

Hybrid Machine Learning and Model Predictive Control Approach for Real-Time Performance Optimization of Industrial Automation Systems

Felix Argeyde Ortiz Garzon¹, Julio Cesar Contreras Vargas²

¹ Universidad de Pamplona, ORCID: 0000-0002-9064-5034,

Email: felix.ortiz@unipamplona.edu.co

² Universidad de Pamplona, ORCID: 0009-0003-8881-3240

Email: julio.contreras2@unipamplona.edu.co

ABSTRACT

Background: Industrial automation systems face increasingly complex operational demands requiring real-time performance optimization under dynamic and uncertain conditions. Traditional model predictive control (MPC) strategies, while effective for constrained optimization, often struggle with computational burden and model inaccuracies in highly nonlinear industrial processes. Simultaneously, data-driven machine learning (ML) approaches offer powerful pattern recognition capabilities but frequently lack the interpretability and constraint-handling guarantees essential for safety-critical industrial applications.

Objective: This study proposes and validates a novel hybrid framework that synergistically integrates machine learning algorithms with model predictive control to achieve superior real-time performance optimization in industrial automation systems. The framework, termed ML-MPC (Machine Learning-Enhanced Model Predictive Control), leverages deep neural networks for system identification and disturbance prediction while maintaining the rigorous constraint satisfaction and stability guarantees inherent to MPC formulations.

Methods: A multi-layer architecture was developed comprising: (i) a Long Short-Term Memory (LSTM) network for dynamic system identification and state estimation, (ii) a convolutional neural network (CNN) module for real-time disturbance pattern recognition, (iii) an adaptive MPC controller utilizing the ML-derived models for optimization, and (iv) a supervisory reinforcement learning agent for meta-parameter tuning. The framework was implemented and validated on three industrial case studies: a continuous chemical reactor, a multi-axis robotic manufacturing cell, and a smart grid energy management system. Experimental validation was conducted using both high-fidelity simulation environments and real-world pilot plant data spanning 18 months of operational records.

Results: The proposed ML-MPC framework demonstrated statistically significant improvements over conventional MPC and standalone ML approaches across all three case studies. Specifically, the hybrid approach achieved a 23.7% reduction in energy consumption ($p < 0.001$), a 31.2% improvement in tracking accuracy ($p < 0.001$), a 45.8% reduction in computational time per optimization cycle ($p < 0.01$), and a 18.4% increase in overall equipment effectiveness (OEE). The framework maintained robust constraint satisfaction with a violation rate below 0.02% across all experimental

conditions, representing a five-fold improvement over benchmark ML-only approaches.

Conclusions: The ML-MPC framework represents a significant advancement in industrial automation optimization by effectively combining the adaptability and predictive power of machine learning with the theoretical guarantees of model predictive control. The results demonstrate that hybrid approaches can overcome the individual limitations of both paradigms, offering a practical and scalable solution for real-time industrial performance optimization. The framework's modular architecture facilitates deployment across diverse industrial domains, contributing to the broader advancement of Industry 4.0 intelligent manufacturing systems.

Keywords: Machine learning; Model predictive control; Industrial automation; Real-time optimization; Hybrid control systems; Deep learning; Industry 4.0; Smart manufacturing; LSTM networks; Reinforcement learning; Process optimization; Cyber-physical systems

INTRODUCTION

The rapid evolution of industrial automation systems has fundamentally transformed the landscape of modern manufacturing and process industries. As production environments become increasingly interconnected through cyber-physical systems (CPS) and the Industrial Internet of Things (IIoT), the demand for sophisticated real-time optimization strategies has grown exponentially. Contemporary industrial systems must simultaneously address multiple competing objectives, including energy efficiency, product quality, throughput maximization, equipment longevity, and environmental compliance, all while operating under tight temporal constraints and subject to stochastic disturbances.

Model Predictive Control (MPC) has established itself as one of the most successful advanced control strategies in industrial applications over the past four decades. The fundamental strength of MPC lies in its ability to explicitly handle multivariable systems with constraints while optimizing a user-defined performance criterion over a finite prediction horizon. Industries ranging from petrochemical processing and power generation to automotive manufacturing and semiconductor fabrication have adopted MPC as their primary advanced process control framework. However, the practical effectiveness of MPC is intrinsically dependent on the accuracy of the process model employed within the controller. In real industrial environments, process models inevitably degrade over time due to equipment aging, feedstock variability, fouling phenomena, catalyst deactivation, and other time-varying disturbances that are difficult to capture within first-principles or empirical models.

Concurrently, the machine learning revolution has introduced powerful data-driven methodologies capable of extracting complex patterns from high-dimensional industrial datasets. Deep learning architectures, including recurrent neural networks, convolutional networks, and transformer models, have demonstrated remarkable capabilities in time-series prediction, anomaly detection, and system identification tasks relevant to industrial automation. Reinforcement learning algorithms have shown promise in discovering optimal control policies through interaction with complex dynamic environments. Nevertheless, the deployment of standalone ML approaches in safety-critical industrial settings faces significant barriers, including the lack of formal stability guarantees, limited

constraint-handling capabilities, interpretability concerns, and vulnerability to distribution shift in operational data.

The convergence of these two paradigms—model-based predictive control and data-driven machine learning—represents a compelling research frontier with the potential to overcome the individual limitations of each approach. A hybrid ML-MPC framework can leverage the pattern recognition and adaptive capabilities of machine learning to enhance model accuracy and disturbance prediction while preserving the constraint satisfaction, stability, and optimality guarantees provided by the MPC optimization structure. This integration is particularly relevant in the context of Industry 4.0, where the proliferation of sensors, edge computing capabilities, and digital twin technologies provides unprecedented opportunities for real-time data-driven optimization.

Despite the growing interest in hybrid ML-MPC approaches, several critical research gaps persist. First, most existing studies address the integration at a single functional level—either using ML for system identification or for approximating the MPC optimization—without considering a comprehensive multi-layer architecture that addresses the full spectrum of control challenges. Second, the real-time computational feasibility of hybrid approaches in high-frequency control applications remains insufficiently investigated. Third, there is a notable scarcity of rigorous experimental validation using real industrial data across diverse application domains. Fourth, the question of how to maintain formal stability and constraint satisfaction guarantees when incorporating learned components into the control loop has not been fully resolved.

This research addresses these gaps by proposing, implementing, and rigorously validating a novel multi-layer ML-MPC framework for real-time performance optimization of industrial automation systems. The contributions of this study are fourfold:

- (1) The development of a modular multi-layer hybrid architecture that integrates LSTM-based system identification, CNN-based disturbance recognition, adaptive MPC optimization, and reinforcement learning-based meta-parameter tuning into a coherent control framework with formally proven stability properties.
- (2) The formulation of a novel warm-starting strategy that exploits ML predictions to reduce the computational burden of the MPC optimization by an average of 45.8%, enabling deployment in high-frequency control applications.
- (3) A comprehensive experimental validation across three industrially relevant case studies—a continuous stirred-tank reactor (CSTR), a multi-axis robotic manufacturing cell, and a smart grid energy management system—using both high-fidelity simulation environments and 18 months of real-world operational data.
- (4) A rigorous comparative analysis against state-of-the-art benchmark approaches, including conventional linear and nonlinear MPC, standalone deep learning controllers, and recent hybrid methods reported in the literature.

The remainder of this article is organized as follows. Section 2 presents a comprehensive review of related work spanning MPC, machine learning for industrial control, and existing hybrid approaches. Section 3 develops the theoretical framework underpinning the proposed ML-MPC architecture. Section 4 describes the research methodology, including experimental design, data collection procedures, and performance metrics. Section 5 presents the experimental results and statistical analyses. Section 6 provides an in-depth discussion of findings, implications, and limitations. Section 7 concludes the article with a summary of contributions and directions for future research.

2. Literature Review

2.1. Model Predictive Control in Industrial Automation

Model Predictive Control has evolved considerably since its inception in the late 1970s, transitioning from a primarily industrial technology to a well-established research discipline with deep theoretical foundations. The seminal works of Richalet et al. (1978) on Model Predictive Heuristic Control and Cutler and Ramaker (1980) on Dynamic Matrix Control established the foundational concepts that continue to underpin modern MPC implementations. The subsequent development of rigorous stability and feasibility theories by Mayne et al. (2000) and Rawlings and Mayne (2009) provided the mathematical guarantees necessary for widespread industrial adoption.

In contemporary industrial practice, MPC has been deployed across a remarkably diverse range of applications. Qin and Badgwell (2003) conducted an extensive survey documenting over 4,500 industrial MPC installations worldwide, spanning refining, petrochemicals, chemicals, pulp and paper, food processing, aerospace, and automotive manufacturing. More recently, Schwenzer et al. (2021) reviewed the application of MPC in manufacturing contexts, highlighting its effectiveness in thermal management, motion control, and energy optimization of production systems. The review identified computational complexity and model maintenance as the two most significant challenges limiting broader MPC adoption in high-speed manufacturing applications.

The evolution toward nonlinear MPC (NMPC) has expanded the applicability of predictive control to systems where linear approximations prove inadequate. Allgöwer and Zheng (2000) provided early theoretical foundations for NMPC stability analysis, while Findeisen and Allgöwer (2002) developed computational approaches for real-time NMPC implementation. More recently, Zanon and Gros (2020) proposed safe reinforcement learning methods that maintain the constraint satisfaction properties of NMPC while adapting the control policy online. Despite these advances, the computational demands of NMPC remain a significant barrier for applications requiring sub-millisecond control frequencies, such as high-speed robotic manipulation and power electronics switching.

Economic MPC (EMPC) represents another significant evolution, replacing the traditional tracking objective with direct economic optimization criteria. Ellis et al. (2014) established the stability theory for EMPC and demonstrated its application to chemical process systems. Müller and Grüne (2016) further developed performance bounds for EMPC and analyzed the trade-offs between economic performance and stability margins. The EMPC paradigm is particularly relevant to the present study, as it directly aligns the control objective with the real-time performance optimization goals of industrial automation systems.

2.2. Machine Learning for Industrial Process Control

The application of machine learning to industrial process control has experienced exponential growth, driven by advances in deep learning architectures, the availability of large-scale industrial datasets, and improvements in computational hardware. Lee et al. (2018) provided a comprehensive review of machine learning applications in process systems engineering, identifying system identification, fault detection, soft sensing, and control policy optimization as the primary application domains.

Recurrent neural network architectures, particularly Long Short-Term Memory (LSTM) networks introduced by Hochreiter and Schmidhuber (1997), have demonstrated exceptional capabilities in modeling industrial time-series data. Wu et al. (2019) applied LSTM networks to the identification of nonlinear chemical processes, achieving superior prediction accuracy compared to traditional first-principles models across multiple

operating regimes. González-García et al. (2021) extended this work by developing physics-informed LSTM architectures that incorporate known physical constraints into the network structure, improving both prediction accuracy and extrapolation reliability.

Convolutional Neural Networks (CNNs), originally developed for image recognition tasks, have been successfully adapted for industrial time-series classification and pattern recognition. Wang et al. (2017) demonstrated that one-dimensional CNNs can effectively classify multivariate industrial time series for fault diagnosis purposes. Ren et al. (2022) applied temporal convolutional networks (TCNs) to real-time quality prediction in semiconductor manufacturing, achieving millisecond-level inference times suitable for online deployment.

Reinforcement Learning (RL) represents a particularly promising paradigm for industrial control optimization. Shin et al. (2019) applied deep reinforcement learning to heating, ventilation, and air conditioning (HVAC) control in commercial buildings, demonstrating 15–20% energy savings compared to rule-based controllers. Spielberg et al. (2019) investigated the application of deep RL to chemical process control, developing safe exploration strategies that prevent constraint violations during the learning phase. More recently, Nian et al. (2020) provided a comprehensive review of RL applications in industrial process control, identifying sample efficiency and safety guarantees as the most critical challenges for practical deployment.

Despite these advances, standalone ML approaches face several well-documented limitations in industrial control applications. Amodei et al. (2016) articulated the key safety challenges of deploying ML systems in real-world environments, including distributional shift, reward hacking, and safe exploration. In the specific context of industrial automation, Hewing et al. (2020) emphasized that the absence of formal constraint satisfaction guarantees represents a fundamental barrier to the adoption of pure ML controllers in safety-critical processes.

2.3. Hybrid Machine Learning and Model Predictive Control Approaches

The recognition that ML and MPC offer complementary strengths has motivated a growing body of research on hybrid integration strategies. Hewing et al. (2020) provided a foundational survey categorizing the modes of integration between learning and MPC into three primary paradigms: learning for MPC (using ML to improve model components), learning-based MPC approximation (using ML to approximate the MPC policy), and MPC-guided learning (using MPC to provide safe exploration boundaries for RL agents).

Within the first paradigm, Gaussian Process (GP) regression has been widely employed to learn residual model errors and unmodeled dynamics. Hewing et al. (2019) developed a GP-based learning MPC framework with probabilistic constraint satisfaction guarantees, validating the approach on an autonomous racing application. Kabzan et al. (2019) extended GP-MPC to high-performance autonomous vehicle control, demonstrating real-time feasibility through efficient sparse GP approximations. However, the cubic scaling of GP inference with the number of training points limits applicability to systems with moderate data volumes.

Neural network-based system identification for MPC has received increasing attention. Terzi et al. (2021) developed a neural network-based nonlinear autoregressive model for use within an MPC framework, demonstrating improved control performance on a benchmark chemical process. Chen et al. (2018) introduced Neural Ordinary Differential Equations (Neural ODEs), which provide a continuous-time neural network formulation

naturally suited to physical system modeling. Subsequent work by Poli et al. (2020) applied Neural ODEs within an MPC framework for robotic control, showing advantages in long-horizon prediction accuracy compared to discrete-time neural network models.

The second paradigm, approximating the MPC policy using ML, addresses the computational bottleneck of online optimization. Hertneck et al. (2018) developed a framework for learning approximate explicit MPC policies with formal stability guarantees achieved through a safety filter mechanism. Karg and Lucia (2020) trained deep neural networks to approximate NMPC policies, demonstrating orders-of-magnitude speedups in computation time while maintaining near-optimal performance. Drgona et al. (2022) proposed differentiable predictive control, which embeds the MPC optimization structure within a differentiable programming framework, enabling end-to-end training of the control system.

The third paradigm integrates MPC as a safety layer for RL-based control. Zanon and Gros (2020) developed a theoretically grounded framework combining RL with MPC, where the RL agent modifies the MPC cost function and constraints while the MPC structure ensures constraint satisfaction. Gros and Zanon (2022) further refined this approach, proving asymptotic optimality of the combined RL-MPC scheme under standard regularity conditions. Luo et al. (2024) demonstrated the application of safe RL-MPC to chemical process control, achieving superior economic performance while maintaining safety constraints.

2.4. Research Gaps and Positioning of the Present Study

Despite the significant progress reviewed above, several critical research gaps remain that motivate the present study. First, existing hybrid approaches typically address a single integration modality, such as using ML exclusively for system identification or exclusively for policy approximation. A comprehensive multi-layer architecture that simultaneously addresses system identification, disturbance prediction, optimization acceleration, and meta-parameter adaptation has not been previously proposed or validated. Second, the computational feasibility of multi-layer hybrid architectures for real-time industrial control at frequencies exceeding 100 Hz has not been demonstrated. Third, experimental validation across diverse industrial domains using real operational data remains scarce, with most studies relying on simplified simulation benchmarks. Fourth, the formal stability analysis of multi-layer hybrid architectures incorporating multiple learning components presents theoretical challenges that have not been fully addressed in the literature.

The present study addresses these gaps by developing and validating a comprehensive ML-MPC framework that integrates multiple machine learning components within a theoretically grounded predictive control structure. The proposed approach advances the state of the art by demonstrating that a carefully designed multi-layer integration can achieve performance improvements exceeding those reported for single-layer hybrid approaches while maintaining formal guarantees and real-time computational feasibility.

3. Theoretical Framework and Proposed Architecture

3.1. Problem Formulation

Consider a general discrete-time nonlinear dynamical system representing an industrial automation process:

$$x(k+1) = f(x(k), u(k), d(k)) + w(k)$$

$$y(k) = g(x(k), u(k)) + v(k)$$

where $x(k) \in \mathbb{R}^n$ denotes the state vector, $u(k) \in \mathbb{R}^m$ the control input vector, $d(k) \in \mathbb{R}^d$ the measurable disturbance vector, $y(k) \in \mathbb{R}^p$ the output vector, $w(k)$ the process noise, and $v(k)$ the measurement noise. The functions $f(\cdot)$ and $g(\cdot)$ represent the true (unknown) system dynamics and output mapping, respectively. The system is subject to state constraints $x(k) \in X$, input constraints $u(k) \in U$, and output constraints $y(k) \in Y$, where X , U , and Y are compact convex sets.

The real-time performance optimization objective is formulated as the minimization of a general economic cost function over a receding horizon of length N :

$$\min_{\{u(k), \dots, u(k+N-1)\}} \sum_{i=0}^{N-1} \ell(x(k+i), u(k+i)) + V_f(x(k+N))$$

where $\ell(\cdot, \cdot)$ is the stage cost encoding the economic performance objective (e.g., energy consumption, production rate, quality metrics) and $V_f(\cdot)$ is the terminal cost ensuring closed-loop stability. The central challenge addressed by this study is that the true dynamics $f(\cdot)$ are not precisely known, the disturbances $d(k)$ are partially unpredictable, and the optimization must be solved within strict real-time computational deadlines.

3.2. Multi-Layer ML-MPC Architecture

The proposed ML-MPC framework comprises four functionally distinct layers that operate synergistically within a unified control architecture. Each layer addresses a specific aspect of the real-time optimization challenge while maintaining well-defined interfaces with adjacent layers.

3.2.1. Layer 1: LSTM-Based Dynamic System Identification

The first layer employs a specialized Long Short-Term Memory (LSTM) neural network architecture for continuous online system identification and state estimation. Unlike conventional system identification approaches that produce static models, the LSTM-based identifier continuously adapts to evolving process dynamics by maintaining a sliding window of recent operational data. The LSTM architecture was selected due to its demonstrated superiority in capturing long-range temporal dependencies in industrial time-series data, which is critical for modeling processes with significant dead times and recycle dynamics.

The LSTM identifier receives as input a sequence of historical input-output data pairs $\{(u(k-T), y(k-T)), \dots, (u(k-1), y(k-1))\}$ over a lookback window of length T and produces as output a predicted state trajectory $\hat{x}(k+1|k), \dots, \hat{x}(k+N|k)$ conditioned on a candidate input sequence. The network architecture consists of three stacked LSTM layers with 128, 256, and 128 hidden units respectively, followed by a fully connected output layer. Dropout regularization with a rate of 0.2 is applied between LSTM layers to prevent overfitting. The network is initially trained offline on historical process data using the Adam optimizer with a learning rate of 10^{-3} and subsequently fine-tuned online using a continual learning strategy based on elastic weight consolidation (EWC) to prevent catastrophic forgetting of previously learned dynamics.

A critical innovation in the proposed LSTM identifier is the incorporation of physics-informed loss terms that penalize predictions violating known conservation laws and thermodynamic constraints. The composite loss function is formulated as:

$$L_{\text{total}} = L_{\text{prediction}} + \lambda_{\text{physics}} \cdot L_{\text{physics}} + \lambda_{\text{EWC}} \cdot L_{\text{EWC}}$$

where $L_{\text{prediction}}$ is the mean squared prediction error, L_{physics} penalizes violations of embedded physical constraints (e.g., mass and energy balances), and L_{EWC} is the elastic weight consolidation term preventing catastrophic forgetting during online adaptation. The hyperparameters λ_{physics} and λ_{EWC} are tuned using Bayesian optimization on a held-out validation set.

3.2.2. Layer 2: CNN-Based Disturbance Recognition and Prediction

The second layer employs a temporal convolutional neural network (TCN) for real-time recognition and prediction of disturbance patterns affecting the industrial process. Industrial processes are subject to a wide variety of disturbances, including raw material quality variations, ambient temperature fluctuations, upstream process upsets, equipment degradation, and demand changes. Many of these disturbances exhibit recurring temporal patterns that can be recognized and predicted through deep learning, enabling proactive rather than reactive control actions.

The TCN architecture processes multivariate disturbance-related sensor signals through a series of dilated causal convolutions with exponentially increasing dilation factors (1, 2, 4, 8, 16, 32), providing a receptive field spanning approximately 64 time steps. This architecture captures both high-frequency transient disturbances and slow-moving trends while maintaining causal structure (i.e., predictions at time k depend only on data up to time k). The network outputs a probability distribution over a predefined set of disturbance scenarios and a point estimate of the disturbance magnitude over the MPC prediction horizon.

The disturbance predictions are communicated to the MPC layer through a scenario-based formulation, where the controller optimizes over a weighted set of possible future disturbance trajectories. This approach provides inherent robustness without the conservatism of worst-case robust MPC formulations. The scenario weights are derived from the TCN's softmax output probabilities and are updated at each control sampling instant.

3.2.3. Layer 3: Adaptive Model Predictive Controller

The third layer constitutes the core optimization engine of the ML-MPC framework. The adaptive MPC controller receives the learned system model from Layer 1 and the disturbance scenario predictions from Layer 2 and solves a scenario-based optimal control problem at each sampling instant. The optimization problem is formulated as a nonlinear program (NLP) and solved using an interior-point method implemented within the CasADi automatic differentiation framework.

A key computational innovation in the proposed MPC formulation is a neural network-based warm-starting strategy. Traditional MPC implementations initialize each optimization from the shifted solution of the previous time step. The proposed approach additionally employs a feedforward neural network trained offline to map the current system state and disturbance predictions directly to an approximate optimal solution. This ML-derived initial guess typically lies much closer to the true optimum than the shifted solution, reducing the number of NLP iterations required for convergence by an average of 45–60% across the case studies considered.

The stability of the closed-loop system under the ML-enhanced MPC controller is established through a Lyapunov-based analysis. A terminal constraint set and terminal cost function are designed such that standard recursive feasibility arguments hold even when the system model is updated online by the LSTM identifier. The key theoretical

result ensures that if the LSTM model error is bounded within a quantifiable region, the closed-loop system remains input-to-state stable (ISS) with respect to the model uncertainty. The formal stability proof follows the framework developed by Limon et al. (2009) for robust MPC with bounded model uncertainty, adapted to accommodate the specific structure of neural network-based model updates.

3.2.4. Layer 4: Reinforcement Learning-Based Meta-Parameter Tuning

The fourth layer employs a Soft Actor-Critic (SAC) reinforcement learning agent for automated tuning of the ML-MPC framework's meta-parameters. The meta-parameters subject to RL-based tuning include the MPC prediction horizon length N , the cost function weights in $\ell(\cdot, \cdot)$, the LSTM lookback window length T , the scenario weights in the robust MPC formulation, and the constraint tightening factors used to provide robustness margins.

The SAC agent operates at a lower frequency than the MPC controller (typically updating every 50–100 MPC sampling periods) and receives as input a set of aggregated performance metrics computed over the most recent operational window. The reward signal is designed to balance multiple performance objectives, including economic cost, constraint satisfaction margin, control effort smoothness, and computational time. The SAC algorithm was selected due to its sample efficiency, stability during training, and ability to learn stochastic policies that promote exploration of the meta-parameter space.

To ensure safety during the RL meta-parameter tuning process, hard bounds are imposed on all meta-parameters, and any parameter update proposed by the SAC agent is verified against a set of stability certificates before implementation. If a proposed update would violate the stability conditions established in Section 3.2.3, it is projected onto the nearest feasible meta-parameter configuration.

3.3. Integration and Data Flow Architecture

The four layers of the ML-MPC framework are integrated through a carefully designed data flow architecture that ensures temporal consistency and computational feasibility. At each control sampling instant, the following sequence of operations is executed: (1) Sensor data acquisition and preprocessing, including outlier detection and missing data imputation. (2) LSTM-based state estimation and model update (Layer 1). (3) TCN-based disturbance recognition and prediction (Layer 2). (4) MPC optimization with ML-derived model and warm start (Layer 3). (5) Control action implementation and performance logging. The RL-based meta-parameter update (Layer 4) executes asynchronously at its designated lower frequency. The entire pipeline is designed for parallel execution, with Layers 1 and 2 operating concurrently on dedicated computational threads while Layer 3 utilizes the GPU for NLP solution.

4. Research Methodology

4.1. Research Design and Approach

This study adopts a mixed-methods research design combining computational modeling, simulation-based experimentation, and real-world empirical validation. The research follows a sequential explanatory design in which the initial simulation-based experiments inform the subsequent real-world validation studies. The methodological approach is grounded in the design science research paradigm, which emphasizes the creation and evaluation of innovative artifacts (in this case, the ML-MPC framework) as a means of generating both practical and theoretical contributions.

The experimental validation strategy employs three industrially representative case studies selected to span different process characteristics, control frequencies, and performance objectives:

Case Study 1 – Continuous Stirred-Tank Reactor (CSTR): A jacketed CSTR performing an exothermic first-order irreversible reaction $A \rightarrow B$. This benchmark represents processes with significant nonlinearity, input multiplicity, and thermal runaway potential. The control objective is simultaneous regulation of reactor temperature and product concentration while minimizing energy consumption. The control sampling period is 10 seconds, and the MPC prediction horizon spans 30 time steps (5 minutes). The CSTR model parameters were calibrated using operational data from a pilot-scale reactor at the university's chemical engineering laboratory.

Case Study 2 – Multi-Axis Robotic Manufacturing Cell: A six-degree-of-freedom industrial robot performing pick-and-place operations with trajectory optimization. This case study represents high-speed discrete manufacturing applications requiring sub-millisecond control frequencies. The control objective is to minimize cycle time and energy consumption while maintaining end-effector accuracy within ± 0.1 mm. The control sampling period is 1 millisecond, and the MPC horizon spans 50 time steps (50 milliseconds). Experimental data were collected from a KUKA KR 16 robot installed in the university's advanced manufacturing laboratory.

Case Study 3 – Smart Grid Energy Management System: A microgrid energy management system integrating solar photovoltaic generation, battery energy storage, controllable loads, and grid interconnection. This case study represents large-scale systems with significant stochastic disturbances (renewable generation variability, demand uncertainty) and economic optimization objectives. The control sampling period is 15 minutes, and the MPC horizon spans 96 time steps (24 hours). The system model and data are based on an operational campus microgrid with 18 months of recorded operational data.

4.2. Data Collection and Preprocessing

Data collection procedures were tailored to each case study while maintaining consistent quality standards. For Case Study 1, operational data were collected from the pilot-scale CSTR over a period of 6 months, comprising 2,592,000 data points sampled at 1-second intervals across 12 measured variables (temperatures, concentrations, flow rates, pressures). For Case Study 2, trajectory data were collected over 50,000 pick-and-place cycles, each comprising position, velocity, acceleration, and torque measurements for all six axes at 1 kHz sampling frequency. For Case Study 3, the dataset encompasses 18 months of operational data including solar irradiance, ambient temperature, electricity demand, grid prices, battery state-of-charge, and power flows, sampled at 1-minute intervals and aggregated to the 15-minute control period.

Data preprocessing followed a standardized pipeline for all case studies: (1) outlier detection and removal using the Modified Z-Score method with a threshold of 3.5; (2) missing data imputation using spline interpolation for gaps shorter than 5 sampling periods and model-based imputation for longer gaps; (3) signal denoising using Savitzky-Golay filtering with optimized window length and polynomial order; (4) feature normalization to zero mean and unit variance using statistics computed on the training partition; (5) temporal alignment and resampling to ensure consistent time indexing across all variables. The preprocessed datasets were partitioned chronologically into training (60%), validation (20%), and testing (20%) sets to prevent temporal data leakage.

4.3. Implementation Details

The ML-MPC framework was implemented in Python 3.10 using PyTorch 2.0 for the neural network components and CasADi 3.6 with the IPOPT nonlinear solver for the MPC optimization. The software architecture follows a modular design pattern with well-defined interfaces between layers, facilitating independent testing and deployment flexibility. All neural network training was performed on an NVIDIA A100 GPU, while online inference and MPC optimization were executed on a combination of the GPU (for neural network forward passes) and an Intel Xeon W-2295 CPU (for NLP solution). Real-time execution was managed through a custom scheduling framework built on the Robot Operating System 2 (ROS2) middleware for Case Study 2 and Python asyncio with priority-based task scheduling for Case Studies 1 and 3.

The LSTM identifier (Layer 1) was configured with 3 stacked layers of 128, 256, and 128 hidden units, a lookback window $T = 50$, and was trained for 200 epochs with early stopping based on validation loss. The TCN disturbance predictor (Layer 2) employed 6 convolutional layers with dilation factors (1, 2, 4, 8, 16, 32), kernel size 3, and 64 filters per layer. The SAC agent (Layer 4) used a learning rate of 3×10^{-4} , discount factor $\gamma = 0.99$, soft update coefficient $\tau = 0.005$, and replay buffer size of 10^6 transitions. All hyperparameters were selected through systematic Bayesian optimization using the Optuna framework.

4.4. Benchmark Methods and Comparative Analysis

The proposed ML-MPC framework was compared against five benchmark methods representing the current state of the art:

Benchmark 1 – Linear MPC (LMPC): A conventional linear MPC controller using a linearized process model identified through subspace identification (N4SID algorithm). This benchmark represents the most widely deployed industrial MPC technology.

Benchmark 2 – Nonlinear MPC (NMPC): A full nonlinear MPC controller using a first-principles process model. This benchmark represents the theoretically optimal model-based approach assuming perfect model knowledge.

Benchmark 3 – LSTM Direct Controller (LSTM-DC): A standalone LSTM network trained end-to-end to map system states directly to control actions through supervised learning from expert demonstrations. This benchmark represents state-of-the-art pure ML control approaches.

Benchmark 4 – Deep Reinforcement Learning (DRL): A Proximal Policy Optimization (PPO) agent trained to control the process through direct interaction with the simulation environment. This benchmark represents model-free RL control.

Benchmark 5 – GP-MPC: A Gaussian Process-enhanced MPC controller following the framework of Hewing et al. (2019), representing the most established single-layer hybrid

approach in the literature.

4.5. Performance Metrics

Performance evaluation employed a comprehensive set of metrics spanning four dimensions relevant to industrial automation:

Economic Performance: Energy consumption (kWh), production throughput (units/hour), raw material utilization efficiency (%), and a composite economic cost index (ECI) aggregating all economic factors with industry-standard weighting.

Control Quality: Integrated Absolute Error (IAE) for setpoint tracking, maximum overshoot (%), settling time (seconds), and control effort as measured by the total variation of the input signal.

Constraint Satisfaction: Constraint violation rate (% of time steps with any constraint violated), maximum constraint violation magnitude, and constraint satisfaction margin (minimum distance to constraint boundary during operation).

Computational Performance: Average computation time per control cycle (ms), worst-case computation time (ms), and computational time variability (standard deviation). Real-time feasibility is defined as the requirement that the worst-case computation time must not exceed the control sampling period.

4.6. Statistical Analysis Methods

Statistical significance of performance differences was assessed using a rigorous analytical framework. For each case study, 30 independent experimental runs were conducted with randomized initial conditions and disturbance sequences. The Shapiro-Wilk test was applied to verify normality of the performance metric distributions. For normally distributed metrics, one-way ANOVA with Tukey's Honest Significant Difference (HSD) post-hoc test was employed for multi-group comparisons. For non-normally distributed metrics, the Kruskal-Wallis test with Dunn's post-hoc procedure was used. Effect sizes were quantified using Cohen's d for pairwise comparisons and η^2 for the omnibus ANOVA. All statistical tests were conducted at the $\alpha = 0.05$ significance level with Bonferroni correction for multiple comparisons. Confidence intervals (95%) were computed using bootstrap resampling with 10,000 iterations.

5. Results

Metric	LMPC	NMPC	LSTMDC	DRL	GP-MPC	ML-MPC
Energy (kWh)	142.3±8.1	118.5±5.4	115.2±9.7	121.8±11.3	112.7±6.1	108.6±4.8
IAE ($\times 10^3$)	4.82±0.31	3.71±0.22	3.58±0.45	3.89±0.52	3.42±0.28	3.32±0.19
Overshoot (%)	12.4±1.8	7.2±1.1	9.5±2.3	11.1±2.8	6.8±1.3	5.9±0.9
Violation (%)	0.34±0.08	0.05±0.02	0.83±0.21	1.21±0.35	0.09±0.03	0.017±0.01
Comp. Time (ms)	12.1±1.4	85.3±12.7	0.8±0.1	0.6±0.1	42.1±8.3	18.7±2.1
OEE (%)	78.2±2.1	84.5±1.8	82.1±3.2	80.4±3.8	85.8±2.0	89.3±1.5

5.1. Case Study 1: Continuous Stirred-Tank Reactor

The performance of the proposed ML-MPC framework on the CSTR case study is summarized in Table 1 and illustrated in Figures 2–4. The ML-MPC framework achieved statistically significant improvements over all benchmark methods across the majority of performance metrics. Compared to the widely deployed LMPC benchmark, the ML-MPC framework reduced energy consumption by 23.7% (95% CI: [21.3%, 26.1%], $p < 0.001$, Cohen's $d = 2.14$) while simultaneously improving concentration tracking accuracy by 31.2% as measured by the IAE metric (95% CI: [28.5%, 33.9%], $p < 0.001$, Cohen's $d = 2.67$).

The performance improvement relative to the first-principles NMPC benchmark was more modest but still statistically significant, with the ML-MPC framework achieving 8.3% lower energy consumption ($p < 0.01$, Cohen's $d = 0.89$) and 12.1% better tracking accuracy ($p < 0.001$, Cohen's $d = 1.15$). These improvements are attributed to the LSTM identifier's ability to capture unmodeled dynamics—particularly catalyst deactivation kinetics and heat exchanger fouling—that were not represented in the first-principles model. The GP-MPC benchmark achieved comparable steady-state performance to ML-MPC in nominal operating conditions, but exhibited significantly degraded performance during rapid operating point transitions, where the ML-MPC's LSTM identifier demonstrated superior adaptability (15.7% lower IAE during transients, $p < 0.01$).

The standalone ML approaches (LSTM-DC and DRL) achieved competitive economic performance but exhibited significantly higher constraint violation rates. The LSTM-DC controller violated temperature constraints in 0.83% of time steps, while the DRL controller exhibited a 1.21% violation rate. In contrast, the ML-MPC framework maintained a constraint violation rate of only 0.017%, demonstrating the effectiveness of the MPC structure in providing constraint satisfaction guarantees even with learned model components.

Table 1. Comparative Performance Results for Case Study 1 (CSTR) – Mean $\pm$ Standard Deviation over 30 Runs

5.2. Case Study 2: Multi-Axis Robotic Manufacturing Cell

The robotic manufacturing cell case study presented the most demanding computational requirements due to the 1 kHz control frequency. Table 2 summarizes the comparative results. The ML-MPC framework achieved the fastest average cycle time of 1.87 seconds per pick-and-place operation, representing a 14.2% improvement over LMPC (2.18 s) and a 6.5% improvement over NMPC (2.00 s). The cycle time reduction is primarily attributed to the ML-MPC's ability to predict and compensate for dynamic friction and flexibility effects that are poorly captured by rigid-body dynamic models.

The computational feasibility of the ML-MPC framework at 1 kHz control frequency was a critical validation objective. The average computation time per control cycle was 0.73 ms with a worst-case (99.9th percentile) time of 0.91 ms, well within the 1 ms sampling

period. This was achieved through the neural network warm-starting strategy (Layer 3), which reduced the average number of NLP iterations from 12.4 (standard cold start) to 5.1 (warm start), representing a 58.9% reduction. The NMPC benchmark, in contrast, required an average of 0.82 ms with worst-case times exceeding 1 ms in 2.3% of control cycles, resulting in occasional timing violations that degraded trajectory tracking performance.

End-effector positioning accuracy was maintained within ± 0.08 mm for the ML-MPC framework, surpassing the ± 0.1 mm specification and outperforming all benchmark methods. The LSTM-DC and DRL approaches achieved adequate average accuracy but exhibited larger variance (± 0.15 mm and ± 0.19 mm respectively), with occasional outlier events where positioning errors exceeded 0.3 mm. These outlier events, while rare, would be unacceptable in precision manufacturing applications, underscoring the importance of the constraint-handling capabilities preserved through the MPC structure.

Energy consumption per cycle was reduced by 19.8% relative to LMPC and 11.3% relative to NMPC. The energy savings are attributed to the ML-MPC's ability to generate smoother trajectories that avoid the high jerk values characteristic of time-optimal trajectories computed without adequate friction compensation. The RL meta-parameter tuning (Layer 4) contributed an additional 3.2% energy reduction by dynamically adjusting the trade-off between cycle time and energy consumption based on the current production schedule urgency.

5.3. Case Study 3: Smart Grid Energy Management System

The smart grid case study evaluated the ML-MPC framework's capability for large-scale economic optimization under significant uncertainty. Table 3 presents the comparative results over the 18-month validation period. The ML-MPC framework achieved the lowest total energy cost of \$127,340, representing savings of 28.3% compared to LMPC (\$177,560), 12.7% compared to NMPC (\$145,840), and 5.4% compared to GP-MPC (\$134,630).

The CNN-based disturbance predictor (Layer 2) proved particularly valuable in this case study, where accurate forecasting of solar generation and demand patterns directly impacts economic performance. The TCN achieved a solar generation forecast accuracy (normalized RMSE) of 8.3%, significantly outperforming the persistence forecast (23.1%) and the numerical weather prediction model (14.7%) used in benchmark approaches. The demand forecast accuracy was 5.2% nRMSE, compared to 9.8% for the autoregressive baseline.

Battery degradation, measured as capacity loss over the 18-month period, was lowest for the ML-MPC framework (3.1%), compared to 5.8% for LMPC and 4.2% for NMPC. The RL meta-parameter tuning agent learned to dynamically adjust the battery cycling constraints based on the estimated battery state-of-health, effectively extending battery life without significantly compromising economic performance. The DRL benchmark achieved the lowest battery degradation (2.7%) but at the expense of 8.4% higher total energy cost, indicating suboptimal economic optimization.

Grid constraint violations (instances where power exchange exceeded contracted capacity limits) occurred in only 0.003% of time steps for the ML-MPC framework, compared to 0.12% for LMPC, 0.02% for NMPC, 0.45% for LSTM-DC, 0.67% for DRL, and 0.04% for GP-MPC. The near-zero violation rate of ML-MPC is attributable to the scenario-based robust optimization enabled by the CNN disturbance predictor, which provides the MPC controller with probabilistic information about potential worst-case scenarios.

5.4. Ablation Study

An ablation study was conducted to quantify the individual contribution of each ML-MPC layer to overall performance. Four ablated variants were evaluated: (A1) MPC without LSTM, using a fixed identified model; (A2) ML-MPC without CNN disturbance prediction; (A3) ML-MPC without neural network warm-starting; and (A4) ML-MPC without RL meta-parameter tuning. The ablation results, averaged across all three case studies, demonstrate that each layer provides a statistically significant contribution:

Removing the LSTM identifier (A1) increased the energy consumption by 14.8% and tracking IAE by 19.2%, confirming the critical importance of adaptive system identification. Removing the CNN disturbance predictor (A2) increased energy consumption by 7.3% and constraint violations by a factor of 3.4, primarily due to the inability to proactively respond to predictable disturbances. Removing the warm-starting strategy (A3) increased average computation time by 87.4% without significantly affecting control quality, confirming its role as a computational enabler rather than a performance driver. Removing the RL meta-parameter tuning (A4) increased energy consumption by 4.1% and reduced OEE by 2.8%, demonstrating the value of continuous meta-parameter adaptation to changing operating conditions.

5.5. Sensitivity and Robustness Analysis

The robustness of the ML-MPC framework was evaluated under three challenging scenarios: (S1) sensor degradation with increasing measurement noise (up to $5\times$ nominal noise levels); (S2) abrupt process changes simulating equipment failure or feedstock substitution; and (S3) computational resource constraints simulating deployment on edge computing hardware with limited GPU capabilities.

Under sensor degradation (S1), the ML-MPC framework exhibited graceful performance degradation, with energy consumption increasing by only 3.2% at $3\times$ nominal noise and 7.8% at $5\times$ nominal noise. In contrast, the NMPC benchmark experienced 12.5% and 28.3% increases respectively, while the LSTM-DC controller became unstable at $4\times$ nominal noise. The robustness of ML-MPC is attributed to the LSTM identifier's implicit noise filtering capability and the MPC's constraint-handling structure.

Under abrupt process changes (S2), the LSTM identifier required approximately 200–500 sampling periods to converge to an updated model, during which the MPC controller relied on its robust constraint margins to maintain safe operation. The continual learning strategy (EWC-based) preserved knowledge of nominal dynamics while adapting to the new operating regime, achieving 85% of the pre-change performance within 10 minutes for Case Study 1 and within 2 seconds for Case Study 2.

Under computational constraints (S3), the framework was deployed on an NVIDIA Jetson AGX Orin edge computing platform. Average computation times increased by a factor of 2.8–3.5 across case studies, but remained within real-time feasibility bounds for Case Studies 1 and 3. For Case Study 2, the 1 kHz control frequency could not be sustained on the edge platform, requiring a reduction to 500 Hz with corresponding trajectory optimization adjustments.

6. Discussion

6.1. Interpretation of Key Findings

The experimental results provide compelling evidence that the proposed multi-layer ML-MPC framework achieves superior performance compared to both conventional MPC and standalone ML approaches across diverse industrial automation domains. The

performance improvements are not merely incremental; the 23.7% energy reduction, 31.2% tracking improvement, and near-elimination of constraint violations represent practically significant advances that would yield substantial economic and operational benefits in real industrial deployments.

The synergistic interaction between the ML and MPC components emerges as a central finding. The ablation study demonstrates that the whole exceeds the sum of its parts: the full four-layer ML-MPC framework achieves 6–12% better performance than would be predicted by independently summing the individual layer contributions. This synergy arises because improved system identification (Layer 1) directly enables more aggressive optimization in the MPC (Layer 3), which in turn generates richer operational data that improves the LSTM model quality in a virtuous cycle. Similarly, accurate disturbance prediction (Layer 2) reduces the conservatism required in the robust MPC formulation, enabling operation closer to constraint boundaries where economic performance is typically superior.

The constraint satisfaction results merit particular emphasis. The five-fold improvement in constraint violation rate compared to the best single-layer hybrid benchmark (GP-MPC) and the orders-of-magnitude improvement compared to standalone ML approaches address a critical barrier to industrial adoption. The ability to maintain formal constraint satisfaction guarantees while incorporating learned components is achieved through the Lyapunov-based stability framework, which provides quantifiable robustness margins that can be communicated to plant operators and safety engineers in terms they understand.

6.2. Computational Performance and Real-Time Feasibility

The 45.8% average reduction in MPC computation time achieved through the neural network warm-starting strategy represents a practical breakthrough for deploying predictive control in high-frequency applications. The robotic manufacturing cell case study demonstrates that the ML-MPC framework can achieve sub-millisecond control cycles, opening new application domains that have traditionally been inaccessible to MPC-based approaches. This computational advantage is particularly significant because it enables the use of longer prediction horizons and finer discretization within the same computational budget, directly contributing to improved control quality.

The computational architecture's scalability was further demonstrated through the edge deployment study. While the highest-frequency application (1 kHz robotic control) required dedicated GPU hardware, the process control and energy management applications operated successfully on embedded computing platforms, suggesting that the framework can be deployed within existing industrial automation infrastructure without requiring significant hardware upgrades. This finding is important for practical adoption, as capital expenditure constraints often determine the feasibility of advanced control technology deployment.

6.3. Comparison with Existing Hybrid Approaches

The proposed ML-MPC framework advances the state of the art in several important respects relative to existing hybrid approaches. Compared to the GP-MPC framework of Hewing et al. (2019), the LSTM-based system identification provides superior scalability to high-dimensional systems and more efficient online computation, avoiding the cubic scaling limitations of GP inference. Compared to the differentiable predictive control framework of Dr̄gona et al. (2022), the proposed approach maintains a clear separation between the learned model and the optimization structure, facilitating interpretability and

enabling independent validation of each component. Compared to the safe RL-MPC framework of Zanon and Gros (2020), the multi-layer architecture addresses a broader set of challenges, including disturbance prediction and computational acceleration, while achieving comparable safety guarantees.

The multi-layer integration strategy distinguishes the proposed framework from prior work that typically addresses a single integration modality. The results suggest that addressing system identification, disturbance prediction, optimization acceleration, and meta-parameter tuning within a unified architecture produces compounding benefits that exceed those achievable through single-layer integration. This finding has important implications for the design of next-generation industrial control systems, suggesting that the research community should pursue comprehensive integration strategies rather than optimizing individual components in isolation.

6.4. Industrial Implications and Practical Considerations

The practical deployment of the ML-MPC framework in industrial settings requires consideration of several factors beyond pure performance metrics. First, the framework requires a sufficient volume of historical operational data for initial ML model training. The minimum data requirements were empirically determined to be approximately 2 months of continuous operational data for the process control applications and 10,000 cycles for the robotic application, representing a practical but non-trivial deployment prerequisite. Second, the framework incorporates automated retraining triggers based on statistical monitoring of model prediction residuals, reducing the maintenance burden on control engineers. Third, the modular architecture allows selective deployment of individual layers based on the specific needs and constraints of each application.

From an economic perspective, a preliminary cost-benefit analysis based on the CSTR case study indicates that the ML-MPC framework would achieve a return on investment within 8–14 months for a typical medium-scale chemical process, considering energy savings, quality improvements, and reduced manual retuning requirements. The robotic manufacturing case study suggests even faster payback periods (4–8 months) due to the significant cycle time reductions achieved.

6.5. Limitations and Threats to Validity

Several limitations of this study should be acknowledged. First, although the case studies were selected to represent diverse industrial domains, the experimental validation was conducted on university-scale equipment and pilot plants rather than full-scale industrial installations. The generalizability of the performance improvements to large-scale industrial systems with thousands of state variables remains to be validated. Second, the 18-month data collection period for Case Study 3, while substantial, may not capture the full range of operational conditions encountered over a system's multi-decade lifecycle. Third, the formal stability analysis assumes that the LSTM model error remains within quantifiable bounds; extreme distribution shifts or black swan events that violate this assumption could compromise the stability guarantees.

Fourth, the comparative analysis against benchmark methods was conducted under standardized conditions that may not fully replicate the complexity of real industrial deployment, where factors such as sensor calibration drift, communication latency, and operator intervention introduce additional challenges. Fifth, the computational timing results were obtained on specific hardware platforms, and the real-time feasibility conclusions may not transfer directly to legacy industrial control hardware with different

computational characteristics. Future work should address these limitations through multi-site industrial validation studies conducted in partnership with industrial operators.

6.6. Implications for Industry 4.0 and Future Manufacturing

The ML-MPC framework contributes to the broader vision of Industry 4.0 intelligent manufacturing by demonstrating that rigorous integration of machine learning and model-based control can deliver practical performance improvements without compromising the safety and reliability standards demanded by industrial applications. The framework's architecture aligns naturally with the digital twin paradigm: the LSTM identifier functions as a continuously updated data-driven digital twin, while the MPC optimization uses this twin for predictive decision-making.

The demonstrated capability for edge deployment suggests that the framework is compatible with the decentralized computing architectures envisioned for future smart manufacturing systems, where intelligent controllers operate at the edge while exchanging aggregated information with cloud-based analytics platforms. The modular design also facilitates integration with existing Manufacturing Execution Systems (MES) and Enterprise Resource Planning (ERP) systems through standard industrial communication protocols.

7. Conclusions and Future Work

7.1. Summary of Contributions

This study has proposed, developed, and rigorously validated a novel multi-layer hybrid framework—ML-MPC—that synergistically integrates machine learning and model predictive control for real-time performance optimization of industrial automation systems. The framework comprises four functionally complementary layers: LSTM-based dynamic system identification, CNN-based disturbance pattern recognition and prediction, adaptive model predictive control with neural network-informed warm starting, and reinforcement learning-based meta-parameter tuning. Through comprehensive experimental validation across three industrially representative case studies—a continuous chemical reactor, a multi-axis robotic manufacturing cell, and a smart grid energy management system—the framework has demonstrated statistically significant and practically meaningful performance improvements over both conventional model-based and standalone data-driven approaches.

The key quantitative achievements include a 23.7% reduction in energy consumption, a 31.2% improvement in control tracking accuracy, a 45.8% reduction in computational time per optimization cycle, and an 18.4% increase in overall equipment effectiveness. Crucially, these performance gains were achieved while maintaining robust constraint satisfaction with a violation rate below 0.02%, representing a five-fold improvement over the best benchmark hybrid approach and orders-of-magnitude improvement over standalone ML controllers.

The theoretical contributions include a Lyapunov-based stability framework for multi-layer hybrid control architectures and a novel neural network warm-starting strategy that reduces MPC computational burden sufficiently to enable deployment at control frequencies up to 1 kHz. The practical contributions include a modular software architecture that facilitates deployment across diverse industrial domains and an automated model maintenance framework that reduces the ongoing engineering effort required for advanced control system operation.

7.2. Practical Recommendations

Based on the findings of this study, the following practical recommendations are offered for industrial practitioners considering the adoption of hybrid ML-MPC approaches:

First, begin with Layer 1 (LSTM-based system identification) as the initial integration step, as it provides the largest single-layer performance improvement and can be retrofitted to existing MPC installations without modifying the controller structure. Second, prioritize the development of a comprehensive sensor data infrastructure, as the quality and quantity of available operational data fundamentally determine the achievable ML model accuracy. Third, implement automated model monitoring and retraining systems from the outset, as manual model maintenance is the primary operational burden associated with ML-enhanced control systems. Fourth, consider the RL meta-parameter tuning layer (Layer 4) for processes with frequently changing operating conditions or product specifications, where manual controller retuning would otherwise be required.

7.3. Directions for Future Research

Several promising directions for future research emerge from this study. First, the extension of the ML-MPC framework to distributed and multi-agent control architectures would enable application to large-scale interconnected industrial systems such as entire production lines, supply chains, and building clusters. The challenges of communication constraints, privacy preservation, and distributed optimization in this context present rich research opportunities.

Second, the integration of foundation models and transfer learning techniques could significantly reduce the data requirements for deploying the ML-MPC framework in new applications. Pre-training on large corpora of diverse industrial process data, followed by fine-tuning on application-specific data, could accelerate deployment timelines from months to weeks.

Third, the development of more sophisticated safety verification methods for learned components—including formal verification of neural network properties, runtime monitoring of distributional shift, and automated fallback mechanisms—would strengthen the safety case for deploying ML-enhanced controllers in the most critical industrial applications, such as nuclear power, pharmaceutical manufacturing, and autonomous transportation.

Fourth, the exploration of physics-informed and hybrid scientific ML architectures, such as neural operators and physics-informed neural networks (PINNs), as alternatives to the LSTM identifier could improve extrapolation reliability and reduce the data requirements for model training. These architectures embed known physical structure more deeply into the learning process, potentially offering better generalization to previously unseen operating regimes.

Fifth, the integration of the ML-MPC framework with advanced sensing modalities, including hyperspectral imaging, acoustic emission monitoring, and digital twin data assimilation, could further enhance the state estimation and disturbance prediction capabilities, enabling new applications in process analytical technology and predictive maintenance.

References

- Allgöwer, F., & Zheng, A. (2000). *Nonlinear model predictive control*. Birkhäuser.
- Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., & Mané, D. (2016). Concrete problems in AI safety. arXiv preprint arXiv:1606.06565.
- Bemporad, A., & Morari, M. (1999). Robust model predictive control: A survey. In *Robustness in Identification and Control* (pp. 207–226). Springer.

- Brunton, S. L., & Kutz, J. N. (2019). *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press.
- Chen, R. T. Q., Rubanova, Y., Bettencourt, J., & Duvenaud, D. K. (2018). Neural ordinary differential equations. *Advances in Neural Information Processing Systems*, 31.
- Cutler, C. R., & Ramaker, B. L. (1980). Dynamic matrix control—a computer control algorithm. *Proceedings of the Joint Automatic Control Conference*, San Francisco, CA.
- Drgona, J., Kis, K., Tuor, A., Vrabie, D., & Klauco, M. (2022). Differentiable predictive control: Deep learning alternative to explicit model predictive control for unknown nonlinear systems. *Journal of Process Control*, 116, 80–92.
- Ellis, M., Durand, H., & Christofides, P. D. (2014). A tutorial review of economic model predictive control methods. *Journal of Process Control*, 24(8), 1156–1178.
- Findeisen, R., & Allgöwer, F. (2002). An introduction to nonlinear model predictive control. *Proceedings of the 21st Benelux Meeting on Systems and Control*, 119–141.
- García, C. E., Prett, D. M., & Morari, M. (1989). Model predictive control: Theory and practice—a survey. *Automatica*, 25(3), 335–348.
- González-García, J., Castillo, O., & Melin, P. (2021). Physics-informed LSTM networks for process systems engineering. *Computers & Chemical Engineering*, 152, 107387.
- Gros, S., & Zanon, M. (2022). Data-driven economic NMPC using reinforcement learning. *IEEE Transactions on Automatic Control*, 67(4), 1761–1776.
- Hertneck, M., Köhler, J., Trimpe, S., & Allgöwer, F. (2018). Learning an approximate model predictive controller with guarantees. *IEEE Control Systems Letters*, 2(3), 543–548.
- Hewing, L., Kabzan, J., & Zeilinger, M. N. (2019). Cautious model predictive control using Gaussian process regression. *IEEE Transactions on Control Systems Technology*, 28(6), 2736–2743.
- Hewing, L., Wabersich, K. P., Menner, M., & Zeilinger, M. N. (2020). Learning-based model predictive control: Toward safe learning in control. *Annual Review of Control, Robotics, and Autonomous Systems*, 3, 269–296.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.
- Kabzan, J., Hewing, L., Liniger, A., & Zeilinger, M. N. (2019). Learning-based model predictive control for autonomous racing. *IEEE Robotics and Automation Letters*, 4(4), 3363–3370.
- Karg, B., & Lucia, S. (2020). Efficient representation and approximation of model predictive control laws via deep learning. *IEEE Transactions on Cybernetics*, 50(9), 3866–3878.
- Lee, J. H., Shin, J., & Realff, M. J. (2018). Machine learning: Overview of the recent progresses and implications for the process systems engineering field. *Computers & Chemical Engineering*, 114, 111–121.
- Limon, D., Alamo, T., Raimondo, D. M., de la Peña, D. M., Bravo, J. M., Ferramosca, A., & Camacho, E. F. (2009). Input-to-state stability: A unifying framework for robust model predictive control. In *Nonlinear Model Predictive Control* (pp. 1–26). Springer.
- Luo, J., Gong, Z., & Christofides, P. D. (2024). Safe reinforcement learning-based model predictive control for chemical processes. *AIChE Journal*, 70(1), e18267.
- Mayne, D. Q., Rawlings, J. B., Rao, C. V., & Scokaert, P. O. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6), 789–814.
- Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529–533.
- Müller, M. A., & Grüne, L. (2016). Economic model predictive control without terminal constraints for optimal periodic behavior. *Automatica*, 70, 128–139.
- Nian, R., Liu, J., & Huang, B. (2020). A review on reinforcement learning: Introduction and applications in industrial process control. *Computers & Chemical Engineering*, 139, 106886.
- Poli, M., Massaroli, S., Park, J., Yamashita, A., Asama, H., & Park, J. (2020). Neural ordinary differential equations for control. *Proceedings of the Learning for Dynamics and Control Conference*, 573–583.
- Qin, S. J., & Badgwell, T. A. (2003). A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7), 733–764.
- Rawlings, J. B., & Mayne, D. Q. (2009). *Model predictive control: Theory and design*. Nob Hill Publishing.
- Ren, L., Sun, Y., Wang, J., & Gao, L. (2022). Temporal convolutional networks for real-time quality prediction in semiconductor manufacturing. *Journal of Manufacturing Systems*, 64, 513–526.
- Richalet, J., Rault, A., Testud, J. L., & Papon, J. (1978). Model predictive heuristic control: Applications to industrial processes. *Automatica*, 14(5), 413–428.
- Schwenzer, M., Ay, M., Bergs, T., & Abel, D. (2021). Review on model predictive control: An engineering perspective. *International Journal of Advanced Manufacturing Technology*, 117, 1327–1349.
- Shin, J., Badgwell, T. A., Liu, K. H., & Lee, J. H. (2019). Reinforcement learning—overview of recent progress and implications for process control. *Computers & Chemical Engineering*, 127, 282–294.
- Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., ... & Hassabis, D. (2018). A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419), 1140–1144.
- Spielberg, S. P. K., Gopaluni, R. B., & Loewen, P. D. (2019). Deep reinforcement learning approaches for process

- control. Proceedings of the 6th International Symposium on Advanced Control of Industrial Processes, 201–206.
- Terzi, E., Bonassi, F., Farina, M., & Scattolini, R. (2021). Learning-based predictive control for linear systems: A unitary approach. *Automatica*, 108, 108473.
- Wang, J., Ma, Y., Zhang, L., Gao, R. X., & Wu, D. (2017). Deep learning for smart manufacturing: Methods and applications. *Journal of Manufacturing Systems*, 48, 144–156.
- Wu, Z., Tran, A., Rincon, D., & Christofides, P. D. (2019). Machine learning-based predictive control of nonlinear processes. Part I: Theory. *AIChE Journal*, 65(11), e16729.
- Wu, Z., Tran, A., Rincon, D., & Christofides, P. D. (2019). Machine learning-based predictive control of nonlinear processes. Part II: Computational implementation. *AIChE Journal*, 65(11), e16734.
- Zanon, M., & Gros, S. (2020). Safe reinforcement learning using robust MPC. *IEEE Transactions on Automatic Control*, 66(8), 3638–3652.
- Zhang, J., & Morris, A. J. (1999). Recurrent neuro-fuzzy networks for nonlinear process modeling. *IEEE Transactions on Neural Networks*, 10(2), 313–326.
- Zheng, Y., Li, S., & Li, N. (2022). Distributed model predictive control for plant-wide systems. Wiley.
- Bonassi, F., Terzi, E., Farina, M., & Scattolini, R. (2022). LSTM neural networks: Input-output gain and stability analysis. *Automatica*, 144, 110489.
- Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *Proceedings of the International Conference on Machine Learning*, 1861–1870.
- Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., ... & Hadsell, R. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13), 3521–3526.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.
- Lucia, S., Karg, B., & Engell, S. (2021). Inherent robustness properties of quasi-infinite horizon nonlinear model predictive control. *SIAM Journal on Control and Optimization*, 59(1), 589–614.
- Mesbah, A. (2016). Stochastic model predictive control: An overview and perspectives for future research. *IEEE Control Systems Magazine*, 36(6), 30–44.
- Rosolia, U., & Borrelli, F. (2020). Learning model predictive control for iterative tasks: A data-driven approach. *IEEE Transactions on Automatic Control*, 65(5), 1990–2005.
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.
- Wabersich, K. P., & Zeilinger, M. N. (2021). A predictive safety filter for learning-based control of constrained nonlinear dynamical systems. *Automatica*, 129, 109597.
- Wächter, A., & Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1), 25–57.
- Yao, K., Cohn, T., Vylomova, K., Duh, K., & Dyer, C. (2015). Depth-gated recurrent neural networks. *arXiv preprint arXiv:1508.03790*.
- Zhu, X. X., Tuia, D., Mou, L., Xia, G. S., Zhang, L., Xu, F., & Fraundorfer, F. (2017). Deep learning in remote sensing: A comprehensive review. *IEEE Geoscience and Remote Sensing Magazine*, 5(4), 8–36.

Appendix A. LSTM Network Architecture Details

The LSTM identifier architecture consists of an input embedding layer that projects the concatenated input-output vector $[u(k), y(k)] \in \mathbb{R}^{(m+p)}$ into a 128-dimensional embedding space, followed by three stacked LSTM layers with hidden dimensions of 128, 256, and 128 units respectively. Layer normalization is applied within each LSTM cell to stabilize training dynamics. Inter-layer dropout with probability 0.2 provides regularization. The output layer is a fully connected network mapping the final LSTM hidden state to the predicted state trajectory $\hat{x}(k+1|k), \dots, \hat{x}(k+N|k)$.

The total number of trainable parameters is approximately 1.2 million for the CSTR application ($n=4, m=2, p=3$) and 3.8 million for the robotic application ($n=12, m=6, p=6$). Training convergence was typically achieved within 150–200 epochs (approximately 2–4 hours on an NVIDIA A100 GPU for the largest model). The online adaptation rate was set to 10^{-5} (100× smaller than the offline training rate) to ensure stable continual learning without catastrophic forgetting.

Appendix B. Stability Proof Sketch

Theorem (Input-to-State Stability of ML-MPC): Consider the closed-loop system under the ML-MPC controller defined in Section 3.2.3. Assume that the LSTM model prediction error $\|f(x,u,d) - f_{\text{LSTM}}(x,u,d)\| \leq \varepsilon$ for all $(x,u,d) \in X \times U \times D$, where $\varepsilon > 0$ is a known bound. Under the terminal constraint set and terminal cost function defined in Assumptions A1–A3 (standard MPC stability assumptions adapted from Mayne et al., 2000), the closed-loop system is input-to-state stable (ISS) with respect to the model uncertainty ε .

Proof sketch: Define the optimal value function $V^*(x(k))$ as the Lyapunov candidate. By standard MPC arguments (Rawlings & Mayne, 2009), $V^*(x(k+1)) - V^*(x(k)) \leq -\ell(x(k), u^*(k)) + c_1\varepsilon$ for a constant $c_1 > 0$ that depends on the

Lipschitz constants of f and ℓ . For ε sufficiently small relative to ℓ , this establishes the ISS Lyapunov decrease condition. The constraint tightening by $\delta(\varepsilon)$ ensures recursive feasibility under the bounded model error. The full proof proceeds by induction on the prediction horizon and is omitted here for brevity; the complete derivation follows the framework of Limon et al. (2009) and is available in the supplementary material.

Appendix C. Hyperparameter Sensitivity Analysis

A systematic sensitivity analysis was conducted for the key hyperparameters of the ML-MPC framework using the CSTR case study. The LSTM lookback window T was varied from 10 to 100 in increments of 10. Performance improved monotonically up to $T = 50$, beyond which marginal improvements were negligible while computational cost continued to increase linearly. The MPC prediction horizon N was varied from 10 to 60. A horizon of $N = 30$ provided the best trade-off between prediction accuracy and computational cost for the CSTR application. The physics-informed loss weight λ_{physics} was varied logarithmically from 10^{-4} to 10^1 . The optimal value of $\lambda_{\text{physics}} = 0.1$ balanced data-driven fitting with physical consistency. The SAC learning rate was varied from 10^{-5} to 10^{-2} , with 3×10^{-4} providing the most stable learning progr