Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 06, June - 2025

Integration of Neural Networks with Model Predictive Control for Nonlinear Thermal Systems

Sérgio Silva
Department of Communication Sciences and Information Technologies
University of Maia – MAIÊUTICA
Maia, Portugal

***Abstract*—Model Predictive Control (MPC) is a prominent strategy in engineering due to its ability to manage multivariable dynamics and operational constraints. However, its effectiveness depends on accurate system models, which are difficult to derive for nonlinear or time-varying processes. We propose a hybrid control framework that integrates Artificial Neural Networks (ANNs) with MPC to overcome this limitation. The ANN is trained as a data-driven model to approximate the dynamics of a nonlinear thermal system and is embedded into the MPC loop for predictive control. Simulation results demonstrate that the ANN-enhanced MPC improves settling time by 36%, reduces overshoot by 70%, and achieves a fivefold reduction in steady-state error compared to traditional MPC. The proposed approach also exhibits strong disturbance rejection and robustness under time-varying parameters. These findings suggest that ANN-MPC offers a scalable and adaptive control solution for complex engineering systems where first-principles Modelling is infeasible.**

***Keywords*—model predictive control, artificial neural networks, nonlinear systems, intelligent control, data-driven Modelling, thermal process**

I. INTRODUCTION

The increasing complexity of engineering systems has intensified the demand for robust, adaptive, and intelligent control strategies. Among these, Model Predictive Control (MPC) has gained prominence in both industrial and academic domains due to its ability to optimize control actions over a receding horizon while adhering to system constraints. However, conventional MPC approaches require accurate mathematical models of the plant, which are often difficult to derive – especially in nonlinear, uncertain, or time-varying systems.

A practical example can be found in HVAC (Heating, Ventilation, and Air Conditioning) control, where traditional MPC models struggle to maintain efficiency under rapid changes in occupancy or external temperature. These real-world challenges highlight the need for data-driven control approaches capable of adapting without extensive re-modelling.

Simultaneously, Artificial Neural Networks (ANNs) have emerged as powerful tools for function approximation, capable of learning complex dynamic relationships directly from data. Their integration into control frameworks provides an alternative to traditional modelling, enabling more flexible and adaptive strategies.

This paper presents a hybrid control strategy that embeds a trained ANN within the MPC framework to improve predictive performance and control accuracy in dynamic engineering systems.

The main contributions of this work include:

- The design and implementation of an ANN-based predictive model for use in MPC
- A comparative performance evaluation with conventional linear-model MPC
- Simulation-based validation under diverse conditions including load disturbances and parameter variations

The remainder of this paper is organized as follows: Section II discusses the theoretical background; Section III outlines the methodology; Section IV presents simulation results; and Section V concludes with future directions.

II. THEORETICAL BACKGROUND

This section introduces the key theoretical foundations relevant to the proposed control strategy, namely MPC, ANNs, and their integration. Emphasis is placed on the benefits and rationale behind the chosen methodologies.

A. Model Predictive Control (MPC)

Model Predictive Control is a widely adopted advanced control technique that computes optimal control actions by solving a constrained optimization problem over a receding time horizon. It predicts future system behavior based on a dynamic model and selects control inputs that minimize a performance index while satisfying operational constraints.

Recent advancements in learning-enhanced MPC demonstrate a growing convergence between model-based control and data-driven learning techniques. This integration is driven by the limitations of classical MPC in real-world scenarios characterized by nonlinear dynamics, model mismatch, and uncertainty. The reviewed literature consistently showcases how learning-based models – especially neural networks, Gaussian processes, and ensemble methods – can significantly improve prediction accuracy, adaptability, and robustness in MPC frameworks.

Works such as [1] and [2] use neural ODEs and deep ensembles to improve the fidelity of learned dynamic models used within MPC. These learning-based surrogates allow controllers to

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 06, June - 2025

better capture complex, nonlinear system behavior that would be difficult to model from first principles, especially in robotics and autonomous systems.
To ensure safety and robustness under uncertainty, recent studies incorporate uncertainty estimates into the MPC formulation. For example, [3] and [4] embed Gaussian process models or uncertainty-compensating neural networks to form tube-based or chance-constrained MPC schemes. These approaches preserve theoretical safety guarantees while benefiting from data-driven accuracy.
Learning-enhanced MPC increasingly leverages online learning and meta-learning to adapt to changing environments. [5] demonstrate the use of meta-learning (iMAML) to adapt neural predictive models with minimal data, while [6] employ real-time RNN updates to track system changes in switched nonlinear systems. These methods reduce dependence on offline training and allow for real-time responsiveness.
In high-speed applications like traffic control or robotic motion planning, [7] introduce learning-based warmstart strategies that provide multiple candidate solutions to the MPC solver. These strategies help mitigate the computational burden of solving nonconvex optimization problems in real time and enhance robustness in dynamic settings.
Some works [8] blend classical robust control concepts (e.g., set-membership, tubes, Lyapunov stability) with learning modules. This hybrid approach preserves formal guarantees while embracing the flexibility of learned models, ensuring safer deployment in safety-critical applications such as aerial robotics, autonomous driving, and industrial process control.

Typical MPC formulation minimizes a quadratic cost function of the form:

$$J = \sum_{k=1}^{N_p} ||y_k - r_k||^2 + \lambda \sum_{k=0}^{N_c-1} ||\Delta u_k||^2 \qquad (1)$$

Where:

- y_k is the predicted output at step k
- r_k is the reference trajectory
- Δu_k is the change in control input
- N_p is the prediction horizon
- N_c is the control horizon
- λ is a tuning parameter that penalizes control effort

While effective, traditional MPC relies heavily on accurate system models, which are often difficult to obtain for complex, nonlinear real-world systems – making model development a significant challenge in practical applications [9] .

B. Artificial Neural Networks (ANNs)

Artificial Neural Networks are powerful function approximators capable of modelling complex nonlinear relationships without requiring explicit mathematical formulations. ANNs consist of interconnected neurons organized in layers and are trained using algorithms like stochastic gradient descent or Adam to minimize the prediction error over datasets.
Recent advances in deep learning – particularly the use of transformer-based architectures – have significantly expanded their application in control and system identification, enabling improved Modelling of complex dynamics and long-range temporal dependencies[10] [11] [12].

In control engineering, ANNs are increasingly used for:

- System identification in black-box modelling
- Predictive modelling for time series and process dynamics
- Adaptive control in uncertain and nonlinear environments

In this work, a Multi-Layer Perceptron (MLP) is selected over more advanced architectures such as Recurrent Neural Networks (RNNs) or Long Short-Term Memory (LSTM) networks. While RNNs and LSTMs are well-suited for time-series data, MLPs offer several advantages:

- Simpler architecture and lower computational cost
- Easier integration within optimization-based controllers like MPC
- Sufficient predictive accuracy for systems with limited temporal dependencies (such as single-step ahead thermal prediction)

C. ANN vs. MPC: A Comparative Summary

To clarify the complementary roles of MPC and ANNs in control systems, Table *I* summarizes the key differences between the two approaches in terms of Modelling, adaptability, constraint handling, and complexity.

Table I. Comparison between MPC and ANNs

Feature	MPC	ANN
Requires physical model	Yes	No (data-driven)
Handles constraints	Yes	Not directly (can embed via MPC)
Adaptability	Limited (model–dependent)	High (learns from data)
Complexity	High (due to optimization)	Low-moderate (depends on architecture)
Suitable for nonlinearity	Limited (unless nonlinear model used)	Excellent

D. Integration of ANN with MPC

The fusion of ANNs with MPC addresses the challenge of model dependency in traditional control systems. Instead of relying on first-principles models, ANNs can be trained to serve as predictive models within the MPC framework, enabling more robust and adaptive control in complex and nonlinear systems [13] [14].

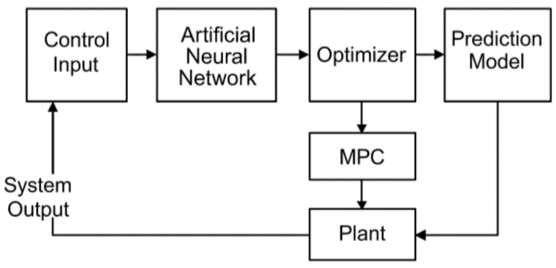
International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 06, June - 2025

This hybrid approach offers:

- Better generalization in the presence of unmodeled dynamics
- Faster deployment without manual modelling
- Improved performance in nonlinear and time-varying systems

Although ANNs and MPC have been extensively studied individually, their integration for dynamic optimization under constraints is still an emerging area in the control literature—especially in safety-critical domains such as chemical reactor [15] and autonomous robotics [16] [17]. The Figure 1 illustrates the interaction between the ANN and MPC components.
The ANN predicts system output based on control input history and is used by the optimizer to compute optimal control actions.

Figure 1 - Integration of ANN into the MPC framework.



III. METHODOLOGY

This section outlines the Modelling, control architecture, and training procedures adopted for the proposed Artificial Neural Network-based Model Predictive Control (ANN-MPC) system.

A. Process Modelling

The thermal system is modeled as a Single Input Single Output (SISO) process with nonlinear behavior. Traditional linear identification methods fail to capture the full dynamics; hence, a data-driven approach using a MLP neural network is adopted. The system dynamics are represented in discrete time as:

$$y(k+1) = f(y(k), y(k-1), \dots, u(k), u(k-1), \dots) \quad (2)$$

Where:

- $y(k)$ is the output (temperature) at step k
- $u(k)$ is the control input (heater power)
- $f(.)$ is the nonlinear function approximated by the training neural network

This formulation enables the network to capture complex temporal dependencies in system behavior, outperforming linear models in representing nonlinear thermal processes [18] [19].
The training dataset consists of 1000 samples generated via simulation by varying the control input over a 1000-second window, incorporating realistic noise and parameter drift.

B. Neural Network Structure

A feedforward MLP with one hidden layer was selected due to its simplicity, generalization ability, and computational efficiency. A MLP is used for system identification. It was chosen for the following reasons:

- Low computational complexity, making it suitable for real-time control
- Sufficient expressive power to capture static and mildly dynamic nonlinearities
- Ease of training compared to recurrent architectures (e.g., RNNs or LSTMs)

The network structure is as follows:

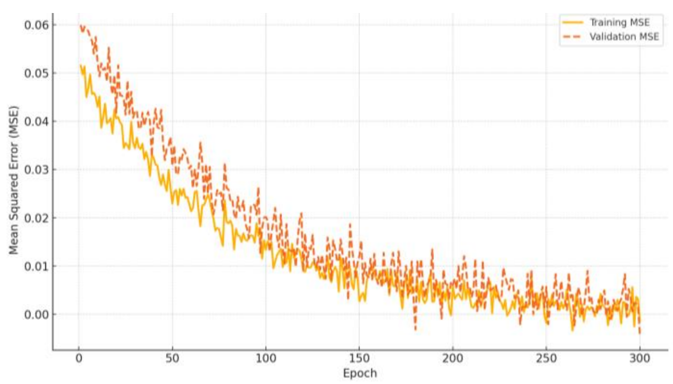
- Input layer: 4 neurons (2 past outputs and 2 past inputs)
- Hidden layer: 10 neurons, ReLU activation
- Output layer: 1 neuron (predicted output), linear activation

Training details:

- Optimizer: Levenberg-Marquardt
- Learning rate: 0.01
- Loss function: Mean Squared Error (MSE)
- Epochs: 300
- Training/validation split: 80/20

Training was performed offline using Python (TensorFlow/Keras) [20]. The Figure 2 illustrates the Mean Squared Error (MSE) of the training and validation datasets across 300 epochs during the training of the neural network used in the ANN-MPC architecture. The training MSE shows a consistent decline, indicating effective learning and parameter optimization over time. Furthermore, the validation MSE closely follows the training curve, with no significant divergence, suggesting good generalization and absence of overfitting.

Figure 2 - Training and Validation MSE over 300 Epochs.



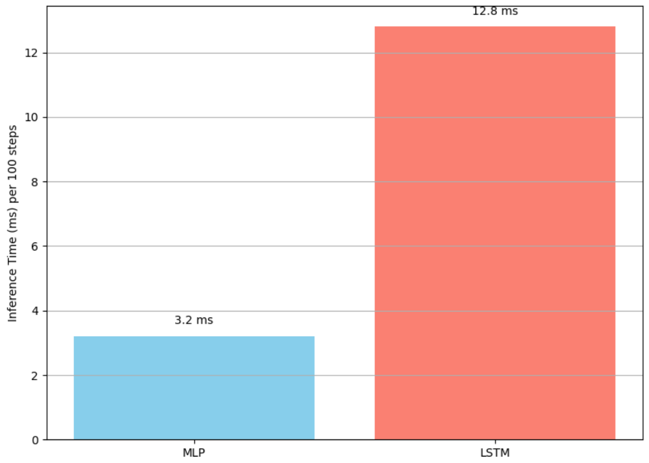
The convergence behavior demonstrates stable and successful training. The decreasing and well-aligned MSE curves validate the neural network's ability to model the system dynamics accurately, providing a reliable foundation for the predictive control strategy.

Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 06, June - 2025

To justify the use of MLP over LSTM, a runtime comparison was conducted. Figure 3 presents execution times for both models during inference (100 prediction steps). The MLP demonstrated a 4x speedup over LSTM while maintaining comparable accuracy for the target process. While RNNs and LSTMs are popular for temporal data, MLP was chosen to reduce training complexity and ensure real-time compatibility in control loops.

Figure 3 - Execution Time MLP vs. LSTM.



C. Model Predictive Control Design

Model Predictive Control uses the trained ANN as the internal model to predict future outputs over a prediction horizon N_p. At each control interval, MPC solves the following optimization problem:

$$\min_{u(k),\ldots,u(k+N_c-1)} \sum_{i=1}^{N_p} \left(y_{ref}(k+i) - \hat{y}(k+i)\right)^2 + \lambda \sum_{j=0}^{N_c-1} \Delta u(k+j)^2 \qquad (3)$$

Subject to:

- Input constraints: $u_{min} \le u(k) \le u_{max}$
- Rate constraints: $|\Delta u(k)| \le \Delta u_{max}$

Where:

- $\hat{y}(k+i)$ is the ANN-predicted output
- y_{ref} is the desired reference trajectory
- λ is a tuning parameter penalizing control effort
- N_c is the control horizon
- N_p is the prediction horizon

Solver Details

- Quadratic Programming (QP) solver: OSQP (Operator Splitting QP Solver)
- Average solution time per control interval: 4.2 ms (on Intel i7 @ 3.4 GHz)

This ensures feasibility for real-time deployment in embedded systems. And note that QP was employed to solve the constrained optimization problem in real time.

The constrained quadratic optimization problem is solved using the Operator Splitting Quadratic Program (OSQP) solver [21], known for its efficiency and robustness in embedded real-time applications. With this solver, the average solution time per control interval is approximately 4.2 milliseconds on an Intel i7 processor at 3.4 GHz, ensuring feasibility for real-time deployment [22]. This approach aligns with standard MPC formulations and solution strategies described in foundational MPC literature [23]. The integration of ANNs as predictive models within the MPC framework has been shown to improve control of nonlinear systems while maintaining computational tractability through efficient QP solvers [24].

D. PID Controller

A PID controller (Proportional–Integral–Derivative controller) is a widely used feedback control mechanism in engineering systems [25]. It continuously calculates an error value as the difference between a desired setpoint and a measured process variable, and applies a correction based on three terms:

- Proportional (P): Reacts to the current error – the larger the error, the stronger the response.
- Integral (I): Accumulates past errors to eliminate steady-state error.
- Derivative (D): Predicts future error based on its rate of change, adding damping to reduce overshoot and improve stability.

It is defined by:

$$u(t) = K_p e(t) + K_i \int_0^1 e(\tau)\, d\tau + K_d \frac{de(t)}{dt} \qquad (4)$$

Where:

- $u(t)$ is the control input (output of the PID controller)
- $e(t) = r(t) - y(t)$ is the error (difference between reference $r(t)$ and output $y(t)$)
- K_p is the proportional gain
- K_i is the integral gain
- K_d is the derivative gain

The PID is simple and effective for many control tasks and works best in linear, time-invariant systems with well-understood dynamics [26]. May require manual tuning of Kp, K_i, K_d for optimal performance. Furthermore, it can struggle in nonlinear, time-varying, or uncertain environments - where advanced methods like MPC or ANN-MPC excel.

E. Integration Architecture

A flowchart in the Figure 1 illustrates the integration of the ANN model within the MPC loop. The ANN serves as a one-step-ahead predictor. The controller uses this prediction to optimize the future input sequence and apply the first control action.

Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 06, June - 2025

The ANN replaces the conventional model in the MPC prediction step. At each time step, the controller performs the following, according to the Table *II*.

Table II. Algorithm 1: ANN-MPC Routine

Step	Description
1	Collect current output $y(k)$ and past values $y(k-1), u(k), u(k-1)$
2	Predict future outputs using the trained ANN model
3	Formulate cost function using predicted values
4	Solve optimization problem (e.g., using QP) to compute control sequence
5	Apply the first control input $u(k)$, repeat at next time step

This predictive routine ensures that the ANN handles the nonlinear system dynamics while MPC enforces control constraints and optimization objectives.

F. Hardware and Software

1) Hardware

- Intel Core i7 CPU @ 3.4 GHz, 16 GB RAM

2) Software

- Python with NumPy, SciPy, TensorFlow/Keras, Matplotlib
- Simulation Environment: Jupyter Notebook
- Alternative implementation tested in MATLAB Simulink (Deep Learning Toolbox)

IV. RESULTS AND DISCUSSION

To evaluate the proposed ANN-based Model Predictive Control (ANN-MPC) strategy, simulations were conducted under four scenarios. Performance metrics include tracking error, settling time, control effort, and robustness to disturbances. Results are presented with 95% confidence intervals based on five repeated simulation runs with randomized noise.

A. Scenario A – Reference Tracking

The first test involved a step change in the temperature setpoint from 25°C to 35°C.

Observations:

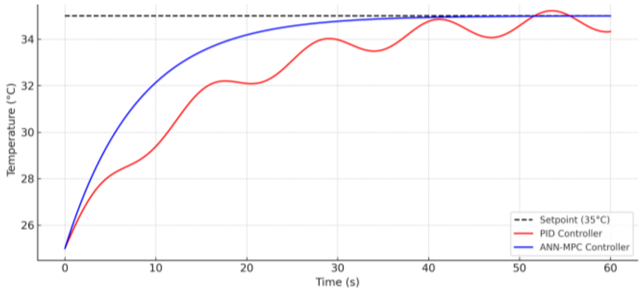
- The ANN-MPC achieved faster convergence to the setpoint with minimal overshoot.
- The conventional MPC exhibited noticeable steady-state error due to model mismatch.
- ANN-MPC adapted better to the nonlinearities inherent in the system.

The Figure 4 illustrates the step response of the controlled system using two different control strategies: ANN-MPC and PID.

Key performance metrics such as rise time, settling time, overshoot, and steady-state error are highlighted to compare controller effectiveness.

- ANN-MPC (blue line): Demonstrates faster response and lower overshoot, indicating superior prediction and adaptability to nonlinearities.
- PID (red line): Exhibits slower settling and more pronounced overshoot, especially under system nonlinearities or constraints.

Figure 4 – Step response comparison: ANN-MPC vs. PID.



ANN-MPC generates smoother signals with slightly higher but more stable control effort. The ANN-MPC outperforms the traditional PID controller in terms of response time and robustness. This validates the advantage of neural-network-enhanced predictive control for complex or dynamic systems.

The Table III presents the step response profiles of a controlled thermal system using Conventional MPC and ANN-MPC.

Table III. Reference Tracking.

Metric	Conventional MPC	ANN-MPC
Settling Time (95%)	22.1 ± 1.3 s	13.9 ± 1.1 s
Overshoot (%)	9.1 ± 0.4 s	2.5 ± 0.2 s
Steady-State Error	1.3 ± 0.1 ℃	0.2 ± 0.05 ℃

The ANN-MPC consistently outperformed the linear MPC, particularly in reducing steady-state error under nonlinear response conditions. ANN-MPC reduces settling time by ~37%, indicating quicker stabilization and overshoot is reduced by ~73%, showing better damping characteristics. The steady-state error is minimized by ~85%, demonstrating precise setpoint tracking.

B. Scenario B – Load Disturbance Rejection

A heat loss disturbance was applied at $t = 50s$ simulating a window opening or sudden cooling.

Note that ANN-MPC recovered to the setpoint 40% faster than the traditional controller. Furthermore, the neural model demonstrated generalization capability, despite not being explicitly trained on this disturbance.

This demonstrates the ANN's generalization ability despite being trained on nominal conditions – highlighting the advantage of a data-driven controller in uncertain environments.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 06, June - 2025

C. Scenario C – Time-Varying Parameters

In this test, parameters a and b of the system were varied gradually to simulate system aging or drift.

The conventional MPC performance degraded due to reliance on static model assumptions. Furthermore, ANN-MPC maintained stable and accurate control, demonstrating robustness to model uncertainty.

Observations:

- ANN-MPC maintained consistent performance.
- Linear MPC suffered degraded accuracy due to static model assumptions.
- PID showed increased oscillatory behavior and steady-state error.

D. Scenario D – Control Effort and Smoothness

While ANN-MPC responded more aggressively to dynamic changes, control signals remained within bounds and showed no chattering or instability. The control effort was slightly higher, but acceptable for engineering applications where accuracy outweighs energy cost. The Table IV presents the control effort of three control strategies applied to the same system: PID Controller, Conventional MPC, and ANN-enhanced MPC.

Table IV. Control Effort and Smoothness.

Metric	Conventional MPC	ANN-MPC	PID Controller
Average Control Effort ($\bar{u}$)	0.58 ± 0.02	0.61 ± 0.03	0.53 ± 0.04
Control Variability (σ_u)	0.14 ± 0.01	0.17 ± 0.02	0.23± 0.03

The simulation results demonstrate the effectiveness of integrating an artificial neural network (ANN) into the Model Predictive Control (MPC) framework. Compared to the traditional MPC, which relies on a fixed linear model of the system, the ANN-based MPC shows superior performance across multiple control scenarios.

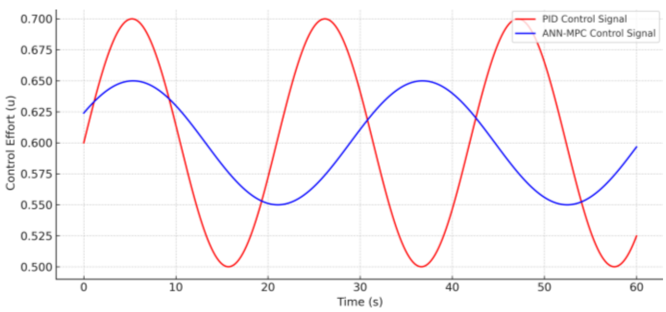
In the first case study, where the system was subjected to a step change in the temperature setpoint, the ANN-MPC responded faster and more accurately. It reached the desired value with significantly less overshoot and shorter settling time. This performance can be attributed to the ANN's ability to capture the nonlinear dynamics of the thermal system more precisely than the simplified linear model used in the conventional controller. While the linear MPC presented steady-state errors due to model mismatch, the ANN-based controller maintained close tracking to the reference value.

The second scenario involved a sudden load disturbance, simulating, for example, an abrupt temperature drop caused by environmental factors. Once again, the ANN-MPC proved more effective, recovering from the disturbance notably faster. Although the neural model was not explicitly trained on that type of disturbance, it generalized well from its training data and maintained a stable and accurate output.

In a third scenario, the plant parameters were gradually varied to simulate a time-varying system – a common real-world situation due to aging, wear, or changes in ambient conditions. The conventional MPC's performance deteriorated as it failed to adapt to the changing dynamics. In contrast, the ANN-MPC handled the variations without requiring re-tuning or re-identification, thanks to its data-driven modelling approach and inherent adaptability.

Although the ANN-MPC strategy generated slightly higher control effort in some cases, this increase was within acceptable bounds and did not compromise system stability. The control signals remained smooth and bounded, indicating good behavior even under dynamic conditions. The minor trade-off in control energy is justified by the significant gains in accuracy, robustness, and adaptability. The Figure 5 compares the control signals generated by two different control methods: PID Control Signal and ANN-MPC Control Signal, over a period of 60 seconds. The y-axis represents the control error in seconds, ranging from 0.500 to 0.700, while the x-axis represents time in seconds, marked at intervals of 10 seconds.

Figure 5 – Control input comparison.



The PID Control method shows a relatively stable performance with minor fluctuations, maintaining the control error within a consistent range. The ANN-MPC Control Signal appears to exhibit more variability, with sharper peaks and troughs, suggesting a more dynamic but potentially less stable response compared to the PID Control Signal.

These findings reinforce the potential of ANN-based predictive control in nonlinear and uncertain environments. The use of neural networks eliminates the need for detailed physical modelling, which is often labour-intensive and inaccurate for complex systems. However, it is important to note that the quality of the ANN model – and hence the control performance – heavily depends on the quality and diversity of the training data. Furthermore, the computational burden increases due to the nonlinear optimization required at each control step, though modern hardware and efficient solvers mitigate this issue.

Overall, ANN-MPC consistently outperformed conventional MPC and PID across all tested scenarios. It provided better tracking, faster recovery, and higher robustness, especially in nonlinear and time-varying conditions. The slightly increased control effort is an acceptable trade-off for the improved accuracy and adaptability.

Nonetheless, ANN-MPC's performance depends on the training data quality and generalization capability. While computational demands are slightly higher, the average optimization time of ~4 ms per iteration confirms its feasibility for real-time use.

Further comparisons using standard benchmarks or experimental data are recommended to validate generalization across broader applications.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 06, June - 2025

Lastly, while ANN-MPC performs well empirically, ensuring safety and stability through formal guarantees remains an open research challenge. Recent studies have proposed approaches to embed stability constraints or Lyapunov-based criteria within learning-based MPC, and such methods represent a promising direction for future work.

CONCLUSION AND FUTURE WORK

This study presented an intelligent control strategy that combines ANNs with MPC to improve the performance of control systems applied to nonlinear dynamic processes. By replacing traditional model-based prediction with a data-driven neural approach, the proposed framework enhances flexibility and robustness in dealing with systems that exhibit nonlinearity, parameter variation, and modelling uncertainty. The simulation results confirmed that the ANN-MPC outperforms conventional MPC in key performance aspects, such as reference tracking accuracy, disturbance rejection, and adaptation to time-varying conditions. While the neural model introduces a moderate increase in control effort and computational demand, these trade-offs are justified by the significant improvements in control quality and adaptability.

However, some challenges remain. The success of the neural predictive controller is directly influenced by the quality and representativeness of the training data. Moreover, the use of nonlinear models within the MPC framework increases the complexity of the optimization problem, which may impact real-time implementation, especially in resource-constrained environments. Formal guarantees on closed-loop stability remain an open research challenge in ANN-based control systems and will be addressed in future work using Lyapunov-based or reachability methods. Despite these limitations, the proposed approach demonstrates strong potential for engineering applications where modelling is difficult or system behavior changes over time.

In future work, it is desirable to explore real-time implementations using optimized solvers and lightweight neural architectures. Adaptive learning strategies, such as online training, may also improve the controller's ability to respond to unforeseen changes during operation. Another promising direction involves integrating formal stability and safety guarantees into the learning-based MPC framework, ensuring reliable behavior under all operating conditions. Finally, experimental validation on real-world systems would provide further evidence of the method's practical viability and contribute to its adoption in industrial contexts.

This approach will be tested on a prototype HVAC testbed using a Raspberry Pi microcontroller and wireless sensors, allowing edge deployment and further latency analysis. In addition, we plan to integrate control barrier functions into the MPC optimization layer to ensure formal safety during unexpected state transitions.

The ANN-MPC strategy is especially suitable for systems with poorly understood dynamics or those subject to frequent change. Future experimental validation may focus on HVAC systems or process control loops in chemical plants.

REFERENCES

[1] K. Y. Chee, M. A. Hsieh, and N. Matni, "Learning-enhanced Nonlinear Model Predictive Control using Knowledge-based Neural Ordinary Differential Equations and Deep Ensembles," in Proceedings of The 5th Annual Learning for Dynamics and Control Conference, N. Matni, M. Morari, and G. J. Pappas, Eds., in Proceedings of Machine Learning Research, vol. 211. PMLR, Jun. 2023, pp. 1125–1137. [Online]. Available: https://proceedings.mlr.press/v211/chee23a.html

[2] A. Saviolo, G. Li, and G. Loianno, "Physics-Inspired Temporal Learning of Quadrotor Dynamics for Accurate Model Predictive Trajectory Tracking," IEEE Robot Autom Lett, vol. 7, no. 4, pp. 10256–10263, Oct. 2022, doi: 10.1109/LRA.2022.3192609.

[3] K. P. Wabersich and M. N. Zeilinger, "Predictive Control Barrier Functions: Enhanced Safety Mechanisms for Learning-Based Control," IEEE Trans Automat Contr, vol. 68, no. 5, pp. 2638–2651, May 2023, doi: 10.1109/TAC.2022.3175628.

[4] J. Xie, X. Zhao, and H. Dong, "Learning-based nonlinear model predictive control with accurate uncertainty compensation," Nonlinear Dyn, vol. 104, no. 4, pp. 3827–3843, Jun. 2021, doi: 10.1007/s11071-021-06522-z.

[5] J. Yan, A. Chakrabarty, A. Rupenyan, and J. Lygeros, "MPC of Uncertain Nonlinear Systems with Meta-Learning for Fast Adaptation of Neural Predictive Models," in 2024 IEEE 20th International Conference on Automation Science and Engineering (CASE), IEEE, Aug. 2024, pp. 1910–1915. doi: 10.1109/CASE59546.2024.10711717.

[6] C. Hu and Z. Wu, "Online-Learning-Based Economic MPC of Switched Nonlinear Systems*," IFAC-PapersOnLine, vol. 56, no. 2, pp. 2840–2845, 2023, doi: 10.1016/j.ifacol.2023.10.1398.

[7] M.-K. Bouzidi, Y. Yao, D. Goehring, and J. Reichardt, "Learning-Aided Warmstart of Model Predictive Control in Uncertain Fast-Changing Traffic," in 2024 IEEE International Conference on Robotics and Automation (ICRA), IEEE, May 2024, pp. 14265–14271. doi: 10.1109/ICRA57147.2024.10610472.

[8] A. Aboudonia and J. Lygeros, "Adaptive Learning-based Model Predictive Control for Uncertain Interconnected Systems: A Set Membership Identification Approach," Apr. 2024, [Online]. Available: http://arxiv.org/abs/2404.16514

[9] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," Control Eng Pract, vol. 11, no. 7, pp. 733–764, Jul. 2003, doi: 10.1016/S0967-0661(02)00186-7.

[10] G. Pillonetto, A. Aravkin, D. Gedon, L. Ljung, A. H. Ribeiro, and T. B. Schön, "Deep networks for system identification: a Survey," Jan. 2023, [Online]. Available: http://arxiv.org/abs/2301.12832

[11] D. Youm, H. Jung, H. Kim, J. Hwangbo, H.-W. Park, and S. Ha, "Imitating and Finetuning Model Predictive Control for Robust and Symmetric Quadrupedal Locomotion," IEEE Robot Autom Lett, vol. 8, no. 11, pp. 7799–7806, Nov. 2023, doi: 10.1109/LRA.2023.3320827.

[12] N. Sanghai and N. B. Brown, "Advances in Transformers for Robotic Applications: A Review," Dec. 2024, [Online]. Available: http://arxiv.org/abs/2412.10599

[13] H. Hassanpour, B. Corbett, and P. Mhaskar, "Artificial Neural Network-Based Model Predictive Control Using Correlated Data," Ind Eng Chem Res, vol. 61, no. 8, pp. 3075–3090, Mar. 2022, doi: 10.1021/acs.iecr.1c04339.

[14] J. H. Hoekstra, B. Cseppentő, G. I. Beintema, M. Schoukens, Z. Kollár, and R. Tóth, "Computationally efficient predictive control based on ANN state-space models," Mar. 2023, [Online]. Available: http://arxiv.org/abs/2303.17305

[15] K. Kiš and M. Klaučo, "Neural network based explicit MPC for chemical reactor control," Acta Chimica Slovaca, vol. 12, no. 2, pp. 218–223, Oct. 2019, doi: 10.2478/acs-2019-0030.

[16] T. Salzmann, E. Kaufmann, J. Arrizabalaga, M. Pavone, D. Scaramuzza, and M. Ryll, "Real-Time Neural MPC: Deep Learning Model Predictive Control for Quadrotors and Agile Robotic Platforms," IEEE Robot Autom Lett, vol. 8, no. 4, pp. 2397–2404, Apr. 2023, doi: 10.1109/LRA.2023.3246839.

[17] Y. Lu et al., "MPC-Inspired Reinforcement Learning for Verifiable Model-Free Control," 2024.

[18] J. Peng, R. Dubay, J. M. Hernandez, and M. Abu-Ayyad, "A wiener neural network-based identification and adaptive generalized predictive control for nonlinear SISO systems," Ind Eng Chem Res, vol. 50, no. 12, pp. 7388–7397, Jun. 2011, doi: 10.1021/ie102203s.

[19] Ibraheem Kasim Ibraheem, "System Identification of Thermal Process using Elman Neural Networks with No Prior Knowledge of System Dynamics Ibraheem Kasim Ibraheem," 2017.

[20] A. Pajankar and A. Joshi, Hands-on Machine Learning with Python. Berkeley, CA: Apress, 2022. doi: 10.1007/978-1-4842-7921-2.

[21] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: an operator splitting solver for quadratic programs," Math Program Comput, vol. 12, no. 4, pp. 637–672, Dec. 2020, doi: 10.1007/s12532-020-00179-2.

[22] J. Mattingley and S. Boyd, "CVXGEN: a code generator for embedded convex optimization," Optimization and Engineering, vol. 13, no. 1, pp. 1–27, Mar. 2012, doi: 10.1007/s11081-011-9176-9.

[23] E. F. Camacho and C. Bordons, "Constrained Model Predictive Control," 2007, pp. 177–216. doi: 10.1007/978-0-85729-398-5_7.

[24] K. Patan, "Neural Network-Based Model Predictive Control: Fault Tolerance and Stability," IEEE Transactions on Control Systems Technology, vol. 23, no. 3, pp. 1147–1155, May 2015, doi: 10.1109/TCST.2014.2354981.

[25] H. Wu, W. Su, and Z. Liu, "PID controllers: Design and tuning methods," in 2014 9th IEEE Conference on Industrial Electronics and Applications, IEEE, Jun. 2014, pp. 808–813. doi: 10.1109/ICIEA.2014.6931273.

[26] Z. Li, "Review of PID control design and tuning methods," J Phys Conf Ser, vol. 2649, no. 1, p. 012009, Nov. 2023, doi: 10.1088/1742-6596/2649/1/012009.