framework bridges the gap between academic learning and practical implementation, demonstrating how control theory translates to operational decision-making.

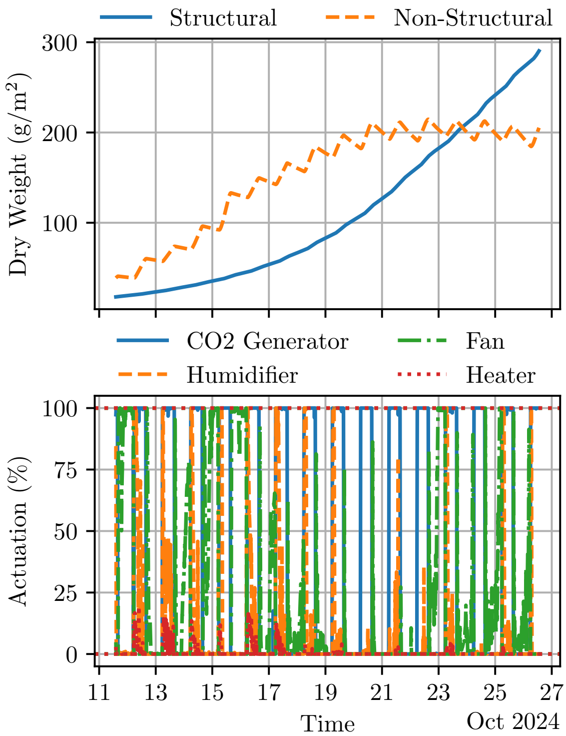

Figure 3.3: 15 days of simulation displaying periodic utilization of the CO_2 generator during daytime. The heater is used mostly during the initial growth phase. Both fan and humidifier are used extensively due to lower cost, mixing the humid air inside the greenhouse.

3.3 Scalable Deployment of Risk-Aware Microgrid Energy Management

While the greenhouse application demonstrates EMPC principles in an educational context, microgrid energy management presents a more challenging commercial-scale deployment scenario.[1] Microgrids—local networks of electricity sources, storage, and loads maintaining connection to the main power grid—must navigate complex interactions with energy markets, balance markets, energy suppliers, and grid operators. Economic considerations drive operation, with revenue stacking from multiple sources: price arbitrage, peak shaving, renewable self-consumption, and imbalance settlement.

Traditionally, power system reliability has been assessed through deterministic metrics such as Loss of Load Probability (LOLP) and Expected Energy Not Supplied (EENS), which quantify the frequency and magnitude of supply shortfalls, respectively. The increasing penetration of renewable energy sources, with their inherent stochastic generation profiles, challenges these conventional measures, as they fail to adequately capture the tail risks associated with correlated forecast errors and extreme weather events. Conditional Value-at-Risk (CVaR) addresses this limitation as a coherent risk measure that explicitly quantifies expected losses in the worst-case tail of the distribution, providing a more comprehensive assessment of reliability under uncertainty [19]. This motivates the CVaR-based formulation adopted in this section.

This section presents a comprehensive energy management system (EMS) that integrates risk-aware Model Predictive Control with a scalable software architecture for commercial deployment [1]. The risk-aware control problem and its tractable two-stage linear-program formulation are summarized first to ground the deployment analysis that follows. The principal emphasis, however, lies on the software architecture, the cost-driven matching of computational tasks to cloud service models, and the commercial validation across heterogeneous microgrid installations—the dimensions that determine whether risk-aware MPC remains economically viable once deployed at scale.

[1] The risk-aware MPC formulation summarized in this section was developed jointly with T. Ábelová as part of [117], where the CVaR-based two-stage methodology constitutes her primary contribution. The present section concentrates on the deployment architecture, the computation-to-service mapping, and the commercial validation, which constitute the present author's principal contributions to that work.

3.3.1 Microgrid Model and Control Hierarchy

The microgrid comprises four asset groups connected to a common AC bus, as illustrated in Figure 3.4: energy sources (primarily renewable), loads (controllable and non-controllable), storage systems, and the utility grid connection. Effective management requires decision-making across multiple timescales, from millisecond-level power electronics control to day-ahead market participation [1].

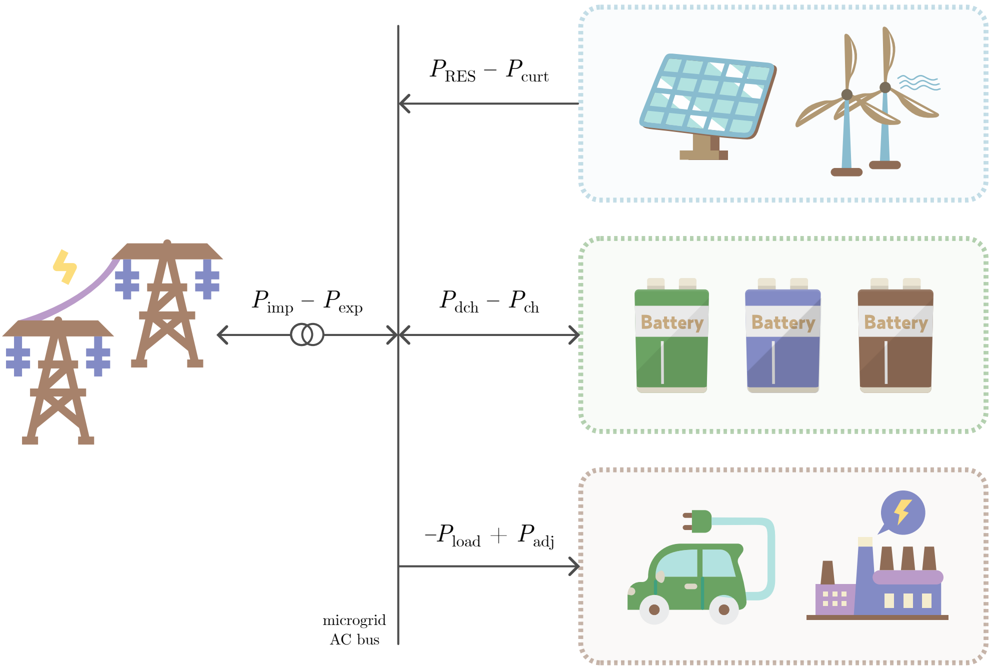

Figure 3.4: Scheme of a microgrid with four distinct asset groups connected to a common AC bus. Energy sources, loads, storages, and the utility grid connection are managed by the EMS.

3.3.1.1 Control Hierarchy

The EMS implements a four-layer control hierarchy, with each layer responsible for specific tasks at different timescales:

Business layer: Operating on timescales of months to years, this layer determines operational goals (cost saving, carbon reduction) and functionalities (price arbitrage, self-consumption maximization) based on contractual agreements and microgrid composition. Decisions require approval by business entities and define objectives for lower layers.

Energy management layer: This layer focuses on real-time decisions regarding energy quantities allocated to each unit or asset group, operating with a predictive horizon of hours to days at 15-minute granularity aligned with billing intervals and market settlement periods. Model Predictive Control serves as the core control strategy, enabling optimization over time while respecting operational constraints.

Power management layer: Operating at second-level timescales, this layer validates and distributes setpoints received from the energy management layer. Rule-based controllers handle fast microgrid dynamics and ensure operational limits are maintained. This layer also disaggregates group-level setpoints to individual units and manages state-of-charge balancing among storage systems.

Device control layer: At millisecond resolution, primary controllers embedded in power electronic converters regulate output voltage and current using PID control, ensuring precise tracking of power references from upper layers.

3.3.1.2 Power Balance and Storage Model

The microgrid operates under energy conservation, requiring balance between production and consumption at every time step:

$$\begin{aligned} P_{\mathrm{imp},k} + P_{\mathrm{dch},k} + P_{\mathrm{RES},k} + P_{\mathrm{adj},k} = \\ P_{\mathrm{exp},k} + P_{\mathrm{ch},k} + P_{\mathrm{load},k} + P_{\mathrm{curt},k}, \end{aligned} \tag{3.12}$$

where subscripts denote import, discharge, renewable generation, load adjustment, export, charging, load demand, and curtailment, respectively.

Energy storage systems provide flexibility through bidirectional power flow. The discrete battery model describes state-of-charge evolution:

$$\boldsymbol{E}_{k+1} = \boldsymbol{E}_k + \Delta T \left(\eta_{\mathrm{ch}} P_{\mathrm{ch},k} - \frac{1}{\eta_{\mathrm{dch}}} P_{\mathrm{dch},k} \right), \tag{3.13}$$

subject to capacity constraints $\underline{\beta} E_{\mathrm{avail}} \leq \boldsymbol{E}_k \leq \overline{\beta} E_{\mathrm{avail}}$, where η_{ch} and η_{dch} are charging and discharging efficiencies, and $\underline{\beta}$, $\overline{\beta}$ define allowable state-of-charge bounds.

3.3.1.3 Revenue Model

The microgrid revenue model encompasses costs and opportunities from external power systems and internal asset operations. For power exchange with the utility grid, costs

c_{imp}, c_{exp}, c_{short}, and c_{long} apply to import, export, shortage, and surplus, respectively. Variable costs over time enable price arbitrage—drawing more energy when prices are low, exporting when prices are high.

Imbalance settlement deserves particular attention. Microgrids must submit day-ahead schedules P_{DA} to grid operators and face penalties or receive compensation for deviations. The imbalance mechanism creates opportunities: when the grid is in surplus, parties in shortage receive compensation; when the grid is in shortage, parties contributing surplus are incentivized. Strategic positioning relative to system imbalance can generate significant revenue.

Deviations from schedule result in shortage or surplus:

$$P_{\text{short},k} = \max\big(0, P_{\text{imp},k} - P_{\text{exp},k} - P_{\text{DA},k}\big), \tag{3.14a}$$

$$P_{\text{long},k} = \max\big(0, P_{\text{DA},k} - P_{\text{imp},k} + P_{\text{exp},k}\big). \tag{3.14b}$$

The controller actively manages these deviations to exploit imbalance pricing when favorable conditions arise.

3.3.2 Risk-Aware MPC Methodology

In volatile environments deterministic MPC proves insufficient, robust MPC becomes overly conservative, and conventional scenario-based stochastic MPC scales poorly with the number of scenarios—making it impractical for real-time execution on non-commercial solvers. The risk-aware methodology summarized below addresses this gap through a two-stage optimization process that retains the uncertainty-aware decision quality of stochastic MPC at a computational cost comparable to a single-scenario deterministic solve. The deterministic stage maximizes total profit J over the prediction horizon,

$$J = \sum_{k=0}^{N-1} \left(\Delta T\, \boldsymbol{c}_k^\top \boldsymbol{u}_k - \boldsymbol{q}_k^\top \boldsymbol{e}_k\right), \tag{3.15}$$

where $\boldsymbol{c}_k$ are cost coefficients for control actions $\boldsymbol{u}_k$, $\boldsymbol{q}_k$ are penalty weights for constraint violations $\boldsymbol{e}_k$, and ΔT is the time step duration. The objective is maximized subject to battery dynamics, state and power-balance constraints, and initial conditions, with the receding-horizon principle applied throughout.

To account for forecast uncertainty, multiple scenarios $s \in \mathcal{S}$ represent potential realizations of renewable generation and load demand, sampled by Monte Carlo within forecasted confidence intervals. Rather than optimizing expected profit (which exposes the system to unacceptable downside risk) or worst-case profit (which forfeits revenue to protect against rare outcomes), the framework employs Conditional Value at Risk:

$$\mathrm{CVaR}_\alpha \triangleq \mathbb{E}\big[J_s \mid J_s \leq \mathrm{VaR}_\alpha\big], \tag{3.16}$$

where VaR_α is the profit threshold not exceeded with probability α. The parameter α provides a continuously tunable risk dial: $\alpha = 0\%$ collapses CVaR to the worst-case outcome (fully conservative), $\alpha = 100\%$ approaches the best-case scenario (tail risk ignored), and intermediate values calibrate risk aversion to operator tolerance. Adopting the linearization of [78], maximization of CVaR is cast as a linear program through auxiliary variables y (the VaR threshold) and z_s (the shortfall in scenario s below the threshold):

$$\max_{y,z} \quad \left(y - \frac{1}{1-\alpha} \sum_{s=1}^{N_S} \pi_s z_s \right) \tag{3.17a}$$

$$\text{s.t.} \quad z_s \geq y - J_s, \quad \forall s, \tag{3.17b}$$

$$z_s \geq 0, \quad \forall s, \tag{3.17c}$$

with π_s the probability of scenario s. Non-anticipatory constraints $u_{i,0} = u_{j,0}\ \forall i,j \in \mathcal{S}$ enforce that the first-stage decision is consistent across all scenarios.

The two-stage procedure (Figure 3.6) exploits the structural separation between the stochastic and deterministic problems. A scenario-based stochastic MPC is solved infrequently on a relaxed linear model to obtain the profit distribution and identify the α-scenario—the weighted average over scenarios in $\mathcal{S}_\alpha = \{s \in \mathcal{S} \mid J_s \leq \mathrm{VaR}_\alpha\}$,

$$\boldsymbol{w}_{s_\alpha} = \frac{\sum_{s \in \mathcal{S}_\alpha} \pi_s \boldsymbol{w}_s}{\sum_{s \in \mathcal{S}_\alpha} \pi_s}, \tag{3.18}$$

where $\boldsymbol{w}_s$ denotes the uncertain parameter realization under scenario s. The α-scenario remains valid for the entire forecast-update interval, allowing the deterministic MPC to be solved frequently on the full nonlinear model against this single risk-adjusted scenario, with the first action applied in receding-horizon fashion. Figure 3.5 contrasts this approach against purely stochastic and sequential two-stage alternatives, showing that parallelization of the two stages yields the shortest delay between measurement and applied control.

3.3.3 Software Architecture for Scalable Deployment

The MPC methodology described above resolves the algorithmic question of how to make risk-aware decisions efficiently for a single microgrid. Commercial viability of risk-aware EMS, however, depends on a separate and equally hard question: how to deliver

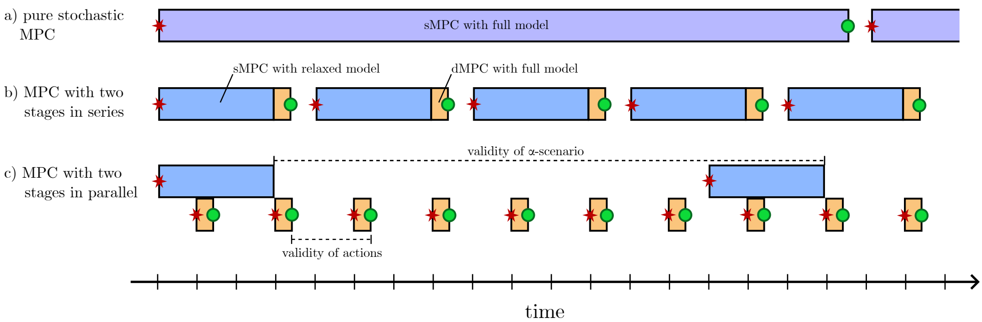

Figure 3.5: Comparison of MPC approaches. Blocks represent computation time, red stars denote measurement timestamps, and green circles denote application of optimal control actions. The proposed approach (c) achieves the shortest delay between measurement and applied control.

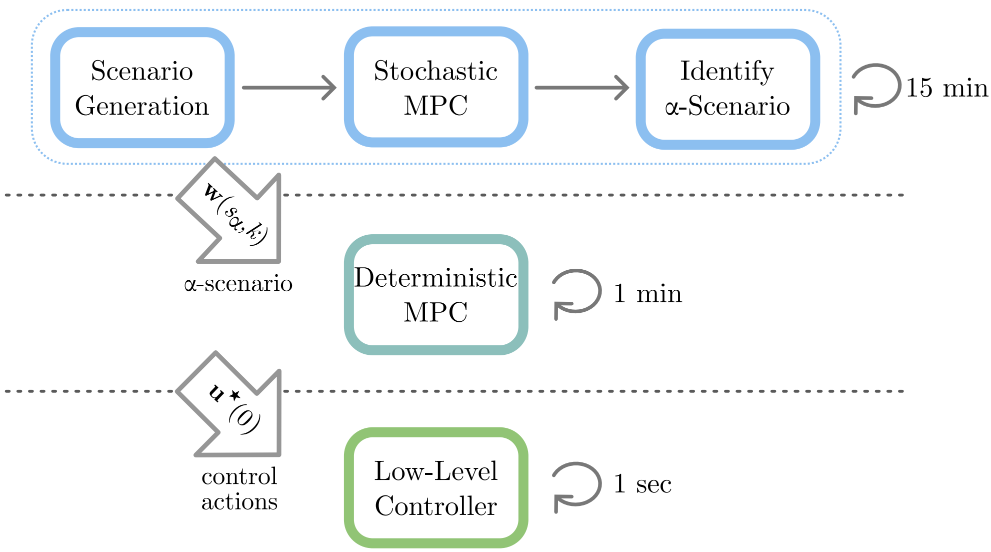

Figure 3.6: Major steps of the two-stage MPC procedure at different timescales, forming a hierarchical structure. Output from deterministic MPC are action commands valid for a fixed interval (e.g., 1 minute), serving as setpoints for low-level controllers.

that control capability simultaneously to many microgrids of varying composition without per-site infrastructure investments scaling linearly with the customer base. This subsection presents the software architecture developed to answer that question, structured as Microgrid Control as a Service (MaaS) and illustrated in Figure 3.7. The architecture is not a peripheral implementation detail; it is the mechanism through which the risk-aware methodology becomes deployable in practice.

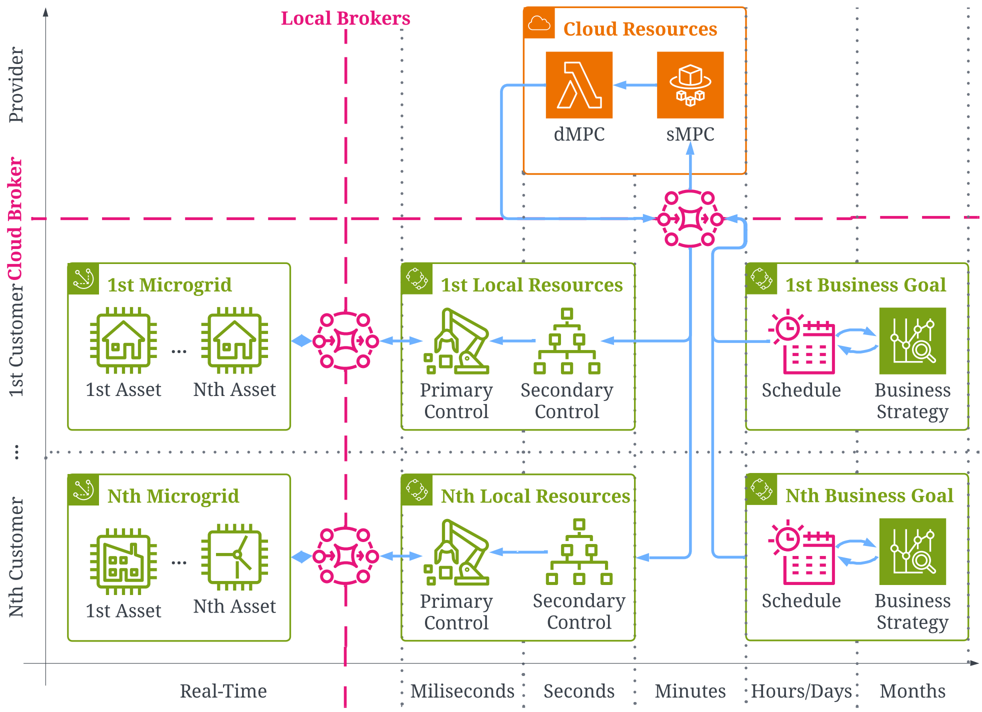

Figure 3.7: Scalable MaaS architecture. Individual customers operate their microgrids with varying numbers of assets using local control units, while the MaaS provider offers cloud-based computation to reduce capital expenditure and optimize operational costs based on customer business goals.

3.3.3.1 From Algorithmic Cost Structure to Architectural Split

Risk-aware MPC is only commercially feasible because the two-stage procedure exposes a structural asymmetry that the architecture can exploit. The stochastic stage requires substantial computational resources—tens of seconds per execution on a non-commercial solver—but is invoked infrequently and tolerates latency on the order of forecast-update intervals (15 minutes). The deterministic stage, by contrast, must complete within seconds and execute every minute to keep the controller's response aligned with the system imbalance settlement period. Were both stages deployed on the same compute substrate, the platform would face a fundamental dilemma: provisioning for the stochastic peak load wastes resources during the 14 minutes between its invocations, while provisioning for the deterministic stage's steady-state demand starves the stochastic stage of capacity when it runs. In either case, additional operational expenditure scales with peak resource demand rather than with average utilization, eroding—and in marginal markets erasing—the revenue gain that risk-aware control unlocks.

Splitting the two stages across distinct service models eliminates this dilemma. The infrequent, resource-intensive stochastic stage maps naturally onto an elastic batch service that bills per-second of execution and idles at zero cost. The frequent, short-lived deterministic stage maps onto a serverless event-driven service with millisecond-level billing and zero reservation overhead. Risk-aware MPC's higher computational footprint is thereby converted from a recurring cost burden into an intermittent workload absorbed by elastic cloud resources, preserving the revenue advantage that risk-awareness provides over deterministic-only operation.

3.3.3.2 Hybrid Cloud-Local Architecture

A second design axis separates control layers by data dependency. Power management and device control layers ($\Delta t \lesssim 1$ s) operate on real-time process data and are deployed on local industrial-grade controllers—PLCs or industrial PCs co-located with the microgrid assets. Local deployment guarantees bounded latency, maintains operational integrity during communication failures, and isolates safety-critical control from third-party faults or network attacks on the cloud side. Business and energy management layers ($\Delta t \in [1, 15]$ min) process historical and forecasted time-series data, and are deployed in the cloud where elastic resources and higher tolerance for latency are acceptable.

This hybrid split is not solely an economic optimization but a safety-relevant design choice. If the cloud-side stochastic stage fails—due to network partition, cloud-provider outage, or transient resource exhaustion—the local deterministic controller continues to execute against the last identified α-scenario, which remains valid until the next forecast update arrives. The controller degrades gracefully from risk-aware to deterministic operation at the most recent risk-adjusted operating point rather than failing open. Similarly, if the deterministic stage briefly exceeds its time budget, the local power-management layer continues to enforce setpoints from the previous cycle and operational limits remain protected. The architecture is therefore fault-tolerant by construction: each layer holds sufficient state to absorb the failure of the layer above.

3.3.3.3 Cost-Driven Cloud Service Selection

Cloud service models trade off provisioning granularity against per-unit cost. Function as a Service (FaaS) provides millisecond-level billing and zero idle charges but bills at a relatively high rate per unit of compute. Container as a Service (CaaS) bills per-second of execution at moderate per-unit cost and is well-suited to longer batch tasks. Infrastructure as a Service (IaaS) reserves whole virtual machines at low per-unit cost but bills continuously, including idle time. Table 3.2 summarises the feature differences relevant to deployment. Mapping the MPC tasks to these services according to their computational profile yields:

Table 3.2: Comparison of cloud compute service models: Function as a Service (FaaS), Container as a Service (CaaS), and Infrastructure as a Service (IaaS).

Feature	**FaaS**	**CaaS**	**IaaS**
Billing period	Millisecond	Second	Hour
Pay for idle time	×	×	✓
Price per unit	High	Medium	Low
Price per request	Low	Medium	High
Task duration fit	Short	Medium	Long
Flexible scaling	✓	✓	×
Startup time	Milliseconds	Seconds	Minutes

- **Deterministic MPC on FaaS.** Sub-second to few-second executions invoked at one-minute intervals (43 200 invocations per microgrid per month) achieve the lowest cost on FaaS, where billing terminates when the function returns and no idle resources are reserved between invocations.
- **Stochastic MPC on CaaS.** Executions of tens of seconds invoked every 15 minutes (2 880 invocations per microgrid per month) achieve the lowest cost on CaaS, where the linear scaling of cost with execution time avoids the per-unit premium of FaaS for non-trivial durations while still avoiding idle charges between invocations.

Figure 3.8 substantiates this mapping by plotting monthly OpEx against request duration for an MPC workload executed every minute over 30 days. FaaS dominates for sub-12-second executions; for durations between roughly 12 and 60 seconds, IaaS becomes the lowest-cost option, since additional instances are provisioned in discrete steps and short tasks fit within a small instance pool; beyond 60 seconds, CaaS scales most gently as execution time grows. The crossover points map directly onto the deterministic and stochastic execution profiles measured in deployment: dMPC at a few seconds falls firmly in the FaaS-cheapest regime, while sMPC at tens of seconds falls in the CaaS-cheapest regime.

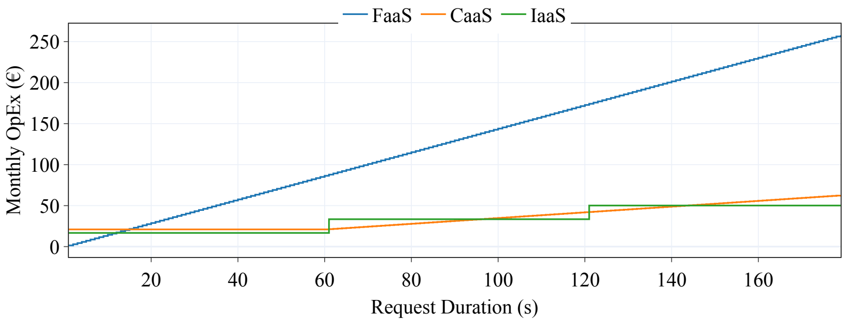

Figure 3.8: Monthly OpEx for MPC workloads of different request durations requiring 2048 MiB of memory and 0.5 vCPU. FaaS has the lowest OpEx for short requests but grows rapidly with duration. CaaS cost is flat up to 60 s before rising linearly. IaaS cost increases stepwise with each additional reserved instance. Pricing based on AWS rates as of March 2025.

Empirical benchmarks of the deployed framework confirm this mapping. Stochastic MPC executions on 50 scenarios reach maximum execution times of 98 s with maximum memory of 736 MB, while deterministic MPC executions complete within 3 s at 180 MB. A 30-day cost analysis indicates that the monthly OpEx contribution of cloud compute remains three orders of magnitude below the revenue generated for the operator. Scenario-count sensitivity for the stochastic stage (Figure 3.9) shows that beyond 25 scenarios CaaS becomes the cheapest substrate for sMPC, with FaaS cost growing exponentially in the number of scenarios; the chosen mapping therefore remains optimal as customers increase risk-mitigation depth.

The use of an open-source LP solver (GLPK) within this architecture is a further deliberate cost choice. Commercial solvers achieve shorter execution times but introduce per-installation licensing fees often amounting to tens of thousands of euros annually—fees that would dominate the cloud OpEx and undermine the economic argument for risk-aware MPC at all but the largest installations. By selecting an open-source solver and tuning the architecture to absorb its longer execution times within the time

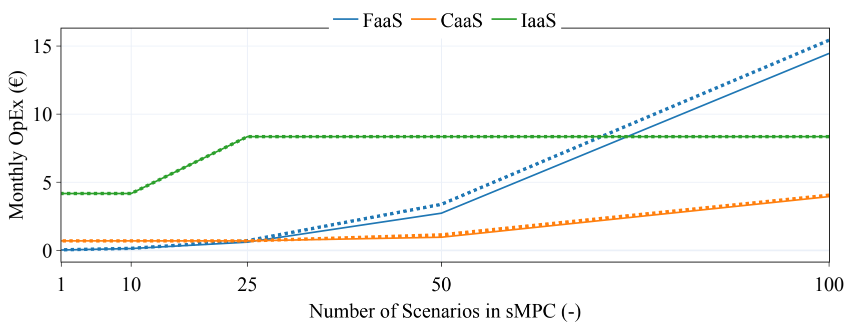

Figure 3.9: Sensitivity of sMPC monthly OpEx to the number of scenarios. CaaS exhibits the lowest sensitivity and becomes the cheapest substrate beyond 25 scenarios; FaaS cost grows exponentially as the number of scenarios increases. Solid lines represent average workload, dashed lines represent maximum workload.

budgets imposed by the two-stage split, the platform sustains commercially viable margins under profit-sharing agreements with microgrid operators.

3.3.3.4 Modularity Across Heterogeneous Microgrids

The decoupling of control layers across the hybrid architecture also enables modularity in the microgrid model itself. Each microgrid is described by parameterized asset abstractions—energy sources, storages, controllable loads, and the utility-grid interface—whose composition is read from a configuration manifest at instantiation time. The MPC stages, the scenario generator, and the cost decomposition operate over these abstractions without code changes when assets are added or removed. Heterogeneous installations differing in PV capacity, battery rating, presence of EV chargers or heat pumps, and the active subset of revenue streams therefore share the same control codebase and the same cloud deployment pipeline. New customers are onboarded by adding a configuration entry and provisioning local control units, not by developing per-site software. This modularity is what converts a single-site control algorithm into a multi-tenant service in which per-microgrid cloud OpEx remains far below the per-microgrid revenue contribution observed at single-site deployment.

3.3.4 Case Study: Commercial Microgrid Deployment

The framework has been deployed in commercial operation across multiple microgrid installations. A representative case study demonstrates results from a commercial building in the Netherlands equipped with a 220 kWp rooftop photovoltaic system, 558 kWh battery energy storage system rated at 180 kW, 200 kW electric vehicle chargers (controllable load), and general building demand (non-controllable load).

3.3.4.1 Risk Aversion Analysis

The impact of risk aversion parameter α was evaluated through closed-loop simulation with 100 Monte Carlo realizations of uncertain parameters. Results in Table 3.3 demonstrate the trade-off between risk and performance: conservative settings ($\alpha = 0\%$) yield the best worst-case outcomes, while aggressive settings ($\alpha = 100\%$) maximize profit under favorable conditions but expose the system to significant losses. The influence of α on scenario selection is illustrated in Figure 3.10.

Table 3.3: Total profit/loss per day from closed-loop simulation, 100 runs for each α level. Bold marks the best α for each percentile.

α (%)	**0**	**25**	**50**	**75**	**100**
Best case (€)	45	45	47	54	**55**
25% quartile (€)	22	23	24	**30**	6
Median (€)	8	10	11	**14**	−37
75% quartile (€)	−1	2	**3**	−40	−89
Worst case (€)	**-163**	−171	−204	−336	−337

For operational deployment, $\alpha = 30\%$ was selected as providing good trade-off—conservative enough to avoid high losses while aggressive enough to achieve reasonable profits.

3.3.4.2 Operational Performance

Over more than three months of commercial operation, the EMS demonstrated effective microgrid management. Figure 3.11 shows an example day comparing grid exchange with and without EMS control, while Figure 3.12 decomposes the corresponding revenue and cost streams. The most significant impact arose from imbalance settlement: optimal control transformed imbalance costs of −600 e into revenue of 2471 e over one month (Table 3.4). Peak-shaving costs were eliminated through proactive battery utilization to satisfy demand during periods when grid import limits would otherwise be exceeded. Figure 3.13 shows the consumption, production, and storage power flow over a month of operation.

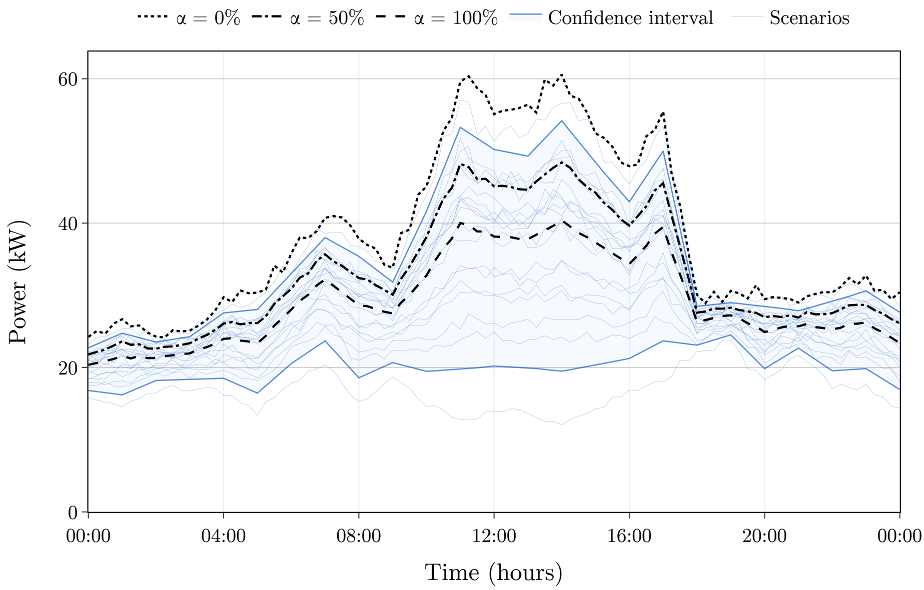

Figure 3.10: Selected scenario of consumption demand P_{cons} for different values of α. Scenarios are generated from the 90% forecasted confidence interval. Lower α selects scenarios closer to the worst case, while higher α approaches the median forecast.

Table 3.4: Comparison of operating revenues and costs over one month with and without EMS actions. Positive values denote revenues, negative values denote costs.

Revenue/cost type		**Without EMS**	**With EMS**
Imbalance	(€)	−600	2471
Day-ahead market	(€)	−326	−405
Battery usage	(€)	0	−210
Peak-shaving	(€)	−267	0
Total profit	(€)	**-1193**	**1856**

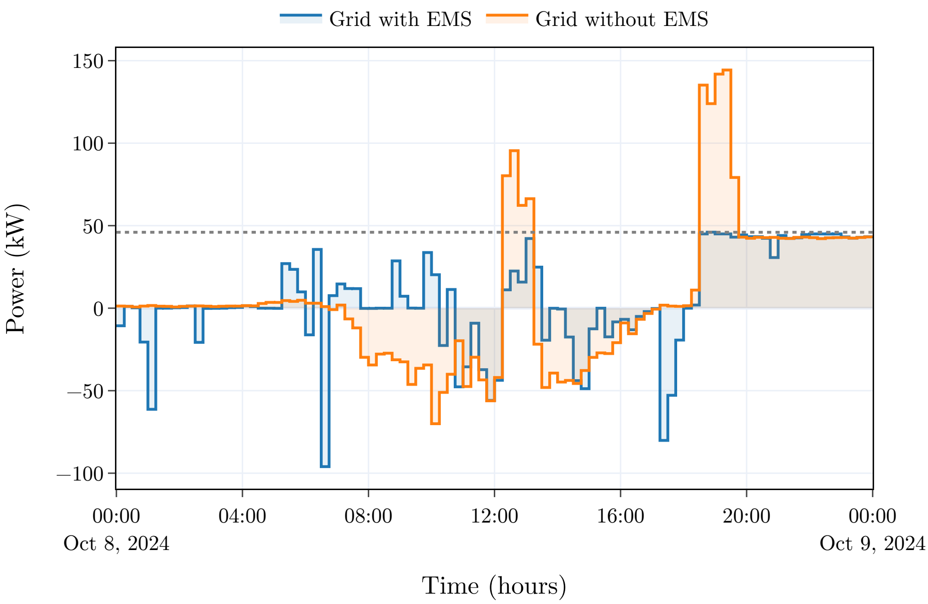

Figure 3.11: Example day of microgrid operation comparing power grid exchange with (realized) and without (estimated) EMS control. The dotted line represents the import limit.

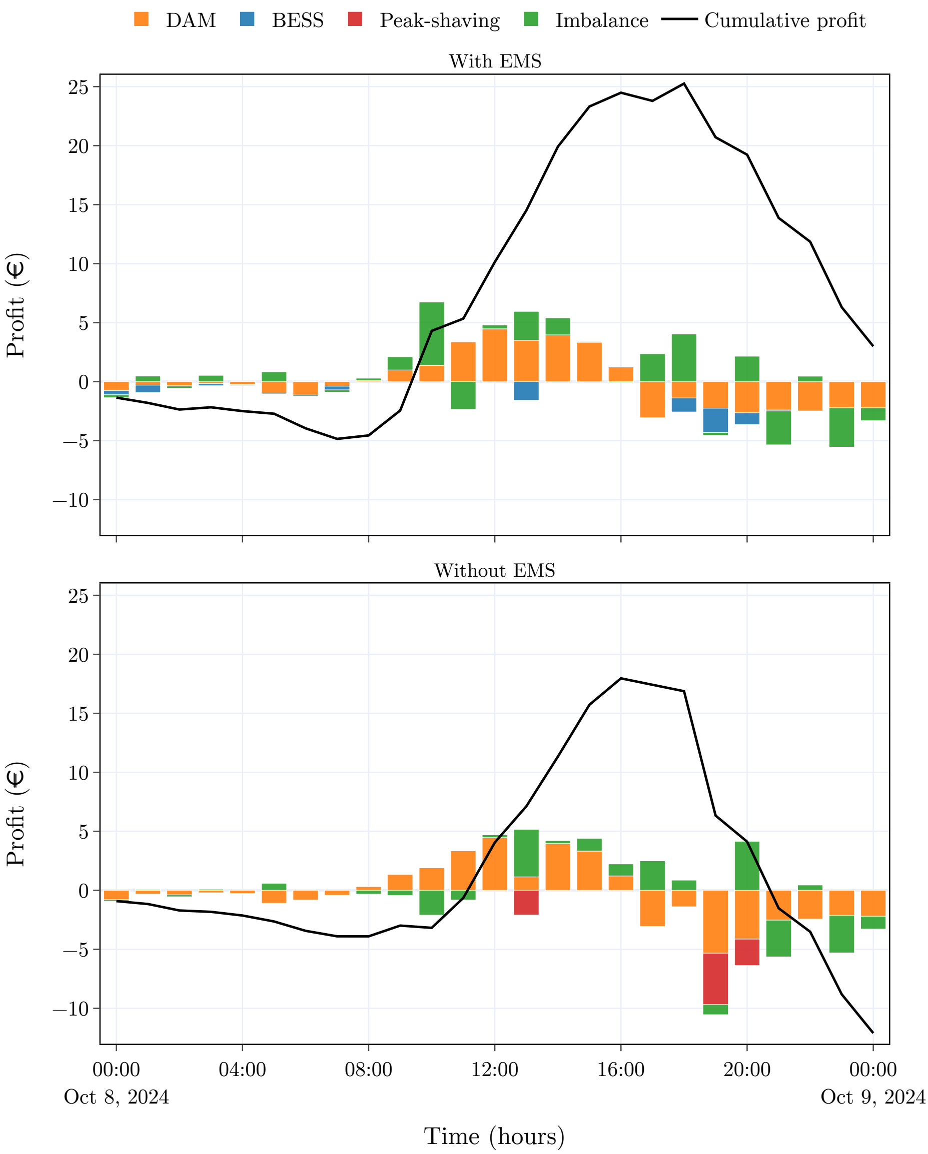

Figure 3.12: Revenue and cost decomposition of microgrid operation with and without EMS control actions. Total profit for the example day was 3.0 €, compared to −12.1 € without predictive control.

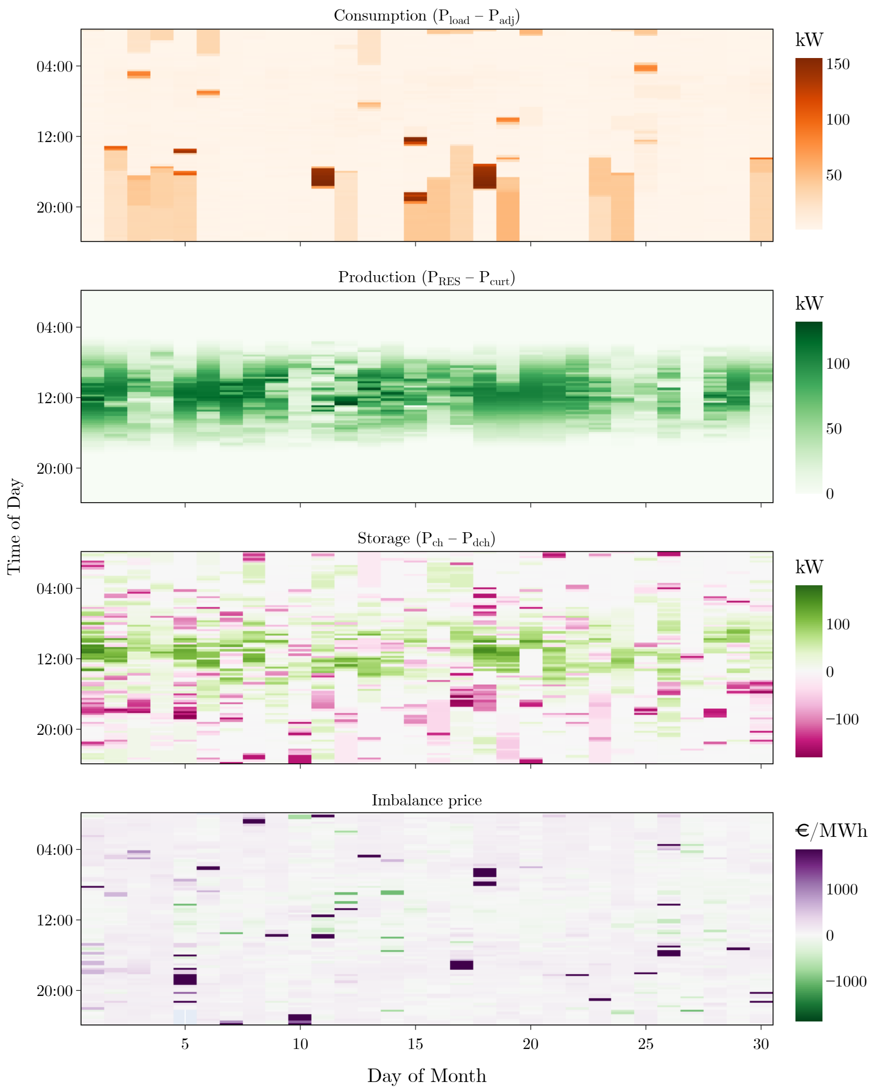

Figure 3.13: Microgrid self-production, consumption, and storage power flow over one month of operation. Imbalance price signal shown in the bottom subplot for reference. Visible patterns in storage power are observed in response to microgrid conditions and market signals.

Total yield of 159 e/kW per year (extrapolated from three months of operation) exceeds investor expectations for battery storage profitability [43]. The EMS achieves this through integrated optimization across multiple revenue streams—day-ahead market, imbalance settlement, and peak shaving—rather than optimizing any single stream in isolation. Benchmarking against comparable frameworks in the Belgian balancing market [83] confirms competitive performance, achieving 17 €/kW/month at average imbalance prices of 67–79 €/MWh.

3.3.4.3 Scalability Across Heterogeneous Installations

The same EMS codebase and cloud deployment pipeline have been instantiated across a diverse portfolio of commercial and industrial microgrids, summarized in Table 3.5. The installations span PV capacities from 150 kWp to 800 kWp, battery storage from 150 kWh to 558 kWh (including configurations with two storage units operated in parallel), and a varying mix of secondary assets including EV chargers and heat pumps. The active subset of revenue streams—price arbitrage, peak shaving, imbalance optimization, and self-sustainability maximization—is determined per installation from the customer's contractual exposure and the local market structure, and is selected through configuration rather than code modification.

The cost-economics of the deployment is preserved across this portfolio. Because cloud OpEx per microgrid is dominated by the deterministic-stage FaaS invocations (whose memory footprint depends weakly on asset count) and the stochastic-stage CaaS executions (whose memory footprint scales with the number of scenarios but is independent of the number of installations served), the marginal OpEx of onboarding an additional microgrid is small relative to the revenue share retained by the MaaS provider. The same three-orders-of-magnitude OpEx-to-revenue ratio measured at single-site deployment therefore applies across the heterogeneous portfolio. This property—rather than the performance of any single installation in isolation—is what establishes risk-aware EMS as a viable commercial offering.

3.4 The Limits of Control

The preceding sections demonstrate that sophisticated control frameworks can significantly improve economic and operational performance under uncertainty. Risk-aware MPC balances profit maximization with downside protection, enabling profitable operation across diverse market conditions. Yet even optimal controllers operate under assumptions that may be violated.

Table 3.5: Overview of microgrid installations operated under the proposed EMS, illustrating the heterogeneity of compositions and functionalities supported by a single codebase. The case-study installation analyzed in this section is listed first.

Facility	**Composition**	**Functionality**
Commercial building	PV 220 kWp; BESS 558 kWh; EV chargers 200 kW	Imbalance optimization, peak-shaving
Commercial building	PV 150 kWp; BESS 150 kWh; EV chargers 132 kW	Price arbitrage, peak-shaving
Industrial park	PV 600 kWp; BESS 2× 150 kWh; heat pumps	Imbalance optimization, peak-shaving
Industrial park	PV 280 kWp; BESS 150 kWh; heat pumps	Imbalance optimization
Industrial park	PV 350 kWp; BESS 150 kWh	Price arbitrage, self-sustainability
Industrial park	PV 500 kWp; BESS 2× 150 kWh	Price arbitrage, peak-shaving
Mountain resort	PV 800 kWp; BESS 558 kWh	Peak-shaving, self-sustainability

Control frameworks assume that system models remain valid, that constraints accurately reflect physical limits, and that forecasts provide useful predictions of future conditions. When these assumptions hold, MPC optimizes effectively. When they fail—due to equipment degradation, parameter drift, or unforeseen operating modes—optimal control may drive systems toward unsafe or unprofitable operation.

Consider a battery storage system whose capacity has degraded beyond the state-of-health estimate embedded in the control model. The MPC may schedule deeper discharge cycles than the actual battery can sustain, accelerating further degradation or triggering protection systems. Similarly, a renewable forecast model may become systematically biased after sensor drift, leading to persistent over- or under-estimation that the controller cannot correct without external observation.

These failure modes highlight a fundamental limitation: control systems optimize based on models and predictions, but cannot independently detect when models and predictions have become invalid. The controller lacks awareness that it is operating under assumptions that no longer reflect reality.

Addressing this limitation requires complementary monitoring capabilities—systems that observe actual behavior, detect deviations from expectations, and alert when control assumptions may no longer hold. When the monitoring layer detects significant changes in system dynamics, it can trigger model updates, alert operators, or modify control constraints to maintain safe operation.

This necessity motivates the monitoring approaches developed in Chapter 4. The monitoring layer does not replace control—it enables control to remain effective over extended deployments by detecting when adaptation is required. Together, risk-aware control and adaptive monitoring form an integrated framework for sustainable operation: control optimizes performance under current assumptions, while monitoring ensures those assumptions remain valid.

The question shifts from "how do we control optimally under uncertainty" to "how do we detect when our uncertainty models themselves have become invalid". With CVaR, we can control at a desired level of risk, but what about significant threats to the system that fall outside the scenarios we anticipated? Chapter 4 addresses these questions through anomaly detection, change-point detection, and adaptive monitoring frameworks that complete the lifecycle perspective introduced in Chapter 1.

Publication mapping.

The contributions of this chapter are documented in the following publications:

- Section 3.2 (greenhouse Economic MPC): [119].
- Section 3.3 (scalable deployment of risk-aware microgrid EMS): [117].

Chapter 4

Real-Time Monitoring and Adaptive Safety

The preceding chapter demonstrated that sophisticated control frameworks can significantly improve economic and operational performance under uncertainty. However, even optimal controllers operate under assumptions that may be violated as systems evolve, degrade, or encounter unforeseen operating conditions. This chapter addresses the complementary challenge of ensuring reliability through real-time monitoring—detecting when control assumptions no longer hold and providing the adaptive capability to maintain safe operation over extended deployments.

Section 4.1 establishes the conceptual bridge from control to monitoring, articulating why autonomous systems require self-monitoring capabilities. Section 4.2 presents a method for real-time outlier detection with dynamic process limits, enabling adaptive thresholding within existing SCADA infrastructure. Section 4.3 extends these concepts to multivariate systems through an Adaptable and Interpretable Detection framework (AID) that provides root cause isolation alongside anomaly detection. Section 4.4 develops change-point detection based on truncated online Dynamic Mode Decomposition with Control (toDMDc), detecting fundamental shifts in system dynamics that indicate when predictive models require updating. Together, these contributions provide a comprehensive monitoring layer that ensures reliability throughout the system lifecycle.

4.1 Ensuring Reliability in Evolving Systems

To close the lifecycle loop introduced in Chapter 1, systems must self-monitor to detect when the assumptions made during design and commissioning are no longer valid. Control systems optimize based on models and predictions, but cannot independently detect when models and predictions have become invalid. The controller lacks awareness that it is operating under assumptions that no longer reflect reality.

Consider the assumptions underlying the risk-aware MPC developed in Chapter 3. The controller assumes that system dynamics remain consistent with the model, that constraint bounds accurately reflect physical limits, and that uncertainty characterizations appropriately capture forecast variability. When these assumptions hold, MPC optimizes effectively. When they fail—due to equipment degradation, parameter drift, or unforeseen operating modes—optimal control may drive systems toward unsafe or unprofitable operation without any indication that something is wrong.

This fundamental limitation motivates dedicated monitoring capabilities that operate alongside control. Effective monitoring must address multiple scales of deviation from expected behavior:

- **Point anomalies:** Individual observations that deviate significantly from expected values, potentially indicating sensor faults, data corruption, or transient disturbances.
- **Collective anomalies:** Sequences of observations that together indicate abnormal behavior, even if individual points appear normal—characteristic of gradual degradation or developing faults.
- **Change points:** Fundamental shifts in system dynamics that persist indefinitely, indicating that the system has transitioned to a new operating regime requiring model updates.

This progression — from detecting anomalies through diagnosing faults to predicting remaining useful life — constitutes the domain of Prognostics and Health Management (PHM) [116]. The open challenges identified in that field directly motivate the monitoring methods developed in this chapter.

Industrial deployment imposes additional constraints on monitoring approaches. Methods must operate on streaming data with bounded memory and computational requirements. They must adapt to legitimate operational changes without human intervention while maintaining sensitivity to genuine anomalies. They must integrate with existing SCADA infrastructure rather than requiring wholesale replacement. And they must provide interpretable outputs that enable operators to understand and trust detection decisions.

The following sections develop monitoring methods that address these requirements, progressing from univariate dynamic thresholding through multivariate anomaly detection to dynamics-aware change-point detection.

4.2 Real-Time Outlier Detection with Dynamic Process Limits

Classical SCADA systems rely on static thresholds to detect when process variables exceed acceptable bounds. However, appropriate bounds depend on operating conditions, environmental factors, and equipment state—factors that static thresholds cannot accommodate. This section presents a method for computing dynamic process limits using an online outlier detection algorithm capable of handling concept drifts in real-time [93].

4.2.1 Problem Formulation

Traditional alerting mechanisms detect situations where signal values deviate from constraints. An alert is triggered when variables exceed upper or lower process limits. These limits are typically predefined and fixed based on equipment design specifications and historical operating data. However, factors such as aging, environmental changes, and operational context call for dynamic adjustment of these limits. Several concrete failure modes illustrate this fragility. Sensor drift causes the baseline of normal behavior to shift gradually over time, rendering fixed bounds either too permissive—masking developing faults—or too restrictive—triggering spurious alarms. Environmental variation means that a threshold valid for one installation may be inappropriate for another operating under different ambient conditions, or even for the same installation across seasons. Furthermore, measurement noise across nominally identical equipment produces variance in outputs that a single static bound cannot accommodate, resulting in false alarms on some units while genuine anomalies on others go undetected.

Setting appropriate limits for an ever-growing number of signal measurements is time-consuming. For signals where no prior information about correct process ranges exists—particularly those subject to external factors unknown at setup time—static limits prove inadequate. The challenge is to provide adaptive operation constraints that respond to changing conditions while maintaining compatibility with existing process automation infrastructure.

4.2.2 Inverse CDF-Based Dynamic Thresholding

The proposed approach computes dynamic process limits using an inverse cumulative distribution function (ICDF). Rather than determining whether an observation is anomalous given fixed thresholds, this method calculates the threshold values that correspond to a specified anomaly probability—effectively solving the inverse problem of finding what input would produce an alert-triggering observation.

The method assumes that data collected over a moving window follow a Gaussian distribution. This assumption enables efficient online computation while allowing the influence of trends to be mitigated through appropriate window size selection. The approach comprises four components: initialization, online training, online prediction, and dynamic limit computation.

4.2.2.1 Model Initialization

Initial conditions set the mean $\mu_0 = x_0$ and variance $s_0^2 = 1$. The score threshold q is constant, typically set to 3σ corresponding to 99.73% of the normal distribution. Two user-defined parameters govern behavior: the expiration period t_{e} defining the window for time-rolling computations, and the time constant t_{c} determining adaptation speed.

The expiration period can be adjusted to change the proportion of expected anomalies, enabling relaxation (longer period) or tightening (shorter period) of thresholds. The time constant influences which anomalous points contribute to model updates—only when the rolling average of anomaly scores over window $\boldsymbol{y} = \{y_{k-t_{\mathrm{c}}}, \ldots, y_k\}$ exceeds threshold q:

$$\frac{\sum_{y \in \boldsymbol{y}} y}{n(\boldsymbol{y})} > q, \tag{4.1}$$

where $n(\boldsymbol{y})$ represents the dimensionality of $\boldsymbol{y}$. This condition enables discrimination between isolated outliers and genuine change points.

4.2.2.2 Online Training

The model learns incrementally, updating the mean and variance of the underlying Gaussian distribution with each arriving sample. Computation of moving statistics employs Welford's algorithm for numerical stability:

$$\mu_k = \mu_{k-1} + \frac{x_k - \mu_{k-1}}{n}, \tag{4.2}$$

$$s_k^2 = s_{k-1}^2 + \frac{(x_k - \mu_{k-1})(x_k - \mu_k) - s_{k-1}^2}{n}. \tag{4.3}$$

After the expiration period, samples are forgotten through inverse Welford updates that revert their contribution. This bounded-memory approach enables indefinite online operation while maintaining responsiveness to recent conditions.

4.2.2.3 Online Prediction

In the prediction phase, the z-score is computed and transformed through the error function to evaluate the cumulative distribution function. Observations whose anomaly scores fall outside the threshold-bounded region are flagged:

$$y_k = 1 - 2 \cdot \mathcal{N}\left(x_k; \mu_k, s_k^2\right), \tag{4.4}$$

where $\mathcal{N}$ denotes the Gaussian CDF. The algorithm marks data points whose corresponding anomaly score indicates probability outside the 3σ region.

4.2.2.4 Dynamic Process Limits

Normal process operation is constrained online using the ICDF. The constant threshold q and current distribution parameters pass through the percent-point function to obtain the signal value that would trigger an alert at the current time instance:

$$x_q = \mathcal{N}^{-1}\left(q; \mu_k, s_k^2\right). \tag{4.5}$$

Upper and lower bounds are computed separately by applying the same procedure to the distribution fitted on positive and negative deviations, respectively. These dynamic limits can directly replace static thresholds in existing SCADA alarm systems.

4.2.3 Case Study: Battery Energy Storage System

The method was validated on temperature monitoring of a battery energy storage system (BESS) where tight temperature control is essential for optimal performance and maximum lifespan. The goal was to identify anomalous events, remove corrupted data, and provide adaptive process limits from a self-learning model.

Figure 4.1 illustrates one month of average battery cell temperature measurements. With expiration period set to 7 days and time constant to 5 hours, the model captured anomalous patterns from diverse sources: manipulation events, BESS relocation causing distribution change points, calibration-induced peaks, faulty measurements, packet loss, and temperature control tests.

The model's ability to detect anomalous behavior enabled effective dynamic thresholding. Real-valued threshold mechanisms provided up-to-date upper and lower bounds that adapted quickly to the change point on March 7th, when the BESS was relocated. The system mitigated effects of anomalies on the signal distribution while the user-defined parameters governed adaptation speed and limit tightness.

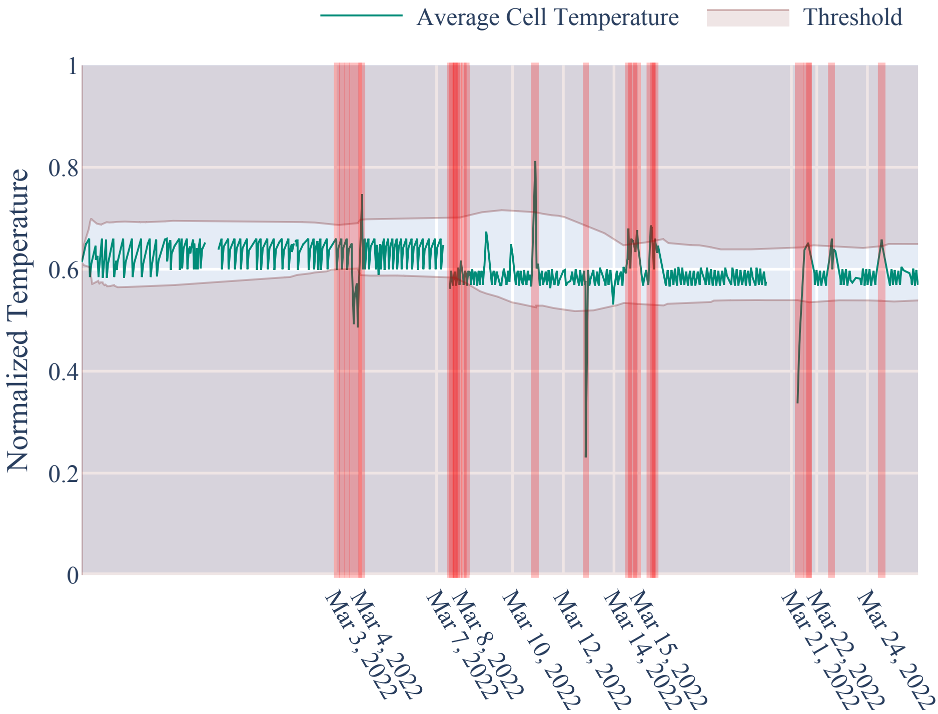

Figure 4.1: Average battery cell temperature measurements for one month. The model captured anomalous patterns from diverse sources: manipulation events, BESS relocation causing distribution change points, calibration-induced peaks, faulty measurements, packet loss, and temperature control tests.

4.2.4 Case Study: Power Inverter Temperature

A second case study demonstrated applicability to inverter temperature monitoring, where high-load periods cause rapid heating. Technical documentation provides static limits based on continuous output ratings, but aging and ambient conditions may render conservative limits impractical.

Figure 4.2 depicts one month of inverter operation. The dynamic threshold mechanism tracked normal operation while tightening limits as oscillations vanished. After packet loss events, the detector deliberately adapted the range of normal operation. Comparison with Half-Space Trees and One-class SVM demonstrated improved adaptation to changing conditions while retaining low false positive rates.

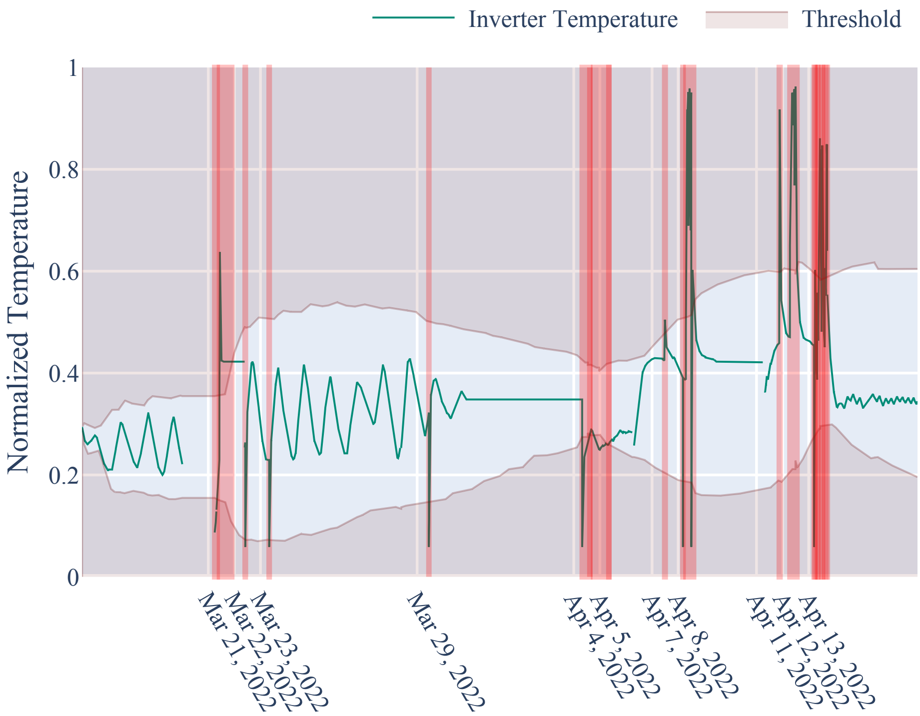

Figure 4.2: Inverter temperature measurement (green line) and predicted anomalous events (red vertical rectangles) over one month. The reddish area bounded by the red line represents the region of anomalous behavior as determined by the dynamic process limits.

4.3 Adaptive Anomaly Detection for SCADA Systems

While dynamic process limits address univariate monitoring, industrial systems typically involve multiple interrelated measurements. The Adaptable and Interpretable Detection framework (AID) extends the previous concepts to multivariate systems, providing root cause isolation alongside anomaly detection through conditional probability analysis [94].

4.3.1 Framework Overview

AID models the system as a dynamic multivariate normal distribution, enabling effective handling of non-stationary effects and interactions between process variables. The framework addresses change points, concept drift, and seasonal effects through integration of adaptable self-supervised learning with interpretable diagnostics.

The approach provides three complementary capabilities:

1. **Conditional probability assessment:** Evaluating normality of operating conditions by computing conditional distributions that consider relationships between signals.
2. **Change point detection:** Identifying abrupt changes due to concept drift, enabling faster adaptation to new operating conditions without human intervention.
3. **Dynamic operating limits:** Establishing adaptive limits that integrate seamlessly with existing SCADA infrastructure.

4.3.2 Online Detection

In the online detection phase, AID distinguishes between normal and anomalous observations based on the model of the system's normal behavior. The detection pipeline is event-triggered upon the arrival of a new set of measurements.

For each element of the process observation vector $\boldsymbol{x}_k$ at time step k, AID computes the conditional mean and conditional variance given the current dynamic joint normal distribution [36]. These calculations yield univariate conditional distributions for individual signals and features. For the j-th element of $\boldsymbol{x}_k$, the conditional mean is obtained from the joint distribution parameters as $\mu_{j|rest} = \mu_j + \boldsymbol{\Sigma}_{j,\neg j}\boldsymbol{\Sigma}^{-1}_{\neg j,\neg j}(\boldsymbol{x}_{\neg j} -$

$\boldsymbol{\mu}_{\neg j}$), where $\neg j$ denotes all elements except j, and the conditional variance is $\sigma^2_{j|rest} = \Sigma_{jj} - \boldsymbol{\Sigma}_{j,\neg j}\boldsymbol{\Sigma}^{-1}_{\neg j,\neg j}\boldsymbol{\Sigma}_{\neg j,j}$. These conditional distributions assess the abnormality of each signal with respect to its relationships with other elements, so AID inherently considers interactions between input signals.

The determination of anomalous behavior is governed by the parameter T, a user-defined probabilistic threshold that sets the boundary between normal and anomalous behavior. Whenever an anomaly is detected within any signal or feature, it triggers an alert regarding the overall system's anomalous behavior. Individual signal anomaly determinations serve additionally as a diagnostic tool for root cause isolation (see Section 4.3.5).

The detection mechanism applies to both point anomalies and collective anomalies. Collective anomalies, by their sustained duration and deviation, may serve as precursors to concept drift. To identify concept drift, AID introduces the adaptation period t_a. Given the predicted system anomaly state y_k over a window of past observations $\boldsymbol{y}_k = \{y_{k-t_\mathrm{a}}, \ldots, y_k\}$ bounded by t_a, the following test determines anticipated change points:

$$\frac{\sum_{y \in \boldsymbol{y}_k} y}{n(\boldsymbol{y}_k)} > T, \tag{4.6}$$

where $n(\boldsymbol{y}_k)$ denotes the dimensionality of $\boldsymbol{y}_k$. Over the adaptation period, change points can be distinguished from collective anomalies and point anomalies due to their minimum duration, while T allows for some overlap with previous normal conditions.

4.3.3 Online Learning

AID follows an incremental self-learning approach, enabling online model updates as new samples arrive. Self-learning selects only relevant data for training to maintain the model's long-term relevancy and stability—a property particularly valuable for streaming data where human supervision introduces unacceptable computational delays.

A grace period t_g enables model calibration in the initial stages after deployment. During this period, when normality in samples is expected, the model learns from all observations. Subsequently, the detector makes autonomous decisions through two self-supervised mechanisms.

First, the model is updated when the observation at time step k is marked normal in the detection phase. In the dynamic multivariate setting, the updated parameters are the mean vector $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$, governed by Welford's online algorithm.

Samples beyond the expiration period t_e are disregarded during a second pass, with their effect reverted using inverse Welford's algorithm on the bounded internal buffer.

Second, an adaptation mechanism enables learning from anomalous samples when a change point is detected. This mechanism operates under the assumption that detected change points represent new operational states with limited overlap with previous ones, as specified by (4.6). It facilitates rapid adaptation to evolving data patterns without human intervention. The selection of t_a is thus crucial for determining adaptation speed.

4.3.4 Dynamic Limits Acquisition

The threshold T applied to the dynamic multivariate normal distribution creates a confidence hyperellipse at the T probability level. Such a hyperellipse cannot effectively bound individual signals, as it depends on values that other jointly distributed variables take. However, by computing the conditional distribution for each element of the process observation vector $\boldsymbol{x}_k$, the conditional density function for individual signals can be obtained. Applying threshold T on individual conditional probabilities establishes a hypercube defined by lower and upper threshold values, denoted as $\boldsymbol{x}_\mathrm{l}$ and $\boldsymbol{x}_\mathrm{u}$, derived from the inverse CDF of the conditional distributions with updated model parameters:

$$\boldsymbol{x}_{\mathrm{l},k} = \mathcal{N}^{-1}\left(\tfrac{1-T}{2};\, \mu_{j|\mathrm{rest}},\, \sigma^2_{j|\mathrm{rest}}\right), \tag{4.7}$$

$$\boldsymbol{x}_{\mathrm{u},k} = \mathcal{N}^{-1}\left(\tfrac{1+T}{2};\, \mu_{j|\mathrm{rest}},\, \sigma^2_{j|\mathrm{rest}}\right). \tag{4.8}$$

These dynamic operating limits may supplement static limits used by SCADA monitoring systems, accounting for spatial factors such as multipoint measurements, temporal factors such as aging, and actual environmental conditions influencing sensor operation. Any violation of the limits is also detected as an anomaly.

4.3.5 Diagnostics

Root cause isolation is a crucial aspect of diagnostics. Using its ability to detect anomalies in individual signals and features, AID isolates the root cause of anomalies while considering their mutual relationships. This is achieved by computing the conditional probability of individual signals and features given the rest of the process observation vector $\boldsymbol{x}_k$. The dynamic process limits further enhance diagnosis by providing context: the extent of deviation from normal operation and the direction of deviation. AID's interpretability enables domain experts to understand why certain anomalies are flagged and allows operators to assess the system's state by visualizing limits and deviations, detecting the speed at which a process variable approaches its limits before an anomaly occurs.

4.3.6 Algorithm Summary

Algorithm 1 summarizes the AID workflow. Upon receiving a new observation, the algorithm performs detection via conditional probability assessment, computes dynamic operating limits, checks sampling regularity, and updates the model either when the observation is normal or when a change point is detected. Expired samples are reverted to maintain bounded memory.

Algorithm 1 Online Detection and Identification Workflow (AID)

Input: expiration period t_e
Output: system anomaly y_i, signal anomalies $\boldsymbol{y}_{\mathrm{s},k}$, sampling anomaly $y_{\mathrm{t},k}$, change-point $y_{\mathrm{c},k}$, lower thresholds $\boldsymbol{x}_{\mathrm{l},k}$, upper thresholds $\boldsymbol{x}_{\mathrm{u},k}$

Initialization: $k \leftarrow 1$; $n \leftarrow 1$; $T \leftarrow 3\sigma$; $\boldsymbol{\mu} \leftarrow \boldsymbol{x}_0$; $\boldsymbol{\Sigma} \leftarrow \boldsymbol{I}_{d\times d}$
loop
  $\boldsymbol{x}_k, t_k \leftarrow$ RECEIVE ()
  $\boldsymbol{y}_{\mathrm{s},k} \leftarrow$ PREDICT $(\boldsymbol{x}_k, T)$ {Conditional anomaly per signal}
  $y_i \leftarrow$ PREDICT $(\boldsymbol{y}_{\mathrm{s},k})$ {System-level anomaly}
  $\boldsymbol{x}_{\mathrm{l},k},\ \boldsymbol{x}_{\mathrm{u},k} \leftarrow$ GET $(T, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ {Dynamic limits via (4.7), (4.8)}
  $y_{\mathrm{t},k} \leftarrow$ PREDICT $(t_k - t_{k-1})$ {Sampling anomaly}
  if $(y_i = 0)$ **or** (4.6) **then**
    $\boldsymbol{\mu}, \boldsymbol{\Sigma} \leftarrow$ UPDATE $(\boldsymbol{x}_k, \boldsymbol{\mu}, \boldsymbol{\Sigma}, n)$ {Welford's algorithm}
    $y_{\mathrm{c},k} \leftarrow \mathbf{1}[(4.6)]$
    $n \leftarrow n + 1$
    for $\boldsymbol{x}_{k-t_\mathrm{e}}$ **do**
      $\boldsymbol{\mu}, \boldsymbol{\Sigma} \leftarrow$ REVERT $(\boldsymbol{x}_{k-t_\mathrm{e}}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, n)$ {Inverse Welford's}
      $n \leftarrow n - 1$
    end for
  end if
  $k \leftarrow k + 1$
end loop

4.3.7 Model Initialization

The model initialization requires two hyperparameters: the expiration period t_e and the threshold T. The expiration period determines the window size for time-rolling computations, directly influencing relaxation (longer t_e) or tightening (shorter t_e) of dynamic signal limits. The grace period t_g, which defaults to t_e, allows model calibration during which system anomalies are not flagged. The grace period can take any value smaller than t_e when fast detection after integration is required.

The adaptation period t_a manages detection of significant drifts in the data-generating process. A shorter t_a results in faster adaptation to new operating conditions while making the system vulnerable to prolonged collective anomalies. A longer t_a slows adaptation but allows longer alerts regarding collective anomalies. In most cases, $t_a = \frac{1}{4}t_e$ offers optimal performance.

As a general rule, t_e should be determined based on the slowest observed dynamics within the multivariate system. The threshold T defaults to the three-sigma probability (99.73%). Adjusting this threshold tunes the trade-off between precision and recall. The presence of one non-default interpretable hyperparameter facilitates quick adaptation of AID across a broad range of use cases.

4.3.8 Case Study: Industrial Battery Energy Storage

AID was validated on the TERRA industrial-scale battery energy storage system, depicted in Figure 4.3, with 151 kWh capacity distributed among 10 modules with 20 Li-ion NMC cells [94]. The system reports temperature measurements at 6 positions of each module, with tight control required for safety and lifespan optimization.

Figure 4.3: The studied TERRA energy storage unit with open doors (left) and closed doors (right). The system has an installed capacity of 151 kWh distributed among 10 modules.

When TERRA was relocated from inside a building to an outside power socket, the system encountered a significant change in operating conditions. Without change point adaptation (Figure 4.4), the model could not capture new behavior as measurements remained outside safe limits learned from indoor operation. The system flagged anomalies for approximately three days until the expiration period enabled gradual adaptation.

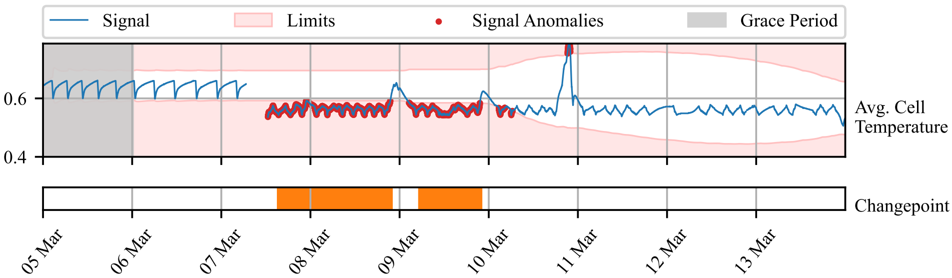

Figure 4.4: Detection of anomalies using model **without** adaptation to change points in normalized average cell temperature of TERRA observed over nine days. The model alerts anomalies (red dots) for approximately three days. The dynamic operating limits (light red area) remain stagnant during the detected change point period (orange bars), triggered t_a hours after anomaly onset.

With change point adaptation enabled (Figure 4.5), the mechanism detected the change within hours and accelerated adaptation by allowing learning from anomalous data representing the changed behavior. Adaptation completed approximately six times faster, demonstrating the value of self-supervised change point handling for long-term deployment.

Observing multiple variables simultaneously, where some may be influenced less by the change, further enhances detection. Figure 4.6 shows the multivariate results combining average cell temperature with a physics-based model of battery temperature, which captures an outlier on 9th March that could not be detected in the univariate setting.

4.3.9 Case Study: Hardware Fault Detection

A second case study validated AID's ability to detect hardware faults in production operation. A cooling fan fault occurred on module 9 of the TERRA system on 23rd August 2023 during commercial deployment. Each module contains 20 cells measured

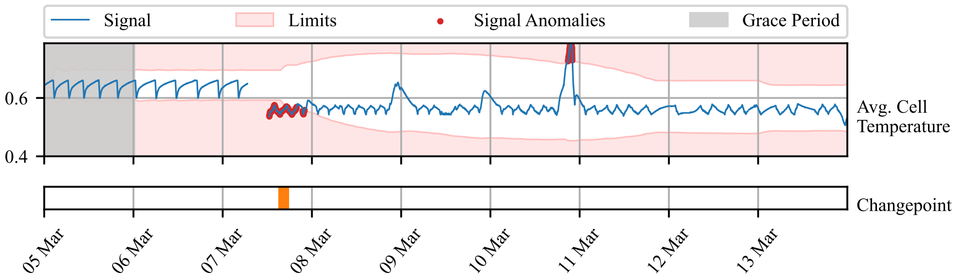

Figure 4.5: Detection of anomalies using model **with** adaptation to change points over the same measurement as in Figure 4.4. The change point triggers adaptation to changed behavior more than two days sooner, reducing the anomaly alert period from approximately three days to ten hours.

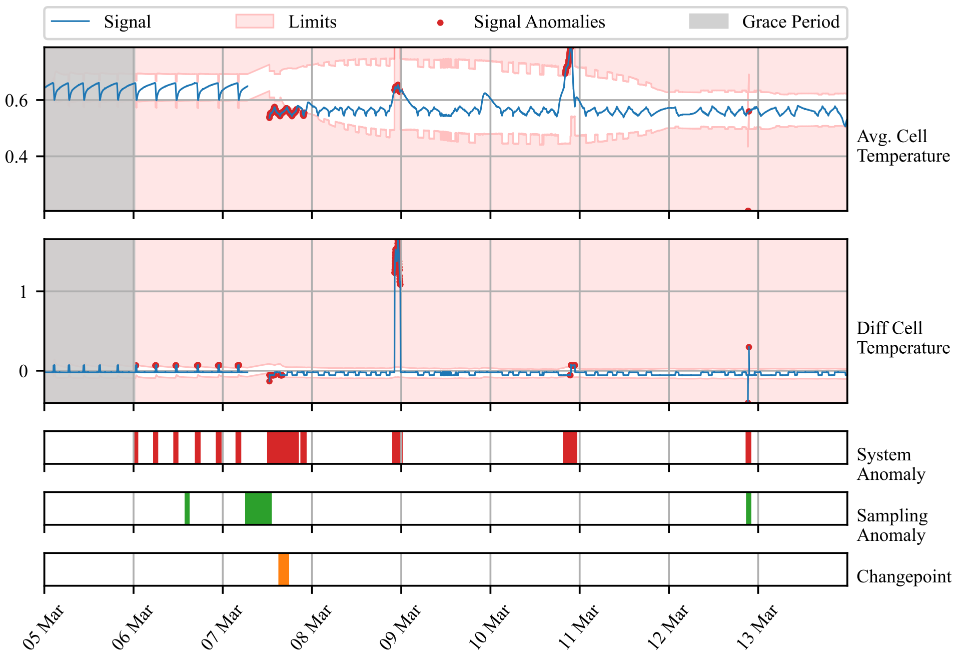

Figure 4.6: Multivariate TERRA measurements over 9 days showing average temperature (blue line) with dynamic operating limits (light red area). Orange bars denote detected change points, green bars indicate sampling anomalies, and red bars mark periods of signal anomalies. The grace period is grayed out.

by 6 spatially distributed sensors (Figure 4.7). Using temperature deviations from the average across all modules, AID successfully identified the fault at the moment of occurrence.

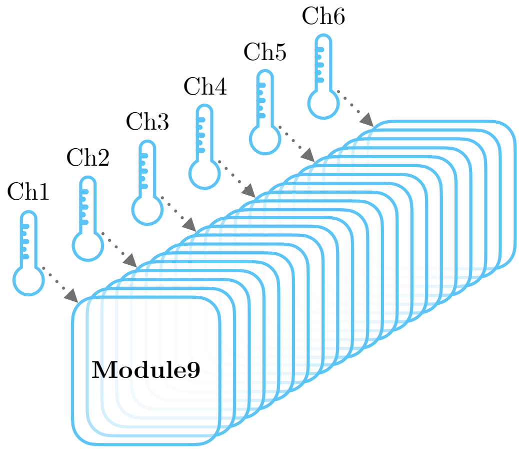

Figure 4.7: Module 9 with 20 cells and 6 sensors measuring the temperature at every 4th cell. Spatially distributed measurements enable root cause localization of thermal anomalies.

Multiple channels independently flagged the fault, providing root cause localization. As shown in Figure 4.8, channels 1, 2, and 3 detected the fault through sharp increases in temperature deviation. The framework detected not only the initial fault occurrence but subsequent intermittent operation periods as the fan alternated between faulty and operational states. This case demonstrates practical utility for enhancing safety of energy storage systems through proactive fault detection.

4.3.10 Benchmark Comparison

Benchmarking against adaptive unsupervised methods—One-Class SVM (OC-SVM) and Half-Space Trees (HS-Trees) [68]—on the Skoltech Anomaly Benchmark (SKAB) dataset [46] demonstrated competitive performance. Table 4.1 summarizes the results for models optimized using the F1 score. AID achieved highest precision (41%), recall (80%), and F1 score (54%) among evaluated methods, with significantly lower false positive rate despite a 30-fold higher computational latency per sample (1.45 ms versus 0.05 ms for HS-Trees and OC-SVM)—an absolute cost that remains well within

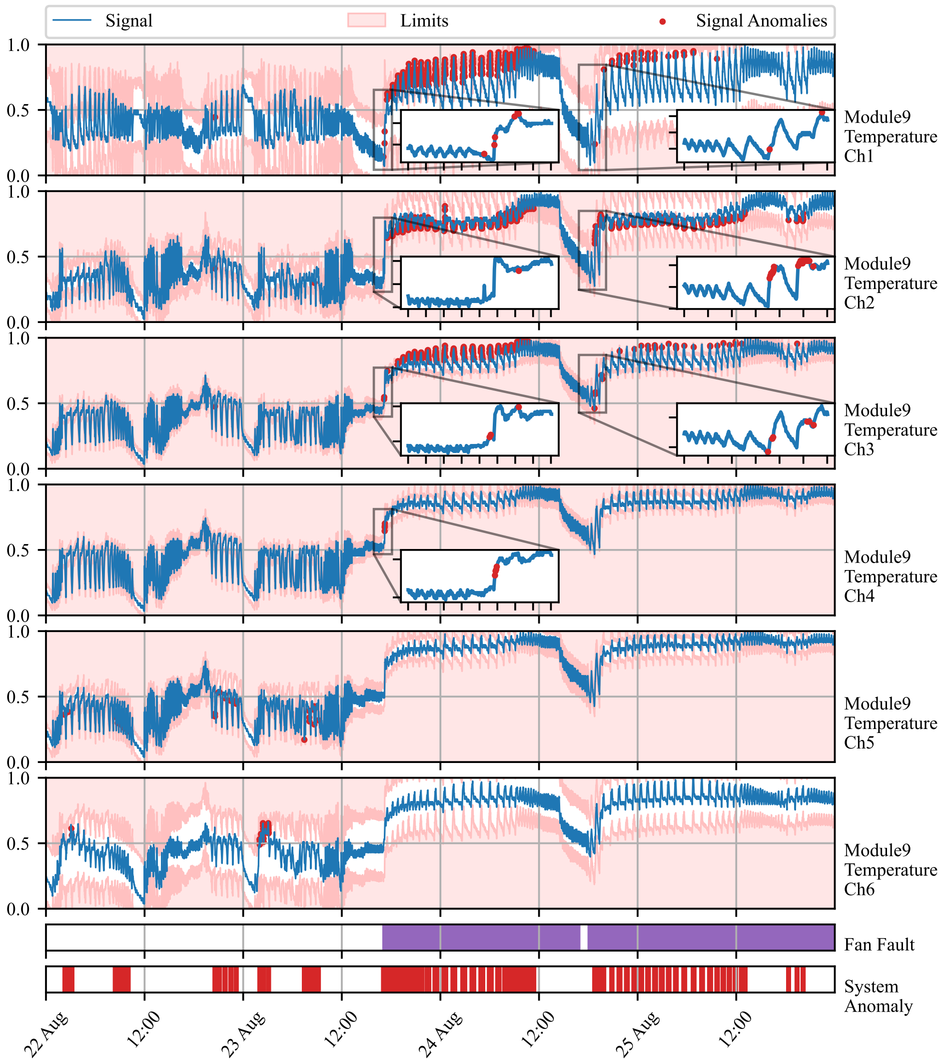

Figure 4.8: Temperature deviations of module 9 channels from the average across all modules over 4 days around the fan fault. Signal anomalies (red dots) and dynamic operating limits (light red area) are shown. Purple bars mark true fan fault periods. The zoomed views reveal sharp temperature increases detected by channels 1, 2, and 3.

real-time monitoring requirements. While AID's 47% false positive rate is the lowest among evaluated methods, it remains high for direct deployment as a standalone alarm trigger. This is a common trade-off in adaptive unsupervised methods, where distinguishing brief collective anomalies from normal transients is inherently difficult. In practice, the significantly higher precision and recall mitigate this limitation by enabling reliable alarm prioritization—operators can trust that flagged events are more likely genuine, even if some normal segments are also flagged.

Table 4.1: Evaluation of models optimized for F1 score on SKAB dataset [46]. Best values in bold; macro values in parentheses.

Metric	**AID**	**HS-Trees**	**OC-SVM**
Precision [%]	**41** (59)	36 (51)	39 (54)
Recall [%]	**80** (59)	74 (51)	63 (54)
F1 [%]	**54** (53)	48 (44)	48 (52)
AUC [%]	**59**	51	54
FPR [%]	**47**	56	48
Avg. Latency [ms]	1.45	**0.05**	**0.05**

Scalability analysis on TERRA temperature data (up to 60 features) confirmed applicability to multivariate industrial systems. Table 4.2 summarizes the latency statistics across varying numbers of features. Mean latency remained below one second for systems with up to 60 features at 30-second sampling intervals, suitable for typical industrial monitoring requirements.

4.4 Detecting System Changes in Real-Time

While AID detects anomalies relative to learned normal behavior, change-point detection (CPD) addresses a more fundamental question: when have the underlying system dynamics changed in ways that invalidate predictive models? This section develops CPD based on truncated online Dynamic Mode Decomposition with Control (toDMDc) [123], detecting shifts in dynamics that indicate when control models require updating.

Table 4.2: Latency analysis of AID with varying number of features.

Features	**Detection** $\mu \pm \sigma$ [ms]	**Detection + Limits** $\mu \pm \sigma$ [ms]
1	0.37 ± 0.26	0.63 ± 0.38
10	2.25 ± 0.92	5.24 ± 0.98
20	5.46 ± 2.16	14.7 ± 2.27
30	10.9 ± 4.31	34.3 ± 4.50
40	20.7 ± 8.15	69.5 ± 8.57
50	97.3 ± 47.4	297 ± 59.4
60	142 ± 71.2	468 ± 111

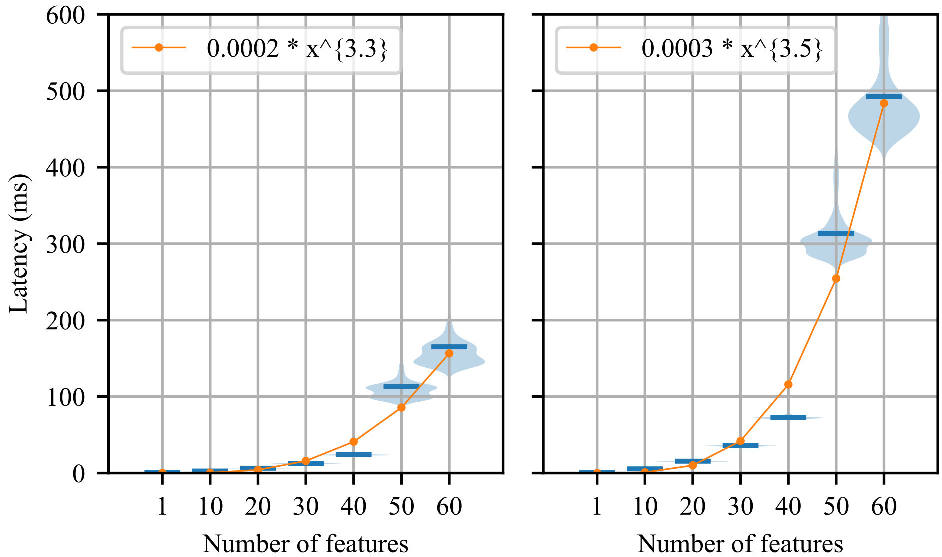

Figure 4.9: Latency distribution of AID for varying numbers of features. Violin plots depict the distribution, while horizontal bars show mean latency. The significantly smaller minimum latency corresponds to evaluation during the grace period.

4.4.1 Motivation and Approach

Many industrial systems are safety-critical, where monitoring abrupt and gradual changes is crucial for ensuring reliability. Traditional monitoring methods like Statistical Process Control rely on independent and identically distributed assumptions often violated in industrial data characterized by autocorrelation and non-stationarity. Static SCADA thresholds cannot adapt to dynamic changes from aging or environmental shifts.

Change points may be characterized by shifts in dynamics more effectively captured in the frequency domain than the time domain. A change in oscillation frequency or damping ratio indicates fundamental alteration of system behavior even if mean values remain constant. Subspace-based methods address this by finding low-dimensional descriptions of data and monitoring whether incoming observations align with the learned subspace [69].

The key insight is that projecting data onto modes of a low-rank subspace increases reconstruction error when non-conforming patterns appear. Transient dynamics cannot be adequately captured by the stationary subspace, making reconstruction error a natural detection statistic. Long-term deployment requires sequential updates to the subspace in streaming fashion, enabling adaptation to slow changes while detecting abrupt transitions.

4.4.2 Mathematical Framework

A change point occurs when incoming data cannot be adequately reconstructed by the current subspace, indicating a shift in underlying dynamics. The reconstruction error for data matrix $\bar{\boldsymbol{X}}_k$ projected onto modes $\boldsymbol{\Phi}_k$ is:

$$\mathcal{E}(\bar{\boldsymbol{X}}_k, \boldsymbol{\Phi}_k) = \|\bar{\boldsymbol{X}}_k - \boldsymbol{\Phi}_k \boldsymbol{\Phi}_k^\top \bar{\boldsymbol{X}}_k\|_F^2. \tag{4.9}$$

Comparing reconstruction errors between a reference (base) window $\bar{\boldsymbol{X}}_\mathrm{B}$ and a test window $\bar{\boldsymbol{X}}_\mathrm{T}$ yields the detection statistic:

$$Q_k = \max\left(0, \frac{\mathcal{E}(\bar{\boldsymbol{X}}_\mathrm{T}, \boldsymbol{\Phi}_k)/|\bar{\boldsymbol{X}}_\mathrm{T}|}{\mathcal{E}(\bar{\boldsymbol{X}}_\mathrm{B}, \boldsymbol{\Phi}_k)/|\bar{\boldsymbol{X}}_\mathrm{B}|} - 1\right), \tag{4.10}$$

where $|\cdot|$ denotes the number of snapshots. Under the null hypothesis (no change), both windows have similar reconstruction errors yielding $Q_k \approx 0$. Under the alternative (change present), the test window has significantly higher error yielding $Q_k > 0$. The magnitude provides an interpretable measure of change severity.

4.4.3 Algorithm Structure

The CPD-DMD algorithm comprises three fundamental steps executed for each arriving batch of measurements, illustrated in Figure 4.10:

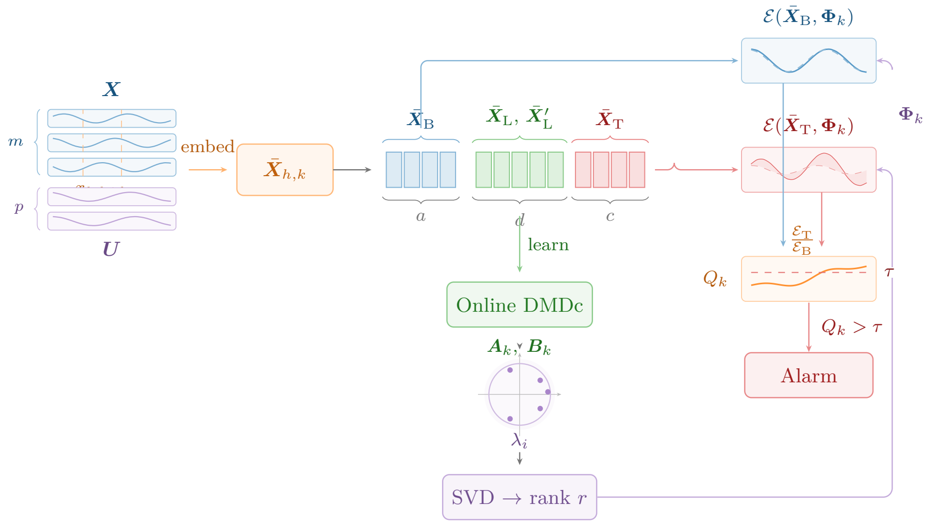

Figure 4.10: CPD-DMD algorithm pipeline. Incoming state and control streams are embedded into augmented snapshots, then partitioned into base (blue), learning (green), and test (red) windows. The learning path updates the DMDc operator and extracts modes $\boldsymbol{\Phi}_k$ via SVD truncation. The detection path projects base and test windows onto the DMD subspace, comparing reconstruction errors to produce the detection statistic Q_k. An alarm is raised when Q_k exceeds threshold τ.

4.4.3.1 Data Stream Management

Incoming snapshots are formed into time-delayed embeddings of predefined lag h to capture temporal dynamics. For systems with control inputs, snapshots are either compensated by known control effects or augmented with control action embeddings:

$$\bar{\boldsymbol{X}}_{h,k} = \begin{bmatrix} \boldsymbol{X}_{h,k} \\ \boldsymbol{\Theta}_{h,k} \end{bmatrix}, \tag{4.11}$$

where $\boldsymbol{X}_{h,k}$ contains state embeddings and $\boldsymbol{\Theta}_{h,k}$ contains control embeddings.

Four parameters a, b, c, d define required snapshot sets: base set $\bar{\boldsymbol{X}}_{\mathrm{B}}$, test set $\bar{\boldsymbol{X}}_{\mathrm{T}}$, and learning pair $\{\bar{\boldsymbol{X}}_{\mathrm{L}}, \bar{\boldsymbol{X}}'_{\mathrm{L}}\}$. The base set provides reference behavior, the test set contains recent observations for comparison, and the learning pair enables subspace updates.

4.4.3.2 Detection Procedure

Detection executes before learning to avoid information leaks from updating the subspace with potentially anomalous data:

1. Project base and test matrices onto DMD subspace: $\tilde{\bar{\boldsymbol{X}}}_{\mathrm{B}} = \boldsymbol{\Phi}^\top \bar{\boldsymbol{X}}_{\mathrm{B}}$, $\tilde{\bar{\boldsymbol{X}}}_{\mathrm{T}} = \boldsymbol{\Phi}^\top \bar{\boldsymbol{X}}_{\mathrm{T}}$
2. Reconstruct full state and compute squared Euclidean distances
3. Normalize by snapshot counts
4. Compute detection statistic as error ratio

When the error ratio falls below 1, the reconstructed test set captures more information than the reconstructed base set—indicating decreased noise variance or end of transient states. This phenomenon is truncated to zero as the minimum matching error.

4.4.3.3 Learning Procedure

The learning procedure updates the DMD model with new snapshots and forgets old ones to track time-varying characteristics. Online DMD maintains computational efficiency through rank-one updates rather than batch recomputation:

$$\boldsymbol{P}_k = \boldsymbol{P}_{k-1} - \frac{\boldsymbol{P}_{k-1}\bar{\boldsymbol{x}}_k\bar{\boldsymbol{x}}_k^\top \boldsymbol{P}_{k-1}}{1 + \bar{\boldsymbol{x}}_k^\top \boldsymbol{P}_{k-1}\bar{\boldsymbol{x}}_k}, \tag{4.12}$$

$$\boldsymbol{A}_k = \boldsymbol{A}_{k-1} + (\bar{\boldsymbol{x}}'_k - \boldsymbol{A}_{k-1}\bar{\boldsymbol{x}}_k)\bar{\boldsymbol{x}}_k^\top \boldsymbol{P}_k, \tag{4.13}$$

where $\boldsymbol{P}_k$ is the precision matrix and $\boldsymbol{A}_k$ approximates the DMD operator. The same update with negative signs enables forgetting old snapshots, maintaining bounded memory.

4.4.4 Parameter Selection

Hyperparameters have intuitive interpretations based on system knowledge:

- **Learning window** d**:** Should capture representative operating conditions including typical cycles and variations
- **Time delays** h**:** Should capture system dynamics—set based on known time constants

- **Base and test windows a, c:** Control detection delay and sensitivity—larger windows reduce noise but increase delay
- **Rank r:** Controls subspace dimensionality—should capture dominant dynamics while rejecting noise

Detection delay equals the test window size c, providing predictable and interpretable timing. The peak of change-point statistics occurs exactly c snapshots after the actual change, enabling precise change point localization.

4.4.5 Experimental Validation

4.4.5.1 Synthetic Step Changes

Controlled experiments validated detection behavior on synthetic data with known change points. The dataset comprised 10,000 samples with nine abrupt step changes at intervals of 1,000 samples, each of increasing magnitude $\Delta_i = 0.5i$ for $i = 1, \ldots, 9$, corrupted by Gaussian noise with $\sigma = 1$. Hyperparameters were set to $r = 2$, $a = 100$, $c = 100$, $d = 300$, and $h = 80$. The proposed method accurately detected all significant changes with detection delay matching the theoretical c snapshots.

Comparison with online SVD-based methods [47] demonstrated lower noise in detection statistics, aiding recognition of subtle changes (Figure 4.11). The CPD-DMD statistics directly relate to data dissimilarity—absolute difference reflects change magnitude while ratio reflects relative significance—providing interpretable outputs that deep learning methods lack [21].

4.4.5.2 Nonlinear Two-Tank System

Validation on a simulated nonlinear two-tank system with input delay demonstrated applicability to controlled systems. The system was simulated at 0.1 Hz sampling frequency for 12,000 snapshots. Random valve steps up to 0.03 m^3/s were applied every 200 snapshots, with input delay of 20–30 snapshots. States were corrupted by Gaussian noise with variance 0.35. Three types of changes were introduced: an artificial sensor bias (unit step applied to tank level observations during snapshots 4,000–5,000), a doubled valve response simulating a control valve offset (snapshots 7,600–8,600), and a linear trend added to system states (snapshots 9,800–12,000) to simulate increasing offset from equipment degradation.

Hyperparameters were selected based on system knowledge: learning window $d = 2{,}000$

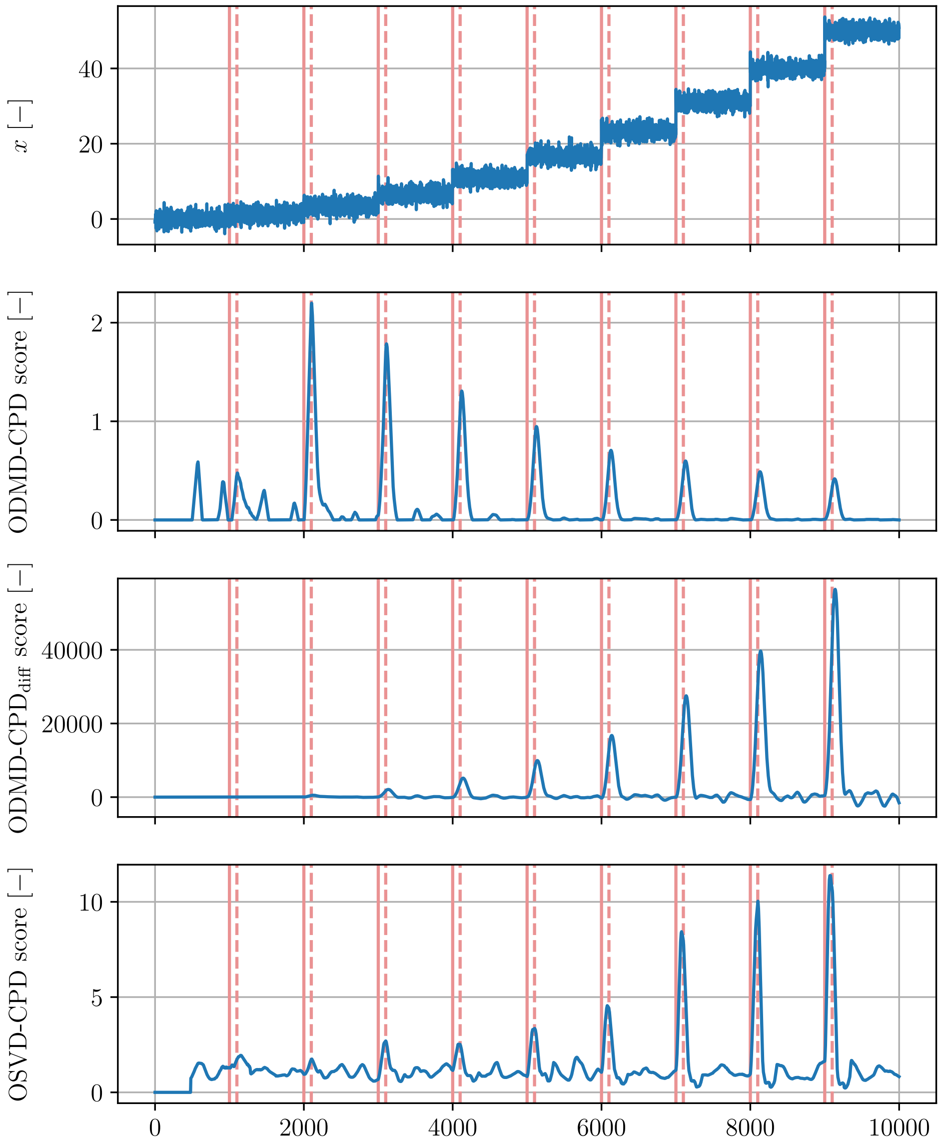

Figure 4.11: Step change detection in synthetic data (1). Change scores for the proposed CPD-DMD method using error ratio (2) and error difference (3), compared with an online SVD-based reference method (4). CPD-DMD detects minor changes missed by the reference method.

snapshots to capture responses across multiple control actions, time-delay embeddings $h = 200$ for state dynamics with $h_{\mathrm{d}} = 30$ to reduce dimensionality, control action embeddings of 30 snapshots to capture delayed effects, polynomial degree 2 for nonlinear dynamics, and $q = 2$, $p = 1$ for noise mitigation. The proposed method detected all change points including subtle linear trends that higher-noise SVD-based methods struggled to distinguish (Figure 4.12). Time-delayed embeddings captured the delayed response to control inputs without requiring explicit delay identification.

4.4.5.3 BESS HVAC Fault Detection

Real-world validation on the TERRA battery energy storage system detected transition from normal to faulty HVAC operation when a cooling fan failed. The method accurately identified the fault onset with detection delay matching the configured window size.

Multiple peaks preceding the confirmed fault aligned with anomalies detected by the AID framework in the same deployment [94], suggesting these may represent fault precursors valuable for predictive maintenance. The error difference formulation detected both transitions to faulty operation (positive peaks) and recovery periods (negative peaks), providing richer diagnostic information (Figure 4.13).

4.4.5.4 Benchmark Comparison

Evaluation on the SKAB [46] and CATS [30] benchmark datasets compared performance against nine reference methods including deep learning approaches (Conv-AE, LSTM-AE, MSCRED) and statistical methods (T-squared, PCA, Isolation Forest), using NAB metrics [2]. Figure 4.14 illustrates the evaluation protocol: the scoring window is placed after the change point to avoid crediting early detections that would constitute false positives.

On SKAB (Table 4.3), CPD-DMD achieved the highest standard NAB score (34.29) and lowest false negative score (42.54), outperforming all reference methods. On CATS (Table 4.4), comprising 5 million snapshots with 200 anomalies, CPD-DMD achieved the second-best standard score while providing significantly better false positive performance and completing evaluation within one hour compared to 24 hours for the top-performing MSCRED method. Notably, MSCRED's standard score (37.19) exceeds the perfect detector reference (30.21), which is a red flag: a score above the perfect detector suggests the method fires before faults occur, whether due to random alerts or persistent alarms from prior anomalies that were not cleared, rather than genuine early detection.

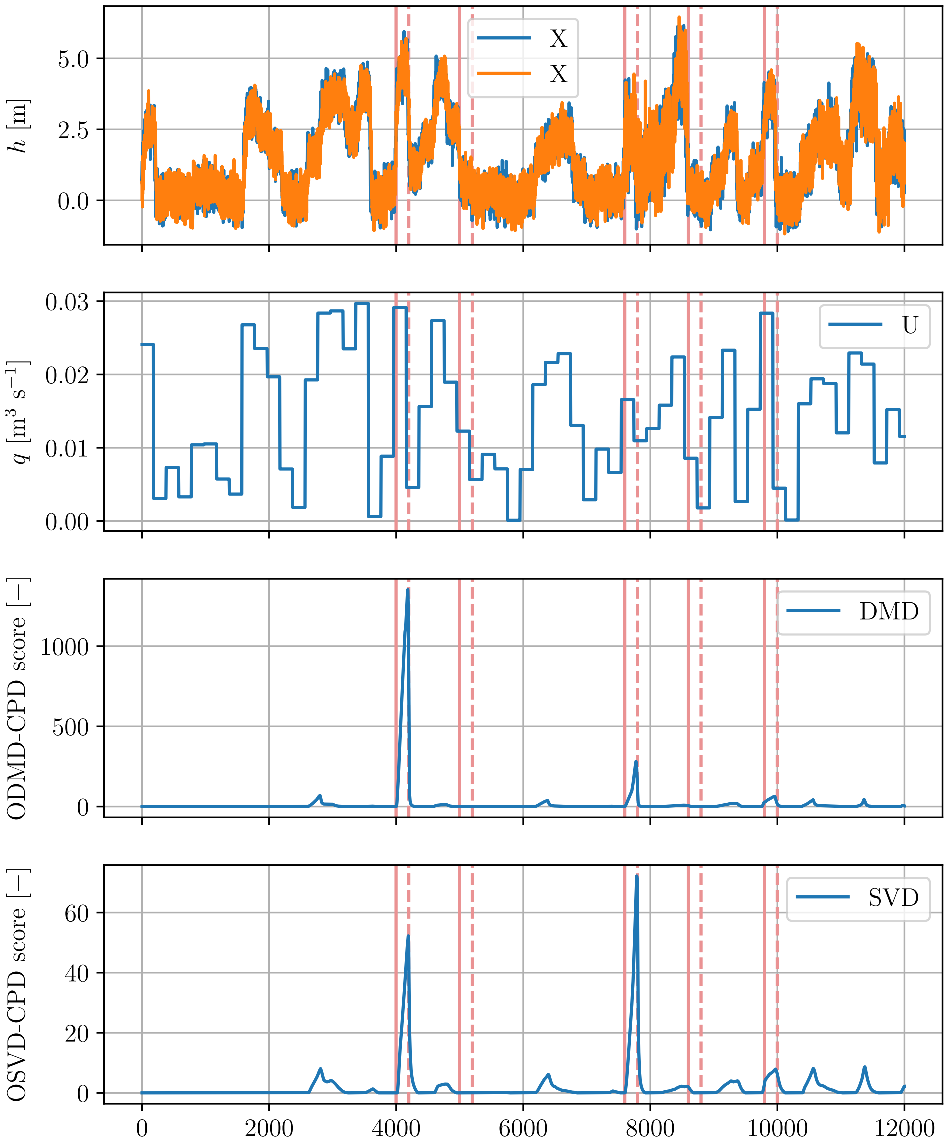

Figure 4.12: Change-point detection on the two-tank system (1) with delayed input (2). DMD-CPD score (3) exhibits lower noise than the SVD-based reference method (4), aiding recognition of the slow drift from snapshot 9800.

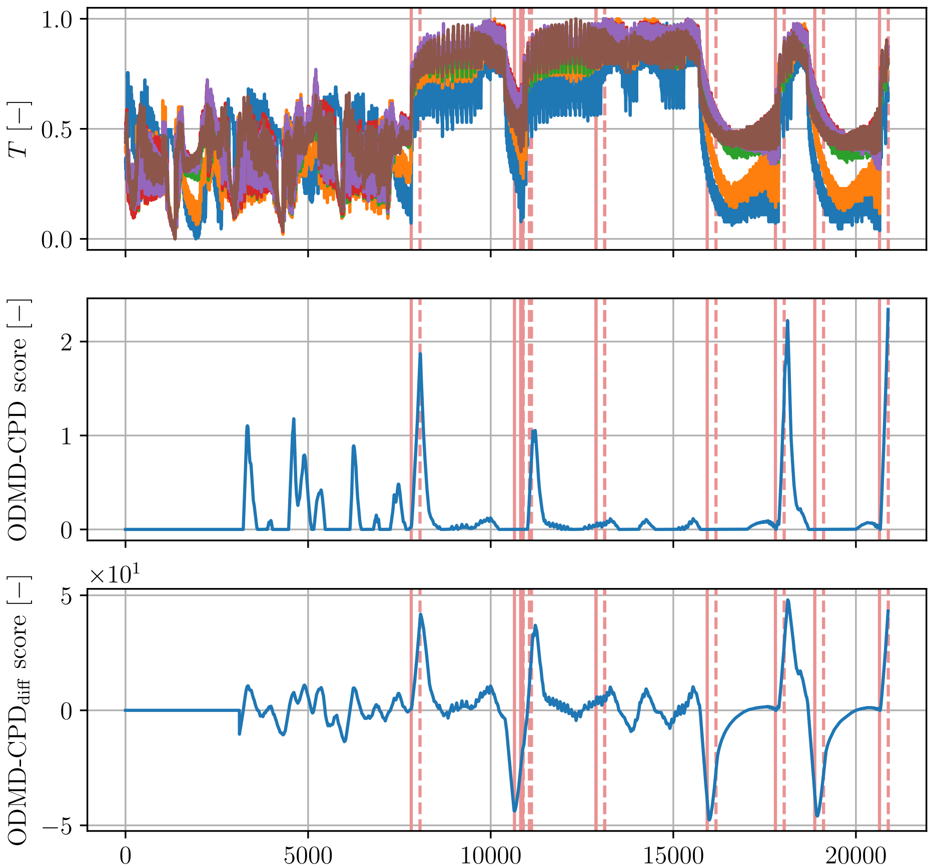

Figure 4.13: Faulty HVAC operation in BESS results in altered temperature (1). The proposed method displays multiple peaks in the CPD score (2). The error difference formulation (3) increases prominence of transitions toward the novel state and assigns negative peaks to transitions back toward the original state.

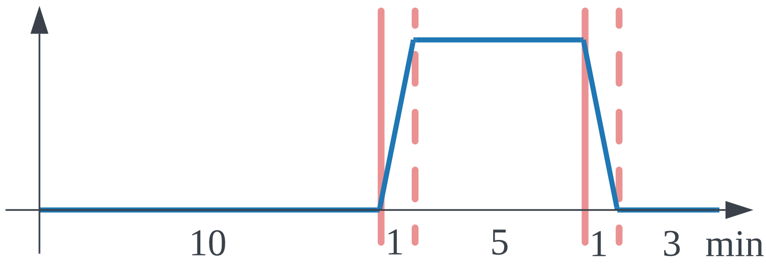

Figure 4.14: SKAB evaluation protocol. After 10 minutes of normal behavior, the system transitions to a new operating point (solid red line) over 1 minute (dashed red line), maintains it for 5 minutes, then returns. The scoring window starts at the transition onset to penalize premature detections.

Table 4.3: Comparison on SKAB dataset [46] using NAB metrics. Best scores in bold.

Algorithm	**NAB (std)**	**NAB (low FP)**	**NAB (low FN)**
Perfect detector	54.77	54.11	56.99
CPD-DMD ($t = 0$)	**34.29**	23.21	**42.54**
CPD-DMD ($t = 0.0025$)	33.43	**23.28**	41.71
MSCRED	32.42	16.53	40.28
Isolation forest	26.16	19.50	30.82
T-squared+Q (PCA)	25.35	14.51	31.33
Conv-AE	23.61	21.54	27.55
LSTM-AE	23.51	20.11	25.91
T-squared	19.54	10.20	24.31
Null detector	0.00	0.00	0.00

Table 4.4: Comparison on CATS dataset [30] using NAB metrics. Best scores in bold.

Algorithm	**NAB (std)**	**NAB (low FP)**	**NAB (low FN)**
Perfect detector	30.21	29.89	31.28
MSCRED	**37.19**	13.46	**47.18**
CPD-DMD ($t = 0$)	25.66	**20.62**	29.84
Isolation forest	17.81	15.84	20.00
T-squared	15.15	14.98	15.71
T-squared+Q (PCA)	11.80	11.40	12.30
LSTM-AE	11.39	11.26	11.69
Null detector	0.00	0.00	0.00

4.4.6 Integration with Control Systems

The change-point detection framework integrates naturally with the control systems developed in Chapter 3. When CPD detects fundamental shifts in system dynamics, it can trigger several responses:

- **Model updates:** Initiate reidentification of system dynamics for control model refresh
- **Constraint modification:** Adjust operating limits to reflect changed system capabilities
- **Operator alerts:** Notify human operators for situations requiring manual assessment
- **Control mode switching:** Transition to conservative fallback control until system behavior is understood

This integration completes the lifecycle perspective: control optimizes performance based on current models while monitoring ensures those models remain valid. When monitoring detects significant changes, it enables appropriate adaptation rather than continued operation under invalid assumptions.

4.5 Summary and Integration

This chapter developed three complementary monitoring approaches addressing different aspects of system reliability:

Dynamic Process Limits provide adaptive thresholding for univariate signals, enabling SCADA systems to adjust alarm bounds based on current operating conditions. The ICDF-based approach requires minimal configuration (two interpretable parameters) while providing seamless integration with existing alarm infrastructure.

AID Framework extends to multivariate systems, providing root cause isolation through conditional probability analysis. Self-supervised adaptation enables long-term deployment without manual retraining, while dynamic limits account for spatial correlations between measurements. The framework successfully detected hardware faults in production battery storage systems.

CPD-DMD addresses fundamental shifts in system dynamics using online DMD subspace tracking. The reconstruction error statistic provides interpretable change severity measures, while streaming updates enable adaptation to evolving systems. Benchmark validation demonstrated competitive performance with computational efficiency suitable for real-time deployment.

Together, these methods provide a comprehensive monitoring layer that ensures control systems remain reliable as conditions evolve. The monitoring approaches detect when control assumptions become invalid, enabling appropriate responses—model updates, constraint adjustments, or operator alerts—rather than continued optimization under incorrect assumptions.

The interpretability of all three methods represents a deliberate design choice. Industrial operators must understand and trust monitoring decisions to take appropriate action. Statistical foundations (Gaussian distributions, reconstruction error) provide clear explanations for detection decisions, while dynamic limits visualization enables operators to assess system state before anomalies occur.

Chapter 5 develops the modeling foundations that support both control and monitoring: interpretable data-driven models that capture system dynamics while maintaining transparency required for safety-critical applications.

Publication mapping.

The contributions of this chapter are documented in the following publications:

- Section 4.2 (ICDF-based dynamic process limits): [121].
- Section 4.3 (AID framework): [122].
- Section 4.4 (CPD-DMD / toDMDc): [123].

Chapter 5

Interpretable Data-Driven Modeling

The preceding chapters have established frameworks for risk-aware control and real-time monitoring that enable safe and economically optimal operation of industrial systems. However, these frameworks fundamentally depend on accurate system models that capture both the underlying dynamics and the physical principles governing system behavior. Advanced control strategies such as Model Predictive Control require fast, linear models for real-time optimization, yet industrial systems are inherently nonlinear. Monitoring systems require understanding of the underlying physics and dynamics, not just raw data patterns. Pure black-box models lack the safety guarantees and interpretability needed for industrial deployment, where operators must understand and trust the system's behavior.

This chapter addresses the challenge of developing interpretable, physically consistent models that bridge the gap between data-driven learning and physical understanding. Section 5.1 establishes the conceptual foundation, explaining why linearizable and physically consistent models are essential for both control and monitoring applications. Section 5.2 presents an LSTM-enhanced Deep Koopman approach for learning linear representations of nonlinear systems with unknown dynamics and input delays, without requiring pre-defined dictionaries. Section 5.3 develops a physics-informed Dynamic Mode Decomposition with Control (piDMDc) framework for predictive maintenance applications, demonstrating how physical constraints can be embedded into data-driven models to ensure consistency with known principles. Finally, Section 5.4 reflects on how validated, physically consistent models serve as the foundation for the predictive controllers developed in earlier chapters.

5.1 The Need for Linearizable and Physically Consistent Models

The transition from theoretical control frameworks to practical industrial deployment requires confronting a fundamental modeling challenge: advanced control algorithms demand fast, linear models for real-time optimization, yet industrial systems exhibit rich nonlinear dynamics. Simultaneously, monitoring systems must provide interpretable insights that align with physical understanding, enabling operators to diagnose issues and trust automated decisions. These dual requirements—computational efficiency through linearization and physical interpretability through consistency—motivate the development of data-driven modeling approaches that bridge the gap between black-box learning and first-principles understanding.

Model Predictive Control frameworks, as developed in Chapter 3, solve optimization problems at each time step to determine optimal control actions. The computational complexity of these optimizations scales with the complexity of the underlying system model. Nonlinear models require nonlinear optimization solvers, which may fail to converge within real-time constraints or may converge to suboptimal solutions. Linear models, in contrast, enable quadratic programming formulations that can be solved efficiently and reliably, making real-time control feasible even for large-scale systems. However, linearization around a fixed operating point often fails to capture the full operational envelope of industrial systems, which must operate across varying conditions.

The Koopman operator theory provides a mathematical framework for addressing this challenge. Rather than approximating nonlinear dynamics with local linearizations, the Koopman approach seeks to find a linear representation of the system in an embedded—or lifted—state space. If such a representation exists, the original nonlinear dynamics can be captured exactly by a linear operator acting on the lifted states. This linear representation enables the use of efficient linear control algorithms while preserving the ability to model nonlinear behavior. The challenge lies in discovering appropriate lifting functions that transform the original state space into a space where linear dynamics emerge.

Traditional approaches to Koopman operator identification, such as Extended Dynamic Mode Decomposition (eDMD), rely on pre-defined dictionaries of basis functions—polynomials, trigonometric functions, or other hand-crafted features—to construct the lifted state space. This dictionary-based approach requires domain expertise to select appropriate basis functions, and the resulting models may be high-dimensional if many basis functions are needed to capture the dynamics. Moreover, when the true nonlinear terms are unknown—as is often the case in complex industrial systems—dictionary selection becomes a trial-and-error process that may fail to capture essential dynamics.

Deep learning approaches offer an alternative: rather than relying on pre-defined dictionaries, neural networks can learn appropriate lifting functions directly from data. Deep Koopman Operator (DKO) methods use autoencoder architectures to learn both the lifting (encoding) and unlifting (decoding) transformations, enabling dictionary-free identification of linear representations. However, these methods face challenges when systems exhibit time delays, as the history of the system must be incorporated into the model, potentially leading to high-dimensional lifted spaces.

Beyond computational efficiency, industrial applications demand models that respect physical principles. Energy conservation, mass balance, and other fundamental laws must be preserved in the identified models, not only to ensure physical consistency but also to enable extrapolation beyond training data. Pure data-driven models, trained solely to minimize prediction error on historical data, may violate these principles in ways that are not apparent during training but become critical when the system operates under novel conditions. Physics-informed approaches address this challenge by constraining the model structure to respect known physical laws, ensuring that predictions remain physically plausible even when extrapolating.

Physics-informed machine learning (PIML) encompasses multiple strategies for incorporating domain knowledge into data-driven models, which can be organized into three principal categories [29]. Observational bias augments or filters training data using physics knowledge, for instance by generating synthetic samples that cover underrepresented operating regions. Learning bias introduces physics-based loss terms that penalize violations of known laws during training, guiding the optimization toward physically consistent solutions. Inductive bias restricts the model to physically plausible solution spaces through architectural constraints or prior assumptions on model structure. The physics-informed DMDc approach presented in Section 5.3 draws primarily on inductive bias: the constrained optimization over matrix manifolds (5.20) enforces physical consistency by restricting the learned operator to a structure-preserving subspace, while the incorporation of electrochemical features through feature engineering (5.24) constitutes an observational bias that encodes domain-specific knowledge of the chlor-alkali process.

The following sections present two complementary approaches to these challenges. The LSTM-enhanced Deep Koopman method addresses the dictionary-free identification problem for systems with time delays, learning compact linear representations without requiring prior knowledge of nonlinear terms. The physics-informed DMDc approach demonstrates how physical constraints can be embedded into data-driven models, ensuring consistency with known principles while maintaining the flexibility to learn from operational data.

5.2 Dictionary-Free Identification of Nonlinear Systems with Input Delays

Industrial systems often exhibit complex nonlinear dynamics with time delays that complicate both modeling and control. Input delays arise from transport phenomena, actuator response times, and communication latencies, while state-dependent delays may emerge from process dynamics themselves. Traditional identification methods struggle with these challenges: dictionary-based approaches require prior knowledge of nonlinear terms, while standard deep learning methods may fail to capture temporal dependencies efficiently. This section presents an LSTM-enhanced Deep Koopman approach that learns linear representations of nonlinear systems with unknown dynamics and input delays, without requiring pre-defined dictionaries.

5.2.1 The Koopman Operator Framework

The foundation of the proposed approach lies in approximating the Koopman operator $\mathcal{K}$, a linear operator that represents the dynamics of nonlinear systems in an embedded space. Formally, we aim to find matrices in the state equation:

$$\boldsymbol{z}_{k+1} = \boldsymbol{A}_{\mathcal{K}}\boldsymbol{z}_k + \boldsymbol{B}_{\mathcal{K}}\boldsymbol{u}_k, \tag{5.1}$$

where $\mathcal{K} = \begin{bmatrix} \boldsymbol{A}_{\mathcal{K}} & \boldsymbol{B}_{\mathcal{K}} \end{bmatrix}$ is the solution to the optimization problem:

$$\arg\min_{\mathcal{K}} \sum_{k=0}^{n} \|\boldsymbol{z}_{k+1} - \mathcal{K} \begin{bmatrix} \boldsymbol{z}_k \\ \boldsymbol{u}_k \end{bmatrix} \|_F, \tag{5.2}$$

where $\boldsymbol{z}_k \in \mathbb{R}^p$ represents the vector of lifted states, $\boldsymbol{z}_{k+1}$ corresponds to states in the next step, and $\|\cdot\|_F$ denotes the Frobenius norm. This formulation provides the basis for deriving a linear approximation of the underlying dynamics, even for systems with time delays.

The proposed approach leverages the power of a long short-term memory (LSTM) layer to encode the history of a system, providing a novel method for data-based identification of the Koopman operator approximation. This approach can accurately identify linear models for processes with time delays, building upon the Deep Koopman Operator (DKO) approach [59] while addressing its limitations for delayed systems.

LSTM-enhanced Deep Koopman is designed to approximate the Koopman operator for systems with time delays. Thanks to the LSTM layer that processes the system's history denoted $\boldsymbol{H}$, we can effectively capture the time delays in the system. Time delays are encoded as the hidden states of the last LSTM layer denoted $\boldsymbol{h}_k$. This encoding enables the model to obtain relevant information about the system's past without requiring the entire history to be explicitly included in the lifted state space.

For example, using the LSTM layer, we can obtain a chosen number (a hyperparameter) of hidden states. These hidden states are then used as input to the encoder part of our approach. This way, we can reduce the amount of input data to the model compared to the Deep Koopman model, which would take the whole history of the system as input. This reduction is both more efficient and computationally less expensive.

Compared to DMD, where we would use the history of measurements directly, we can have a smaller Koopman matrix $\mathcal{K}$. Considering, for example, the last l_{his} measurements of the system states $\boldsymbol{x} \in \mathbb{R}^n$, the $\boldsymbol{A}_{\mathcal{K}}$ would have the size $l_{\text{his}}n \times l_{\text{his}}n$ and the $\boldsymbol{B}_{\mathcal{K}}$ would have the size $l_{\text{his}}n \times m$, where m is the number of inputs to the system $\boldsymbol{u} \in \mathbb{R}^m$. This is often a matrix that is more expensive to compute and store if we are dealing with a long history and many states compared to the LSTM-enhanced deep Koopman approach. On the other hand, the LSTM layer can capture the relevant information about the system's past and reduce the amount of input data in the model, resulting in a more compact representation.

5.2.2 Implementation of LSTM-enhanced Deep Koopman

LSTM-enhanced Deep Koopman was implemented in the Python programming language using the NeuroMANCER [23] library. The NeuroMANCER library does not provide the use of LSTM layers, so we implemented them into the library to support our approach.

The schematic of LSTM-enhanced Deep Koopman architecture can be seen in Figure 5.1. The model is built on the autoencoder architecture of the DKO. Before prediction, we compute the history $\boldsymbol{H}$ of our system:

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{x}_{k-\eta_H} & \cdots & \boldsymbol{x}_{k-1} \\ \boldsymbol{u}_{k-\eta_H} & \cdots & \boldsymbol{u}_{k-1} \end{bmatrix}, \tag{5.3}$$

where η_H is the number of time steps in the history of the system. We set this parameter to be higher than our observed or estimated time delay. The $\boldsymbol{H}$, consisting of state and input data, is first encoded using the LSTM layer. The hidden states of the last LSTM layer are then concatenated with the current state as input to the autoencoder part. The autoencoder part remains the same as for the DKO.

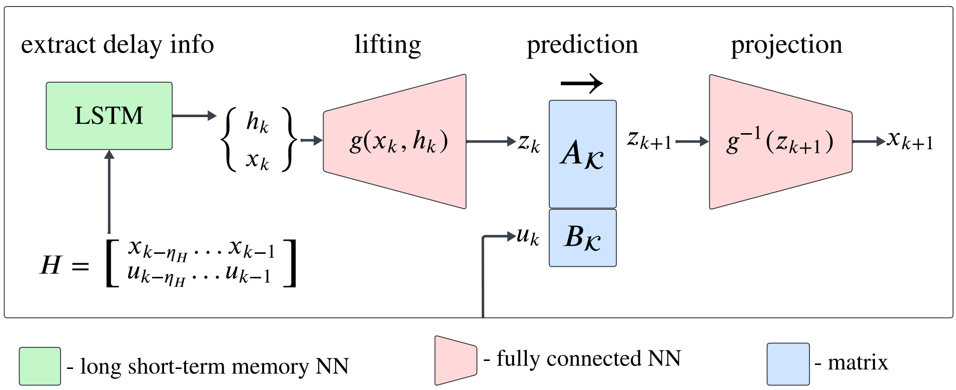

Figure 5.1: Schematic of LSTM-enhanced Deep Koopman architecture. The history of states and inputs is processed through an LSTM layer to extract hidden states, which are concatenated with the current state and fed into the encoder-decoder architecture for learning linear representations of nonlinear dynamics.

5.2.3 Loss Function Formulation

In the training of the LSTM-enhanced Deep Koopman model, we use the same loss function as in the training of the DKO model. This loss function combines the reconstruction loss, one-step output prediction loss, output trajectory prediction loss, and latent trajectory prediction loss, as described in [59]. This function is one of the most important parts of the training process. Compared to DMD, where the loss function is only the mean squared error between the predicted and true future states for a one-step prediction, the Deep Koopman methods also use a multi-step prediction in a loss function. In identifying a system, by using the multi-step prediction, we can better capture the system's dynamics with input delays as opposed to only one-step prediction loss.

Reconstruction loss is the mean squared error between the state $\boldsymbol{x}_k$ and the encoded (lifted) and decoded (unlifted) state $g^{-1}(g(\boldsymbol{x}_k, \boldsymbol{h}_k))$:

$$\mathcal{L}_{\text{rec}} = \left\| \boldsymbol{x}_k - g^{-1}(g(\boldsymbol{x}_k, \boldsymbol{h}_k)) \right\|^2 . \tag{5.4}$$

This loss is the basic loss function for the autoencoder part of the model. It ensures that we can correctly encode and decode the state of the system.

One step output prediction loss is the mean squared error between the predicted state $\hat{\boldsymbol{x}}_{k+1}$ and the true state $\boldsymbol{x}_{k+1}$:

$$\mathcal{L}_{\text{step}} = \left\| \hat{\boldsymbol{x}}_{k+1} - \boldsymbol{x}_{k+1} \right\|^2 , \tag{5.5a}$$

$$\hat{\boldsymbol{x}}_{k+1} = g^{-1}(\boldsymbol{A}_{\mathcal{K}} g(\boldsymbol{x}_k, \boldsymbol{h}_k) + \boldsymbol{B}_{\mathcal{K}} \boldsymbol{u}_k), \tag{5.5b}$$

where $\boldsymbol{u}_k$ is the input flow rate at time k, and the $\boldsymbol{A}_{\mathcal{K}}$ and $\boldsymbol{B}_{\mathcal{K}}$ are the identified Koopman matrices.

Output trajectory prediction loss is the mean squared error between the predicted trajectory $\hat{\boldsymbol{x}}_{k+1:k+N_\mathrm{L}}$ and the true trajectory $\boldsymbol{x}_{k+1:k+N_\mathrm{L}}$:

$$\mathcal{L}_{\text{pred}} = \sum_{i=1}^{N_\mathrm{L}} \left\| \hat{\boldsymbol{x}}_{k+i} - \boldsymbol{x}_{k+i} \right\|^2 , \tag{5.6a}$$

$$\hat{\boldsymbol{x}}_{k+i} = g^{-1}\left(\hat{\boldsymbol{z}}_{k+i}\right), \tag{5.6b}$$

$$\hat{\boldsymbol{z}}_{k+i} = \boldsymbol{A}_{\mathcal{K}} \hat{\boldsymbol{z}}_{k+i-1} + \boldsymbol{B}_{\mathcal{K}} \boldsymbol{u}_{k+i-1}, \tag{5.6c}$$

$$\hat{\boldsymbol{z}}_k = g(\boldsymbol{x}_k, \boldsymbol{h}_k), \tag{5.6d}$$

where N_L is the prediction horizon in the loss function specified by the user. In this case, the predicted trajectory $\hat{\boldsymbol{x}}_{k:k+N_\mathrm{L}}$ is predicted based on the $\boldsymbol{x}_k$, $\boldsymbol{h}_k$ and the input trajectory $\boldsymbol{u}_{k:k+N_\mathrm{L}-1}$. This loss function is used to capture the dynamics of the system and also input delays. By using the multi-step prediction loss, we can better capture the dynamics of the system as opposed to only one-step prediction loss.

Latent trajectory prediction loss is the mean squared error between the lifted predicted trajectory $\hat{\boldsymbol{z}}_{k+1:k+N_\mathrm{L}}$ and the lifted true trajectory $\boldsymbol{z}_{k+1:k+N_\mathrm{L}}$:

$$\mathcal{L}_\mathrm{lpred} = \sum_{i=1}^{N_\mathrm{L}} \left\| \hat{\boldsymbol{z}}_{k+i} - \boldsymbol{z}_{k+i} \right\|^2, \tag{5.7a}$$

$$\boldsymbol{z}_{k+i} = g(\boldsymbol{x}_{k+i}, \boldsymbol{h}_{k+i}), \tag{5.7b}$$

$$\hat{\boldsymbol{z}}_{k+i} = \boldsymbol{A}_\mathcal{K} \hat{\boldsymbol{z}}_{k+i-1} + \boldsymbol{B}_\mathcal{K} \boldsymbol{u}_{k+i-1}, \tag{5.7c}$$

$$\hat{\boldsymbol{z}}_{k} = g(\boldsymbol{x}_{k}, \boldsymbol{h}_{k}), \tag{5.7d}$$

where this loss function works with the same principle as the output trajectory prediction loss, but in the space of lifted states.

The final loss function is a weighted sum of all the loss functions:

$$\mathcal{L} = w_\mathrm{rec}\mathcal{L}_\mathrm{rec} + w_\mathrm{step}\mathcal{L}_\mathrm{step} + w_\mathrm{pred}\mathcal{L}_\mathrm{pred} + w_\mathrm{lpred}\mathcal{L}_\mathrm{lpred}, \tag{5.8}$$

where w_rec, w_step, w_pred, and w_lpred are the weights for the reconstruction loss, one-step output prediction loss, output trajectory prediction loss, and latent trajectory prediction loss, respectively. They are set by the user and are used to balance the importance of the loss functions in the training process.

5.2.4 Validation on Two-Tank System

This section presents a comparison of the performance of LSTM-enhanced Deep Koopman with eDMD on a simulated two-tank system with input delays. The continuous-time system is described by the following equations:

$$\begin{aligned} \frac{\partial\, h_1(t)}{\partial\, t} &= q(t-\tau) - \frac{k_1}{F_1}\sqrt{h_1(t)} \\ \frac{\partial\, h_2(t)}{\partial\, t} &= \frac{k_1}{F_2}\sqrt{h_1(t)} - \frac{k_2}{F_2}\sqrt{h_2(t)}, \end{aligned} \tag{5.9}$$

where $h_1(t)$ and $h_2(t)$ are the water levels in tanks 1 and 2, respectively, $q(t)$ is the input flow rate, τ is the input delay, and k_1, k_2 are the flow rate constants of the valves and F_1, F_2 are the cross-sectional areas of the tanks.

The experimental data were acquired by simulating this system at sampling period $T_\mathrm{s} = 10\,\mathrm{s}$ with input delay $\tau = 20T_\mathrm{s}$. Random changes in the input flow rate from the interval $q_\mathrm{min} = 0.0\,\mathrm{m^3s^{-1}}$, $q_\mathrm{max} = 0.03\ \mathrm{m^3s^{-1}}$ were applied to the system. The simulation yielded $4 \cdot 10^5$ samples. Gaussian noise with standard deviation of 0.1 was added to the data to simulate real-world conditions.

The data were split into training and testing sets with a ratio of 50:50. The training set was used to train the models, while the testing set was used to evaluate their performance. The models were trained to predict the water levels in tanks 1 and 2 for the whole testing set based on the applied input flow rate and the initial state of the system.

For reference, we use eDMD with a dictionary of polynomials up to degree 2, the correct governing nonlinear dynamics (square root of the water levels), and time-delayed embeddings of the input flow rate composed of the previous 20 samples. For comparison, eDMD without the square root of the water levels was also used, simulating a situation where the true nonlinear terms are not known. Therefore, they are not included in the dictionary, which is often the case in more complex systems. Lifted states were obtained by concatenating the dictionary terms and time-delayed embeddings, and the Koopman matrix was learned using eDMD.

The LSTM-enhanced Deep Koopman model consists of an LSTM layer with one hidden layer with 8 units to extract information about the time delays in the system from the history of the system. The concatenated information is then transformed into lifted states using an encoder layer with a fully connected network. The lifting network consists of an input layer with 10 neurons representing inputs $\boldsymbol{x}$ and $\boldsymbol{h}$, two hidden layers, each with 60 neurons, and a lifted state layer consisting of 40 neurons representing lifted states $\boldsymbol{z}$. After lifting the states, there is a prediction layer represented with the matrices $\boldsymbol{A}_{\mathcal{K}}$ (40×40) and $\boldsymbol{B}_{\mathcal{K}}$ (40×1). After the prediction layer, the lifted states are projected back to 2 states representing the water level in the tanks, using the decoder part of the autoencoder with two hidden layers, each containing 60 neurons.

For the weights in (5.8), we set $w_{\mathrm{rec}} = 0$, $w_{\mathrm{step}} = 1$, $w_{\mathrm{pred}} = 10$, and $w_{\mathrm{lpred}} = 1$. The model was trained using PyTorch for 1500 epochs with a batch size of 100 using the Adam optimizer with a learning rate of 0.001.

Figure 5.2 shows the predicted and actual water levels in tanks 1 and 2 for the two-tank system using LSTM-enhanced Deep Koopman and eDMD with and without the square root of the water levels. All three algorithms were initialized with access to the same $\boldsymbol{H}$. The results demonstrate that LSTM-enhanced Deep Koopman outperforms eDMD, which does not include true nonlinear behavior, in terms of prediction accuracy, capturing the system's dynamics more effectively over the testing set. eDMD without the square root of the water levels exhibits systematic overestimation of the influence of input flow rate step changes, leading to a positive bias in the prediction of water levels. At the same time, LSTM-enhanced Deep Koopman provides more accurate and consistent predictions. The linear model generated by LSTM-enhanced Deep Koopman displays non-minimum phase behavior, which is actually a possible approach in the identification of systems with input delays. Meanwhile, eDMD with the square root of the water levels provides accurate prediction, as it includes the true nonlinear terms in the dictionary, thanks to prior knowledge of the system's dynamics.

The performance of the models was evaluated using the median absolute error (MAE) between the predicted and actual water levels in tanks 1 and 2. The results are presented in Table 5.1. LSTM-enhanced Deep Koopman achieved a significantly lower MAE compared to eDMD without square root, indicating its superior performance in modeling the system with input delays in cases where true nonlinear dynamics are not known. Meanwhile, LSTM-enhanced Deep Koopman provides comparable results to eDMD with square root, demonstrating its effectiveness in capturing the system's dynamics without prior knowledge of the true nonlinear terms, demonstrating the capabilities of the LSTM-enhanced Deep Koopman model in modeling nonlinear systems with input delays. The degradation in performance was, in this case, slightly above 6% in MAE. It is important to note that the LSTM-enhanced Deep Koopman model is composed of a smaller number of lifted states compared to eDMD. This is thanks to the fact that the history of the system is encoded and does not need to be introduced whole to the model during lifted states generation. This results in an almost 4 times smaller Koopman matrix, which is computationally less expensive to compute and store.

Table 5.1: Performance comparison of LSTM-enhanced Deep Koopman and eDMD on the two-tank system with input delays with best performance normalized to 100%

Model	**MAE** [%]
eDMD (known dynamics)	100
LSTM-enhanced Deep Koopman	106
eDMD (unknown dynamics)	349

Lastly, we compare the eigenvalues of the Koopman operator for the two-tank system using LSTM-enhanced Deep Koopman and eDMD. Figure 5.3 shows that LSTM-enhanced Deep Koopman provides more accurate and consistent eigenvalues with original eigenvalues compared to eDMD. This is an important contribution as the predictions could better align with reality and provide more accurate and consistent results.

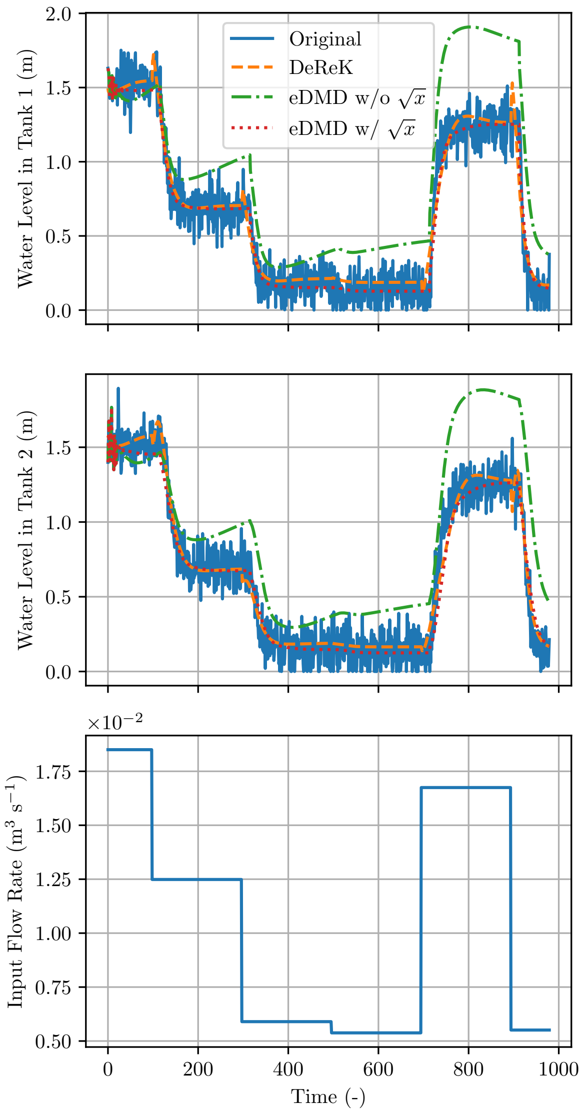

Figure 5.2: Predicted and actual water levels in tanks 1 and 2 for the two-tank system. Gray shows the noisy data, as used for training, while the blue shows the signal without noise. The sequence shows a snippet of the testing set which was unseen during system identification.

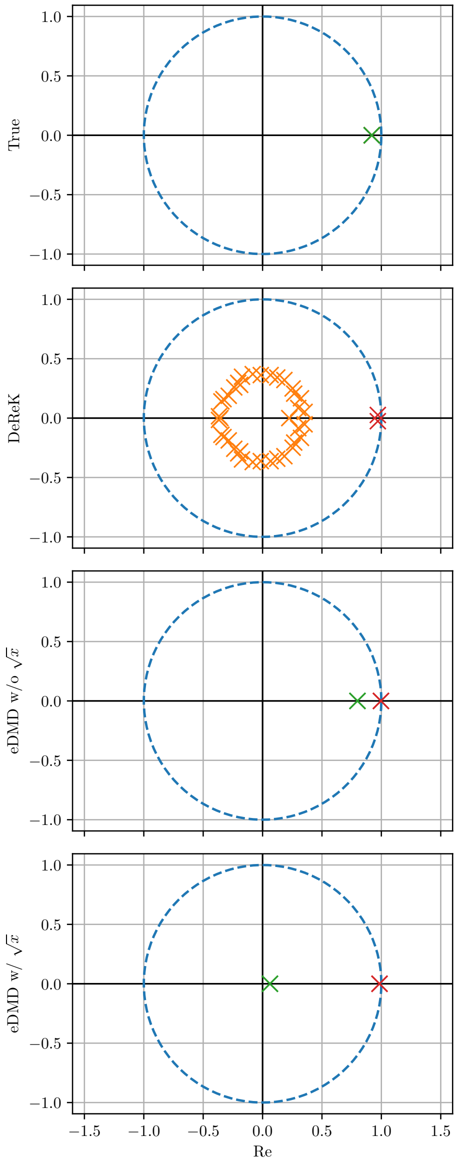

Figure 5.3: Comparison of eigenvalues of the Koopman operator for the two-tank system. Red is the most dominant eigenvalue, green is the second most dominant eigenvalue, and orange are the rest of the eigenvalues. In all cases, all the eigenvalues are stable. The true eigenvalues of the system are given for linearization around mean water levels.

5.3 Physics-Informed Modeling for Predictive Maintenance

While dictionary-free approaches enable identification of nonlinear systems without prior knowledge, industrial applications often require models that respect known physical principles. This requirement becomes particularly critical in predictive maintenance applications, where models must extrapolate beyond training data to predict degradation trajectories and remaining useful life. This section presents a physics-informed Dynamic Mode Decomposition with Control (piDMDc) framework for predictive maintenance, demonstrating how physical constraints can be embedded into data-driven models to ensure consistency with known principles while maintaining the flexibility to learn from operational data.

Remaining Useful Life prediction is a core discipline within the broader Prognostics and Health Management (PHM) framework, alongside fault detection, diagnostics, and health management. Physics-based and data-driven models for PHM typically suffer from two major challenges: physics-based models cannot capture all degradation mechanisms due to incomplete knowledge of failure physics, while data-driven models trained on limited operational histories may not cover the full degradation trajectory, leading to unreliable extrapolation [17]. The piDMDc approach presented here addresses both challenges simultaneously: physics constraints ensure consistency with known electrochemical principles governing the chlor-alkali process, while the data-driven DMD framework learns system dynamics directly from operational measurements, compensating for incomplete first-principles knowledge.

5.3.1 Problem Statement

5.3.1.1 Mechanism of Impurity Buildup

The operation of a chlor-alkali electrolyzer cell is fundamentally based on interrelated electrochemical reactions that yield chlorine at the anode, while also generating hydrogen and hydroxide ions OH^- at the cathode. The generation of OH^- is critical, as it sets the stage for the precipitation of sparingly soluble brine impurities, particularly Ca^{2+} and Mg^{2+}. The formation of calcium and magnesium hydroxides as

$$\mathrm{Ca^{2+}} + 2\,\mathrm{OH^-} \longrightarrow \mathrm{Ca(OH)_2},$$

$$\mathrm{Mg^{2+}} + 2\,\mathrm{OH^-} \longrightarrow \mathrm{Mg(OH)_2},$$

obstructs ion transport and contributes to voltage instability. Meanwhile, their interaction with dissolved CO_2 and ${SO_4}^{2-}$ yields carbonates and sulfates that further exacerbate scaling phenomena and disrupt hydraulic balance. Collectively, these reactions elucidate the dual role of the electrolyzer environment: while optimized for Cl_2 production, the concurrent generation of OH^- fosters impurity-induced degradation, thereby impairing system performance and longevity.

Crucially, this degradation exhibits a self-reinforcing positive feedback mechanism: as impurity deposits accumulate on the electrode surface, they create a rougher, more uneven topology that exposes additional nucleation sites and increases the effective surface area available for further deposition. This positive feedback loop produces exponential rather than linear growth in impurity mass, making early detection critical for timely maintenance intervention.

5.3.1.2 Electrolyzer Degradation Mechanisms

A fundamental challenge in the predictive maintenance of chlor-alkali electrolyzers is estimating impurity buildup when direct observation is impossible. This necessitates inference-based state estimation from observable variables without direct ground truth. Conventionally, the deviation between the measured and predicted resistance is used as a proxy for the degradation of the system [106, 97, 8]. With impurity buildup being the dominant degradation mechanism following an exponential trend [71], we leverage a key insight: a well-trained data-driven model with good generalization properties will capture normal system dynamics but systematically underestimate resistance growth due to impurity effects. This systematic bias provides a quantifiable indicator of degradation progression, enabling predictive maintenance despite unmeasurable states.

5.3.1.3 Laboratory Analysis Estimation

Laboratory analyses of brine impurity concentrations, which occur hourly or daily, challenge the elucidation of their effects from sparse observations. These measurements offer high accuracy, yet miss critical transient dynamics needed for accurate modeling. To address these challenges, the state estimation must be designed to reconcile the temporal resolution of the data with the underlying dynamics of the buildup of impurities. This may involve data fusion techniques or adaptive filtering approaches that effectively bridge the gap between continuous process dynamics and discrete measurement instances. Ultimately, the objective is to ensure that the degradation model captures the underlying dynamics of the system for both gradual changes and sudden disturbances, enabling more reliable and timely decisions.

5.3.1.4 Formal Definition of the RUL Prediction Problem

Remaining Useful Life (RUL) prediction provides a unified framework and a common language to describe the degradation of the system. It represents the remaining time of the system to reach the failure threshold. Using the previous evolution of the State of Health (SoH), which is a normalized measure of the health of a specific component (spanning from 1, fully healthy, to 0, failed), the RUL model learns the degradation trend. The model then extrapolates this trend into the future.

5.3.1.5 Key Assumptions

The proposed approach relies on the following assumptions:

Assumption 1: The impurities buildup affecting the cell resistance does not influence other operating conditions.

Assumption 2: The cell resistance increases exponentially over time due to impurities buildup [71].

Assumption 3: Impurities buildup constitutes the dominant unmodeled degradation mechanism, with degradation being monotonic and continuous.

Assumption 4: The offset between measured and predicted resistance remains consistent over time when degradation trends are neglected.

5.3.2 Methodology

5.3.2.1 DMD with Control for Resistance Prediction

The DMD with control (DMDc) method extends its application to systems with exogenous control inputs, enabling the identification of both the underlying dynamics and the input-to-state mapping from data [73]. Consider a discrete-time dynamical system with control:

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k, \tag{5.10}$$

where k denotes the discrete time instance, $\boldsymbol{x}_k \in \mathbb{R}^n$ is the state, $\boldsymbol{u}_k \in \mathbb{R}^l$ is the control input, and $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{B} \in \mathbb{R}^{n\times l}$ are the unknown state transition and input matrices, respectively.

When $\boldsymbol{A}$ and $\boldsymbol{B}$ are unknown, DMDc jointly estimates them from snapshot data. Let

$\boldsymbol{X}$, $\boldsymbol{X}'$, and $\boldsymbol{\Upsilon}$ be matrices of state and control snapshots:

$$\boldsymbol{X} = [\boldsymbol{x}_1 \ \boldsymbol{x}_2 \ \cdots \ \boldsymbol{x}_{k-1}], \tag{5.11}$$

$$\boldsymbol{X}' = [\boldsymbol{x}_2 \ \boldsymbol{x}_3 \ \cdots \ \boldsymbol{x}_k], \tag{5.12}$$

$$\boldsymbol{\Upsilon} = [\boldsymbol{u}_1 \ \boldsymbol{u}_2 \ \cdots \ \boldsymbol{u}_{k-1}]. \tag{5.13}$$

The system dynamics can be rewritten as:

$$\boldsymbol{X}' \approx [\boldsymbol{A} \ \boldsymbol{B}]\boldsymbol{\Omega}, \quad \text{where} \quad \boldsymbol{\Omega} = \begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{\Upsilon} \end{bmatrix}. \tag{5.14}$$

To solve for $\boldsymbol{G} = [\boldsymbol{A} \ \boldsymbol{B}]$, the following steps are performed:

1. Compute the truncated SVD of the augmented input matrix $\boldsymbol{\Omega} \approx \tilde{\boldsymbol{U}}\tilde{\boldsymbol{\Sigma}}\tilde{\boldsymbol{V}}^\top$, retaining r singular values.
2. Estimate $\boldsymbol{G}$ via:
$$\boldsymbol{G} \approx \boldsymbol{X}'\tilde{\boldsymbol{V}}\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{U}}^\top. \tag{5.15}$$
3. Partition $\tilde{\boldsymbol{U}}^\top = [\tilde{\boldsymbol{U}}_1^\top \ \tilde{\boldsymbol{U}}_2^\top]$ to recover estimated $\boldsymbol{A}$ and $\boldsymbol{B}$:
$$\tilde{\boldsymbol{A}} = \boldsymbol{X}'\tilde{\boldsymbol{V}}\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{U}}_1^\top, \tag{5.16}$$
$$\tilde{\boldsymbol{B}} = \boldsymbol{X}'\tilde{\boldsymbol{V}}\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{U}}_2^\top, \tag{5.17}$$

where $\tilde{\boldsymbol{U}}_1^\top \in \mathbb{R}^{p\times n}$ and $\tilde{\boldsymbol{U}}_2^\top \in \mathbb{R}^{(r-p)\times n}$ are the left singular vectors of $\tilde{\boldsymbol{U}}^\top$ corresponding to the r largest singular values.

The model is then given by:

$$\boldsymbol{x}_{k+1} \approx \tilde{\boldsymbol{x}}_{\mathrm{k+1,=}}\tilde{\boldsymbol{A}}\boldsymbol{x}_k + \tilde{\boldsymbol{B}}\boldsymbol{u}_k. \tag{5.18}$$

5.3.2.2 Physics-Informed DMDc

Despite their widespread adoption, data-driven approaches, including DMD, often yield models that struggle with extrapolation beyond training datasets and may contradict fundamental physical principles. Physics-informed DMD (piDMD) addresses these limitations [7]. We unify piDMD and DMDc (piDMDc) in a single framework.

Consider the discrete-time system with control:

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k + \mathcal{G}(\boldsymbol{x}_k, \boldsymbol{u}_k), \tag{5.19}$$

where $\mathcal{G}$ captures unmodeled nonlinear interactions between states and inputs. piDMDc jointly addresses two key challenges:

Constrained Matrix Manifolds for $\boldsymbol{A}$ and $\boldsymbol{B}$ When physical principles (e.g., energy conservation, locality, or time dependency) are known, we constrain $\boldsymbol{A}$ and $\boldsymbol{B}$ to lie on prescribed matrix manifolds $\mathcal{M}_A$ and $\mathcal{M}_B$:

$$\arg\min_{\boldsymbol{A}\in\mathcal{M}_A, \boldsymbol{B}\in\mathcal{M}_B} \left\|\boldsymbol{X}' - \boldsymbol{A}\boldsymbol{X} - \boldsymbol{B}\boldsymbol{\Upsilon}\right\|_F, \tag{5.20}$$

where the norm is the Frobenius norm.

For the states representing the resistance of the cells, which may exhibit local effects, we apply tridiagonal matrices for $\mathcal{M}_A$ and shift-equivariant block Hankel matrices for $\mathcal{M}_B$ for time-delayed embeddings. Formally, following [7],

$$\mathcal{M}_A = \left\{\boldsymbol{A} \in \mathbb{R}^{n\times n} \mid A_{ij} = 0 \text{ for } |i-j| > 1\right\}, \tag{5.21}$$

$$\mathcal{M}_B = \left\{\boldsymbol{B} \in \mathbb{R}^{n\times md} \mid \boldsymbol{B}_{ij} = \boldsymbol{B}_{i+1,j+m} \text{ for all admissible } i, j\right\}, \tag{5.22}$$

where d is the number of time-delayed copies of the m-dimensional input. The tridiagonal structure of $\mathcal{M}_A$ encodes spatial locality, restricting each cell's dynamics to depend only on its immediate neighbors, while the block Hankel structure of $\mathcal{M}_B$ enforces translation invariance of the control effect along the time-delay axis—a discrete analogue of shift equivariance. The selection of specific manifold structures is guided by physical principles and canonical structures as outlined in [7].

Feature Engineering Whether the governing mechanisms of subsystems are known or we wish to discover nonlinear dependencies and time-delayed effects, we may augment the control inputs using this knowledge. This introduces new observables $\boldsymbol{\phi}(\boldsymbol{u}_k, k)$, which may include nonlinear combinations of arbitrary elements i and j of $\boldsymbol{u}_k$, results of nonlinear function h applied to input, or previous control actions, which may take the form of

$$\boldsymbol{\phi}(\boldsymbol{u}_k, k) = [u_i(k)u_j(k), h(u_i(k)), u_i(k-1), \dots], \tag{5.23}$$

into the augmented control input vector

$$\bar{\boldsymbol{u}}_k = [\boldsymbol{u}_k; \boldsymbol{\phi}(\boldsymbol{u}_k)]. \tag{5.24}$$

While it is possible to introduce linear combinations, this must be done with caution to avoid the overfitting problem caused by linear correlations.

The extended system becomes:

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \bar{\boldsymbol{B}}\bar{\boldsymbol{u}}(k), \quad \boldsymbol{A} \in \mathcal{M}_A, \bar{\boldsymbol{B}} \in \mathcal{M}_{\bar{B}}, \tag{5.25}$$

effectively capturing both the linear dynamics and nonlinear interactions while maintaining physical consistency through constrained optimization, thus enabling more consistent and physically plausible predictions even with limited training data.

5.3.2.3 Multi-fidelity Model for Impurity Concentration Estimation

Estimating brine impurity concentrations with sparse high-fidelity laboratory measurements can be addressed through semi-supervised regression techniques [25].

Let us denote high-fidelity measurements as $\boldsymbol{u}_{\mathrm{HF}}(k)$ sampled at time instances $\mathcal{K}_{\mathrm{HF}}$ and low-fidelity predictions as $\boldsymbol{u}_{\mathrm{LF}}(k)$ sampled at $\mathcal{K}_{\mathrm{LF}}$, where:

$$|\mathcal{K}_{\mathrm{LF}}| \gg |\mathcal{K}_{\mathrm{HF}}| \quad \text{and} \quad \mathcal{K}_{\mathrm{HF}} \subset \mathcal{K}_{\mathrm{LF}} \tag{5.26}$$

The scarce sampling of $\boldsymbol{u}_{\mathrm{HF}}(k)$ violates the Nyquist-Shannon sampling theorem, creating a challenge for dynamic model development. This problem presents a peculiar circularity: to develop a model that predicts $\boldsymbol{u}_{\mathrm{LF}}(k)$ from $\boldsymbol{u}_{\mathrm{HF}}(k)$, we already need sufficient low-fidelity data for training.

Let us interpolate the high-fidelity data to the desired sampling rate using a linear method and train the data-driven model. For a given interpolation function $\mathcal{I}$, we can obtain:

$$\tilde{\boldsymbol{u}}_{\mathrm{LF}}(k) = \mathcal{I}(\boldsymbol{u}_{\mathrm{HF}}(\mathcal{K}_{\mathrm{HF}}), \boldsymbol{\theta}, k), \quad \forall k \in \mathcal{K}_{\mathrm{LF}}. \tag{5.27}$$

The optimization of model parameters $\boldsymbol{\theta}$ is formulated through a cost function that minimizes the deviation between the high-fidelity measurements and model predictions at given instances:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \sum_{k \in \mathcal{K}_{\mathrm{HF}}} \|\boldsymbol{u}_{\mathrm{HF}}(k) - \tilde{\boldsymbol{u}}_{\mathrm{LF}}(k)\|_2. \tag{5.28}$$

5.3.2.4 Deviation-based Degradation Modeling

The deviation between measured and predicted resistance provides a quantitative indicator of impurity buildup. Let every element of $\boldsymbol{x}_k$ represent the measured resistance of a specific electrolyzer cell at time instance k, and $\tilde{\boldsymbol{x}}(k)$ denote the predicted resistances from a model. We define the resistance deviation $\Delta\boldsymbol{x}_k$ as:

$$\Delta\boldsymbol{x}_k = \boldsymbol{x}_k - \tilde{\boldsymbol{x}}(k). \tag{5.29}$$

This deviation captures the unmeasurable degradation state that even models with good generalization capabilities cannot detect. Based on empirical observations made in [71], we model this deviation growth due to the impurities buildup using an exponential function:

$$\Delta\boldsymbol{x}_k = \Delta\boldsymbol{x}_0 + e^{k/\boldsymbol{\lambda}} + \boldsymbol{\epsilon}_k \tag{5.30}$$

where $\Delta\boldsymbol{x}_0$ is the initial deviation, $\boldsymbol{\lambda}$ represents the cell-specific degradation rate parameter, and $\boldsymbol{\epsilon}_k$ accounts for the process and measurement noise.

The maximum resistance deviation $\Delta\boldsymbol{x}_{\max}$ corresponds to the threshold beyond which the cell is considered failed:

$$\Delta\boldsymbol{x}_{\max} = \Delta\boldsymbol{x}_{k_{\max}}, \tag{5.31}$$

where $k_{\max}$ is the maximum life of the cells. Appropriate selection of degradation model given the physics of the system is crucial to ensure that Assumption 4 holds and accumulation of prediction error due to unmodeled effects is minimized.

5.3.2.5 Mapping from Deviation to SoH and RUL

Let us define $\boldsymbol{SoH}_k \in \mathbb{R}^n$ as the state of health of n cells. We define the failure time k_{f} as the smallest k such that

$$\boldsymbol{SoH}_{k_{\mathrm{f}}} \leq \tau_{\mathrm{f}}, \tag{5.32}$$

where τ_{f} is a predefined failure threshold. Typically, to select a Pareto optimal τ_{f}, a trade-off must be made between the capital expenditures (CapEx) of replacement and the operational expenditures (OpEx) associated with decreased system efficiency [52].

Accordingly, the Remaining Useful Life is given by

$$\mathrm{RUL} = k_{\mathrm{f}} - k. \tag{5.33}$$

In practice, the degradation trajectory $\boldsymbol{SoH}_k$ is estimated from historical operational data using regression techniques. The prediction problem then becomes one of fitting a function $\mathcal{F}$ such that

$$\mathrm{RUL} \approx \mathcal{F}\left(\{\boldsymbol{SoH}_i\}_{i\leq k}\right), \tag{5.34}$$

which extrapolates the current degradation trend into the future.

By estimating confidence intervals around the predicted RUL, the model can account for measurement noise, model inaccuracies, and variations in operational conditions. This probabilistic perspective supports risk-informed maintenance planning and helps determine optimal intervention strategies.

The deviation increases exponentially as the prediction model neglects the effect of impurities buildup and remains consistent while the actual resistance increases [71]. This is the opposite trend of the SoH, which decreases with the degradation. To obtain $\boldsymbol{SoH}_k$, we normalize the deviation to the maximum deviation and subtract it from perfect SoH, using

$$\boldsymbol{SoH}_k = 1 - \frac{\Delta \boldsymbol{x}_k}{\Delta \boldsymbol{x}_{\max}}. \tag{5.35}$$

We make prediction of the $\boldsymbol{SoH}_{k+1}$ using a reflected exponential decay function of the deviation as follows:

$$\widetilde{\boldsymbol{SoH}}(k+1) = 1 - e^{-(k-k_{\max})/\boldsymbol{\lambda}}, \tag{5.36}$$

where $k_{\max}$ is the maximum life of the cells at which $\boldsymbol{SoH}_{k_{\max}} = 0$. The parameters $k_{\max}$ and $\boldsymbol{\lambda}$ are obtained from the past measurements of the deviation using the Trust Region Reflective algorithm, which solves the following constrained optimization problem:

$$\begin{aligned} \min_{k_{\max}, \boldsymbol{\lambda}} \quad & \sum_{i=0}^{k} \|\widetilde{\boldsymbol{SoH}}(i) - \boldsymbol{SoH}_i\|_2 \\ \text{s.t.} \quad & k_{\max} \geq 0, \boldsymbol{\lambda} \geq 0. \end{aligned} \tag{5.37}$$

5.3.2.6 Uncertainty Quantification

Uncertainty quantification is implemented through confidence intervals, computed using the covariance matrix from the parameter estimation. For a given confidence level $\alpha \in [0, 1]$, the confidence intervals for the estimated parameters are given by:

$$\begin{bmatrix} k_{\max} \\ \boldsymbol{\lambda} \end{bmatrix} \pm z_{\alpha/2} \cdot \sqrt{\operatorname{diag}(\boldsymbol{C}_{k_{\max}, \boldsymbol{\lambda}})}, \tag{5.38}$$

where $\boldsymbol{C}_{k_{\max}, \boldsymbol{\lambda}}$ is the covariance matrix of the estimated parameters, and $z_{\alpha/2}$ is the critical value of the standard normal distribution for the desired confidence level.

Given the prediction model, we can analyze the evolution of the $\boldsymbol{SoH}_k$ over the system's useful life. The RUL is then determined as the worst case at which the predicted SoH reaches the predefined failure threshold τ_{f}:

$$\mathrm{RUL} = \min\{k \in \mathbb{N} \mid \widetilde{\boldsymbol{SoH}}(k) \leq \tau_{\mathrm{f}}\}. \tag{5.39}$$

The maintenance is scheduled at the RUL to prevent suboptimal operation and avoid expensive unplanned downtime. Reaching end of life suggests cleaning and replacement of the electrode cells.

5.3.3 Implementation Case Study

We validate the method using historical data from a chlor-alkali electrolyzer with an installed capacity of 3 MW and designed chlorine production of 104 T/year, located in Italy. The electrolyzer comprises 112 cells, each with an active area of $3.2\,\mathrm{m}^2$.

5.3.3.1 Data Collection Setup

Data were collected at 30 min intervals over one year, encompassing regular operation, start-up, and shutdown phases. A simplified block scheme of the electrolyzer (Figure 5.4) illustrates sensor locations. Key challenges included sensor noise, communication delays, and occasional outages. Industrial data were normalized to the [0,1] range to protect proprietary information while preserving relative dynamics. Sequences with zero flow rates were automatically filtered to exclude downtime effects.

5.3.3.2 Experimental Setup

The piDMDc rank was selected via grid search, considering optimal hard thresholding [34], ranks 1–10, and full-rank configurations. First-principles transformations from the chlor-alkali electrolysis model were applied during feature engineering [100]. Hankel embeddings captured time-delayed control effects, followed by output standardization to address scale variations in Frobenius-norm optimization (Eq. (5.20)).

5.3.3.3 Stability Analysis

The identified system matrix $\boldsymbol{A}$ (tridiagonal structure) reveals localized cell interactions, while $\boldsymbol{B}$ with engineered features, as shown in Figure 5.5, better reveals the relationship between the control input and cell dynamics. Constrained (spatially banded) and unconstrained $\boldsymbol{A}$ matrices were compared, showing reduced coupling between distant cells. All poles reside within the unit circle, confirming stability. Sensitivity analysis further quantified dominant input effects on state evolution.

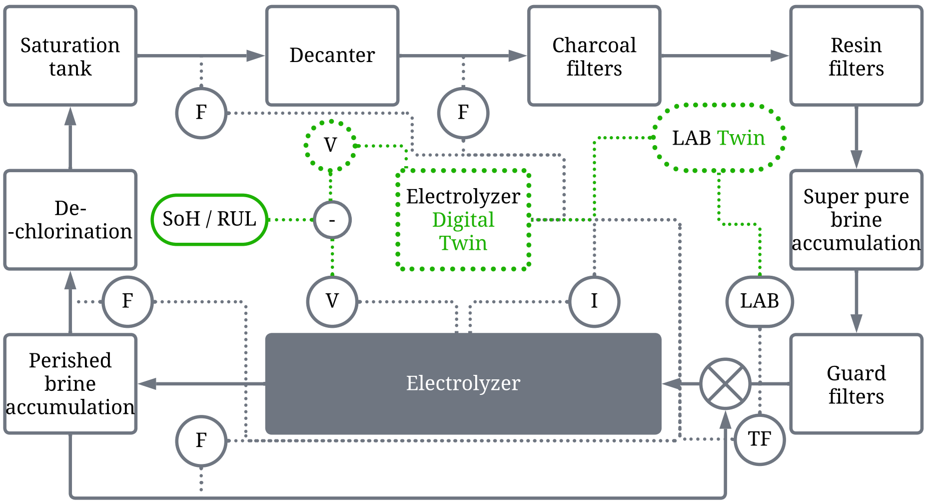

Figure 5.4: Illustrative block scheme of the electrolyzer with sensor locations. The green color indicates our contribution to the overall chlor-alkali electrolysis process.

5.3.3.4 Validity Assessment

Compared to the baseline method, partial least squares (PLS), previously used on the electrolyzer for degradation assessment, piDMDc reduces mean squared error (MSE) by 61 % (from 0.0017 to 0.0002) and improves R^2 scores by 24 % (from 0.74 to 0.96). This performance improvement was consistent across all intervals of regular operation, with particularly notable gains during transient states where the model captured complex dynamics that PLS failed to represent [44].

5.3.3.5 Results

Figure 5.6 (a) shows measured versus predicted mean cell resistance over two weeks of operation in January 2024. Predictions align with measurements during this period. By August 2024 (Figure 5.6 (b)), a growing offset between predicted and measured resistance emerges, consistent with long-term impurity effects.

Fitting the degradation model (Eq. (5.37)) to the state of health data captures the exponential growth of prediction offsets (Figure 5.7). The framework suggests a maintenance deadline of May 11, 2025, with a worst-case failure estimate of March 17, 2025.

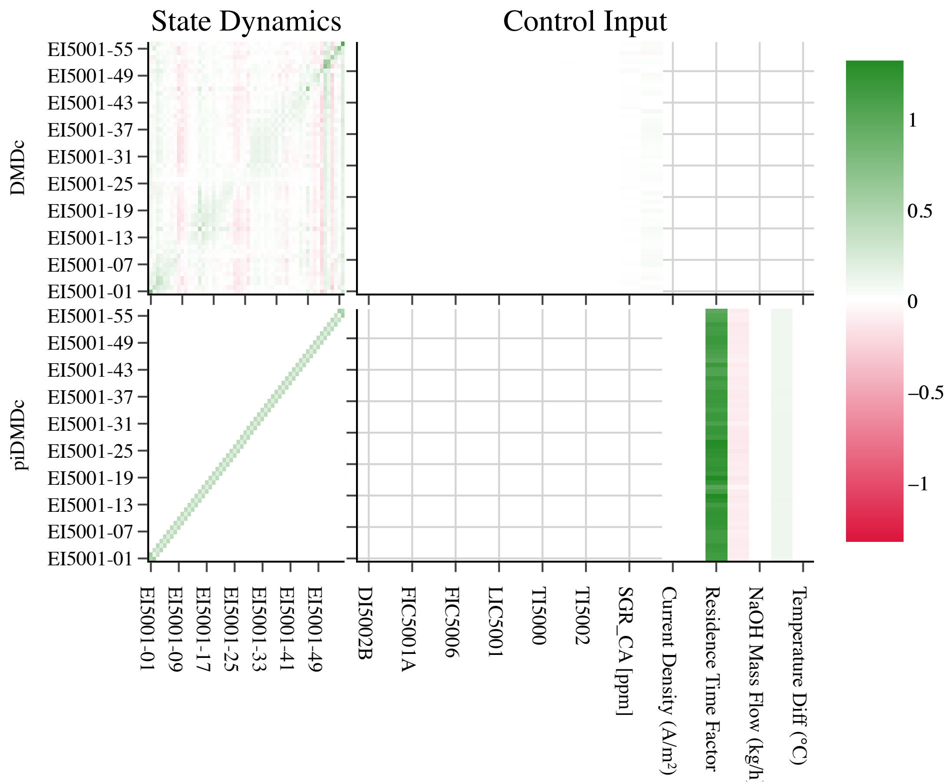

Figure 5.5: Comparison of the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ for the DMDc and piDMDc models. The unconstrained $\boldsymbol{A}$ matrix indicates equal influence of each cell to itself and all other cells. Meanwhile, the constrained $\boldsymbol{A}$ matrix is tridiagonal, capturing interactions between neighboring cells. Matrix $\boldsymbol{B}$ in piDMDc, with engineered features, better reveals the relationship between the control input and cell dynamics than the DMDc model.

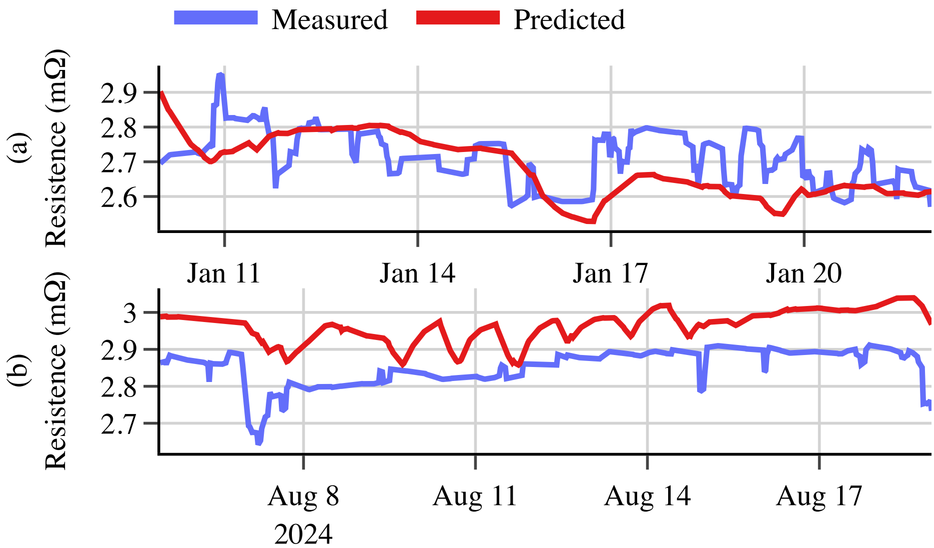

Figure 5.6: Measured versus predicted mean cell resistance over two weeks of operation in January 2024 (a) and August 2024 (b). Consistent predictions are observed in the former while the predictions underestimate the measured resistance in the latter.

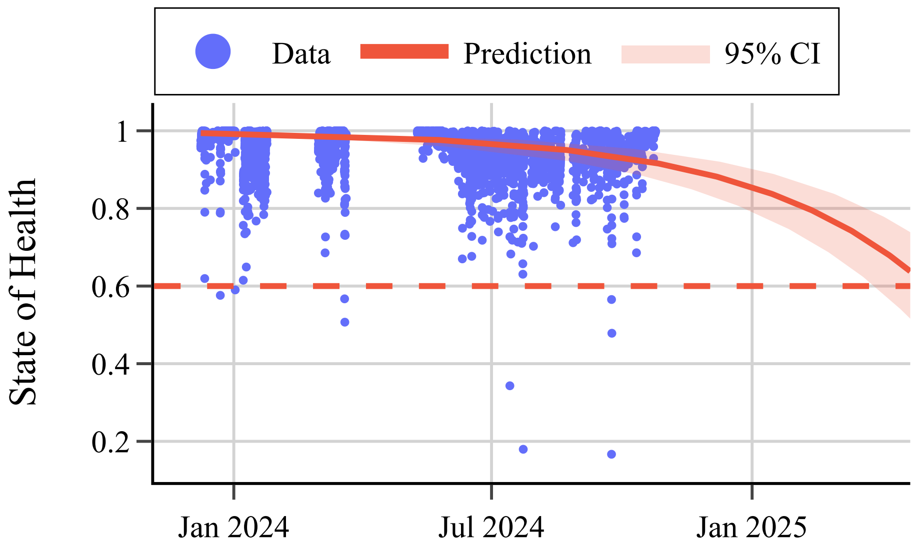

Figure 5.7: Graph depicting RUL as the time difference between the red (maintenance deadline) and gray (last measurement time) vertical dotted lines. Horizontal dashed line represents the desired SoH threshold. Blue points represent the measured state of health, while the red line represents the predicted evolution of SoH with a 95% confidence interval.

5.4 From Modeling to Control: Validated Models as Control Foundations

The interpretable modeling approaches presented in this chapter—LSTM-enhanced Deep Koopman for dictionary-free identification and physics-informed DMDc for degradation-aware modeling—establish the foundation upon which the control frameworks of Chapter 3 operate. The linear representations learned through the Koopman approach enable efficient Model Predictive Control formulations, while the physical consistency ensured by physics-informed constraints provides the safety guarantees necessary for industrial deployment.

The LSTM-enhanced Deep Koopman method addresses a fundamental challenge in control-oriented modeling: systems with time delays and unknown nonlinear dynamics cannot be effectively controlled using traditional linearization approaches, yet nonlinear MPC may be computationally intractable for real-time implementation. By learning compact linear representations in a lifted state space, the method enables the use of efficient linear MPC algorithms while preserving the ability to model nonlinear behavior. The validation on the two-tank system demonstrates that this approach can achieve performance comparable to dictionary-based methods that require prior knowledge of nonlinear terms, while maintaining computational efficiency through reduced dimensionality.

The physics-informed DMDc approach demonstrates how physical constraints can be embedded into data-driven models to ensure consistency with known principles. This consistency is not merely a theoretical nicety—it enables models to extrapolate beyond training data with confidence, a critical requirement for predictive maintenance applications where models must predict degradation trajectories extending months or years into the future. The case study on chlor-alkali electrolyzer degradation demonstrates that physically consistent models can achieve superior prediction accuracy compared to purely data-driven approaches, while providing interpretable insights that enable operators to understand and trust the system's behavior.

These validated, physically consistent models serve as the foundation for the predictive controllers developed in Chapter 3. The linear Koopman representations enable efficient MPC formulations that can execute in real-time, while the physical consistency ensures that control actions respect fundamental constraints and safety limits. The degradation-aware models enable predictive maintenance strategies that optimize both operational performance and maintenance scheduling, balancing immediate economic objectives with long-term reliability requirements.

However, even the most accurate models cannot eliminate all uncertainty. Model predictions may deviate from actual system behavior due to unmodeled dynamics, measurement noise, or novel operating conditions. The monitoring frameworks developed in Chapter 4 address this challenge by continuously comparing model predictions with actual measurements, detecting anomalies that may indicate model degradation, sensor faults, or emerging failure modes. This closed-loop integration of modeling, control, and monitoring creates a comprehensive framework for safe and efficient industrial operation, where validated models enable optimal control, while continuous monitoring ensures that models remain accurate and that control actions achieve their intended objectives.

Publication mapping.

The contributions of this chapter are documented in the following publications:

- Section 5.2 (LSTM-enhanced Deep Koopman): [118].
- Section 5.3 (physics-informed DMDc and electrolyzer RUL): [120].

CHAPTER 6

Conclusions

This dissertation set out to address a fundamental disconnect between theoretical advances in data-driven modeling, optimal control, and operational monitoring on one hand, and the practical demands of sustainable industrial and energy systems on the other. The central thesis—that reliable operation of such systems requires a unified methodological framework bridging these three domains—has been substantiated through seven interconnected contributions spanning risk-aware economic control, real-time adaptive monitoring, and interpretable data-driven modeling.

6.1 Summary of Contributions

The contributions of this dissertation are organized along three pillars, each responding to one of the research questions posed in Chapter 1.

6.1.1 Economic Feasibility Under Risk-Awareness at Scale

Addressing the first research question, this dissertation developed control architectures that embed economic objectives and risk measures directly into the optimization formulation, enabling profitable operation under uncertainty.

The greenhouse climate control framework presented in Chapter 3 demonstrated that Economic Model Predictive Control can simultaneously optimize crop yield, energy consumption, and the social cost of CO_2 emissions within a single economic objective [119]. By integrating a detailed thermodynamic model of lettuce growth with real-time weather and carbon intensity forecasts, the framework establishes the theoretical and educational foundation for multi-objective optimization under time-varying uncertainty.

Building on this foundation, the risk-aware energy management system for microgrids translates the theoretical MPC framework into a fully deployed, revenue-generating solution [117]. The novel two-stage approach—combining stochastic MPC for parameter generation with deterministic MPC for real-time execution—resolves the tension between computational tractability and uncertainty-aware decision-making. Deployment across multiple commercial and industrial facilities validated the framework's scalability and economic impact, with operation yielding profit exceeding 3000 EUR/month and a normalized yield of 159 EUR/kW/year. The modular architecture supports diverse microgrid compositions and revenue streams, including energy arbitrage, peak-shaving, and imbalance settlement.

6.1.2 Safety and Reliability in Long-Term Operation

The second research question was addressed through a suite of adaptive monitoring methods that ensure system reliability across the operational lifecycle, from real-time outlier detection to change-point detection and remaining useful life estimation.

The online inverse cumulative distribution function approach provides a computationally efficient, plug-and-play mechanism for establishing dynamic process limits that adapt to non-stationary operating conditions [121]. This univariate method was extended to the multivariate setting through the Adaptable and Interpretable Framework for Anomaly Detection (AID), which models the normal operation of dynamic systems using adaptive conditional probability distributions [122]. AID provides signal-level root cause isolation, self-supervised adaptation to non-stationarity, and dynamic operating limits that integrate directly with existing SCADA alarm handling infrastructure—capabilities that distinguish it from black-box alternatives requiring labeled training data.

For detecting fundamental shifts in system dynamics, the truncated online Dynamic Mode Decomposition with control (toDMDc) framework combines optimal rank truncation with online system identification to achieve interpretable change-point detection with bounded detection delays [123]. Validation on synthetic benchmarks, nonlinear controlled systems with input delays, and industrial battery energy storage confirmed accurate detection across diverse change types—abrupt, gradual, and drift—at computational cost of $\mathcal{O}(mr^2)$ per update, suitable for real-time deployment.

The reconstruction error of the physics-informed model is further leveraged for remaining useful life estimation in industrial electrolyzers, connecting the monitoring layer back to predictive maintenance and long-term asset management [120].

6.1.3 Credibility and Industrial Adoption

The third research question was addressed by developing interpretable, physics-informed modeling approaches that produce outputs amenable to validation against engineering intuition and regulatory requirements.

The optimal truncation introduced into online DMDc provably improves convergence properties and numerical stability, addressing a prerequisite for trustworthy deployment in time-varying industrial environments [123]. The dictionary-free Deep Koopman method based on LSTM networks overcomes the fundamental limitation of traditional Koopman operator approaches that rely on pre-defined basis functions [118]. Quantitative evaluation demonstrated more than threefold improvement in mean absolute error over extended Dynamic Mode Decomposition when the governing dynamics are unknown, while achieving only 6% worse performance when the true dynamics are incorporated into the dictionary—a favorable trade-off considering that dynamics are rarely known in practice.

Physics-Informed Dynamic Mode Decomposition with Control (piDMDc) reduces resistance prediction errors by 61% compared to baseline, provides stable and interpretable degradation tracking, and bridges the critical gap between sparse laboratory measurements and continuous operational data [120]. Applied to a 3 MW industrial chlor-alkali electrolyzer, this approach establishes a methodology for impurity-driven failure forecasting in electrochemical systems.

6.2 Discussion of Implications

The contributions of this dissertation carry implications at both the methodological and the practical level.

At the methodological level, the results demonstrate that risk-aware control, adaptive monitoring, and interpretable modeling are not independent research directions but complementary components of a unified lifecycle framework. The monitoring methods developed in Chapter 4 detect when assumptions underlying the controllers of Chapter 3 are violated, while the interpretable models of Chapter 5 explain what has changed and provide the dynamical representations on which both controllers and monitors depend. This integration reflects the lifecycle perspective advocated throughout the dissertation: industrial systems must be modeled, controlled, and monitored as a coherent whole.

At the practical level, several contributions have been validated beyond laboratory settings. The microgrid energy management system operates in commercial facilities, generating measurable economic value. The monitoring methods integrate with existing SCADA infrastructure through dynamic operating limits, requiring minimal modification to deployed systems. The physics-informed modeling approach has been applied to utility-scale electrolyzers representing significant capital investments. These deployments confirm that the proposed methods satisfy the constraints of real industrial environments—computational efficiency for real-time execution, interpretability for operator trust, and modularity for diverse system configurations.

Translating these results into concrete economic and environmental terms helps to anchor their significance. In the microgrid case study, the deployed EMS yielded revenue of the order of 3000 €/month and an annualized yield of approximately 159 €/kW, exceeding the 145 €/kW threshold quoted by investors as consistent with a 10% internal rate of return on battery storage assets [117]. In the chlor-alkali application, piDMDc reduced the mean absolute error of resistance prediction by 61% on a 3.36 MW industrial electrolyzer, supporting earlier and more targeted scheduling of cell-stack maintenance and thereby compressing both unplanned downtime and the lifecycle cost of impurity-driven degradation [120]. For the greenhouse application, the economic objective explicitly minimizes a cost composed of energy expenditure and an internal-carbon-price weighting on CO_2 emissions associated with electricity use [119]; the per-installation tonnage of CO_2-equivalent reduction is application-specific and is not quantified in this dissertation, but the optimization framing places climate impact and monetary cost on the same axis, so that improvements in one are not obtained at the silent expense of the other. The quantitative anchors reported here are case-specific and depend on market conditions, asset class, and operating regime; they are intended as evidence that the methods clear practical economic thresholds rather than as universal performance guarantees.

Beyond the deployed installations, the contributions also carry implications for the wider research and engineering community. The toDMDc, piDMDc, and Deep Koopman evaluations extend the public benchmarking surface for online change-point detection and physics-informed system identification by adding industrial reference points to communities whose benchmarks remain predominantly synthetic. The framework components—the dynamic-process-limits formulation, the AID self-supervised update rule, and the piDMDc constraint library—are deliberately designed as reusable building blocks rather than monolithic systems, which lowers the integration cost for follow-on industrial work and invites composition with monitoring, control, and identification stacks beyond the case studies considered here.

6.3 Deployment Lessons Learned

Bringing these methods into operational environments—a commercial microgrid [117], an industrial 3 MW chlor-alkali electrolyzer [120], and battery and HVAC monitoring installations [122, 121]—surfaced practical obstacles that shaped the design choices reported in the preceding chapters. Four lessons stand out.

First, on *latency budgets versus process clocks*: industrial telemetry arrives at SCADA polling rates measured in seconds to minutes, whereas inference of the proposed monitors completes in single-digit milliseconds (e.g., AID averages approximately 1.45 ms per sample, against approximately 0.05 ms for the static-threshold baselines reported in Section 4.3). Real-time integration is therefore bottlenecked by the I/O and SCADA layers rather than by the models themselves, which leaves headroom for more sophisticated estimators than a naive latency budget would admit.

Second, on *communication robustness in cloud–edge split architectures*: the microgrid deployment exposed the need for graceful degradation when the uplink to the cloud control layer is lost. The local controller falls back to the last known-good schedule and resumes coordination once telemetry resyncs, and telemetry buffering with store-and-forward semantics avoids data loss across reconnections. Network reliability, rather than algorithmic optimality, was often the binding constraint on day-to-day performance.

Third, on *fallback and safe-mode strategies*: across deployments, change-point or low-confidence flags trigger a conservative fallback regime (e.g., reverting to static thresholds, holding the previous schedule, or escalating to operator review) rather than blindly trusting a model whose validation envelope may have been left behind. The dynamic-limits and toDMDc methods were designed to coexist with the existing SCADA alarm stack rather than replace it, so that a failure of the adaptive layer cannot disable the underlying safety net.

Fourth, on *integration friction with SCADA HMI and alarm stacks*: backward compatibility with existing alarm interfaces was a binding requirement on every site. The dynamic limits were specifically designed to emit signals interchangeable with the static thresholds they replace, and the AID interpretability layer surfaces evidence in a form that operators trained on traditional rule-based alarms can act on without retraining. Adoption was driven less by raw accuracy than by how well a method slotted into existing operator workflows.

Together, these practical concerns pushed the dissertation's design choices toward modular, interpretable, and low-latency components that degrade gracefully and integrate with rather than displace the incumbent control and alarm infrastructure.

6.4 Limitations

Several limitations of this work are acknowledged.

The risk-aware microgrid controller depends on the accuracy of probabilistic forecasts for renewable generation, load demand, and market prices. While the two-stage MPC formulation mitigates computational burden, the quality of control decisions remains bounded by forecast quality. Furthermore, the cloud-based architecture, though equipped with a local fallback mode, introduces dependence on network connectivity that may reduce optimality during extended outages.

The monitoring methods assume that the underlying processes can be adequately characterized by multivariate Gaussian distributions (AID) or low-rank linear approximations (toDMDc). Strongly non-Gaussian processes or systems exhibiting high-dimensional nonlinear mode interactions may challenge these assumptions. The plug-and-play nature of the dynamic process limits, while advantageous for deployment, may yield elevated false positive rates when applied to systems without prior calibration of time constants.

For the modeling contributions, direct remaining useful life validation of the piDMDc framework awaits comprehensive maintenance records from the industrial partner. More narrowly, the LSTM-enhanced Deep Koopman method of Section 5.2 is validated on a single-input single-output simulated two-tank benchmark with a stationary input delay. Extension to multi-input multi-output targets, to industrial-scale assets, and to operating regimes with non-stationary or state-dependent delays is identified as future work and is required before generalizable claims about Deep Koopman performance on industrial systems can be supported.

The unifying modeling–control–monitoring lifecycle around which this dissertation is organized is advanced as a methodological argument and is validated pillar by pillar across the case studies of Chapter 3–Chapter 5, but no single application closes the loop end-to-end within the scope of the dissertation itself: the controllers deployed in Chapter 3 rely on mechanistic or physics-derived linear models rather than on the data-driven models identified in Chapter 5, and the monitoring-triggered re-identification pathway is described rather than demonstrated in a fully integrated deployment. The bridging discussion in Section 5.4 and the corresponding entry in the

Future Research Directions below identify the closed-loop integration experiment that would settle this gap.

6.5 Future Research Directions

The contributions of this dissertation suggest several avenues for future research, organized around the overarching goal of tightening the integration across the modeling–control–monitoring lifecycle.

A natural extension is the development of closed-loop architectures in which monitoring signals trigger automatic model re-identification and controller reconfiguration. The toDMDc framework already provides real-time detection of dynamics changes; coupling this detection mechanism with the piDMDc or Deep Koopman identification methods would enable autonomous model updates without operator intervention. Integrating these updated models into the risk-aware MPC layer would create a self-correcting control system capable of responding to degradation, environmental shifts, and regime changes.

The interpretable modeling methods can be extended to accommodate more complex physical constraints. Refinement of the piDMDc framework to capture nonlinear cell behavior—particularly voltage outruns during accelerated degradation—would improve remaining useful life predictions. Extending the Deep Koopman approach to higher-dimensional systems and exploring its direct integration as a prediction model within MPC formulations would bridge the gap between identification and control.

The monitoring layer would benefit from advances in non-parametric distributional models to relax the Gaussian assumption underlying AID, improved change-point detection mechanisms with reduced latency for high-dimensional data, and formal integration of detection statistics with risk measures used in the control layer. Establishing theoretical guarantees on the end-to-end performance of the combined monitoring–control pipeline represents a fundamental challenge worthy of sustained investigation.

The modular software architecture developed for the microgrid EMS provides a template for deploying the complete lifecycle framework as a service across heterogeneous industrial systems. Extending this architecture to incorporate the monitoring and modeling components would realize the vision of an integrated, commercially viable platform for sustainable industrial automation.

Beyond these broad directions, one concrete experiment crystallizes the agenda above and merits explicit formulation: the closed-loop integration of the modeling, control, and monitoring layers as a single deployable artifact. A natural next step that closes the gap identified in Section 5.4 and in the Limitations above is a closed-loop demonstration in which a piDMDc model of the chlor-alkali electrolyzer, or a Deep Koopman model of the two-tank target, is substituted into the EMPC of Section 3.2 (resp. the risk-aware EMS of Section 3.3). Such an experiment would settle three open questions: closed-loop stability under model–plant mismatch, recursive feasibility under the learned dynamics, and the value of the structural priors of piDMDc in long-horizon constraint satisfaction.

6.6 Code and Data Availability

In support of reproducibility, the artifacts underlying this dissertation are released under the following terms.

Open code and benchmarks. The synthetic step-change suite, the simulated two-tank benchmark, and the experimental scripts for the SKAB and CATS public datasets used in Chapter 4 are released through the umbrella repository at `https://github.com/MarekWadinger`, with the per-publication repositories linked from the abstracts of the corresponding papers [123, 122, 121, 118, 120, 119].

Industrial datasets. The commercial microgrid telemetry underpinning Section 3.3 and the 3.36 MW chlor-alkali electrolyzer process data underpinning the piDMDc validation in Section 5.3 are subject to operator non-disclosure constraints and cannot be redistributed in raw form. The source publications document the feature schemas, sampling rates, and preprocessing pipelines; where possible, synthetic surrogates derived from the same first- and second-order statistics are released alongside the code where applicable so that the methodology can be reproduced on representative-but-non-identifying data.

6.7 Closing Remarks

This dissertation has argued that sustainable industrial automation demands more than advances in any single domain. Optimal controllers require accurate models; models degrade as systems evolve; and monitoring must detect when models and controllers are no longer valid. By contributing methods across all three domains—interpretable modeling, risk-aware control, and adaptive monitoring—and demonstrating their deployment in real industrial settings, this work establishes a methodological foundation for systems that are not merely optimized at commissioning but remain reliable, profitable, and safe throughout their operational lifetime.

Appendix A

Resumé

Motivácia

Priemyselné a energetické systémy sa nachádzajú v bode zlomu. Prechod k udržateľnosti, integrácia obnoviteľných zdrojov a rastúca prepojenosť výrobných prevádzok vytvárajú procesy, ktoré sú zložitejšie, citlivejšie na poruchy a ekonomicky náročnejšie než kedykoľvek predtým. Súčasne stúpa cena chyby: nesprávne rozhodnutie regulátora môže znamenať stratu miliónov, výpadok kritickej infraštruktúry alebo nezvratné poškodenie zariadenia v hodnote stoviek tisíc eur. V kontexte energetickej transformácie a klimatickej krízy už nestačí prevádzkovať priemyselné systémy iba spoľahlivo — musia byť spoľahlivé, ekonomicky efektívne a environmentálne zodpovedné súčasne.

Napriek tomu, že priemyselné prevádzky generujú nebývalé množstvo dát a teoretický pokrok v oblastiach dátovo riadeného modelovania, optimálneho riadenia a monitorovania je značný, medzi metódami publikovanými v odbornej literatúre a metódami skutočne nasaditeľnými v priemyselnej praxi pretrváva výrazná priepasť. Pokročilé regulátory zlyhávajú, keď sa predpoklady, na ktorých sú postavené, prestanú zhodovať s realitou starnúcich zariadení a meniacich sa prevádzkových podmienok. Monitorovacie systémy zaplavujú obsluhu falošnými poplachmi alebo, naopak, prehliadnu kritické zmeny dynamiky. Modely strojového učenia dosahujú vysokú presnosť, ale ich neinterpretovateľnosť bráni ich akceptácii v bezpečnostne kritických aplikáciách. Výskumné komunity okolo riadenia, monitorovania a modelovania pritom pracujú prevažne oddelene, hoci v reálnej prevádzke tvoria neoddeliteľné komponenty jediného životného cyklu.

Predložená dizertačná práca tvrdí, že dôveryhodná autonómia priemyselných systémov si vyžaduje jednotný metodický rámec, ktorý interpretovateľné dátovo riadené modely, ekonomické riadenie s vedomím rizika a adaptívne monitorovanie v reálnom čase chápe ako vzájomne sa podporujúce zložky jediného životného cyklu — a ktorý je preukázateľne nasaditeľný na existujúcej priemyselnej infraštruktúre bez nutnosti rozsiahlych zásahov.

Dosiahnuté výsledky

Práca prináša sedem vzájomne previazaných príspevkov, ktoré boli validované nie iba simulačne, ale priamo na priemyselne relevantných prípadových štúdiách. Nasledujúce výsledky predstavujú merateľný dopad práce.

Ekonomické riadenie s vedomím rizika. Vrstvený systém riadenia energie pre mikrosiete, kombinujúci stochastické prediktívne riadenie pre generovanie parametrov rizikovej politiky s deterministickým prediktívnym riadením pre rozhodovanie v reálnom čase, bol nasadený v komerčných a priemyselných prevádzkach. Reálna prevádzka dosahuje zisk presahujúci **3000 EUR/mesiac** a anualizovaný výnos **159 EUR/kW** na batériových úložiskách, čo prekračuje prah zodpovedajúci **10 % internej miere návratnosti**. Nelineárne ekonomické prediktívne riadenie klímy v skleníku v rámci jednej účelovej funkcie vyvažuje produkciu plodín, spotrebu energie a sociálne náklady emisií CO_2 a slúži zároveň ako vzdelávacia platforma pre štúdium kompromisov medzi ziskovosťou a environmentálnou stopou.

Interpretovateľné dátovo riadené modelovanie. Hlboký Koopmanov model rozšírený o vrstvy LSTM sa učí lineárne reprezentácie nelineárnych systémov so vstupnými oneskoreniami bez potreby vopred definovaného slovníka pozorovateľných funkcií. V porovnaní s rozšírenou dynamickou modálnou dekompozíciou dosahuje viac než **trojnásobné zlepšenie** strednej absolútnej chyby v prípadoch, keď riadiace rovnice nie sú známe, pri zachovaní iba **6 % penalty** v prípadoch, keď známe sú. Fyzikálne informovaná dynamická modálna dekompozícia s riadením (piDMDc), overená na priemyselnom chlóralkalickom elektrolyzéri s výkonom **3 MW** v Taliansku, znižuje chybu predikcie elektrického odporu o **61 %** oproti základnej línii a umožňuje sledovanie degradácie vyvolanej nečistotami pre potreby prediktívnej údržby.

Adaptívne monitorovanie v reálnom čase. Metóda založená na online inverznej kumulatívnej distribučnej funkcii poskytuje dynamické prevádzkové limity, ktoré sa prispôsobujú nestacionárnym podmienkam bez potreby anotovaných dát či expertného ladenia. Rámec AID (Adaptable and Interpretable Framework for Anomaly Detection) ju rozširuje do mnohorozmerného prostredia a umožňuje izoláciu koreňových príčin až na úrovni jednotlivých signálov. Metóda toDMDc (truncated online Dynamic Mode

Decomposition with Control) spája optimálnu redukciu hodnosti s online identifikáciou systému a dosahuje interpretovateľnú detekciu zmenových bodov s ohraničeným oneskorením pri výpočtovej zložitosti $\mathcal{O}(mr^2)$ na jednu aktualizáciu. Všetky monitorovacie metódy sa integrujú priamo s existujúcimi mechanizmami spracovania alarmov v SCADA systémoch a vyžadujú minimálne zásahy do nasadenej infraštruktúry.

Záver

Spoločne tieto výsledky preukazujú, že interpretovateľné modelovanie, ekonomické riadenie s vedomím rizika a adaptívne monitorovanie nepredstavujú oddelené disciplíny, ale vzájomne sa podporujúce komponenty jediného životného cyklu priemyselných automatizačných systémov: monitorovanie deteguje, kedy boli porušené predpoklady, na ktorých sú postavené regulátory; interpretovateľné modely vysvetľujú, čo a prečo sa zmenilo, a poskytujú dynamické reprezentácie, na ktorých sú regulátory aj monitorovacie metódy závislé; a riadenie s vedomím rizika premieňa toto povedomie o životnom cykle systému na merateľnú ekonomickú a environmentálnu hodnotu. Validácia na komerčných mikrosieťach, batériových úložiskách, skleníkových prevádzkach a chlóralkalickom elektrolyzéri v meradle megawattov ukazuje, že tento rámec je nielen teoreticky koherentný, ale aj prakticky nasaditeľný, čím otvára cestu k širšiemu prijatiu dôveryhodnej automatizácie v priemyselnej a energetickej praxi.

Appendix B

Curriculum Vitae

Marek Wadinger

PhD Candidate, Data-Driven Control for Industrial Systems
Institute of Information Engineering, Automation, and Mathematics,
Faculty of Chemical and Food Technology, Slovak University of Technology in Bratislava

email: marek.wadinger@stuba.sk
linkedin: linkedin.com/in/marek-wadinger
github: github.com/MarekWadinger
ORCID: 0000-0002-0167-6331

Connecting academic research to real-world practice. My work focuses on anomaly and fault detection, online learning, and data-driven decision-making for industrial and energy systems.

Education

2022 – 2026 **PhD, Process Control**
Faculty of Chemical and Food Technology, Slovak University of Technology in Bratislava
Dissertation: *Safety-First Control and Monitoring for Industrial Systems*

Jan – Jun 2024 **PhD Research Stay — Information Technology**
The University of Osaka
Funded by the National Scholarship of the Slovak Republic

2020 – 2022 **MSc, Automation & Information Engineering in Chemistry and Food Industry *(graduated with honors)***
Faculty of Chemical and Food Technology, Slovak University of Technology in Bratislava
Thesis: *Design, Implementation and Optimization of a Classification Algorithm for Identifying Small Molecules Using Annotated Spectral Trees*

2020 – 2021 **MSc Exchange**
Alma Mater Studiorum — Università di Bologna

2017 – 2020 **BSc, Automation, Information Engineering & Management in Chemistry and Food Industry**
Faculty of Chemical and Food Technology, Slovak University of Technology in Bratislava
Thesis: *Predictive Data Analytics Based on Machine Learning*

Research Focus

Anomaly and fault detection; online and streaming machine learning; dynamic mode decomposition; model predictive control; reliability engineering; green hydrogen and microgrids.

Selected Publications

Dec 2025 **Predictive Risk-Aware Control for Microgrids: Operation of a Revenue-Generating Energy Management System.**
With T. Ábelová, M. Kvasnica. *Sustainable Energy, Grids and Networks*, vol. 44, art. 102028.

Dec 2025 **Physics-Informed Dynamic Mode Decomposition with Control for Predictive Maintenance in Chlor-Alkali Electrolysis.**
With L. Galčíková, L. Razzolini, M. Kvasnica, E. Brunazzi, M. Vaccari. *IEEE 64th Conference on Decision and Control*, Rio de Janeiro, pp. 450–455.

Nov 2025 **Carbon Neutral Greenhouse: Economic Model Predictive Control Framework for Education.**
With R. Fáber, E. Pavlovičová, R. Paulen. *European Journal of Control*, vol. 86, art. 101297.

Jun 2025 **Deep Dictionary-Free Method for Identifying Linear Model of Nonlinear System with Input Delay.**
With P. Valábek, M. Kvasnica, M. Kláučo. *25th International Conference on Process Control*, IEEE, pp. 1–6.

Aug 2024 **Change-Point Detection in Industrial Data Streams Based on Online Dynamic Mode Decomposition with Control.**
With M. Kvasnica, Y. Kawahara. Preprint, arXiv:2407.05976.

Jul 2024 **Adaptable and Interpretable Framework for Anomaly Detection in SCADA-Based Industrial Systems.**
With M. Kvasnica. *Expert Systems with Applications*, vol. 246, art. 123200.

Jun 2023 **Real-Time Outlier Detection with Dynamic Process Limits.**
With M. Kvasnica. *Proc. 24th International Conference on Process Control*, IEEE.

Selected Projects

Open source **adaptive-interpretable-ad**
Self-supervised, adaptive, and interpretable anomaly detection with dynamic operating limits. Python.
Companion code to the 2024 *Expert Systems with Applications* paper.

Open source **odmd-subid-cpd**
Change-point detection using online dynamic mode decomposition with control. Python.
Companion code to the 2024 preprint.

Open source **ecompc-greenhouse-platform**
Interactive web-based platform for education in nonlinear economic model predictive control. Python.
Companion code to the 2025 *European Journal of Control* paper.

Open source **tf2ss**
Pure-Python MIMO transfer-function to state-space conversion.
Easy-to-install, MATLAB-consistent alternative to SLYCOT — no Fortran compiler required.

Awards and Honors

2025 **Dean's Praise — doctoral studies**
Faculty of Chemical and Food Technology, Slovak University of Technology in Bratislava
For activities for the benefit of the Faculty in academic year 2024/2025.

2022 **Best Diploma Thesis Award**
Siemens Slovensko

2022 **Dean's Praise — graduate studies**
Faculty of Chemical and Food Technology, Slovak University of Technology in Bratislava
For results in research, technical work, and service to the faculty.

2021 **2nd Place — 23rd Slovak Student Scientific Conference**
Section: Process Control Supported by Information Technologies

2019 – 2021 **Dean's Praise (three consecutive years)**
Faculty of Chemical and Food Technology, Slovak University of Technology in Bratislava
For service to the faculty.

2019	**EBEC Central — 3rd Place, Case Study** European regional round, Budapest. Supervised by Knorr-Bremse.
2019	**EBEC Bratislava — 1st Place, Case Study** Local round, Bratislava.

Leadership and Service

2021 – 2022	**Member, Quality Assurance Internal System Board** Faculty of Chemical and Food Technology, Slovak University of Technology in Bratislava
2019 – 2022	**Student Member, Academic Senate** Faculty of Chemical and Food Technology, Slovak University of Technology in Bratislava
2019 – 2021	**Chairman, CHEM Student Society** Faculty of Chemical and Food Technology, Slovak University of Technology in Bratislava
2020	**Fundraising Team Lead** St. Vincent de Paul Homeless Shelter. Led a team of nine volunteers.
2018	**B2C Team Lead** AIESEC, Comenius University in Bratislava.

Bibliography

[1] S. A. G. K. Abadi, S. I. Habibi, T. Khalili, and A. Bidram. “A model predictive control strategy for performance improvement of hybrid energy storage systems in DC microgrids”. In: *IEEE Access* 10 (2022), pp. 25400–25421.

[2] S. Ahmad, A. Lavin, S. Purdy, and Z. Agha. “Unsupervised real-time anomaly detection for streaming data”. In: *Neurocomputing* 262 (2017). Online Real-Time Learning Strategies for Data Streams, pp. 134–147. ISSN: 0925-2312. DOI: `https://doi.org/10.1016/j.neucom.2017.04.070`. URL: `https://www.sciencedirect.com/science/article/pii/S0925231217309864`.

[3] M. A. Alarcón, R. G. Alarcón, A. H. González, and A. Ferramosca. “Economic model predictive control for energy management of a microgrid connected to the main electrical grid”. In: *Journal of Process Control* 117 (2022), pp. 40–51.

[4] K. Amarasinghe, K. Kenney, and M. Manic. “Toward Explainable Deep Neural Network Based Anomaly Detection”. In: *2018 11th International Conference on Human System Interaction (HSI)*. July 2018, pp. 311–317. DOI: `10.1109/HSI.2018.8430788`.

[5] M. Amoretti, R. Pecori, Y. Protskaya, L. Veltri, and F. Zanichelli. “A Scalable and Secure Publish/Subscribe-Based Framework for Industrial IoT”. In: *IEEE Transactions on Industrial Informatics* 17.6 (June 2021), pp. 3815–3825. ISSN: 1941-0050. DOI: `10.1109/TII.2020.3017227`.

[6] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl. “CasADi – A software framework for nonlinear optimization and optimal control”. In: *Mathematical Programming Computation* 11.1 (2019), pp. 1–36. DOI: `10.1007/s12532-018-0139-4`.

[7] P. J. Baddoo, B. Herrmann, B. J. McKeon, J. N. Kutz, and S. L. Brunton. “Physics-informed dynamic mode decomposition”. In: *Proc. R. Soc. A.*

479.20220576 (2023). DOI: 10.1098/rspa.2022.0576. URL: http://doi.org/10.1098/rspa.2022.0576.

[8] M. Bahr, A. Gusak, S. Stypka, and B. Oberschachtsiek. "Artificial Neural Networks for Aging Simulation of Electrolysis Stacks". In: *Chemie Ingenieur Technik* 92.10 (2020), pp. 1610–1617. DOI: https://doi.org/10.1002/cite.202000089. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cite.202000089. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/cite.202000089.

[9] X. Bao, L. Chen, J. Zhong, D. Wu, and Y. Zheng. "A self-supervised contrastive change point detection method for industrial time series". In: *Engineering Applications of Artificial Intelligence* 133 (2024), p. 108217. ISSN: 0952-1976. DOI: https://doi.org/10.1016/j.engappai.2024.108217. URL: https://www.sciencedirect.com/science/article/pii/S0952197624003750.

[10] N. Barbosa Roa, L. Travé-Massuyès, and V. H. Grisales-Palacio. "DyClee: Dynamic clustering for tracking evolving environments". In: *Pattern Recognition* 94 (2019), pp. 162–186. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2019.05.024. URL: https://www.sciencedirect.com/science/article/pii/S0031320319301992.

[11] L. Batista, L. Da Costa, S. Berriah, and H. Lademann. "A Multi-Expert System for chlorine electrolyzer monitoring". In: *Expert Systems with Applications* 40.8 (2013), pp. 3128–3136. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2012.12.094. URL: https://www.sciencedirect.com/science/article/pii/S095741741201336X.

[12] H. H. Bosman, G. Iacca, A. Tejada, H. J. Wörtche, and A. Liotta. "Ensembles of incremental learners to detect anomalies in ad hoc sensor networks". In: *Ad Hoc Networks* 35 (2015). Special Issue on Big Data Inspired Data Sensing, Processing and Networking Technologies, pp. 14–36. ISSN: 1570-8705. DOI: https://doi.org/10.1016/j.adhoc.2015.07.013. URL: https://www.sciencedirect.com/science/article/pii/S1570870515001481.

[13] L. C. Brito, G. A. Susto, J. N. Brito, and M. A. V. Duarte. "Fault Diagnosis using eXplainable AI: A transfer learning-based approach for rotating machinery exploiting augmented synthetic data". In: *Expert Systems with Applications* 232 (2023), p. 120860. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2023.120860. URL: https://www.sciencedirect.com/science/article/pii/S0957417423013623.

[14] S. L. Brunton, B. W. Brunton, J. L. Proctor, E. Kaiser, and J. N. Kutz. "Chaos as an intermittently forced linear system". In: *Nature Communications* 8 (2017), p. 15595. DOI: 10.1038/ncomms15595.

[15] M. Carletti, C. Masiero, A. Beghi, and G. A. Susto. “Explainable Machine Learning in Industry 4.0: Evaluating Feature Importance in Anomaly Detection to Enable Root Cause Analysis”. In: *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC).* Oct. 2019, pp. 21–26. DOI: 10.1109/SMC.2019.8913901.

[16] V. Chandola, A. Banerjee, and V. Kumar. “Anomaly Detection: A Survey”. In: *ACM Comput. Surv.* 41.3 (July 2009). ISSN: 0360-0300. DOI: 10.1145/1541880.1541882. URL: https://doi.org/10.1145/1541880.1541882.

[17] M. A. Chao, C. S. Kulkarni, K. Goebel, and O. Fink. “Fusing physics-based and deep learning models for prognostics”. In: *Reliability Engineering & System Safety* 217 (2022), p. 107961. DOI: 10.1016/j.ress.2021.107961.

[18] A. A. Cook, G. Mısırlı, and Z. Fan. “Anomaly Detection for IoT Time-Series Data: A Survey”. In: *IEEE Internet of Things Journal* 7.7 (2020), pp. 6481–6494. DOI: 10.1109/JIOT.2019.2958185.

[19] L. C. da Costa Jr., F. S. Thomé, J. D. Garcia, and M. V. F. Pereira. “Reliability-Constrained Power System Expansion Planning: A Stochastic Risk-Averse Optimization Approach”. In: *IEEE Transactions on Power Systems* 36.1 (2021), pp. 97–106. DOI: 10.1109/TPWRS.2020.3007974.

[20] Y. Dai, S. Sun, and L. Che. “Improved DBSCAN-based Data Anomaly Detection Approach for Battery Energy Storage Stations”. In: *Journal of Physics: Conference Series.* Vol. 2351. July 2022, p. 012025. DOI: 10.1088/1742-6596/2351/1/012025.

[21] T. De Ryck, M. De Vos, and A. Bertrand. “Change Point Detection in Time Series Data Using Autoencoders With a Time-Invariant Representation”. In: *IEEE Transactions on Signal Processing* 69 (2021), pp. 3513–3524. ISSN: 1941-0476. DOI: 10.1109/TSP.2021.3087031.

[22] S. Deldari, D. V. Smith, H. Xue, and F. D. Salim. “Time Series Change Point Detection with Self-Supervised Contrastive Predictive Coding”. In: *Proceedings of the Web Conference 2021.* WWW ’21. Ljubljana, Slovenia: Association for Computing Machinery, 2021, pp. 3124–3135. ISBN: 9781450383127. DOI: 10.1145/3442381.3449903. URL: https://doi.org/10.1145/3442381.3449903.

[23] J. Drgona, A. Tuor, J. Koch, M. Shapiro, and D. Vrabie. *NeuroMANCER: Neural Modules with Adaptive Nonlinear Constraints and Efficient Regularizations.* 2023. URL: https://github.com/pnnl/neuromancer.

[24] X. Du, J. Chen, J. Yu, S. Li, and Q. Tan. "Generative adversarial nets for unsupervised outlier detection". In: *Expert Systems with Applications* 236 (2024), p. 121161. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2023.121161. URL: https://www.sciencedirect.com/science/article/pii/S0957417423016639.

[25] E. Esche, T. Talis, J. Weigert, G. Brand Rihm, B. You, C. Hoffmann, and J.-U. Repke. "Semi-supervised learning for data-driven soft-sensing of biological and chemical processes". In: *Chemical Engineering Science* 251 (2022), p. 117459. ISSN: 0009-2509. DOI: https://doi.org/10.1016/j.ces.2022.117459. URL: https://www.sciencedirect.com/science/article/pii/S0009250922000434.

[26] C. Fan, Y. Sun, Y. Zhao, M. Song, and J. Wang. "Deep learning-based feature engineering methods for improved building energy prediction". In: *Applied Energy* 240 (2019), pp. 35–45. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2019.02.052. URL: https://www.sciencedirect.com/science/article/pii/S0306261919303496.

[27] H. Fan, Q. Yuan, and H. Cheng. "Multi-objective stochastic optimal operation of a grid-connected microgrid considering an energy storage system". In: *Applied Sciences* 8.12 (2018), p. 2560.

[28] Y. Fathy, P. Barnaghi, and R. Tafazolli. "An Online Adaptive Algorithm for Change Detection in Streaming Sensory Data". In: *IEEE Systems Journal* 13.3 (Sept. 2019), pp. 2688–2699. ISSN: 1937-9234. DOI: 10.1109/JSYST.2018.2876461.

[29] O. Fink et al. "From physics to machine learning and back: Part I — Learning with inductive biases in prognostics and health management (PHM)". In: *Reliability Engineering & System Safety* 271 (2026), p. 112213. DOI: 10.1016/j.ress.2026.112213.

[30] G. Fleith, F. Harrou, A. Dairi, and Y. Sun. *Controlled Anomalies Time Series (CATS) Dataset.* https://zenodo.org/records/8338435. Zenodo. 2023.

[31] M. Fochesato, C. Peter, L. Morandi, and J. Lygeros. "Peak shaving with hydrogen energy storage: From stochastic control to experiments on a 4 MWh facility". In: *Applied Energy* 376 (2024), p. 123965.

[32] J. Gama, I. Žliobaitundefined, A. Bifet, M. Pechenizkiy, and A. Bouchachia. "A Survey on Concept Drift Adaptation". In: *ACM Comput. Surv.* 46.4 (Mar. 2014). ISSN: 0360-0300. DOI: 10.1145/2523813. URL: https://doi.org/10.1145/2523813.

[33] D. Garreau and S. Arlot. "Consistent change-point detection with kernels". In: *Electronic Journal of Statistics* 12.2 (2018), pp. 4440–4486. DOI: 10.1214/18-EJS1513. URL: https://doi.org/10.1214/18-EJS1513.

[34] M. Gavish and D. L. Donoho. "The Optimal Hard Threshold for Singular Values is $4/\sqrt{3}$". In: *IEEE Transactions on Information Theory* 60.8 (Aug. 2014), pp. 5040–5053. ISSN: 1557-9654. DOI: 10.1109/TIT.2014.2323359.

[35] P. Ge, D. Tang, Y. Yuan, J. M. Guerrero, and E. Zio. "A hierarchical multi-objective co-optimization framework for sizing and energy management of coupled hydrogen-electricity energy storage systems at ports". In: *Applied Energy* 384 (2025), p. 125451. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2025.125451. URL: https://www.sciencedirect.com/science/article/pii/S0306261925001813.

[36] A. Genz. "Numerical Computation Of Multivariate Normal Probabilities". In: *Journal of Computational and Graphical Statistics* 1 (May 2000). DOI: 10.1080/10618600.1992.10477010.

[37] R. Ghosh and M. McAfee. "Koopman operator theory and dynamic mode decomposition in data-driven science and engineering: A comprehensive review". In: *Mathematical Modelling and Numerical Simulation with Applications* 4.4 (2024), pp. 562–594. DOI: 10.53391/mmnsa.1512698.

[38] M. Gupta, R. Wadhvani, and A. Rasool. "Real-time Change-Point Detection: A deep neural network-based adaptive approach for detecting changes in multivariate time series data". In: *Expert Systems with Applications* 209 (2022), p. 118260. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2022.118260. URL: https://www.sciencedirect.com/science/article/pii/S0957417422014026.

[39] V. Hamdipoor, H. N. Nguyen, B. Mekhaldi, J. Parra, J. Badosa, and F. C. Obaldia. "Experimental validation of scenario-based stochastic model predictive control of nanogrids". In: *Control Engineering Practice* 157 (2025), p. 106249.

[40] L. Herre. "Impact of imbalance settlement system design on risk-averse energy storage". In: *2020 17th International Conference on the European Energy Market (EEM)*. IEEE. 2020, pp. 1–6.

[41] L. Herre. "Risk-Averse Aggregator of Controllable Loads as Virtual Battery Providing Multiple Services". In: *2020 17th International Conference on the European Energy Market (EEM)*. IEEE. 2020, pp. 1–6.

[42] S. Hirabaru, T. Matsuda, Y. Hirota, H. Izumikawa, H. Hanano, C. Ono, and T. Takine. “A Change-Point Detection Scheme Based on Subspace Tracking for Mobile Access Traffic”. In: *2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*. Dec. 2016, pp. 818–823. DOI: 10.1109/HPCC-SmartCity-DSS.2016.0118.

[43] Y. Hu, M. Armada, and M. J. Sánchez. “Potential utilization of battery energy storage systems (BESS) in the major European electricity markets”. In: *Applied Energy* 322 (2022), p. 119512.

[44] A. Incorvaia, E. Brunazzi, M. Vaccari, and L. Razzolini. “Digital twin di un elettrolizzatore a membrana per la produzione di cloro-alcalini: sviluppo di un modello data-driven”. Italian. Tesi di laurea magistrale. Università di Pisa, Nov. 2024. URL: https://etd.adm.unipi.it/theses/available/etd-11112024-144554/ (visited on 11/28/2024).

[45] A. Iovine, T. Rigaut, G. Damm, E. De Santis, and M. D. Di Benedetto. “Power management for a DC MicroGrid integrating renewables and storages”. In: *Control Engineering Practice* 85 (2019), pp. 59–79. ISSN: 0967-0661. DOI: https://doi.org/10.1016/j.conengprac.2019.01.009. URL: https://www.sciencedirect.com/science/article/pii/S0967066119300048.

[46] I. D. Katser and V. O. Kozitsin. *Skoltech Anomaly Benchmark (SKAB)*. https://www.kaggle.com/dsv/1693952. 2020. DOI: 10.34740/KAGGLE/DSV/1693952.

[47] Y. Kawahara, T. Yairi, and K. Machida. “Change-Point Detection in Time-Series Data Based on Subspace Identification”. In: *Seventh IEEE International Conference on Data Mining (ICDM 2007)*. Oct. 2007, pp. 559–564. DOI: 10.1109/ICDM.2007.78.

[48] A. Kejariwal. *Introducing practical and robust anomaly detection in a time series*. 2015. URL: https://blog.twitter.com/engineering/en_us/a/2015/introducing-practical-and-robust-anomaly-detection-in-a-time-series.

[49] R. Khodabakhsh and S. Sirouspour. “Optimal control of energy storage in a microgrid by minimizing conditional value-at-risk”. In: *IEEE Transactions on Sustainable Energy* 7.3 (2016), pp. 1264–1273.

[50] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, and A. Anandkumar. “Neural Operator: Learning Maps Between Function Spaces With Applications to PDEs”. In: *Journal of Machine Learning Research* 24.89 (2023), pp. 1–97. URL: http://jmlr.org/papers/v24/21-1524.html.

[51] N. Laptev, S. Amizadeh, and I. Flint. “Generic and Scalable Framework for Automated Time-Series Anomaly Detection”. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’15. Sydney, NSW, Australia: Association for Computing Machinery, 2015, pp. 1939–1947. ISBN: 9781450336642. DOI: 10.1145/2783258.2788611. URL: https://doi.org/10.1145/2783258.2788611.

[52] H. Lee, J. Gu, B. Lee, H.-S. Cho, and H. Lim. “Prognostics and health management of alkaline water electrolyzer: Techno-economic analysis considering replacement moment”. In: *Energy and AI* 13 (2023), p. 100251. ISSN: 2666-5468. DOI: https://doi.org/10.1016/j.egyai.2023.100251. URL: https://www.sciencedirect.com/science/article/pii/S266654682300023X.

[53] L.-J. Li, T.-T. Dong, S. Zhang, X.-X. Zhang, and S.-P. Yang. “Time-delay identification in dynamic processes with disturbance via correlation analysis”. In: *Control Engineering Practice* 62 (2017), pp. 92–101. ISSN: 0967-0661. DOI: https://doi.org/10.1016/j.conengprac.2017.03.007. URL: https://www.sciencedirect.com/science/article/pii/S0967066117300655.

[54] J. Li and Z. Liu. “Attribute-weighted outlier detection for mixed data based on parallel mutual information”. In: *Expert Systems with Applications* 236 (2024), p. 121304. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2023.121304. URL: https://www.sciencedirect.com/science/article/pii/S0957417423018067.

[55] Z. Li, C. Zang, P. Zeng, H. Yu, and H. Li. “Two-stage stochastic programming based model predictive control strategy for microgrid energy management under uncertainties”. In: *2016 International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*. IEEE. 2016, pp. 1–6.

[56] R. B. Lima and P. R. Barros. “Identification of Time-Delay Systems: a State-Space Realization Approach”. In: *IFAC-PapersOnLine* 48.8 (2015). 9th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2015, pp. 254–259. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2015.08.190. URL: https://www.sciencedirect.com/science/article/pii/S2405896315010575.

[57] J. Liu, D. Yang, K. Zhang, H. Gao, and J. Li. “Anomaly and change point detection for time series with concept drift”. In: *World Wide Web* 26.5 (Sept. 2023), pp. 3229–3252. ISSN: 1573-1413. DOI: 10.1007/s11280-023-01181-z. URL: https://doi.org/10.1007/s11280-023-01181-z.

[58] X. Lu, W. Gang, S. Cai, and Z. Tu. “Energy management strategy for a novel multi-stack integrated hydrogen energy storage system based on hybrid rules and optimization”. In: *Applied Energy* 381 (2025), p. 125189. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2024.125189. URL: https://www.sciencedirect.com/science/article/pii/S030626192402573X.

[59] B. Lusch, J. N. Kutz, and S. L. Brunton. “Deep learning for universal linear embeddings of nonlinear dynamics”. In: *Nature Communications* 9.1 (Nov. 2018), p. 4950. ISSN: 2041-1723. DOI: 10.1038/s41467-018-07210-0. URL: https://doi.org/10.1038/s41467-018-07210-0.

[60] M. Markou and S. Singh. “Novelty detection: a review—part 1: statistical approaches”. In: *Signal Processing* 83.12 (2003), pp. 2481–2497. DOI: 10.1016/j.sigpro.2003.07.018.

[61] E. Mashonjowa, F. Ronsse, J. Pieters, and R. Lemeur. “Modelling heat and mass transfer in a naturally ventilated greenhouse in Zimbabwe”. In: *Communications in agricultural and applied biological sciences* 72 (Feb. 2007), pp. 239–43.

[62] S. Matrone, A. Pozzi, E. Ogliari, and S. Leva. “Deep Learning-Based Predictive Control for Optimal Battery Management in Microgrids”. In: *IEEE Access* (2024).

[63] A. Mauroy and J. Goncalves. “Linear identification of nonlinear systems: A lifting technique based on the Koopman operator”. In: *2016 IEEE 55th Conference on Decision and Control (CDC)*. Dec. 2016, pp. 6500–6505. DOI: 10.1109/CDC.2016.7799269.

[64] I. Melnyk, A. Banerjee, B. Matthews, and N. Oza. “Semi-Markov Switching Vector Autoregressive Model-Based Anomaly Detection in Aviation Systems”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’16. San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 1065–1074. ISBN: 9781450342322. DOI: 10.1145/2939672.2939789. URL: https://doi.org/10.1145/2939672.2939789.

[65] C. Meng, S. Griesemer, D. Cao, S. Seo, and Y. Liu. “When physics meets machine learning: a survey of physics-informed machine learning”. In: *Machine Learning for Computational Science and Engineering* 1.1 (2025), p. 20. ISSN: 3005-1436. DOI: 10.1007/s44379-025-00016-0. URL: https://doi.org/10.1007/s44379-025-00016-0.

[66] I. Mezić. “Spectral Properties of Dynamical Systems, Model Reduction and Decompositions”. In: *Nonlinear Dynamics* 41.1 (Aug. 2005), pp. 309–325. ISSN: 1573-269X. DOI: 10.1007/s11071-005-2824-x. URL: https://doi.org/10.1007/s11071-005-2824-x.

[67] I. Mezić and A. Banaszuk. "Comparison of systems with complex behavior". In: *Physica D: Nonlinear Phenomena* 197.1 (2004), pp. 101–133. ISSN: 0167-2789. DOI: `https://doi.org/10.1016/j.physd.2004.06.015`. URL: `https://www.sciencedirect.com/science/article/pii/S0167278904002507`.

[68] J. Montiel et al. "River: machine learning for streaming data in Python". In: *Journal of Machine Learning Research* 22.110 (2021), pp. 1–8. URL: `http://jmlr.org/papers/v22/20-1380.html`.

[69] V. Moskvina and A. Zhigljavsky. "An Algorithm Based on Singular Spectrum Analysis for Change-Point Detection". In: *Communications in Statistics - Simulation and Computation* 32.2 (2003), pp. 319–352. DOI: `10.1081/SAC-120017494`. eprint: `https://doi.org/10.1081/SAC-120017494`. URL: `https://doi.org/10.1081/SAC-120017494`.

[70] Q. P. Nguyen, K. W. Lim, D. M. Divakaran, K. H. Low, and M. C. Chan. "GEE: A Gradient-based Explainable Variational Autoencoder for Network Anomaly Detection". In: *2019 IEEE Conference on Communications and Network Security (CNS)*. June 2019, pp. 91–99. DOI: `10.1109/CNS.2019.8802833`.

[71] Y. Ogata, T. Kojima, S. Uchiyama, M. Yasuda, and F. Hine. "Effects of the Brine Impurities on the Performance of the Membrane-Type Chlor-Alkali Cell". In: *Journal of The Electrochemical Society* 136.1 (Jan. 1989), p. 91. DOI: `10.1149/1.2096621`. URL: `https://dx.doi.org/10.1149/1.2096621`.

[72] H. S. Pannu, J. Liu, and S. Fu. "AAD: Adaptive Anomaly Detection System for Cloud Computing Infrastructures". In: *2012 IEEE 31st Symposium on Reliable Distributed Systems*. 2012, pp. 396–397. DOI: `10.1109/SRDS.2012.3`.

[73] J. L. Proctor, S. L. Brunton, and J. N. Kutz. "Dynamic Mode Decomposition with Control". In: *SIAM Journal on Applied Dynamical Systems* 15.1 (2016), pp. 142–161. DOI: `10.1137/15M1013857`. eprint: `https://doi.org/10.1137/15M1013857`. URL: `https://doi.org/10.1137/15M1013857`.

[74] S. Rafayal, A. Alnaggar, and M. Cevik. "Optimizing electricity peak shaving through stochastic programming and probabilistic time series forecasting". In: *Journal of Building Engineering* 88 (2024), p. 109163.

[75] J. B. Rawlings, D. Q. Mayne, and M. Diehl. *Model Predictive Control: Theory, Computation, and Design*. 2nd ed. Nob Hill Publishing, 2017. ISBN: 978-0-9759377-3-0.

[76] B. Recht. "A Tour of Reinforcement Learning: The View from Continuous Control". In: *Annual Review of Control, Robotics, and Autonomous Systems* 2 (2019), pp. 253–279. DOI: `10.1146/annurev-control-053018-023825`.

[77] R. Reiter, J. Hoffmann, D. Reinhardt, F. Messerer, K. Baumgärtner, S. Sawant, J. Bödecker, M. Diehl, and S. Gros. “Synthesis of model predictive control and reinforcement learning: Survey and classification”. In: *Annual Reviews in Control* 61 (2026), p. 101045. ISSN: 1367-5788. DOI: https://doi.org/10.1016/j.arcontrol.2026.101045. URL: https://www.sciencedirect.com/science/article/pii/S1367578826000015.

[78] R. T. Rockafellar, S. Uryasev, et al. “Optimization of conditional value-at-risk”. In: *Journal of risk* 2 (2000), pp. 21–42.

[79] M. Salehi and L. Rashidi. “A Survey on Anomaly Detection in Evolving Data: [With Application to Forest Fire Risk Prediction]”. In: *SIGKDD Explor. Newsl.* 20.1 (May 2018), pp. 13–23. ISSN: 1931-0145. DOI: 10.1145/3229329.3229332. URL: https://doi.org/10.1145/3229329.3229332.

[80] A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, and P. W. Battaglia. “Learning to Simulate Complex Physics with Graph Networks”. In: *Proceedings of the 37th International Conference on Machine Learning*. ICML ’20. 2020, pp. 8459–8468.

[81] P. J. Schmid. “Dynamic mode decomposition of numerical and experimental data”. In: *Journal of Fluid Mechanics* 656 (2010), pp. 5–28. DOI: 10.1017/S0022112010001217.

[82] T. Siqin, S. He, B. Hu, and X. Fan. “Shared energy storage-multi-microgrid operation strategy based on multi-stage robust optimization”. In: *Journal of Energy Storage* 97 (2024), p. 112785.

[83] R. Smets, K. Bruninx, J. Bottieau, J.-F. Toubeau, and E. Delarue. “Strategic Implicit Balancing with Energy Storage Systems via Stochastic Model Predictive Control”. In: *IEEE Transactions on Energy Markets, Policy and Regulation* 1.4 (2023), pp. 373–385.

[84] T. Stauffer and D. Chastain-Knight. “Do not let your safe operating limits leave you S-O-L (out of luck)”. In: *Process Safety Progress* 40.1 (2021), e12163. DOI: https://doi.org/10.1002/prs.12163. eprint: https://aiche.onlinelibrary.wiley.com/doi/pdf/10.1002/prs.12163. URL: https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/prs.12163.

[85] P. D. Talagala, R. J. Hyndman, and K. Smith-Miles. “Anomaly Detection in High-Dimensional Data”. In: *Journal of Computational and Graphical Statistics* 30.2 (2021), pp. 360–374. DOI: 10.1080/10618600.2020.1807997. eprint: https://doi.org/10.1080/10618600.2020.1807997. URL: https://doi.org/10.1080/10618600.2020.1807997.

[86] C. Tang, B. Abbatematteo, J. Hu, R. Chandra, R. Martín-Martín, and P. Stone. "Deep Reinforcement Learning for Robotics: A Survey of Real-World Successes". In: *Annual Review of Control, Robotics, and Autonomous Systems* 8.Volume 8, 2025 (2025), pp. 153–188. ISSN: 2573-5144. DOI: `https://doi.org/10.1146/annurev-control-030323-022510`. URL: `https://www.annualreviews.org/content/journals/10.1146/annurev-control-030323-022510`.

[87] A. G. Tartakovsky, A. S. Polunchenko, and G. Sokolov. "Efficient Computer Network Anomaly Detection by Changepoint Detection Methods". In: *IEEE Journal of Selected Topics in Signal Processing* 7.1 (Feb. 2013), pp. 4–11. ISSN: 1941-0484. DOI: `10.1109/JSTSP.2012.2233713`.

[88] V. Trovato and B. Kantharaj. "Energy storage behind-the-meter with renewable generators: Techno-economic value of optimal imbalance management". In: *International Journal of Electrical Power & Energy Systems* 118 (2020), p. 105813.

[89] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz. "On dynamic mode decomposition: Theory and applications". In: *Journal of Computational Physics* 279 (2014), pp. 815–834. DOI: `10.1016/j.jcp.2014.06.006`.

[90] E. Van Henten. "Validation of a dynamic lettuce growth model for greenhouse climate control". In: *Agricultural Systems* 45.1 (1994), pp. 55–72. ISSN: 0308-521X. DOI: `https://doi.org/10.1016/S0308-521X(94)90280-1`. URL: `https://www.sciencedirect.com/science/article/pii/S0308521X94902801`.

[91] B. Vanthoor. "A model-based greenhouse design method". PhD thesis. [s.l.]: Wageningen University and Research, 2011. ISBN: 9789085859192. URL: `https://edepot.wur.nl/170301`.

[92] A. Wächter and L. T. Biegler. "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming". In: *Mathematical Programming* 106.1 (Mar. 2006), pp. 25–57. ISSN: 1436-4646. DOI: `10.1007/s10107-004-0559-y`. URL: `https://doi.org/10.1007/s10107-004-0559-y`.

[93] M. Wadinger and M. Kvasnica. "Real-Time Outlier Detection with Dynamic Process Limits". In: *Proceedings of the 2023 24th International Conference on Process Control (PC)*. in press. 2023.

[94] M. Wadinger and M. Kvasnica. "Adaptable and Interpretable Framework for Anomaly Detection in SCADA-based industrial systems". In: *Expert Systems with Applications* 246 (2024), p. 123200. ISSN: 0957-4174. DOI: `https://doi.org/10.1016/j.eswa.2024.123200`. URL: `https://www.sciencedirect.com/science/article/pii/S0957417424000654`.

[95] D. Wagner et al. *Formally Exploring Time-Series Anomaly Detection Evaluation Metrics*. 2025. arXiv: 2510.17562 [cs.LG]. URL: https://arxiv.org/abs/2510.17562.

[96] X. Wang and S.-H. Ahn. "Real-time prediction and anomaly detection of electrical load in a residential community". In: *Applied Energy* 259 (2020), p. 114145. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2019.114145. URL: https://www.sciencedirect.com/science/article/pii/S030626191931832X.

[97] Y. Wang, K. Wu, H. Zhao, J. Li, X. Sheng, Y. Yin, Q. Du, B. Zu, L. Han, and K. Jiao. "Degradation prediction of proton exchange membrane fuel cell stack using semi-empirical and data-driven methods". In: *Energy and AI* 11 (2023), p. 100205. ISSN: 2666-5468. DOI: https://doi.org/10.1016/j.egyai.2022.100205. URL: https://www.sciencedirect.com/science/article/pii/S2666546822000519.

[98] Z. Wang, S. Yang, X. Xiang, A. Vasilijević, N. Mišković, and Đ. Nađ. "Cloud-based mission control of USV fleet: Architecture, implementation and experiments". In: *Control Engineering Practice* 106 (2021), p. 104657. ISSN: 0967-0661. DOI: https://doi.org/10.1016/j.conengprac.2020.104657. URL: https://www.sciencedirect.com/science/article/pii/S0967066120302276.

[99] R. Ward. "EECi/GES: First release of GES software". In: *Energy Efficient Cities Initiative, University of Cambridge* (2019). DOI: 10.5281/zenodo.2594716. URL: https://zenodo.org/record/2594716.

[100] J. Weigert, C. Hoffmann, E. Esche, P. Fischer, and J.-U. Repke. "Towards demand-side management of the chlor-alkali electrolysis: Dynamic modeling and model validation". In: *Computers and Chemical Engineering* 149 (2021), p. 107287. ISSN: 0098-1354. DOI: https://doi.org/10.1016/j.compchemeng.2021.107287. URL: https://www.sciencedirect.com/science/article/pii/S009813542100065X.

[101] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley. "A data–driven approximation of the koopman operator: Extending dynamic mode decomposition". In: *Journal of Nonlinear Science* 25 (2015), pp. 1307–1346.

[102] D. Wilson. "Koopman Operator Inspired Nonlinear System Identification". In: *SIAM Journal on Applied Dynamical Systems* 22.2 (2023), pp. 1445–1471. DOI: 10.1137/22M1512272. eprint: https://doi.org/10.1137/22M1512272. URL: https://doi.org/10.1137/22M1512272.

[103] H. Wu, J. He, M. Tömösközi, Z. Xiang, and F. H. Fitzek. "In-Network Processing for Low-Latency Industrial Anomaly Detection in Softwarized Networks". In: *2021 IEEE Global Communications Conference (GLOBECOM)*. Dec. 2021, pp. 01–07. DOI: `10.1109/GLOBECOM46510.2021.9685489`.

[104] L. Wu, X. Yin, L. Pan, and J. Liu. "Smart energy management: Process structure-based hybrid neural networks for optimal scheduling and economic predictive control in integrated systems". In: *Applied Energy* 380 (2025), p. 124965. ISSN: 0306-2619. DOI: `https://doi.org/10.1016/j.apenergy.2024.124965`. URL: `https://www.sciencedirect.com/science/article/pii/S0306261924023481`.

[105] Y. Wu, B. Xu, and J. Anderson. "Energy Storage Arbitrage Under Price Uncertainty: Market Risks and Opportunities". In: *arXiv preprint arXiv:2501.08472* (2025).

[106] B. Xu, W. Ma, W. Wu, Y. Wang, Y. Yang, J. Li, X. Zhu, and Q. Liao. "Degradation prediction of PEM water electrolyzer under constant and start-stop loads based on CNN-LSTM". In: *Energy and AI* 18 (2024), p. 100420. ISSN: 2666-5468. DOI: `https://doi.org/10.1016/j.egyai.2024.100420`. URL: `https://www.sciencedirect.com/science/article/pii/S2666546824000867`.

[107] W.-T. Yang, M. S. Reis, V. Borodin, M. Juge, and A. Roussy. "An interpretable unsupervised Bayesian network model for fault detection and diagnosis". In: *Control Engineering Practice* 127 (2022), p. 105304. ISSN: 0967-0661. DOI: `https://doi.org/10.1016/j.conengprac.2022.105304`. URL: `https://www.sciencedirect.com/science/article/pii/S0967066122001502`.

[108] J. Ye, Y. Xu, and Q. Wang. "Subspace Change Point Detection Under Spiked Wigner Model". In: *IEEE Transactions on Signal Processing* 71 (2023), pp. 1995–2010. ISSN: 1941-0476. DOI: `10.1109/TSP.2023.3282075`.

[109] E. Yeung, S. Kundu, and N. Hodas. "Learning deep neural network representations for Koopman operators of nonlinear dynamical systems". In: *2019 American Control Conference (ACC)*. IEEE. 2019, pp. 4832–4839.

[110] R. Yu and R. Wang. "Learning dynamical systems from data: An introduction to physics-guided deep learning". In: *Proceedings of the National Academy of Sciences* 121.27 (2024), e2311808121. DOI: `10.1073/pnas.2311808121`. eprint: `https://www.pnas.org/doi/pdf/10.1073/pnas.2311808121`. URL: `https://www.pnas.org/doi/abs/10.1073/pnas.2311808121`.

[111] H. Zhang, C. W. Rowley, E. A. Deem, and L. N. Cattafesta. "Online Dynamic Mode Decomposition for Time-Varying Systems". In: *SIAM Journal on Applied Dynamical Systems* 18.3 (2019), pp. 1586–1609. DOI: `10.1137/18M1192329`.

eprint: https://doi.org/10.1137/18M1192329. URL: https://doi.org/10.1137/18M1192329.

[112] K. Zhang, J. Chen, C.-G. Lee, and S. He. "An unsupervised spatiotemporal fusion network augmented with random mask and time-relative information modulation for anomaly detection of machines with multiple measuring points". In: *Expert Systems with Applications* 237 (2024), p. 121506. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2023.121506. URL: https://www.sciencedirect.com/science/article/pii/S0957417423020080.

[113] R. Zhang, P. Zhou, and J. Qiao. "Anomaly Detection of Nonstationary Long-Memory Processes Based on Fractional Cointegration Vector Autoregression". In: *IEEE Transactions on Reliability* (2023), pp. 1–12. ISSN: 1558-1721. DOI: 10.1109/TR.2023.3314429.

[114] X. Zhang, J. Shi, X. Huang, F. Xiao, M. Yang, J. Huang, X. Yin, A. Sohail Usmani, and G. Chen. "Towards deep probabilistic graph neural network for natural gas leak detection and localization without labeled anomaly data". In: *Expert Systems with Applications* 231 (2023), p. 120542. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2023.120542. URL: https://www.sciencedirect.com/science/article/pii/S0957417423010448.

[115] Y. Zhang, F. Meng, R. Wang, W. Zhu, and X.-J. Zeng. "A stochastic MPC based approach to integrated energy management in microgrids". In: *Sustainable cities and society* 41 (2018), pp. 349–362.

[116] E. Zio. "Prognostics and Health Management (PHM): Where are we and where do we (need to) go in theory and practice". In: *Reliability Engineering & System Safety* 218 (2022), p. 108119. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2021.108119. URL: https://www.sciencedirect.com/science/article/pii/S0951832021006153.

Author's Publications

[117] T. Ábelová, M. Wadinger, and M. Kvasnica. "Predictive risk-aware control for microgrids: Operation of a revenue-generating energy management system". In: *Sustainable Energy, Grids and Networks* 44 (Dec. 2025), p. 102028. ISSN: 2352-4677. DOI: https://doi.org/10.1016/j.segan.2025.102028. URL: https://www.sciencedirect.com/science/article/pii/S2352467725004102.

[118] P. Valábek, M. Wadinger, M. Kvasnica, and M. Klauco. "Deep Dictionary-Free Method for Identifying Linear Model of Nonlinear System with Input Delay". In: *2025 25th International Conference on Process Control (PC)*. June 2025, pp. 1–6. DOI: 10.1109/PC65047.2025.11047273.

[119] M. Wadinger, R. Fáber, E. Pavlovičová, and R. Paulen. "Carbon neutral greenhouse: Economic model predictive control framework for education". In: *European Journal of Control* 86 (Nov. 2025). Special Issue on the European Control Conference 2025, p. 101297. ISSN: 0947-3580. DOI: https://doi.org/10.1016/j.ejcon.2025.101297. URL: https://www.sciencedirect.com/science/article/pii/S0947358025001268.

[120] M. Wadinger, L. Galčíková, L. Razzolini, M. Kvasnica, E. Brunazzi, and M. Vaccari. "Physics-Informed Dynamic Mode Decomposition with Control for Predictive Maintenance in Chlor-Alkali Electrolysis". In: *2025 IEEE 64th Conference on Decision and Control (CDC)*. Dec. 2025, pp. 450–455. DOI: 10.1109/CDC57313.2025.11312597.

[121] M. Wadinger and M. Kvasnica. "Real-Time Outlier Detection with Dynamic Process Limits". In: *2023 24th International Conference on Process Control (PC)*. June 2023, pp. 138–143. DOI: 10.1109/PC58330.2023.10217717.

[122] M. Wadinger and M. Kvasnica. "Adaptable and Interpretable Framework for Anomaly Detection in SCADA-based industrial systems". In: *Expert Systems with Applications* 246 (July 2024), p. 123200. ISSN: 0957-4174. DOI: https://

doi.org/10.1016/j.eswa.2024.123200. URL: https://www.sciencedirect.com/science/article/pii/S0957417424000654.

[123] M. Wadinger, M. Kvasnica, and Y. Kawahara. *Change-Point Detection in Industrial Data Streams based on Online Dynamic Mode Decomposition with Control*. Aug. 2024. arXiv: 2407.05976 [cs.AI]. URL: https://arxiv.org/abs/2407.05976.