Implementation of control with artificial intelligence elements in industrial control systems using high-performance computing*

Ivo Petráš

Abstract— **This work-in-progress paper discusses a possible implementation of a modern control technique based on artificial intelligence in industrial control systems using high-performance computing. The most widespread industrial control systems today include PLCs, and the most widely used devices for high-performance computing are supercomputers. These devices can be conveniently connected to create almost unlimited possibilities for applying artificial intelligence in control, enabling new, more robust control methods that achieve better control quality. The proposed solution is also presented.**

Index Terms— **Artificial Intelligence (AI), Programmable Logic Controller (PLC), High-Performance Computing (HPC)**

I. INTRODUCTION

A physical system does not control itself. Real systems are dynamic, nonlinear, and with disturbances and uncertainties. Control is the only way to keep the process in a stable and safe state. Artificial intelligence (AI) is increasingly being used in control systems with PLCs, although classic PLCs have limited computing power and are mainly designed for deterministic control. However, AI does not replace control, which guarantees stability, security, and deterministic behavior; it only complements it (by optimizing, predicting, and recommending). Therefore, AI is used by combining PLCs with higher-level systems or specialized devices, for instance, high-performance computing on a supercomputer. Here is an overview of the main options [1], [2], [3]:

1) **AI directly in the PLC** is strictly limited but has growing potential. Some manufacturers already offer simple models directly in the PLC or in PLC-compatible edge modules, for example: Siemens S7 + Industrial Edge Beckhoff TwinCAT ML, B&R Automation + AI Edge. They are mainly useful for state classification, simple regression models, and decision-making without strict time requirements.
2) **Process control** optimization using AI can tune controller parameters (e.g., PID) to minimize energy consumption and improve production quality, with AI calculating optimal settings and the PLC applying them in real time.
3) **Digital twin**, where AI model simulates system behavior, allows testing changes without interfering with production, optimizes control training for AI models, PLCs, and runs real processes, communicating with the simulation.
4) **Predictive maintenance** is the most common practical use of AI in industry; it involves collecting data from sensors, and an AI model evaluates trends and anomalies to predict failures before they occur. Implementation is that the PLC sends data to edge devices, and an AI model generates alerts for the PLC to respond to (alarms, shutdowns, mode changes).
5) **Detection of anomalies and failures** where an AI monitors the "normal behavior" of the system. It detects sensor errors, atypical state combinations, and gradual degradations. It has the advantages of avoiding classic limits, independence from fixed boundaries, and adaptability.
6) **Visual quality control** in which a PLC is connected to a camera system with an AI (neural networks). The application of the defect recognition assembly inspection code and text reading PLC performs product sorting and rejects non-conforming pieces.
7) **Operator support** and autonomous decision-making, where AI is in the superior system, for instance, recommends interventions to the operator, explains the causes of failures, suggests optimal procedures, and then the PLC executes approved commands, ensuring safety.

The advantages are higher efficiency, reduced downtime, and improved production quality. The following limitations apply: safety function certification, deterministic PLC requirements for quality data, and cybersecurity. Thus, AI does not replace the PLC but extends its capabilities, increasing the control system's intelligence and elevating decision-making to a higher level. The implementation of AI in PLCs may help an HPC, where HPC is used as a higher-level computing layer for tasks that are extremely computationally demanding. Below is a practical and realistic explanation of how and where HPC makes sense:

- Training AI models, where HPC is ideal for: training deep neural networks, processing huge historical production data, hyperparametric optimization simulations, and optimization calculations.
- Digital twins and process simulations, where HPC enables highly accurate physical simulations (CFD, FEM).
- Model Predictive Control (MPC) with HPC support for complex systems. HPC solves demanding optimization problems in a simplified model or spreadsheet, while PLC makes fast decisions.

*This research was funded in part by the Slovak Grant Agency for Science under grant VEGA 1/0674/23, and by the Slovak Research and Development Agency under contracts No. APVV-18-0526 and No. APVV-22-0508.

Ivo Petráš is with the Technical University of Košice, Faculty of BERG, Němcovej 3, 042 00 Košice, Slovak Republic ivo.petras@tuke.sk.

- Optimization of control strategies for PID or MPC design controllers, testing changes without risk, PLC remains unchanged, and uses simulation results.
- Production and planning optimization, where HPC solves production scheduling, material flow optimization, and logistics.
- Centralized multi-line analysis, where HPC can analyze data from multiple plants, learn global models, and identify systematic errors.

This work-in-progress paper is organized as follows: Section 1 provides an introduction to the problem and an overview. In Section 2, a hardware configuration is presented. Section 3 describes the proposed examples for future experiments. In Section 4, the paper presents a discussion of possible uses, and it concludes with comments and ideas for further work.

II. HARDWARE CONFIGURATIONS

For this work-in-progress, an industrial system (PLC) and an HPC will be used, located approximately 30 m apart in the same building. Thus, there is a minimal latency.

A. Industrial Control System - PLC

Industrial control systems, including PLCs, operating/monitoring devices such as industrial PCs and Power Panels, and uninterruptible power supplies, have all been designed, developed, and manufactured for conventional use or for applications of object/process control with increased safety requirements and safety technology in industry.

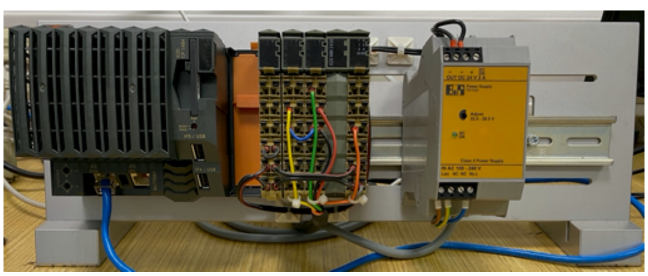

Fig. 1. PLC in the laboratory at the Technical University of Košice.

In Fig. 1, a PLC X20 manufactured by B&R located at the Laboratory of Industrial Control Systems at the Technical University of Košice, Slovak Republic, is depicted. This PLC has a 1.6 GHz CPU, 2 GB of RAM, Ethernet connectivity, and various I/O modules.

B. High-Performance Computing - Supercomputer PERUN

The supercomputer PERUN, operated at the Technical University of Košice, is a technological pillar of the Slovak research infrastructure. It is designed to provide cutting-edge computing power for projects in artificial intelligence, modeling and simulation, big data processing, and next-generation scientific research – launched in November 2025.

The supercomputer PERUN (https://hpc.tuke.sk), shown in Fig. 2, has: 12,800 processor cores and an accelerated computing unit with 208 NVIDIA H200 GPUs, among the

Fig. 2. Supercomputer PERUN at the Technical University of Košice.

most powerful AI accelerators in the world. The system is equipped with 29.3 TB of GPU-allocated memory and 115.2 TB of system memory, enabling it to handle even the most complex and large-scale tasks. High throughput and efficient cooperation among computing nodes are ensured by the InfiniBand network infrastructure at 400 Gbit, which connects the entire system into a single high-performance unit. The total installed performance of the supercomputer PERUN reaches 10.7 PFlops (Rmax), which makes it one of the most modern HPC systems in the EU.

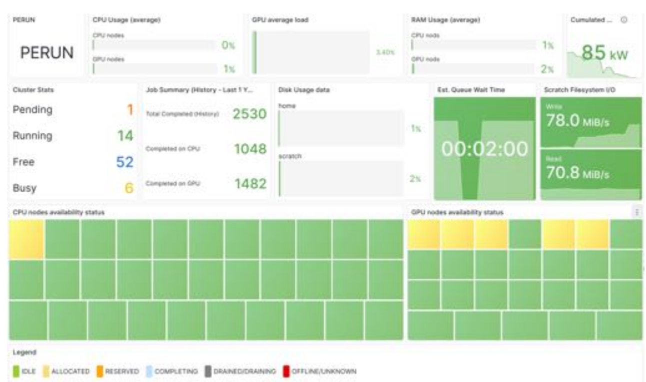

Fig. 3. Design of the PERUN supercomputer dashboard.

In Fig. 3, the dashboard of the supercomputer PERUN accessed on January 22, 2026, is depicted.

Typical work for the HPC layer is model training (predictive maintenance, condition classification, vision), simulations/digital twin (FEM/CFD, multi-body, process models), optimization (planning, parameter setting, PID/MPC design), and model validation and testing (drift, robustness).

III. PROPOSED WORKOUT EXAMPLES

Using the PLC and HPC described in the previous subsection, we may propose a configuration that enables the use of an AI element.

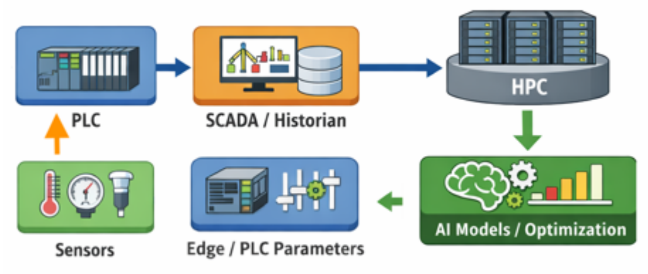

Fig. 4. Typical architecture for using PLCs and HPCs with AI models.

Fig. 4 shows a typical architecture of using PLC and HPC with AI models. Sensors on the controlled object provide signals to the PLC's input modules, which collect the data and send it to the SCADA system. This data is used to train AI models and to run the optimization algorithm on the HPC. Edge layer (online inference + integration) is for low-latency data decision-making and preprocessing. Edge is the "transmission" between the PLC and the HPC, taking raw data from the PLC and sending only approved recommendations (parameters, setpoints, or alarms) to the PLC. Some AI algorithms used in PLC can be found in [4].

Additionally, we provide several objects for further consideration of AI implementation in control using an HPC.

A. Safety Control

Safety control must consistently be implemented on a certified Safety PLC - it is the undisputed basis of every automation, dark factory, and AI-controlled system. However, we should keep in mind that the safety PLC has absolute authority. AI can only recommend, predict, and optimize, but a safety PLC can ignore AI.

Fig. 5. Experimental setup for safety control with PLC.

In Fig. 5, an experimental setup for safety control with a PLC located in the laboratory at the Technical University of Košice is depicted.

B. Control of a Motor

As an example of use in practice, we may demonstrate it for predictive maintenance and optimization, where the PLC controls the motor/drive, collects current/speed/temperature, and the Edge performs an FFT of vibrations and infers "bearing health". Finally, HPC trains a model on months of data and searches for better feature sets. Edge sends a PLC with a health score of 0–100 an alarm: "probable failure within 7 days," with a recommendation to reduce load by a specific value, for instance 10% (only if the guard allows).

Fig. 6. Experimental setup of AC motor, inverter, and PLC.

In Fig. 6, an experimental setup for an AC motor control with a PLC located in the laboratory at the Technical University of Košice is depicted.

C. Temperature Control

Nowadays, temperature control with AI is most often done so that AI does not maintain the temperature directly, but instead helps the classic controller (PID/MPC): it predicts, adapts parameters, compensates for disturbances, and optimizes setpoints. This way, the system remains stable and, at the same time, "intelligent".

Typically, in basic control, AI selects PID "mode" (gain scheduling) or suggests fine retuning. In advanced control, AI adjusts the setpoint trajectory to maintain quality, minimize consumption, and reduce power cycles.

In Fig. 7, an experimental setup for temperature control with a PLC located in the laboratory at the Technical University of Košice is depicted.

For this kind of control, we may use a fractional-order controller with a higher degree of freedom (more parameters).

Fig. 7. Experimental setup for temperature control with PLC.

Instead, the classical PID controller, a PI$^\lambda$D$^\delta$ controller [5] or a fractional-order MPC controller [6] may be used. AI can also help tune/retune the fractional-order controller [7].

The fractional-order PI$^\lambda$D$^\delta$ controller (FOC) has been proposed as a generalization of the PID controller with integrator ${}_0D_t^{-\lambda}$ of real order λ and differentiator ${}_0D_t^{\delta}$ of real order δ. In the time domain, the fractional differential equation of the FOC has the form:

$$u(t) = K_p\, e(t) + T_i\, {}_0D_t^{-\lambda} e(t) + T_d\, {}_0D_t^{\delta} e(t), \qquad (1)$$

where K_p is the proportional constant, T_i is the integration constant and T_d is the differentiation constant. Building on this, we can write a formula for a non-linear fractional-order PI$^\lambda$D$^\delta$ controller (NFOC) of the form:

$$u(t) = f(e)\left[K_p\, e(t) + T_i\, {}_0D_t^{-\lambda} e(t) + T_d\, {}_0D_t^{\delta} e(t)\right], \qquad (2)$$

where $f(e)$ is a non-linear function of the variable e. Various definitions of the non-linear function $f(e)$ can be used, for instance [7]: $f(e) = K_0 + (1 - K_0)|e(t)|$, for $K_0 \in \langle 0, 1\rangle$. Specifically, when $K_0 = 1$, the controller reduces to the linear fractional-order form given in equation (1).

When we use a Grünwald-Letnikov method for the fractional derivative/integral approximation, we may consider the time step of calculation as a sampling period T_s and approximate continuous time t as a set of points with respect to the sampling interval T_s, with $t \approx kT_s$, where k is a position of a point in discrete time. With this in mind, the discrete version of the NFOC takes the following form, as shown in [7]:

$$\begin{aligned} u(k) &= f(e)[K_p e(k) + T_i T_s^{\lambda} \sum_{j=v}^{k} w_j^{(-\lambda)} e(k-j) \qquad (3)\\ &+ T_d T_s^{-\delta} \sum_{j=v}^{k} w_j^{(\delta)} e(k-j)], \quad k = 1, 2, 3, \ldots, \end{aligned}$$

where $w_j^{(-\lambda)}$ and $w_j^{(\delta)}$ are binomial coefficients calculated according to the following relation:

$$w_0^{(\alpha)} = 1, \qquad w_j^{(\alpha)} = \left(1 - \frac{1+\alpha}{j}\right) w_{j-1}^{(\alpha)}. \qquad (4)$$

Additionally, v depends on memory length L_m as follows:

$$\begin{aligned} v &= 0 \quad \text{for} \quad k < (L_m/T_s),\\ v &= k - (L_m/T_s) \quad \text{for} \quad k > (L_m/T_s). \end{aligned}$$

To calculate the control error e, we use the standard formula $e = r - y$, where r is the reference value and y is the measured value.

For implementation purposes, the neural network (NN) with three single neurons is suggested with structure as it is shown in Fig. 8. In our controller (3), each neuron is dedicated to one of three controller parameters: K_p, T_i, and T_d, parameters λ and δ (orders) are included in the weights $w_j^{(\alpha)}$. The output neuron just sums signals from hidden layer neurons to produce the control signal $u(k)$.

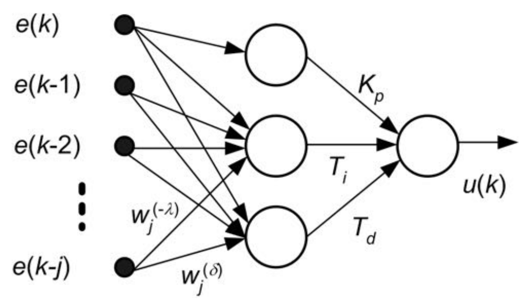

Fig. 8. Structure of the NN for the FOC implementation [7].

Input signals are the history of control errors $e(k)$, with $e(k)$ representing the error at time k and the input vector consisting of the latest L_m samples. The weighted sum of these inputs, plus threshold ϕ, forms the linear function u_j as follows:

$$u_j = \sum_{j=1}^{n} w_j^{(\alpha)} e(k-j) - \phi.$$

The signal u_j is processed by the nonlinear transfer function $f(u_j)$ to generate the output signal of the neuron y_j, i.e., $y_j = f(u_j)$. As a transfer function, a modified sigmoidal function can be used:

$$y_j = f(u_j) = c\frac{1 - e^{-b.u_j}}{1 + e^{-b.u_j}},$$

where c is the saturated level and b is the slope of function.

The parameters of the NFOC can be determined in order to minimize a defined cost function:

$$MSE = \frac{1}{Q}\sum_{k=1}^{Q} e^2(k). \qquad (5)$$

The choice of an appropriate cost function directly affects controller performance. Here, the total mean-squared error (MSE) of the closed-loop system response to a unit-step input is used as the cost function (5). Assuming the closed-loop system error $e(k)$ is calculated for Q samples, the search for controller parameters (tuning) becomes a numerical optimization problem.

Figure 9 illustrates the potential structure of the NN, which may be utilized for NFOC tuning. The NN comprises three

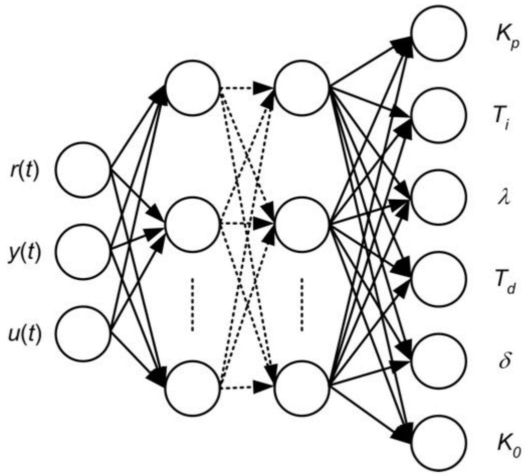

Fig. 9. Structure of the NN for tuning the NFOC parameters [7].

input neurons, six output neurons, and a variable number of hidden layers. The training of this NN will be done entirely in the supercomputer PERUN, as federated or continuous learning, which updates the NFOC's parameters at the Edge.

IV. DISCUSSION AND CONCLUSIONS

All of the described technologies and AI elements can significantly increase the quality of process control and ultimately the quality of products, minimize energy consumption in production, and increase production.

As mentioned in the previous section, three different objects and three different AI techniques will be used in the future to demonstrate the implementation of control with artificial intelligence elements in industrial control systems using high-performance computing, namely:

- safety control in which AI only recommends, predicts, and optimizes decisions, while the PLC can ignore them - AI must never increase risk,
- control of an AC motor, where AI can perform predictive maintenance and optimization,
- temperature control where AI predicts, adapts parameters, compensates for disturbances, and optimizes setpoints for PID/PI$^\lambda$D$^\delta$ or fractional-order MPC control.

Besides the neural network type, the machine learning or deep learning algorithm, and data analytics, communication protocols (OPC UA Server/Client), SCADA, Edge, and network topology must also be considered.

In addition to the above considerations, many other factors, including the rapid development and application of digital technologies such as the industrial internet, robotics and automation, and AI, are accelerating the emergence of the concept of dark factories, especially in China [8].

A dark factory is a manufacturing plant or production line that operates without the permanent presence of humans, requires no lighting or air conditioning for humans, and is fully automated and autonomously controlled by a combination of PLCs, robots, AI, and IT systems. Thus, a dark factory is a system in which a PLC ensures safe movement, Edge AI makes local decisions, and HPC trains the factory to get better every day – without a human having to stand there. Control and control systems are not just necessary - they are the foundation without which AI, HPC, or dark factories would have nothing to control.

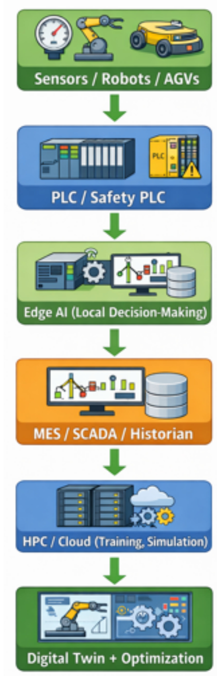

Fig. 10. Typical dark factory architecture.

In Fig. 10 a typical dark factory architecture is shown.

Finally, the HPC, in conjunction with the PLC, serves as the "brain for learning and optimization," while the PLC remains the "nervous system" for safe, real-time control.

REFERENCES

[1] L. Körösi and S. Kajan, "Overview of implementation principles of artificial intelligence methods in industrial control systems," *Journal of Electrical Engineering*, vol. 76, no. 1, pp. 99–105, 2025, doi: 10.2478/jee-2025-0010.

[2] Y. G. Achamyeleh, S. Y. Yu, G. Q. Araya, and M. A. Al Faruque, Bridging the Binary Analysis Gap: A Cross-Compiler Dataset and Neural Framework for Industrial Control Systems. In: Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '2025). Association for Computing Machinery, New York, NY, USA, pp. 5260–5269, doi: 10.1145/3711896.3737373.

[3] E. Carpanzano (Ed.), *Advances in Artificial Intelligence Methods Applications in Industrial Control Systems*, MDPI, 2022, ISBN 978-3-0365-6809-6.

[4] M. Rodriguez, N. Dhameliya, and B. Patel. AI-Driven Programmable Logic Controllers: Enhancing Automation in Industrial Processes. *Robotics Xplore: USA Tech Digest*, vol. 1, no. 1, pp.1–14, 2024.

[5] I. Petráš, "Fractional-Order Controllers: A Practical Approach," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 55, no. 12, pp. 9570–9579, Dec. 2025, doi: 10.1109/TSMC.2025.3614290.

[6] I. Petráš, "Novel Fractional-Order Model Predictive Control: State-Space Approach," *IEEE Access*, vol. 9, pp. 92769–92775, 2021, doi: 10.1109/ACCESS.2021.3093364.

[7] I. Petráš, A note on fractional-order non-linear controller: possible neural network approach to design. In: Proceedings of the International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, Canada, 2016, pp. 603–608, doi: 10.1109/IJCNN.2016.7727255.

[8] Y. Liu and J. Huh, "Smart Factory to Dark Factory in China: A Systematic Survey on Managerial Transformation," *J. Inf. Commun. Converg. Eng.*, vol. 23, pp. 336–354, 2025, doi: 10.56977/jicce.2025.23.4.336.