ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TUPDP102

# LEVERAGING LOCAL INTELLIGENCE TO CERN INDUSTRIAL CONTROL SYSTEMS THROUGH EDGE TECHNOLOGIES

A. Patil, F. Varela, F. Ghawash, B. Schofield, CERN, Geneva, Switzerland,
T. Kaufmann, A. Sundermann, D. Schall, Siemens AT - T DAI DAS, Austria,
C. Kern, Siemens DE - T CED SES, Germany

*Abstract*

Industrial processes often use advanced control algorithms such as Model Predictive Control (MPC) and Machine Learning (ML) to improve performance and efficiency. However, deploying these algorithms can be challenging, particularly when they require significant computational resources and involve complex communication protocols between different control system components. To address these challenges, we showcase an approach leveraging industrial edge technologies to deploy such algorithms. An edge device is a compact and powerful computing device placed at the network's edge, close to the process control. It executes the algorithms without extensive communication with other control system components, thus reducing latency and load on the central control system. We also employ an analytics function platform to manage the life cycle of the algorithms, including modifications and replacements, without disrupting the industrial process.

Furthermore, we demonstrate a use case where an MPC algorithm is run on an edge device to control a Heating, Ventilation, and Air Conditioning (HVAC) system. An edge device running the algorithm can analyze data from temperature sensors, perform complex calculations, and adjust the operation of the HVAC system accordingly. In summary, our approach of utilizing edge technologies enables us to overcome the limitations of traditional approaches to deploying advanced control algorithms in industrial settings, providing more intelligent and efficient control of industrial processes.

## INTRODUCTION

The latest advances in AI and ML, along with time-tested methods like MPC, offer new ways to enhance the functionality of industrial control systems [1]. For instance, these techniques can improve system reliability through anomaly detection, enable energy-efficient operation of complex industrial processes, and extend equipment life through predictive maintenance. Nevertheless, enhancing industrial control systems through such techniques poses several challenges.

One significant challenge is ensuring that the core processes of the system and the algorithms operate independently and do not interfere with one another. This process independence ensures that the demands of complex algorithms do not jeopardize the safe operation of the core process and overburden its resources. Also, deploying complex algorithms on the existing control infrastructure may only be possible if specialized hardware components like GPUs or AI processors are available. These components were relatively uncommon in industrial control setups until recently. However, new control hardware, such as multi-processor PLCs and AI expansion cards, have emerged, making this deployment possible.

Another challenge is the notable disparities between the focus areas of control engineers and data scientists when devising control systems. Control engineers primarily concentrate on industrial communication protocols, control devices, PLC programming, and SCADA development. In contrast, data scientists and software engineers focus on creating new control strategies using Python or C++ and utilize software development tools like package managers and containers. New computing paradigms tailored to industrial control systems have been developed that bridge this divide and integrate information technology (IT) tools into operational technology (OT). Examples include integrating control systems with Cloud computing, High-Performance Computing (HPC), and Edge computing.

This article mainly focuses on solutions that address these challenges and provide local intelligence to a control system, i.e., intelligence close to the process, allowing faster analysis of streamed data and lightening the load on the different layers of the control system by reducing network latency and traffic. We start by comparing various techniques for leveraging local intelligence. We emphasize Industrial Edge Computing as an emerging solution that provides benefits such as separation of concern, simplification of algorithm development, and easy application lifecycle management. Finally, we will share insights from implementing an advanced optimization algorithm on state-of-the-art edge technologies and validating its use in a real-world setting at CERN.

## LEVERAGING INTELLIGENCE TO INDUSTRIAL CONTROL SYSTEMS

In recent years, the capability to deliver intelligence close to industrial processes has progressed from conventional setups such as bare-metal Industrial PCs to more advanced systems like multi-process controllers and edge technologies. Some of these setups are outlined below.

- **Multi-process controllers:** PLC vendors have acknowledged developers' needs to program control components in languages beyond the IEC 61131-3 standard [2], adopting higher-level languages like C++ and Python. For instance, the Siemens S7-1518 Multi-Functional Platform (MFP) has a Linux OS alongside its standard PLC OS that primarily supports C++. Communication between the OSs is via an Ethernet virtual switch, eliminating additional hardware and separating

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TUPDP102

Table 1: Comparison of Different Technologies to Deploy Advanced Control Algorithms

| Criteria | IPC | Edge | AI PLC Extensions | Multi-process controllers |
|---|---|---|---|---|
| **Form factor** | Small box PC to large rack PC 57 x 43 x 18 (cm) | Same as IPC: E.g. IPC427E 27 x 14 x 6 (cm) | PLC Module: E.g. S7-1500 TM NPU 16 x 14 x 4 (cm) | Large PLC: E.g. CPU 1518 MFP 18 x 15 x 13 (cm) |
| **Device & software management** | Manual setup via K8, Compse etc. | Dedicated edge management system | Managed via TIA portal or SCADA | Manual setup via Ansible |
| **Connectivity** | Ethernet, Serial; Extendable with I/O modules | Ethernet, S7, OPC UA, PROFINET etc. | Connectivity via PLC; Additionally USB, Ethernet etc. | TCP for CPU, OPC UA for plant |
| **IT tools & languages** | Unrestricted | Linux-containerized apps | MicroPython | C++, Python |

PLC execution from external code through containerization.

- **Dedicated AI PLC expansion cards:** Another noticeable trend is a shift towards augmenting standard PLCs with specialized extension modules like SIMATIC ET 200MP equipped with dedicated Intel Myriad X chips [3] for real-time analytics and AI tasks. They connect to the PLC via a backplane bus for rapid communication. While they integrate well with existing industrial setups, they have several drawbacks, including limited programmability in environments like MicroPython [4] and constraints on AI model architectures.

- **Bare-metal IPCs:** IPCs have been used in industrial control for several decades. More robust than conventional PCs, they withstand harsh conditions with components resilient to shock, vibration, and extreme temperatures. They're expandable, with many slots and compatibility with modules and peripherals. For instance, Siemens SIMATIC IPC520A Tensorbox PC [5] is optimized for AI with its NVIDIA GPUs. IPCs also prioritize security with features like hardware security modules and secure boot processes, protecting against cyber threats and safeguarding critical data.

- **Industrial Edge Technologies:** Building on IPC, an industrial edge constitutes a hardware and software portfolio to facilitate edge computing in industrial settings. In essence, it situates computation and data storage closer to the network's edge—near the source of data generation. It inherits all the advantages of an IPC and augments it with a management system that orchestrates various devices for software installation and consolidates the data. Since it is our primary focus, we present a more comprehensive discussion in a section on Industrial Edge Ecosystem.

### *Functional Comparison of Different Solutions*

When selecting the best technology to deploy intelligence close to the process, besides cost and performance, other functional criteria must be considered, namely:

1. **Form factor**: Control engineers generally prefer configurations where the additional computing infrastructure for AI or MPC applications can be installed and powered from the same rack that contains the rest of the control equipment. This setup allows connections to devices such as PLCs and sensors over a local network.

2. **Device management**: Managing and monitoring is necessary for a large and complex environment like CERN with thousands of control devices and applications. When adding new devices to run the complex algorithms, it is vital to monitor the condition of these additional devices and manage the lifecycle of the algorithms.

3. **Connectivity**: Ease of integration with existing control equipment is crucial. Having devices read data from controllers or store data in the SCADA system with minimal configuration is preferable.

4. **Support for IT tools and languages**: Since Python has become the de facto programming language for AI and ML applications, deploying Python code or running Jupyter notebooks directly on the devices is beneficial. Moreover, software containerization support can make application deployment and scaling easy.

Table 1 summarizes the key attributes of various technologies according to the criteria discussed above. The analysis points toward edge technology as a favorable choice for our use cases, attributing to its multifaceted advantages. Edge devices, with form factors varying from smaller than a PLC to desktop PC size, provide notable adaptability and integration benefits compared to alternatives like multi-process controllers and AI expansion cards. They execute complex algorithms without interfering with primary control processes, and their modular design permits modifications to the algorithms without operational disruption. Their support for modern programming languages, notably Python, and standard IT tools like containers favor Data scientists and software engineers. Additionally, edge platforms have robust device management capabilities and support industrial protocols for bidirectional connectivity.

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TUPDP102

The subsequent section delves into edge ecosystem components and presents a case study illustrating the practical application of industrial edge in deploying advanced control strategies.

## INDUSTRIAL EDGE ECOSYSTEM

The Industrial Edge Ecosystem serves as a foundation for near real-time data processing and decision-making closer to the source of data generation. Several components constitute the ecosystem as illustrated in Fig. 1: Edge Devices and applications on the field level, management and orchestration of devices and applications, and continuous processes to manage the life cycle of analytics applications.

### *Edge Devices*

Edge devices act as a bridge, linking the operational technology that interfaces with physical equipment, such as controllers and field devices, to the information technology systems responsible for managing digital data. Their primary role is facilitating immediate data processing, reducing latency, improving reliability, and limiting data transfer to centralized data centers.

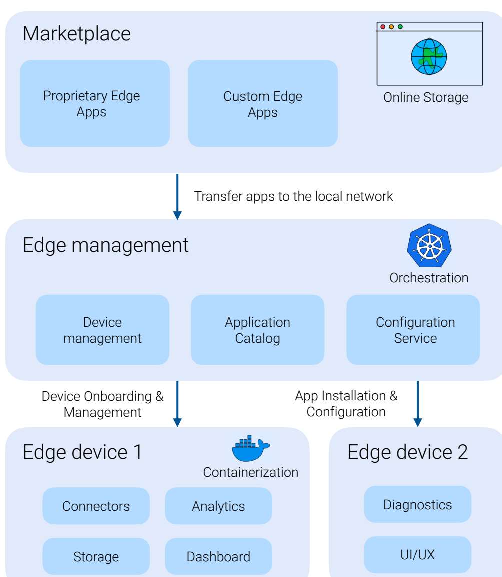


Figure 1: Industrial Edge ecosystem.

### *Edge Applications*

Edge applications are software modules specifically designed to run on Edge devices, often performing particular tasks like data pre-processing, analytics, or control operations. They benefit from the local computing power, allowing them to execute tasks more rapidly than offloading them to a centralized cloud. Some examples of edge applications include - PLC connectors, which exchange data with the PLC, and Data brokers, which share this data with other applications. The Edge applications typically run in a containerized environment, and devices have an orchestration layer to manage these application containers.

### *Analytics Function Platform*

Analytics function platform refers to a set of services tailored to edge computing that combines data analysis and software engineering capabilities. It automates the algorithm life-cycle, including model training, deployment, and monitoring, which is especially important in Edge environments. It ensures that ML and MPC algorithms can be seamlessly integrated into the existing operations, providing a pathway for continuous improvement and real-time adaptability.

### *Edge Management*

Managing the Edge devices requires specialized software for operations, deployments, and resource allocations. They offer centralized management capabilities, allowing seamless deployment, updating, and scaling of edge applications and services across many Edge devices. They also provide an interface to onboard and manage edge devices in the network, which ensures all devices work together and adhere to the operational parameters.

### *Online Marketplace*

The edge applications are typically developed by vendors or third-party developers and distributed via an online marketplace for edge applications. The users can purchase the edge applications via such platforms and transfer them to a local edge management system. They can then install the edge applications on the onboarded edge devices.

## CASE STUDY: MODEL PREDICTIVE CONTROL ON EDGE

Here, we demonstrate how to use an edge ecosystem to enhance a control system by deploying a state-of-the-art MPC algorithm close to the control process. MPC is an advanced control strategy that employs optimization algorithms to identify the optimal control inputs while adhering to predefined constraints. For example, in the context of HVAC systems, MPC aims to find the ideal operational settings for fans, heaters, and coolers to maintain a desired temperature range. It uses a mathematical model, often a set of differential equations, to forecast future states of the system and minimizes a particular objective function, such as the energy costs of running the fans and heaters. Constraints may include system conditions and equipment limitations, such as desired temperature ranges or maximum fan speeds. Due to its predictive capabilities, MPC can manage these constraints and adapt to changing conditions. It offers a significant advantage over less dynamic strategies like PID control, which lack predictive features.

Building on these capabilities, an MPC algorithm [6] was developed to manage Air Handling Units at CERN, highlighting its potential in industrial environments. In our study, we adapt this MPC algorithm to deploy it on a Siemens industrial edge device, IPC427E, bringing optimization benefits closer to the source of data generation. Moreover, we discuss an experiment where we validate the feasibility and

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TUPDP102

efficiency of running the algorithm within an industrial edge ecosystem.

### *Deployment*

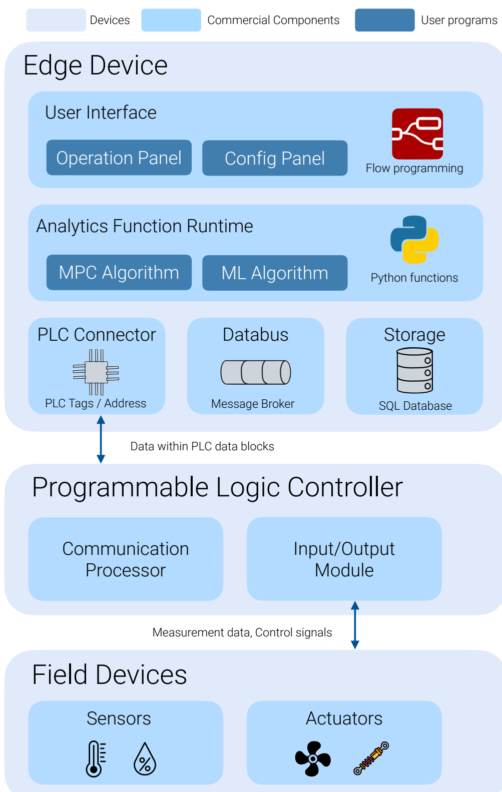


Figure 2: Deployment of MPC on Edge.

Several key aspects of deploying the MPC algorithm on an Edge device are illustrated in Fig. 2 and outlined here.

- **PLC-Edge communication:** Two applications, namely the S7 connector and IE Databus, provide a data communication channel with a Siemens S7-400 PLC, transferring data stored in PLC data blocks back and forth with the edge applications. The connector has industrial S7 protocol drivers underneath, facilitating this PLC-Edge communication. The IE Databus is an MQTT broker that facilitates data exchange between the algorithm and edge applications using the MQTT protocol. It publishes the data to any subscribing application, thus allowing for topic-specific data routing. The MPC algorithm can read the temperature sensor data and output the optimal control signals for fan speed and dampers via these two edge applications. The modular approach for data communication enables seamless coordination between the PLC and the edge ecosystem, spanning multiple devices and applications.
- **Execution of Python Code:** Python functions run on an edge device via a streamlined operational process managed through an analytics function platform designed for edge computing. The platform handles code deployment, process monitoring, and function updates. We package and build the MPC algorithm into an executable function that can be invoked by sending signals via the MQTT protocol. Moreover, we log significant operational data through these functions, such as running times, control inputs, and system states, to a storage service. This data can be accessed through the user interface for troubleshooting or further optimization.
- **User interaction:** A Web User Interface facilitates user interaction with the MPC algorithm. We develop the User Interface (UI) using IE Flow Creator, based on Node-RED [7], a flow-based programming tool that runs directly on the Edge device. Flow Creator provides a browser-based editor that makes it simple to wire together nodes, including a wide range of input and output nodes catering to various data protocols. Each node is essentially a reusable javascript code snippet, and the combination of nodes forms a web UI, using which the operators can modify control inputs, view system states, and even tweak algorithmic parameters or constraints, all without requiring direct interaction with the underlying code. The flexibility and user-friendliness of Flow Creator make it a fitting choice for enabling a dynamic interaction between the operators and the MPC algorithm, thereby improving both transparency and control in operations.

To conclude, deploying complex algorithms like MPC within an Industrial Edge offers distinct advantages over options like centralized servers due to these features of the edge ecosystem. Modular interface applications such as S7 connectors allow us to set up communication with PLCs and other devices easily. Development tools like Flow Creator enable us to test and fine-tune algorithms rapidly. Furthermore, the analytics function platform streamlines the deployment process, ensuring the algorithm can be updated and rolled out efficiently across multiple devices. This combination of easy development and deployment accelerates improvements in the control process, making the edge a favorable choice over other options.

### *Validating Usage in the CERN Industrial Control System.*

With the abovementioned steps, we can deploy a prototypical edge application running an MPC algorithm. However, we must fulfill specific performance requirements to deploy it for optimizing an actual HVAC control process. For example, we need to verify that the execution times are short enough for the process to run smoothly without interruptions. Transitioning from a prototype to real-world deployment requires rigorous performance validation studies. Confirming that the MPC algorithm running on an edge device identifies optimal solutions for all the possible input scenarios is crucial. In instances of extended execution times, it's essential to pinpoint the problematic input sets so that the edge device can delegate the control to the operator or an alternate

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TUPDP102

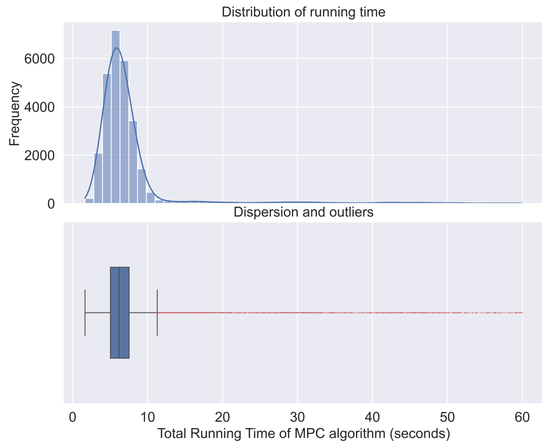


Figure 3: Plots summarizing the running time of the MPC algorithm.

controller. We employ a comprehensive validation process to rigorously evaluate the edge device's performance under these conditions.

In our validation study, we select a uniform set of close to 28,000 inputs that adhere to the model's constraints and then measure the time required for the MPC algorithm to process each input profile. Recognizing that each MPC algorithm instance is independent of different inputs, we leverage parallel execution to compute these running times efficiently. For this purpose, we utilize Simple Linux Utility for Resource Management (SLURM) [8] workload management on an HPC cluster, ensuring a thorough exploration of the feasible input space. SLURM is a widely used workload manager favored by many TOP500 supercomputers. It enables the efficient scheduling of jobs and resource allocation through a queuing system, allowing for task prioritization and utilizing available computing power to its fullest extent. Its scalability and flexibility make it ideal for comprehensively exploring the feasible input space in our experiment. Notably, the HPC environment is configured to mirror the conditions of the Edge device, thereby ensuring that our performance data remains applicable to real-world operational settings.

### *Analysis*

Following the execution of the algorithm under various conditions, we perform a statistical analysis of the running times to gain insights into the system's performance. The initial phase of this analysis involves the calculation of fundamental statistics, including mean, median, standard deviation, and other pertinent statistical metrics. These calculations serve to provide an initial snapshot of the algorithm's performance. The values are listed in Table 2. Notably, for 95 percent of the inputs, the running times were less than 17 seconds. Subsequently, we visualize the Kernel Density Estimation applied to the sample data, as demonstrated in the accompanying plot in Fig. 3. Together, these steps showcase the algorithm's adaptability to diverse input scenarios, underscoring its versatility when deployed on edge devices.

Moreover, we conduct an outlier analysis on the running time data points to pinpoint potential bottlenecks, signifying input conditions associated with longer-than-desired running times. The second distribution plot in Fig. 3 visually represents these outliers, marking them in red for easy identification.

These validation studies establish the groundwork for the practical viability of our edge-deployed MPC algorithm in real-world industrial settings and provide invaluable insights for future optimizations. For instance, we can implement a warm start mode to expedite execution and reduce running times. Exploring alternative numerical solvers, such as WORHP [9] or SNOPT, represents another avenue for potential improvement. Additionally, we can investigate using dedicated computing resources, such as GPUs, to parallelize specific tasks within the MPC algorithm, further enhancing its efficiency.

Table 2: Summary of Statistical Analysis

| Statistic | Value (Seconds) |
|---|---|
| Mean | 7.65 |
| Standard Deviation | 6.91 |
| Median | 6.18 |
| 95th percentile | 16.10 |

## CONCLUSIONS AND OUTLOOK

In this paper, we explored the use of edge technologies to deploy advanced control algorithms close to industrial processes. Through a detailed comparative analysis, we identified the unique benefits of industrial edge compared to other solutions, notably scalability and simplified deployment. We presented the various components that make up the edge ecosystem, including the connector application that streamlines communication among control devices and the analytics function platform that manages the lifecycle of the algorithms. Moreover, using the Siemens industrial platform as a case study, we illustrated a practical application by outlining the steps to deploy a cutting-edge MPC algorithm for HVAC system optimization. Finally, we conducted validation studies on an HPC cluster to substantiate our findings. We confirmed that the Siemens industrial edge platform meets operational requirements under various system conditions.

Based on our comprehensive studies, we recommend the adoption of industrial edge technologies as a means to augment industrial control systems with advanced control strategies.

## ACKNOWLEDGMENTS

This research paper is a component of the Edge Computing project, funded by Siemens AG Technology in partnership with CERN OpenLab. We thank DAI DAS-AT and CED SES-DE departments within Siemens Technologies for making this research possible.

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TUPDP102

## REFERENCES

[1] F. M. Tilaro, B. Bradu, M. Gonzalez-Berges, F. Varela, and M. Roshchin, "Model Learning Algorithms for Anomaly Detection in CERN Control Systems", in *Proc. ICALEPCS'17*, Barcelona, Spain, Oct. 2017, pp. 265–271. doi:10.18429/JACoW-ICALEPCS2017-TUCPA04

[2] K. H. John and M. Tiegelkamp, "Programming Industrial Automation Systems", vol. 166, Springer Berlin, Heidelberg, 2010.

[3] https://www.intel.com/content/www/us/en/products/details/processors/movidius-vpu/movidius-myriad-x/products.html

[4] https://micropython.org/

[5] https://support.industry.siemens.com/cs/document/109814967/delivery-release-tensorbox-simatic-ipc520a

[6] F. Ghawash *et al.*, "Model Predictive Control of Air Handling Unit for a Single Zone Setup", in *Proc. AdCONIP'22*, Vancouver, BC, Canada, Aug. 2022, pp. 158–163. doi:10.1109/adconip55568.2022.9894128

[7] https://nodered.org

[8] A. B. Yoo, M. A. Jette, and M. Grondona, "SLURM: Simple Linux Utility for Resource Management", *Workshop on Job Scheduling Strategies for Parallel Processing*, vol. 2826, Springer Berlin, Heidelberg, 2003, pp. 44–60. doi:10.1007/10968987_3

[9] C. B*ü*skens and D. Wassel, "The ESA NLP solver WORHP", *Modeling and Optimization in Space Engineering*, vol. 73, Springer, New York, NY, USA, pp. 85–110. 10.1007/978-1-4614-4469-5_4