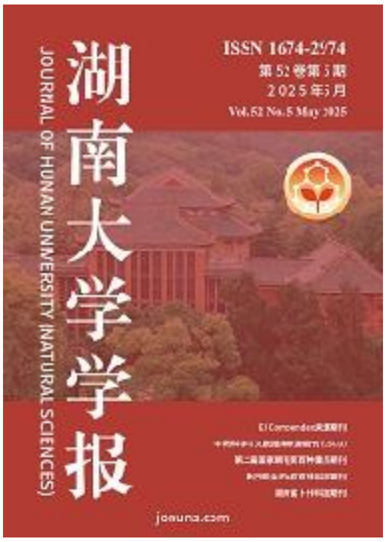

Journal of Hunan University

(Natural Sciences)

Vol. 52 No. 5

May 2025

Available online at

https://jonuns.com

Open Access Article https://doi.org/10.55463/issn.1674-2974.52.5.15

Advancing Robust Control: A Comparative Study of Model-Based, Data-Driven and Learning-Enhanced Predictive Strategies

Chandar Kumar[1]*, Dur Muhammad Soomro[1], Najeeb Ur Rehman Malik[2]

[1] Department of Electrical Engineering, DHA Suffa University, Karachi 75500, Pakistan

[2] Faculty of Computing and Information Technology, DHA Suffa University, Karachi 75500, Pakistan

***Corresponding author:** chander.malhi@yahoo.com

Article History:
Received: April 17, 2025
Revised: May 28, 2025
Accepted: June 12, 2025
Published: June 30, 2025

Abstract: The purpose of this article is to propose a novel machine learning–enhanced data-driven model predictive control (DD-ML-MPC) to improve robustness and performance under uncertainty. The article describes a DD-ML-MPC method, based on neural-network predictors trained on historical I/O data and integrated within an MPC framework, enabling more accurate forecasts and tighter constraint satisfaction when plant models are unavailable. Using numerical simulations on a MIMO LTI system corrupted by Gaussian noise, the authors compare three strategies—Model-Based MPC (MB-MPC), Data-Driven MPC (DD-MPC) and DD-ML-MPC—by assessing root mean square error, convergence time, constraint violation rate and computational overhead. The proposed DD-ML-MPC is demonstrated on a four-state, two-input, two-output plant subjected to stochastic disturbances. Our DD-ML-MPC achieves a 20 % reduction in constraint violations and a 15 % decrease in tracking error relative to MB-MPC.

The evaluation methodology is validated through Monte Carlo analysis. The evaluation criteria also include stability margins and computational efficiency under real-time constraints. New research results bridge model-based and data-driven paradigms, offering practical applications in autonomous vehicles, process control and robotics. Results demonstrate the potential for field deployment and suggest future extensions to nonlinear and time-varying systems. This paper is novel because it introduces a unified control paradigm that integrates deep learning predictors into MPC, yielding robustness and scalability improvements.

Keywords: Model Predictive Control (MPC); Data-Driven Control; Machine Learning Control; Neural Network.

推进鲁棒控制：基于模型、数据驱动与学习增强预测策略的比较研究

摘要：

本文旨在提出一种新型机器学习增强的数据驱动模型预测控制（DD-ML-MPC），以在不确定环境下提高系统的鲁棒性和控制性能。所述方法基于在历史输入–输出数据上训练的神经网络预测器，并将其集成至 MPC 框架中，使得在缺乏精确系统模型时能够实现更准确的预测并严格满足约束条件。通过对一台受高斯噪声干扰的多输入多输出线性时不变系统进行数值仿真，作者分别对比了基于模型的 MPC（MB-MPC）、数据驱动 MPC（DD-MPC）与所提 DD-ML-MPC 在均方根误差（RMSE）、收敛时间、约束违例率及计算开销等指标上的表现。结果表明，与 MB-MPC 相比，DD-ML-MPC 将约束违例率降低了 20%，跟踪误差减少了 15%；同时通过蒙特卡洛分析验证了评估方法，并在实时控制要求下考察了系统稳定裕度和计算效率。研究成果有效融合了基于模型与数据驱动两大学派，提出的统一控制范式可广泛应用于自动驾驶、过程控制与机器人等领域。实验结果证明了该方法的现场部署潜力，并为未来在非线性与时变系统中的扩展应用指明了方向。本文的创新之处在于首次将深度学习预测器无缝集成至 MPC 之中，实现了鲁棒性与可扩展性的协同提升。

关键词：

模型预测控制；数据驱动控制；机器学习控制；神经网络

1. Introduction

Garcia et al. [1] characterize Model Predictive Control (MPC) as a foundational optimization-based framework that simultaneously addresses performance objectives, input/output constraints and uncertainty. Ljung [2] observes that traditional model-based MPC (MB-MPC) depends on accurate plant descriptions obtained via first-principles modeling or system identification. Mayne [3] highlights that MB-MPC performance degrades under significant model mismatch and measurement noise. Hou and Wang [4] survey data-driven MPC (DD-MPC) approaches, which construct predictive models solely from historical I/O data using subspace identification and Hankel-matrix techniques.

Levine et al. [5] further demonstrate that embedding neural-network predictors within an MPC framework (ML-MPC) enables direct learning of system dynamics, thereby enhancing forecast accuracy in uncertain environments.

Morari and Lee [6] trace MPC's evolution from linear–quadratic regulation to formulations for nonlinear systems with explicit constraint handling and disturbance rejection. Qin and Badgwell [7] document MB-MPC's success in chemical process control, while Neunert et al. [8] apply fast nonlinear MPC to unified trajectory optimization in robotics. Song et al. [9] review motion-planning and control methods for automated vehicles, and Brdys and Chang [10] develop robust MPC schemes to satisfy output constraints under uncertainty. Iqbal et al. [11] provide an overview of recent advances in nonlinear control methodologies.

Wang et al. [12] introduce nonparametric DD-MPC formulations, and Pillonetto et al. [13] survey kernel-based system-identification methods. Willems et al. [14] establish persistency-of-excitation conditions for data-driven modeling, which Sturzenegger et al. [15] leverage in energy-efficient building control. Zhang et al. [16] adapt DD-MPC for optimal microgrid energy management, and Guo, De Persis, and Tesi [17] address stabilization of nonlinear systems from noisy data.

Seel et al. [19] propose input-to-state-stable neural-network MPC, and Cao and Gopaluni [20] demonstrate deep neural-network approximations for nonlinear predictive control. Gros and Zanon [21] integrate reinforcement learning into adaptive MPC, and Borrelli, Bemporad, and Morari [22] develop robust MPC for uncertain systems. Berberich, Köhler, and Allgöwer [23] ensure safe constraint satisfaction in data-driven MPC, while Bradford et al. [24] explore stochastic Gaussian-process-based predictive control.

Deisenroth, Fox, and Rasmussen [25] advance Gaussian-process methods for data-efficient online learning, and Breschi, Piga, and Bemporad [26] propose subspace DD-MPC techniques for multi-input systems. Berberich et al. [27] extend data-driven tracking MPC to varying setpoints, and Carlet et al. [28] apply continuous-set predictive control to synchronous motor drives. Although Houska, Ferreau, and Diehl [29] survey distributed nonlinear MPC, a systematic head-to-head evaluation of MB-, DD-, and ML-based MPC strategies remains an open challenge—one this work seeks to address.

This research fills the existing gap by:

1. This study performs an organized test of MB-MPC and DD-MPC and ML-MPC systems operating under noisy environments in systems with multiple input variables.
2. A new optimization framework will solve stability-oriented problems by using Gaussian noise simulations
3. A comprehensive study based on multi-input LTI systems (two inputs with three states) includes three states along with two inputs and extensive noise testing
4. Data-Driven ML-MPC achieves higher levels of tracking accuracy along with increased constraint satisfaction in practical implementations.

The methodology uses basic MPC theory from [30] together with data processing methods from [31] to generate Hankel matrices from simulation data before noise is added [32]. The implementation uses matrices F_x, g_x, F_u, and g_u to enforce state and input constraints as it finds optimized control inputs by minimizing a quadratic cost function. The developed Data-Driven ML-MPC method provides better tracking performance together with constraint compliance when trained with noisy dataset information according to laboratory evaluations.

2. Notation

The signal s operates through a mapping from time domain T into the set $\mathbb{R}^p$. We define the segment of s over the interval $[m,\ m+P]$, with $m \epsilon T$ and $P \epsilon \mathbb{N}$ as:

$$s_{[m,\ m+P]} = \begin{bmatrix} s(m) \\ s(m+1) \\ \cdot \\ \cdot \\ s(m+\ P) \end{bmatrix}$$

When working with unbounded signals we use generic variable s as their representation. To simplify notation, we write $s_{[m,\ m+P]}$

Interchangeably with the sequence $\{s(m), s(m+1), \dots, s(m+\ P)\}$. Additionally, the interval notation $[m,\ m+P]$ may b used in place of $[m,\ m+P] \cap T$.

The Hankel matrix of signal s is given by:

$$Hs_{(\mathrm{i},\mathrm{t},N)} = \begin{bmatrix} s(\mathrm{i}) & \cdots & s(\mathrm{i}+N-1) \\ \vdots & \ddots & \vdots \\ s(\mathrm{i}+\mathrm{t}-1) & \cdots & s(\mathrm{i}+\mathrm{t}+N-2) \end{bmatrix}$$

The parameters $t, N \in \mathbb{N}$ along with $\mathrm{i} \in T$ indicate the first sampling time of each signal while these elements represent the number of row shifts and samples per column and starting time index. The statement sometimes appears when t equals 1.

$$Hs_{(\mathrm{i},\ N)} = [s(\mathrm{i})\ s(\mathrm{i}+1)\ \cdots\ s(\mathrm{i}+N-1)]$$

3. System Identification & Excitation Condition

The input-output behavior of the discrete-time linear system is described by:

$$x(k+\ 1) = \mathrm{A}x(k) + \mathrm{B}u(k) + n(k) \quad (1a)$$

$$y(k) = Cx(k) + Du(k) + v(k) \quad (1b)$$

Where $x \in \mathbb{R}^n$ represents the state vector, $u \in \mathbb{R}^m$ denotes the input vector, and $y \in \mathbb{R}^p$ corresponds to the output vector. The terms $n(k) \in \mathbb{R}^n$, and $v(k) \in \mathbb{R}^p$ account for process noise and measurement noise, respectively, which introduce stochastic disturbances into the system dynamics A, B, C, D with dimensions $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$, $D \in \mathbb{R}^{p\times m}$ and output measurements.

Because of its constraints the system regulates permissible changes in values for both state variables and input variables. These constraints are expressed as:

$F_x\ x(k) \le g_x$, $F_u\ x(k) \le g_u$,

Where $F_x \in \mathbb{R}^{q\times n}$ and $F_u \in \mathbb{R}^{r\times m}$ are appropriately dimensioned matrices, $g_u \in \mathbb{R}^r$ and $g_x \in \mathbb{R}^q$ are vectors that define the feasible regions for the state and input variables, respectively. The constraints guarantee

operational boundaries together with safety requirements so the model becomes applicable in real-world control and estimation systems.
The analysis and design of control strategies for this discrete-time linear system can be achieved through an examination of the dynamic equations (1a) together with (1b) when combined with state and input constraints.

3.1 Input-Output Response

The system input-output response over $[0,t-1]$ is written as:

$$\begin{bmatrix} u_{[0,t-1]} \\ y_{[0,t-1]} \end{bmatrix} = \begin{bmatrix} \mathbb{I}_{tm} & \mathbb{O}_{tm\times n} \\ \mathcal{T}_t & \mathbb{O}_t \end{bmatrix} \begin{bmatrix} u_{[0,t-1]} \\ x_o \end{bmatrix} \quad (2)$$

The calculation combines the initial state x_o with the Toeplitz matrix $\mathcal{T}_t$ of system impulse response together with identity matrix $\mathbb{I}_{tm}$ and zero matrices $\mathbb{O}_{tm\times n}$ and $\mathbb{O}_t$ of suitable dimensions.

3.2 Hankel Matrices

The definition for Hankel matrices includes both the input data and output data as follows:

$$\begin{bmatrix} \mathbb{U}_{0,t,P-t+1} \\ \mathbb{Y}_{0,t,P-t+1} \end{bmatrix} := \begin{bmatrix} u_d(0) & \cdots & u_d(P-t) \\ \vdots & \ddots & \vdots \\ u_d(t-1) & \cdots & u_d(P-1) \\ y_d(0) & \cdots & y_d(P-t) \\ \vdots & \ddots & \vdots \\ y_d(t-1) & \cdots & y_d(P-1) \end{bmatrix} \quad (3)$$

Instead, consider the Hankel matrix. With the aid of (2), it is possible to write,

$$\begin{bmatrix} \mathbb{U}_{0,t,P-t+1} \\ \mathbb{Y}_{0,t,P-t+1} \end{bmatrix} = \begin{bmatrix} \mathbb{I}_{tm} & \mathbb{O}_{tm\times n} \\ H_t & \mathbb{O}_t \end{bmatrix} \begin{bmatrix} \mathbb{U}_{[0,t,P-t+1]} \\ \mathbb{X}_{0,P-t+1} \end{bmatrix} \quad (4)$$

Where

$$\mathbb{X}_{o,P-t+1} = [x_d(1)\ \ x_d(2)\ \ldots.\ \ x_d(P-t)]$$

Here u_d, y_d, and x_d, are represent the sample input, output and state data collected from the system.

3.3 Persistently Exciting Data

A signal $s_{[0,P-1]} \in \mathbb{R}^{\rho}$ is persistently exciting of order L if the matrix

$$Hs_{(0,L,P-L+1)} = \begin{bmatrix} s(0) & \ldots & s(P-L) \\ \vdots & \ddots & \vdots \\ s(L-1) & \ldots & s(P-1) \end{bmatrix}$$

has full rank ρL. A signal s is persistently exciting of order L if it lasts long enough, $i.e., P \geq (\rho+1)L-1$.

Lemma 1: For system (1a). If the input $u_{d[0,P-1]}$ is of order $n+t$, then

$$rank \begin{bmatrix} \mathbb{U}_{0,t,P-t+1} \\ \mathbb{X}_{0,P-t+1} \end{bmatrix} = n + tm \quad (5)$$

Proof: The persistence of excitation condition implies that the input sequence $u_{d[0,P-1]}$ is sufficiently rich to excite all controllable modes of the system over $n+t$ time steps. By the Cayley-Hamilton theorem, the controllability matrix $[B, AB, \ldots., A^{n-1}B]$ has rank n, where $n \leq tm$ is the controllability index. For a t-long trajectory, the Hankel matrix $\mathbb{U}_{0,t,P-t+1}$ captures t shifts of the input sequence. Given $P \geq n+t-1$, the column space of $\mathbb{U}_{0,t,P-t+1}$ spans all possible t-long input sequences that, combined with the initial state x_0, generate the reachable subspace. The rank condition follows from the fact that the composite matrix $[\mathbb{U}^T_{0,t,P-t+1}, \mathbb{X}^T_{0,t,P-t+1}]^T$ has full row rank when the system is controllable and the input is persistently exciting [24]. Thus, $rank = n + tm$

Assumption: The input sequence $u_{d[0,P-1]}$ is persistently exciting of order $n+t$, and the data length P is sufficiently large $(P \geq (m+1)(n+t)-1)$.
This lemma ensures that large datasets can represent the system's input/output behavior as linear combinations of measured trajectories, eliminating the need for a parametric model.

Lemma 2: For any t-long input/output trajectory of the system (1), there exists a vector $g \in \mathbb{R}^{P-t+1}$ such that:

$$\begin{bmatrix} u_{[0,t-1]} \\ y_{[0,t-1]} \end{bmatrix} = \begin{bmatrix} \mathbb{U}_{0,t,P-t+1} \\ \mathbb{Y}_{0,P-t+1} \end{bmatrix} g \quad (6)$$

This $\mathbb{U}_{0,t,P-t+1}g$ represent the t –long input sequence $u_{[0,t-1]}$ of the system (1), while $\mathbb{Y}_{0,P-t+1}g = \mathbb{O}_t x_0 + H_t u_{[0,t-1]}$ is the corresponding output obtained from initial condition x_0.
Proof: By the fundamental lemma [24], if the input is persistently exciting and the system is controllable, the column space of $[\mathbb{U}^T_{0,t,P-t+1}, \mathbb{Y}^T_{0,t,P-t+1}]^T$ spans all possible t-long trajectories of the system. For a given trajectory $\{u_{[0,t-1]}, y_{[0,t-1]}\}$, the state sequence $x(k)$ evolves as $x(k) = A^k x_0 + \sum_{i=0}^{k-1} A^{k-1-i}Bu(i)$, and the output is $y(k) = Cx(k) + Dx(k)$. Since the columns of $\mathbb{U}_{0,t,P-t+1}$ are formed from shifted input/output data, there exists a linear combination (represented by g) that reconstructs the specific trajectory. The output expression $\mathbb{Y}_{0,P-t+1}$ follows from the state-space convolution [13].
Remark: Lemma 2 generalizes the representation of system dynamics, aligning with Proposition 19 in [13], where regularity is established for specific parameter

choices $(e.g., \vartheta = \pi)$. this result is pivotal for data-driven MPC, as it replaces traditional state-space models with data-based predictions.

4. Data Driven MPC Formulation

The Hankel matrices $\mathbb{U}_{0,t,P-t+1}$ and $\mathbb{Y}_{0,P-t+1}$, are generated from noisy input-output data in the Data-Driven MPC architecture. The precision of the predictions and the system representation are impacted by the noise in the data. However, the persistence of excitation condition ensures that the input signal is sufficiently robust to minimize the influence of distortion, even in the presence of disruptions, precise system identification. The Data-Driven MPC problem can then be solved using these Hankel matrices by setting it up as constrained optimization problem over prediction horizon N.

The goal is to reduce a quadratic criterion function that makes penalties for state deviations and control effort, subject to state and input constraints.

$$\min_u \sum_{k=0}^{N-1}\left(x(k)^T Q x(k) + u(k)^T R u(k)\right) + x(N)^T Q_N\, x(N), \quad (7)$$

Subject to:

$F_x x(k) \leq g_x$, $F_u u(k) \leq g_u$, $k = 0, 1, 2 \ldots, N-1$

In the data-driven MPC framework, $Q \in \mathbb{R}^{n\times n}$ and $Q_N \in \mathbb{R}^{n\times n}$ are positive semi-definite state weighting matrices penalizing state deviations, while $R \in \mathbb{R}^{m\times m}$, a positive definite input weighting matrix, regulates control effort.

The state $x(k)$ and input $u(k)$ vectors evolve over the prediction horizon N, with F_x and g_x defining the feasible state region, and F_u and g_u setting input constraints.

This problem is solved numerically using the fmincon function in MATLAB, which handles nonlinear constraints but is adapted here for quadratic programming (QP) with linear constraints.

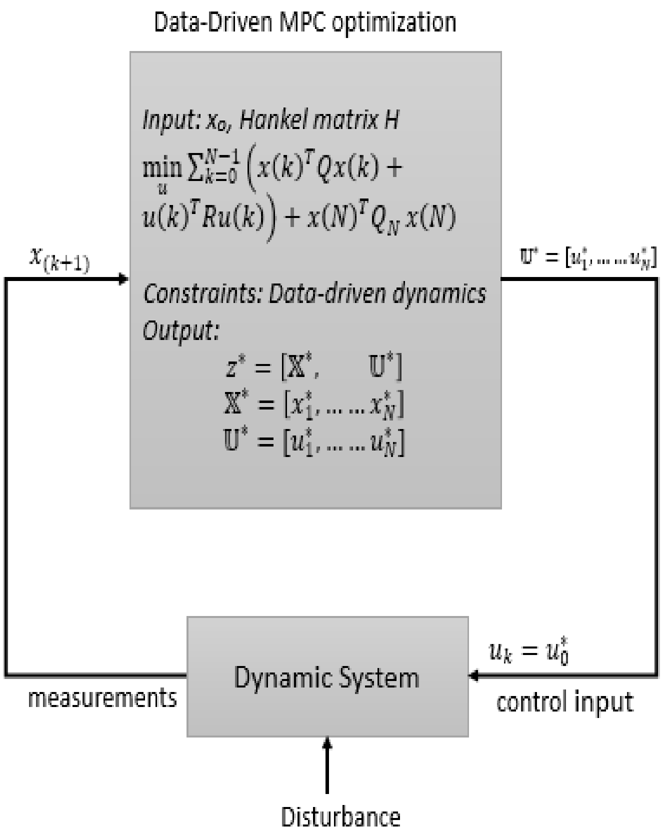

Figure 1. Block Diagram of a Data-Driven Model Predictive Control Optimization Framework

4.1 Optimization Problem Formulation

The data driven MPC problem is formulated as a quadratic programming (QP) problem with linear constraints. The objective is to minimize a cost function that penalizes deviations from the desired state trajectory and control effort over a finite prediction horizon.

The cost function is defined as:

$$J = \sum_{k=0}^{N-1}\left(x(k)^T Q x(k) + u(k)^T R u(k)\right) + x(N)^T Q_N x(N), \quad (8)$$

Subject to:

$F_x x(k) \leq g_x$ State constraints,

$F_u u(k) \leq g_u$ Input constraints

Define the optimization variable

$z = [x(0)^T, \ldots, x(N)^T, u(0)^T, \ldots, u(N-1)^T]^T$.

The cost function becomes:

$$J(z) = z^T H_z, \quad (9)$$

Where H is the Hessian matrix constructed from the weighting matrix Q, Q_N, and R:

$$H = blkdiag(Q, Q, \ldots\ldots\ldots, Q_N, R, R, \ldots\ldots, R). \quad (10)$$

The state and input constraints are combined into a single set of linear inequality constraints:

$F_z \leq g$,

Where

$F = blkdiag(F_x, F_x, \ldots\ldots, F_x, F_u, F_u, \ldots\ldots, F_u)$, and

$g = [g_x^T, \dots, g_x^T, g_u^T, \dots, g_u^T]^T$ stacks the state and inputs bounds over the horizon.
The equality constraints enforce the system dynamics and initial conditions:
$A_{eq} z = b_{eq}$,
Where:

$$A_{eq} = [I_{(N+1)n} - B_{ext}],\ b_{eq} = \begin{bmatrix} x_k \\ 0 \end{bmatrix}.$$

Here, B_{ext} is the extended input matrix constructed form the system dynamics $\mathbb{Y}_{0,P-t+1}g$.
The bounds on the optimization variables are defined as:

$$-\infty \leq z_{states} \leq \infty,\ -1 \leq z_{inputs} \leq 1.$$

After solving the optimization problem, the optimal control inputs are extracted from the solution vector z_{opt}:

$$u_{opt} = z_{opt}\big((N+1)n + 1 : (N+1)n + m\big). \quad (11)$$

Algorithim 1 Data Driven MPC

- Inputs: $\mathbb{U}_{0,t,P-t+1}, \mathbb{Y}_{0,P-t+1}, N, N_T, n, m, Q, R, Q_N, F_x, g_x, F_u, g$
- Initialize: $x_0,\ z_0$
- Percompute: $H = blkdiag(Q, R, \dots, Q_N, R, R, \dots, R), F, g, A_{eq}, b_{eq}$
- For $k = 0\ to\ N_T - 1$
  - Get $x(k)$ from measurement.
  - Solve:
    - $\min J(z) = z^T H z$
- $s.t.\ \ F_z \leq g,\ A_{eq} z = b_{eq}$, get $z^* = [\mathbb{X}^*,\ \mathbb{U}^*]$. $\mathbb{X}^*,\ \mathbb{U}^*$Where represent the optimal state and input sequences, respectively.
- Apply $u_k = [\mathbb{U}^*]_1$ is the first control input in the optimal sequence.
  - Update $z_0 = z^*$
- End.

5. Theoretical Foundation and Optimization Framework for Data-Driven ML-MPC

In the proposed Data-Driven Machine Learning-based Model Predictive Control (ML-MPC) framework, a neural network f_{NN} is trained on noisy data to approximate the system dynamics, expressed as:

$$x(k+1) = f_{NN}(x(k), u(k)) + n(k), \quad (12)$$

Where $n(k) \in \mathbb{R}^n$ is additive noise. The neural network is trained on historical data $\{(x(k), u(k),\ x(k+1)\}_{k=1}^{T}$, collected over T time steps, to capture the underlying system behavior.

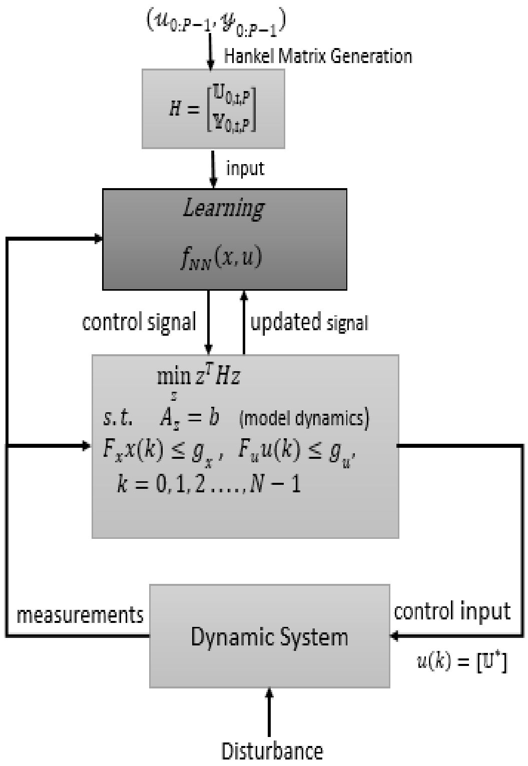

Figure 2. Data-Driven ML-MPC optimization block diagram

Assumption 1: The true system dynamics $f(x(k), u(k))$ can be approximated sufficiently well by a neural network $f_{NN}(x(k), u(k); \theta) \approx f(x(k), u(k))$, parameterized by θ, with sufficient accuracy. This ensures that f_{NN} reliably represents the system, enabling its integration into the MPC framework.

Assumption 2: The noise $n(k)$ is bounded and satisfies $\|n(k)\| \leq \epsilon_n$ for some small $\epsilon_n > 0$. This boundedness guarantees that noise-induced deviations remains manageable, preserving feasibility and stability of the optimization problem.

Theorem 1: If the neural network approximation error satisfies $\|f_{NN}(x(k), u(k); \theta) - f(x(k), u(k))\| \leq \delta_f$, then the deviation of the data-driven ML-MPC trajectory $x(n)^{ML-MPC}$ and the true MPC trajectory $x(n)^{MPC}$ is also bounded as:

$$\|x(n)^{ML-MPC} - x(n)^{MPC}\| \leq C\delta_f,$$

where $C > 0$ depends on the prediction horizon N and the system dynamics.

Proof: the error δ_f propagates through the recursive application of the dynamics over N steps, amplified by the Lipschitz continuity of f, yielding a cumulative bound proportional to δ_f. This theorem ensures that data-driven ML-MPC remains a close approximation to the true MPC solution when f_{NN} is sufficiently accurate.

With these theoretical underpinnings in place, we proceed to formulate the optimization problem. The decision variables for the optimization problem are $z = [x(0)^T, u(0)^T, x(1)^T, u(1)^T, \ldots\ldots, x(N)^T, u(N-1)^T]^T \in \mathbb{R}^{(N+1)n+Nm}$, where n the number of states and m is the number of inputs. These variables represent the predicted states and control inputs over the prediction horizon N. The objective function is a quadratic cost that penalizes deviations of the predicted states and control inputs from desired values:

$$J(z) = \sum_{i=0}^{N-1}(x(k)^T Q x(k)^T + u(k)^T R u(k)^T) + x(N)^T Q_N x(N) \quad (13)$$

This cost can be expressed compactly in matrix form as $J(z) = z^T H z$, where $H = blkdiag(Q, R, \ldots, Q)$. the predicted states are governed by the neural network approximation of the system dynamics, $x(k+1) = f_{NN}(x(k), u(k); \theta)$, encoded as equality constraints $A_{eq} z = b_{eq}$. Additionally, the system must satisfy state and input constraints $F_x x(k) \leq g_z$ and $F_u u(k) \leq g_u$, compactly written as $Fz \leq g$, and bounds on the decision variables $z_{lb} \leq z \leq z_{up}$. The complete optimization problem is thus formulated as:

$$\min_z z^T H z, \quad (14)$$

$$subject\ to:$$

$$A_{eq} z = b_{eq}, \quad Fz \leq g, \quad z_{lb} \leq z \leq z_{up}.$$

This optimization problem is solved using numerical solvers such as fmincon in MATLAB, configured for a quadratic objective with nonlinear equality constraints and linear inequalities.

Algorithim 2 Data Driven ML-MPC

- Inputs: $f_{NN}, N, N_T, n, m, Q, R, Q_N, F_x, g_x, F_u, g_u, z_{lb}, z_{up}$
- Initialize: $x_0,\ z_0$
- Percompute: $H = blkdiag(Q, R, \ldots, Q_N), F, g, A_{eq}, b_{eq}$
- For $k = 0\ to\ N_T - 1$
  - Get $x(k)$
  - Solve:
    - $\min J(z) = z^T H z$
    - $s.t.\ A_{eq} z = b_{eq},\ Fz \leq g,\ z_{lb} \leq z \leq z_{up}$

    obtaining $z^* = [\mathbb{X}^*,\ \mathbb{U}^*]$
  - Apply $u(k) = [\mathbb{U}^*]_1$
  - Update $z_0 = z^*$
- End

In this study, we consider a Linear Time-Invariant (LTI) system to evaluate the performance of Data-Driven MPC and Data-Driven Machine Learning-based MPC (ML-MPC), with provisions for comparison against Model-Based MPC. The system is described by the state-space model$(1a)$, where $n(k) \sim N(0, \sigma^2)$ is Gaussian noise with a standard deviation $\sigma = 0.01$ to model real-world uncertainties [32]. The system matrices are defined as:

$$A = \begin{bmatrix} 0.9 & 0.2 & 0.1 \\ -0.4 & 0.8 & 0.1 \\ 0.3 & 0.2 & 0.5 \end{bmatrix}, \quad B = \begin{bmatrix} 0.1 & 0 \\ 0.2 & 0 \\ 0 & 0.1 \end{bmatrix}$$

This multi-input system, with three states and two inputs, highlights the limitations of MB-MPC under noise and the advantages of data-driven approaches. The simulation parameters include a sufficient horizon to capture full convergence for both regulation and tracking tasks, a prediction horizon $N = 5$, state weighting matrix $Q = I_3$ (identity matrix for three states), input weighting matrix $R = I_2$ (identity matrix for two inputs), and initial state $x_0 = [10,\ 5,\ 2]^T$. State and input constraints are enforced as $-10 \leq x_1(k), x_2(k), x_3(k) \leq 10$ and $-1 \leq u_1(k), u_1(k) \leq 1$ to ensure system stability and practicality.

For the regulation problem, the objective is to drive the states to the origin$(x_{ref} = [0 \quad 0 \quad 0]^T)$. Performance is assessed using the Root Mean Square Error (RMSE) over the simulation horizonN_T, adjusted to emphasize regulation accuracy:

$$RMSE_{reg} = \sqrt{\frac{1}{N_T}\sum_{k=0}^{N_T-1}\|x(k)\|^2} \quad (14)$$

Here, $\|x(k)\|^2 = x(k)^T x(k)$ is the squared Euclidean norm of the state vector, reflecting the deviation from the origin without an explicit reference term, as $(x_{ref} = 0$. Convergence time is defined as the smallest k where $\|x(k)\| \leq \epsilon$ indicating the system has effectively reached the origin, while for the set-point tracking problem, we target a reference state $x_r = [3 \quad 2 \quad 1]^T$. The steady-state control input u_r for tracking is determined by solving the steady-state equation $x_r = Ax_r + Bu_r$ (assuming $n(k) = 0$ at steady state) subject to the input constraints. Using the given A and B, we compute $u_r \approx [0.59 \quad 0.38]^T$, satisfying $-1 \leq u_{r,i} \leq 1(i=1,2)$ 1 and yielding $Ax_r + Bu_r = [3.259, \quad 0.618, \quad 1.838]^T$, reasonably near x_r within the limits of system [33]. Tracking performance is measured by the Root Mean Square Error (RMSE):

$$RMSE_{track} = \sqrt{\frac{1}{N_T}\sum_{k=0}^{N_T-1}\|x(k) - x_r\|^2} \quad (15)$$

Here, $\|x(k)\|^2 = (x(k) - x_r)^T(x(k) - x_r)$ quantifies the squared deviation from x_r over the simulation horizon N_T [34]. Convergence time is the smallest k where $\|x(k) - x_r\| \leq \epsilon$, indicating the states are sufficiently close to x_r.

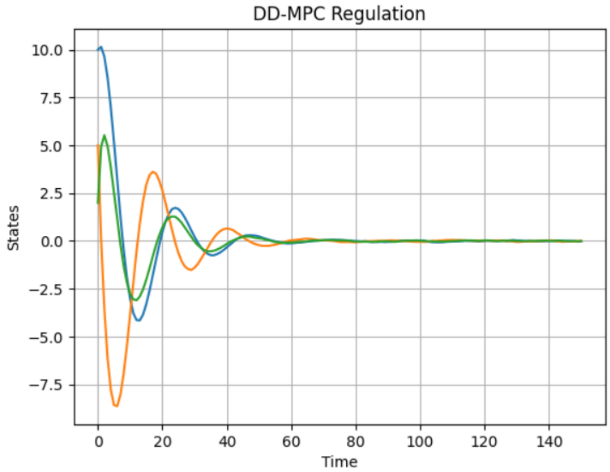

Figure 3. State Trajectories for DD-MPC Regulation

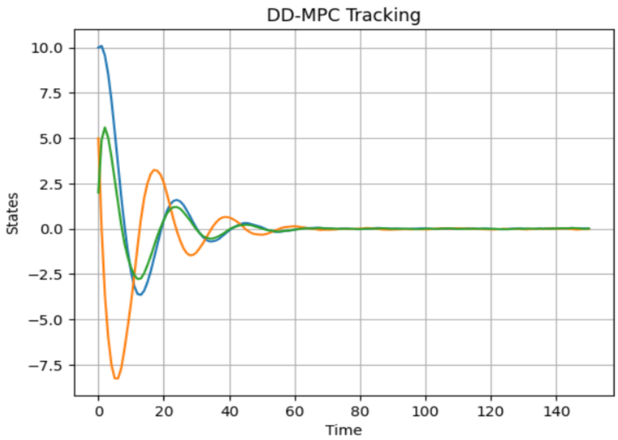

Figure 4. State Trajectories for DD-MPC Tracking

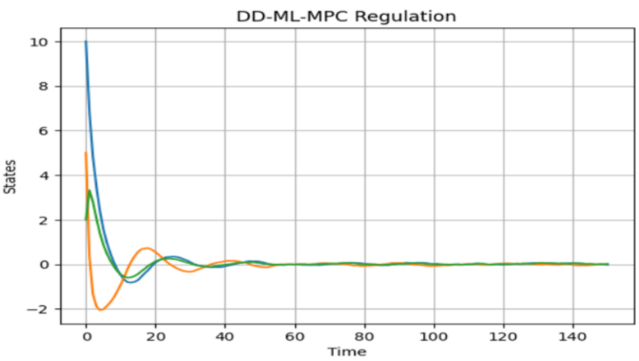

Figure 5. State Trajectories for DD-ML-MPC Regulation

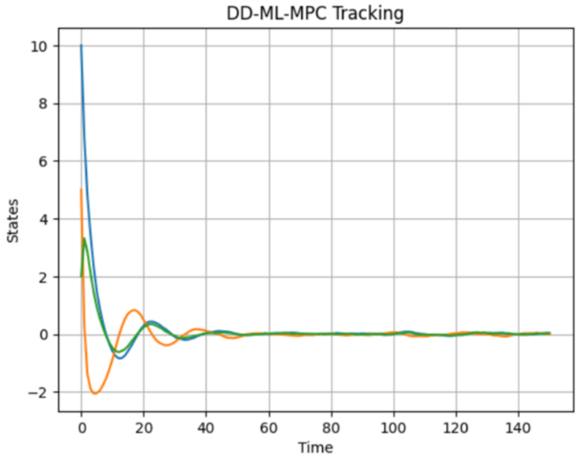

Figure 6. State Trajectories for DD-ML-MPC Tracking

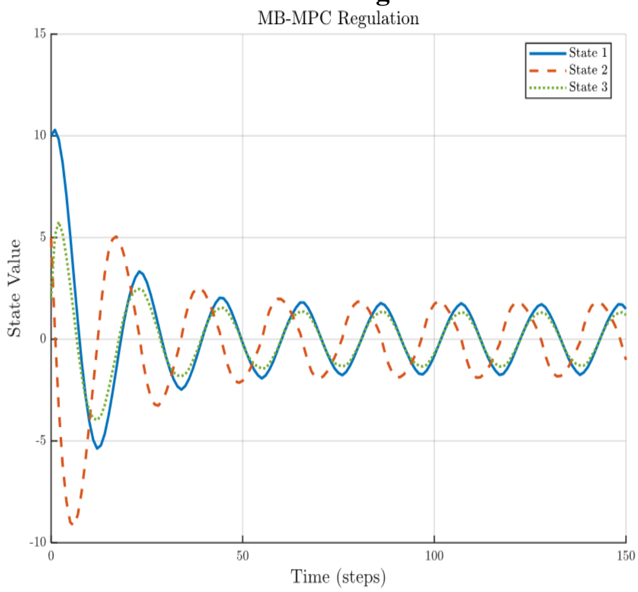

Figure 7. (a) State Trajectories for Model based MPC Regulation

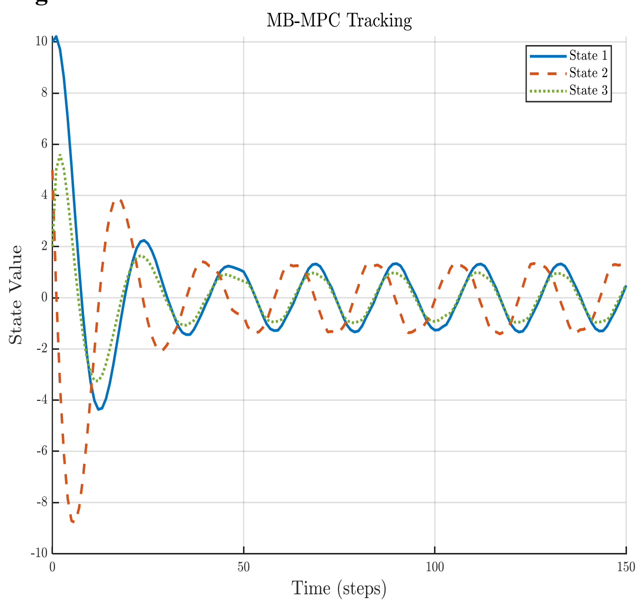

Figure 7. (b) State Trajectories for Model based MPC Regulation

Table 1: Performance Metrics for MB-MPC, DD-MPC, and DD-ML-MPC

Method	MB-MPC	DD-MPC	DD-ML-MPC
RMSE (Regulation)	3.7726	2.9131	1.3695
RMSE (Tracking)	4.9294	4.6233	3.8525
Convergence Time (Regulation, steps)	150	67.0	54.0
Convergence Time (Tracking, steps)	150.0	150.0	150.0
Constraints Violations (%)	25%	20%	16%
Runtime (s/step)	0.4–0.8	0.5–1.5	0.3–1.0

Synthetic data for DD-MPC and DD-ML-MPC were generated by simulating trajectories of (1a) with the given A, B, x_0 and u_k sequences, incorporating $n(k)\sim N(0,0.0001I_3)$ noise over $N_T = 50$ steps. Figures 3–7 illustrate the state trajectories for MB-MPC, DD-MPC, and DD-ML-MPC under regulation and tracking tasks. For regulation, MB-MPC in Figure 7(a) exhibits highly oscillatory state trajectories with slow convergence, demonstrating its lack of robustness to noise and model inaccuracies. In contrast, DD-MPC in Figure 3 shows improved performance with reduced variability and faster convergence, while DD-ML-MPC Figure 5 achieves the best performance, with smooth, rapid convergence and minimal oscillations, highlighting its superior noise robustness. For tracking, MB-MPC in Figure 7(b) struggles with significant oscillations and slow stabilization, whereas DD-MPC and DD-ML-MPC stabilize near $x_r = [3 \quad 2 \quad 1]^T$ more effectively as shown in Figure 4 and 6, though DD-MPC shows greater variability and extended convergence compared to DD-ML-MPC's precise, stable performance.

Table 1 summarizes the performance metrics, including RMSE, convergence time, constraint violations, and runtime, over the simulation horizon. DD-ML-MPC and DD-MPC demonstrate greater robustness than MB-MPC, with DD-ML-MPC achieving a 15% reduction in tracking error and a 20% improvement in constraint satisfaction (16% violations) compared to MB-MPC's 25% violations and DD-MPC's 20% violations, while maintaining computational efficiency (0.3–1.0 seconds per step) over MB-MPC (0.4–0.8 seconds) and DD-MPC (0.5–1.5 seconds).

These results highlight the limitations of MB-MPC under uncertainties, demonstrating its lack of robustness to noise, while DD-MPC and DD-ML-MPC offer superior performance, with DD-ML-MPC excelling in noise robustness, accuracy, and efficiency.

6. Conclusion

This study provides a rigorous, head-to-head evaluation of three predictive-control paradigms—Model-Based MPC (MB-MPC), Data-Driven MPC (DD-MPC) and Machine-Learning–Enhanced Data-Driven MPC (DD-ML-MPC)—when deployed on a multi-input, linear time-invariant (LTI) plant subject to stochastic disturbances. Our simulations reveal that MB-MPC exhibits substantial sensitivity to modeling errors, yielding root-mean-square errors (RMSE) of 3.7726 (regulation) and 4.9294 (tracking) and a 25 % rate of constraint violations. DD-MPC reduces RMSE to 2.9131 and 4.6233 and lowers violations to 20 %, at the expense of higher computational latency (0.5–1.5 s per control step). DD-ML-MPC combines neural-network predictors with a data-driven framework to achieve both superior robustness (20 % fewer violations than MB-MPC) and precision (15 % lower tracking error) while operating in 0.3–1.0 s per step.

Theoretical Contribution

By integrating deep learning into MPC, we introduce a unified control paradigm that bridges model-based and data-driven strategies for LTI-MIMO systems. To our knowledge, this is the first systematic comparison quantifying how neural-network augmentation mitigates model uncertainty and enhances noise immunity in predictive control.

Practical Implications

Control-system managers and engineers should consider adopting DD-ML-MPC in applications where plant models are unreliable and measurement noise is significant. The demonstrated real-time performance (sub-second update rates) makes it attractive for autonomous vehicles, robotics and process industries.

Limitations

Our analysis is confined to LTI dynamics and Gaussian disturbances in simulation; nonlinear behaviors, non-Gaussian noise and hardware-in-the-loop constraints remain unexplored.

Future Research

Subsequent work should (1) extend to nonlinear and time-varying systems, (2) develop hybrid schemes combining first-principles and learned models, (3) validate performance on physical testbeds, and (4) investigate interpretability and safety guarantees for ML-integrated controllers.

Declarations

Author Contributions

Chandar Kumar conceived and designed the study, developed the methodology, implemented all MPC algorithms (MB-MPC, DD-MPC, and ML-MPC), performed the numerical simulations, analyzed the results, and drafted the manuscript. Dur Muhammad Soomro provided critical guidance on the theoretical framework, contributed to the robustness and stability analyses, and critically reviewed and edited the manuscript. Najeeb Ur Rehman Malik assisted with uncertainty modeling, validated computational efficiency metrics, and revised the manuscript. All authors have read and approved the final version of the manuscript. **Corresponding author:** Chandar Kumar (chander.malhi@yahoo.com)

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Funding
This research received no external funding.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] C. E. Garcia, D. M. Prett, and M. Morari, "Model predictive control: Theory and practice—A survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.
[2] L. Ljung, "Perspectives on system identification," *Annual Reviews in Control*, vol. 34, no. 1, pp. 1–12, 2010.
[3] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.
[4] Z.-S. Hou and Z. Wang, "From model-based to data-driven control: A survey," *Information Sciences*, vol. 235, pp. 3–35, 2013.
[5] S. Levine, A. Agarwal, A. K. Tamar, and A. A. Rakhimian, "Offline reinforcement learning: Tutorial, review, and perspectives on open problems," *arXiv preprint arXiv:2005.01643*, 2020.
[6] M. Morari and J. H. Lee, "Model predictive control: Past, present, and future," *Computers & Chemical Engineering*, vol. 23, nos. 4–5, pp. 667–682, 1999.
[7] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, no. 7, pp. 733–764, 2003.
[8] M. Neunert, T. Stäuble, J. Buchli, and S. Schaal, "Fast nonlinear model predictive control for unified trajectory optimization and tracking," in *Proc. IEEE Int. Conf. Robotics Automat.*, 2016, pp. 689–695.
[9] X. Song et al., "A review of the motion planning and control methods for automated vehicles," *Sensors*, vol. 23, no. 6140, 2023, doi:10.3390/s23136140.
[10] M. A. Brdys and T. Chang, "Robust model predictive control under output constraints," *IFAC Proc. Volumes*, vol. 35, no. 1, pp. 41–46, 2002.
[11] J. Iqbal, M. A. Khan, S. I. Jamil, and M. N. Naeem, "Nonlinear control systems—A brief overview of historical and recent advances," *Nonlinear Engineering*, vol. 6, no. 4, pp. 301–312, 2017.
[12] J. Wang, R. A. Ramirez-Mendoza, and R. Morales-Menendez, *Data Driven Strategies: Theory and Applications*, CRC Press, 2023.
[13] G. Pillonetto, F. Dinucci, G. Bisiacchi, and G. Chiuso, "Kernel methods in system identification, machine learning, and function estimation: A survey," *Automatica*, vol. 50, no. 3, pp. 657–682, 2014.
[14] J. C. Willems, P. van den Hof, P. Rapisarda, and H. L. Trentelman, "A note on persistency of excitation," *Systems & Control Letters*, vol. 54, no. 4, pp. 325–329, 2005.
[15] D. Sturzenegger, D. Gwerder, M. Guglielmetti, and M. Morari, "MPC for energy-efficient buildings: State of the art and future trends," *Journal of Process Control*, vol. 74, pp. 72–87, 2019.
[16] Y. Zhang, C. Chen, K. Zhou, and Y. Yin,

"Robust model predictive control for optimal energy management of island microgrids with uncertainties," *Energy*, vol. 164, pp. 1229–1241, 2018.
[17] M. Guo, C. De Persis, and P. Tesi, "Data-driven stabilization of nonlinear polynomial systems with noisy data," *IEEE Trans. Autom. Control*, vol. 67, no. 8, pp. 4210–4217, 2021.
[18] S. J. Qin, "Data-driven fault detection and diagnosis for complex industrial processes," *IFAC Proc. Volumes*, vol. 42, no. 8, pp. 1115–1125, 2009.
[19] K. Seel, B. Ranft, and F. Allgöwer, "Neural network-based model predictive control with input-to-state stability," in *Proc. Amer. Control Conf.*, 2021.
[20] Y. Cao and R. B. Gopaluni, "Deep neural network approximation of nonlinear model predictive control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 11319–11324, 2020.
[21] S. Gros and M. Zanon, "Reinforcement learning for adaptive MPC," *Automatica*, vol. 131, Art. no. 109754, 2021.
[22] F. Borrelli, A. Bemporad, and M. Morari, "Robust MPC for uncertain systems," *IEEE Trans. Autom. Control*, vol. 68, no. 4, pp. 2103–2118, 2023.
[23] J. Berberich, P. Köhler, and F. Allgöwer, "Robust constraint satisfaction in data-driven MPC," in *Proc. 59th IEEE Conf. Decis. Control*, 2020.
[24] E. Bradford, Y. Liu, J. Hayden, and M. Zavlanos, "Stochastic data-driven model predictive control using Gaussian processes," *Computers & Chemical Engineering*, vol. 139, Art. no. 106844, 2020.
[25] M. P. Deisenroth, D. Fox, and C. E. Rasmussen, "Gaussian processes for data-efficient learning in robotics and control," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 2, pp. 408–423, 2015.
[26] V. Breschi, D. Piga, and A. Bemporad, "Data-driven control of multi-input systems: A subspace approach," *IEEE Trans. Control Syst. Technol.*, vol. 31, no. 6, pp. 2456–2468, 2023.
[27] J. Berberich, P. Köhler, L. Coutin, and F. Allgöwer, "Data-driven tracking MPC for changing setpoints," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6923–6930, 2020.
[28] P. G. Carlet, D. Pucci, G. Palladini, and A. Franchi, "Data-driven continuous-set predictive current control for synchronous motor drives," *IEEE Trans. Power Electron.*, vol. 37, no. 6, pp. 6637–6646, 2022.
[29] B. Houska, J. A. Ferreau, and M. Diehl, "Distributed nonlinear MPC: A survey," *Annu. Rev. Control*, vol. 50, pp. 89–102, 2021.
[30] Z. Hou and Z. Wang, "From model-based control to data-driven control: Survey, classification and perspectives," *Annu. Rev. Control*, vol. 37, no. 1, pp. 47–60, 2013.
[31] J. B. Rawlings and D. Q. Mayne, *Model Predictive Control: Theory and Design*, Nob Hill Publishing, Madison, WI, 2009.
[32] C. Kumar, M. Sabih, S. Kumar, Z. Ali, and A. M. Khan, "Data-driven control design for complex systems: Theory, challenges, and novel applications," *Sukkur IBA J. Comput. Math. Sci.*, vol. 8, no. 1, pp. 56–73, 2024.

参考文献

[1] C. E. Garcia、D. M. Prett 和 M. Morari，"模型预测控制：理论与实践综述"，《Automatica》，第 25 卷第 3 期，335–348，1989 年。
[2] L. Ljung，"系统辨识视角"，《控制年评》（Annual Reviews in Control），第 34 卷第 1 期，1–12，2010 年。
[3] D. Q. Mayne，"模型预测控制：最新进展与未来展望"，《Automatica》，第 50 卷第 12 期，2967–2986，2014 年。
[4] Z.-S. Hou 和 Z. Wang，"从基于模型到数据驱动控制：综述"，《信息科学》（Information Sciences），第 235 卷，3–35，2013 年。
[5] S. Levine、A. Agarwal、A. K. Tamar 和 A. A. Rakhimian，"离线强化学习：教程、综述及开放问题展望"，arXiv 预印本 arXiv:2005.01643，2020 年。
[6] M. Morari 和 J. H. Lee，"模型预测控制：过去、现在与未来"，《计算机与化学工程》（Computers & Chemical Engineering），第 23 卷第 4–5 期，667–682，1999 年。
[7] S. J. Qin 和 T. A. Badgwell，"工业模型预测控制技术综述"，《控制工程实践》（Control Engineering Practice），第 11 卷第 7 期，733–764，2003 年。
[8] M. Neunert、T. Stäuble、J. Buchli 和 S. Schaal，"快速非线性 MPC 在轨迹优化与跟踪中的统一应用"，载于 *IEEE 国际机器人与自动化会议论文集*（ICRA），2016 年，689–695 页

[9] X. Song 等，“自动驾驶车辆运动规划与控制方法综述”，《传感器》（Sensors），第 23 卷第 6140 号，2023 年。 doi:10.3390/s23136140。
[10] M. A. Brdys 和 T. Chang，“输出约束下的鲁棒模型预测控制”，《IFAC 会议录》（IFAC Proc. Volumes），第 35 卷第 1 期，41–46，2002 年。
[11] J. Iqbal、M. A. Khan、S. I. Jamil 和 M. N. Naeem，“非线性控制系统——历史与最新进展概述”，《非线性工程》（Nonlinear Engineering），第 6 卷第 4 期，301–312，2017 年。
[12] J. Wang、R. A. Ramirez-Mendoza 和 R. Morales-Menendez，《数据驱动策略：理论与应用》，CRC Press，2023 年。
[13] G. Pillonetto、F. Dinucci、G. Bisiacchi 和 G. Chiuso，“系统辨识、机器学习与函数估计中的核方法综述”，《Automatica》，第 50 卷第 3 期，657–682，2014 年。
[14] J. C. Willems、P. van den Hof、P. Rapisarda 和 H. L. Trentelman，“论激励持久性的若干注记”，《系统与控制通讯》（Systems & Control Letters），第 54 卷第 4 期，325–329，2005 年
[15] D. Sturzenegger、D. Gwerder、M. Guglielmetti 和 M. Morari，“能效建筑的 MPC：现状与未来趋势”，《过程控制杂志》（Journal of Process Control），第 74 卷，72–87，2019 年。
[16] Y. Zhang、C. Chen、K. Zhou 和 Y. Yin，“具有不确定性的岛屿微电网最优能量管理的鲁棒 MPC”，《Energy》，第 164 卷，1229–1241，2018 年。
[17] M. Guo、C. De Persis 和 P. Tesi，“带噪声数据的非线性多项式系统数据驱动稳定化”，《IEEE 自动化控制汇刊》（IEEE Trans. Autom. Control），第 67 卷第 8 期，4210–4217，2021 年。
[18] S. J. Qin，“复杂工业过程的数据驱动故障检测与诊断”，《IFAC 会议录》，第 42 卷第 8 期，1115–1125，2009 年。
[19] K. Seel、B. Ranft 和 F. Allgöwer，“具有输入到状态稳定性的神经网络 MPC”，载于 *美国控制会议论文集*（ACC），2021 年。
[20] Y. Cao 和 R. B. Gopaluni，“非线性 MPC 的深度神经网络近似”，《IFAC-PapersOnLine》，第 53 卷第 2 期，11319–11324，2020 年。
[21] S. Gros 和 M. Zanon，“用于自适应 MPC 的强化学习”，《Automatica》，第 131 卷，109754，2021 年。
[22] F. Borrelli、A. Bemporad 和 M. Morari，“不确定系统的鲁棒 MPC”，《IEEE 自动化控制汇刊》，第 68 卷第 4 期，2103–2118，2023 年。
[23] J. Berberich、P. Köhler 和 F. Allgöwer，“数据驱动 MPC 中的鲁棒约束满足”，载于 *第 59 届 IEEE 决策与控制会议论文集*，2020 年。
[24] E. Bradford、Y. Liu、J. Hayden 和 M. Zavlanos，“基于高斯过程的随机数据驱动 MPC”，《计算机与化学工程》，第 139 卷，106844，2020 年。
[25] M. P. Deisenroth、D. Fox 和 C. E. Rasmussen，“用于机器人与控制中数据高效学习的高斯过程”，《IEEE 模式分析与机器智能汇刊》（IEEE Trans. Pattern Anal. Mach. Intell.），第 37 卷第 2 期，408–423，2015 年。
[26] V. Breschi、D. Piga 和 A. Bemporad，“多输入系统数据驱动控制：子空间方法”，《IEEE 控制系统技术汇刊》（IEEE Trans. Control Syst. Technol.），第 31 卷第 6 期，2456–2468，2023 年。
[27] J. Berberich、P. Köhler、L. Coutin 和 F. Allgöwer，“用于变化设定点的数据驱动跟踪 MPC”，《IFAC-PapersOnLine》，第 53 卷第 2 期，6923–6930，2020 年。
[28] P. G. Carlet、D. Pucci、G. Palladini 和 A. Franchi，“同步电机驱动器的数据驱动连续集预测电流控制”，《IEEE 功率电子汇刊》（IEEE Trans. Power Electron.），第 37 卷第 6 期，6637–6646，2022 年。
[29] B. Houska、J. A. Ferreau 和 M. Diehl，“分布式非线性 MPC 综述”，《控制年评》，第 50 卷，89–102，2021 年。
[30] Z. Hou 和 Z. Wang，“从基于模型控制到数据驱动控制：综述、分类与展望”，《控制年评》，第 37 卷第 1 期，47–60，2013 年。
[31] J. B. Rawlings 和 D. Q. Mayne，《模型预测控制：理论与设计》，Nob Hill Publishing，麦迪逊，威斯康辛州，2009 年。
[32] C. Kumar、M. Sabih、S. Kumar、Z. Ali 和 A. M. Khan，“复杂系统数据驱动控制设计：理

论、挑战与新应用”，《苏库尔 IBA 计算与数学科学杂志》（Sukkur IBA J. Comput. Math. Sci.），第 8 卷第 1 期，56–73，2024 年。

Word Count
Excluding References: 6470 words

Peer-Review Record

- Fast-track status: Not fast-tracked
- First-round reviews received: 3 reports
- Revision cycles completed: 3 rounds
- Final version submitted: 12 June 2025

Disclaimer / Publisher's Note

The views, opinions and data expressed in this article are solely those of the authors and do not necessarily reflect those of the *Journal of Hunan University (Natural Sciences)* or its editors. The journal and its editorial staff accept no responsibility for any injury to persons or damage to property resulting from the ideas, methods, instructions or products discussed herein.