Advances in Science and Technology Research Journal

Advances in Science and Technology Research Journal, 2026, 20(9), 1–12
https://doi.org/10.12913/22998624/221538
ISSN 2299-8624, License CC-BY 4.0

Received: 2026.03.24
Accepted: 2026.07.10
Published: 2026.08.01

Edge dynamic matrix control for energy-efficient control of heat distribution system

Gabriel Malanowski[1], Malgorzata Michalczyk[2], Tomasz Klopot[1*]

[1] Department of Automatic Control and Robotics, Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland

[2] Department of Measurement and Control Systems, Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland

* Corresponding author's e-mail: tomasz.klopot@polsl.pl

ABSTRACT

The dynamic nature of thermal processes means that predictive algorithms are an obvious choice for controlling processes of this type. Unfortunately, the product variability commonly found in real-world processes forces changes in the operating point. This, in turn, forces changes in the controller settings. Unfortunately, even for commonly used PI controllers, this is not usually done in practice. This approach results in a deterioration of the control loop efficiency indices and, consequently, the efficiency of the process and even the quality of the product. Therefore, the implementation of a predictive controller in the control system that would not require personnel to re-parameterize in the event of a change in the operating point seems very attractive. The linear version of the DMC predictive controller meets these expectations, as it features a low computational complexity of the control law formula. This feature allows for its implementation in a PLC. It should be emphasized that the components of the control law for a specific operating point are the result of complex calculations that are difficult to perform in a PLC. A change in the operating point forces them to be re-determined. Therefore, only the combination of a PLC and an industrial computer (IPC) in an edge computing architecture allows the full use of the aforementioned advantages of the DMC predictive controller. This paper presents a predictive edge dynamic matrix control (EDMC) algorithm designed to control heat sources operating as a part of heat distribution systems. The EDMC algorithm is implemented partially in a PLC and partially in an edge device. This cooperation significantly increases the system's available computational power and makes this solution possible to implement in industry. In addition, the publication presents a comparison of the performance quality offered by the EDMC system described in relation to the commonly used PI controller.

Keywords: edge computing, DMC controller, PLC controller, process control.

INTRODUCTION

Current changes in the production paradigm, which is increasingly being based on the availability of energy from sustainable sources, are placing new demands on control systems. These requirements will necessitate the use of more efficient systems, including control loops. Statistics show that only around 20% of control systems operate correctly [1], which makes the concept of effective data acquisition and processing a particularly important issue. Nowadays, the efficient utilization of computer networks poses an increasing challenge for system designers. It is estimated that two years ago, the number of Internet of Things (IoT) devices connected to the network and constantly generating data was 43 billion. Importantly, this trend is clearly on the rise [2].

This rapid development has necessitated the advancement of concepts enabling efficient and secure data transmission and processing, such as edge computing. Edge computing, alongside cloud computing, is considered one of the primary architectures used today for data storage and processing. Its fundamental principle lies in performing computations in proximity to the data source. Local processing enhances system reliability by allowing operations to continue even in the event of a disruption in connectivity with the central server. Ensuring high availability and fault tolerance in industrial communication networks often requires complex architectures, such as redundant server configurations, which must be carefully evaluated for performance and recovery times [3]. By utilizing edge computing, the critical control loops are kept local, minimizing the reliance on extensive and complex network infrastructure. Edge computing also provides enhanced data security by reducing transmission distance and eliminating the need to transfer information to cloud service providers [4]. Another compelling argument driving the development of edge computing technology is the reduction of energy consumption. Large-scale data centers require significant electrical power to operate efficiently [5]. By offloading a portion of computations to edge nodes, it is possible to substantially decrease energy usage. The main drawback associated with the nature of edge computing is its limited computational power and resources. Edge devices are not nearly as powerful as large cloud servers. Moreover, there is a lack of competitive solutions to cloud computing that would fully unlock the potential of this relatively new concept [6]. Existing literature addresses the application of edge computing in model predictive control (MPC) systems for autonomous devices [7]. These works highlight the substantial potential of this architecture to enhance control performance. However, there is a lack of studies addressing the use of edge computing in process control systems, particularly those involving predictive control algorithms.

To clearly position the proposed edge dynamic matrix control (EDMC) system, it is necessary to compare it with modern MPC developments and alternative computation offloading strategies. The demand for higher control quality has led to the rapid development of advanced predictive algorithms, such as adaptive MPC, event-triggered MPC, and learning-based (LB) MPC [8–10]. While these modern developments significantly improve control performance, they inherently increase the computational requirements.

To address this within the direct control layer, various embedded MPC solutions have been explored. Researchers have successfully implemented predictive controllers on specialized hardware, including microcontroller-based MPC [11,12] and FPGA-based predictive controllers [13], achieving rapid execution times. However, in standard industrial cases, replacing widely adopted Programmable Logic Controllers (PLCs) with custom microcontrollers or FPGAs is often impractical due to standardization, maintenance, and industrial programming language standards. The attempts to embed MPC directly into PLCs [14,15] ensure deterministic execution, but severely consume limited memory and restrict dynamic re-parameterization.

To overcome PLC hardware limitations without changing the automation architecture, computation offloading strategies are utilized. One active research trend is the hybrid cloud-PLC MPC [16,17], where optimization tasks are sent to remote servers. However, cloud solutions may introduce significant latency, jitter, and cybersecurity vulnerabilities [4,5], which pose severe risks to local control loops. Recently, edge computing has emerged as a safer alternative. Yet, existing edge-based solutions often offload the entire control law calculation to external devices [18]. While resolving the computational issue, it still leaves the control loop vulnerable to local network disruptions between the PLC and the edge device.

Consequently, a significant research gap exists in the practical implementation of edge-based predictive control for industrial processes. To bridge this gap, this article proposed a novel distributed architecture for the automatic parameterization of heat distribution control systems. The proposed EDMC concept assumes a strict division of tasks: the edge device performs only the complex computations necessary to determine the DMC controller parameters, while the critical, time-sensitive control law is implemented directly on the PLC. This approach ensures that even in the event of communication failures, the PLC continues control using the last known parameters.

In summary, the main contributions of this paper are as follows:

1. Distributed EDMC architecture: proposing a novel architecture that isolates the computationally heavy task of DMC parameterization on an edge device, while the linear control law is executed directly on a PLC, ensuring process safety during communication disruptions.
2. Industrial-grade implementation: demonstrating a practical integration using commonly used industrial hardware and protocols.

3. Industrial validation: practical evaluation of the proposed system on a thermal plant – heat distribution system.

DESCRIPTION OF EDGE COMPUTING SYSTEM

The proposed edge system architecture (Figure 1) is based on the cooperation of a PLC controller and an edge device, with each component responsible for a different part of the control process. The PLC, which implements the linear control law, handles data acquisition and generates control signals for the investigated plant. It cooperates with the edge device, which is responsible for data processing in the controller parameterization process. This device determines the parameters K^u and K^e, which are essential for control system's operation. Devices within the system communicate using the MQTT protocol – a lightweight, publish/subscribe-based communication protocol. Its selection was motivated by its widespread adoption in industrial internet of things (IIoT) systems, which aligns with the edge architecture under investigation.

In the studied architecture, a PC-class computer can serve as the edge device. However, industrial environments often require operation under harsh conditions, such as high levels of dust or exposure to moisture. Reliability is the most critical feature when implementing such systems in practice. Therefore, solutions dedicated to these environments are sought, offering appropriate physical and software-level protection.

Taking the above into consideration, it was decided to test a solution in the form of a compact industrial computer designed for operation in industrial environments, which has the potential for wide practical application. Furthermore, and importantly, this approach allows for full integration of the proposed solution into industrial environments.

DMC ALGORITHM

The DMC algorithm is one of the most well-known examples of model predictive control. Its initial implementations date back to the 1970s and were developed within the oil industry, where there was a need for an effective control method that accounted for nonlinearities and signal constraints [19]. The primary premise of predictive control methods is the optimization of a cost function to determine the appropriate control action at each sampling instant. These methods utilize a finite prediction horizon over which the dynamic behavior of the process is forecasted. The DMC algorithm, based on the step response of the process, minimizes the performance index $\mathcal{J}$ (1) to determine future control increments Δu over the prediction horizon H_p.

$$\mathcal{J}(k) = \sum_{p=1}^{H_P}\left(y^{sp}(k+p|k) - y(k+p|k)\right)^2 + \\ + \lambda \sum_{p=1}^{H_C-1}\left(\Delta u(k+p|k)\right)^2 \tag{1}$$

where: $y^{sp}(k+p|k)$ – set point at time instant $k+p$ predicted at time instant k, $y(k+p|p)$ – controlled variable at time instant $k+p$ predicted at time instant k, $\Delta u\,(k+p|k)$ – control increment at time instant $k+p$ predicted at time instant k, H_c – control horizon, λ – move suppression coefficient that penalizes control signal variability.

Parameter λ plays a crucial role in the cost function due to its ability to model the dynamics of the controller's response – as λ increases, the response becomes more conservative. Other tuning parameters of the controller include sampling period τ_s, dynamic horizon H_D prediction horizon H_p, control horizon H_c, and window horizon H_w, which is related to process transport delay and defines the number of step response samples omitted in the parameterization process.

The analytical form of the DMC controller for single-input single-output (SISO) systems, representing the solution to the presented optimization problem, can be expressed as [20]:

1. Collection of step response data

The first step involves collecting the step response of the process around the specified operating point and fitting a model using one of the available methods (e.g., two-point method) to approximate it by a first-order plus dead time (FOPDT) model (2), based on the determination of the gain k, time constant τ, and delay time τ_0:

$$K(s) = \frac{k}{s\tau + 1}e^{-s\tau_0} \tag{2}$$

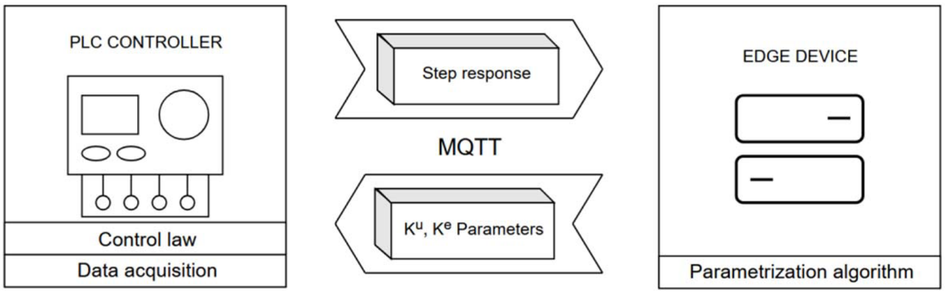

Figure 1. Proposed edge system architecture

2. Controller tuning

The second step is to select the controller parameters based on the identified FOPDT model approximation (2) around the operating point. This can be done using the heuristic methods presented, for example, in [21,22] or stochastic optimization [23,24].

3. Determination of controller parameters: K^e and K^u

The third step involves collecting the step response vector *s* of the process around the specified operating point using the previously determined sampling period τ_s and dynamic horizon length H_D. This is necessary for the correct determination of the parameters used in the control law, which in turn ensures proper controller operation. The elements of the vector *s* are used to define the dynamic matrix $\boldsymbol{M}$ (3) and the matrix $\boldsymbol{M}^P$, which predicts the output based on past control increments (4):

$$\boldsymbol{M} = \begin{bmatrix} s_{H_w} & s_{H_w-1} & \cdots & 0 & \cdots & 0 \\ s_{H_w+1} & s_{H_w} & \cdots & s_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ s_{H_C} & s_{H_C-1} & \cdots & s_{H_C-H_w} & \cdots & s_1 \\ s_{H_C+1} & s_{H_C} & \cdots & s_{H_C-H_w+1} & \cdots & s_2 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ s_{H_P} & s_{H_P-1} & \cdots & s_{H_P-H_w} & \cdots & s_{H_P-H_C+1} \end{bmatrix} \quad (3)$$

$$\boldsymbol{M}^P = \begin{bmatrix} s_{1+H_w}-s_1 & s_{2+H_w}-s_2 & \cdots & s_{H_D-1+H_w}-s_{H_D-1} \\ s_{2+H_w}-s_1 & s_{3+H_w}-s_2 & \cdots & s_{H_D+H_w}-s_{H_D-1} \\ \vdots & \vdots & \ddots & \vdots \\ s_{H_P+1}-s_1 & s_{H_P+2}-s_2 & \cdots & s_{H_P+H_D-1}-s_{H_D-1} \end{bmatrix} \quad (4)$$

The definition of the parameter H_w, which is responsible for excluding the response elements prior to the transport delay from the cost function, allows for a reduction in the size of the matrices $\boldsymbol{M}$ and $\boldsymbol{M}^P$, thereby decreasing the computational load in subsequent steps.

The determination of the parameter K^e is based on the formulas (5) and (6):

$$\boldsymbol{K} = [\boldsymbol{M}^T\boldsymbol{M} + \lambda\boldsymbol{I}]^{-1}\boldsymbol{M}^T \quad (5)$$

where: $\boldsymbol{I}$ is an identity matrix, and the first row of matrix $\boldsymbol{K}$ is of the form $[K_{1,1} \quad K_{1,2} \quad \cdots \quad K_{1,H_P-H_w+1}]$:

$$K^e = \sum_{p=H_w}^{H_P} K_{1,p-H_w+1} \quad (6)$$

The parameter K^u, in the form of a vector, can be determined based on (4, 7):

$$K^u = K_1\boldsymbol{M}^P \quad (7)$$

4. Definition of the control law

The final step involves performing control based on the linear control law defined as [21]:

$$\Delta u(k) = K^e\left(y^{sp}(k) - y(k)\right) - \sum^{H_D-1} K_j^u \Delta u(k-j) \quad (8)$$

$$u(k) = u(k-1) + \Delta u(k) \quad (9)$$

where: $\Delta u(k-j)$ – vector of past control increments with a length equal to the dynamic horizon $H_D - 1$, $u(k)$ – controller output signal calculated at time k, $u(k-j)$ – previous controller output signal at time $k-1$).

An additional constraint that can be implemented involves enforcing allowable ranges for both newly calculated control increments and the controller output:

$$\Delta u_{min} \leq \Delta u(k) \leq \Delta u_{max} \quad (10)$$

$$u_{min} \leq u(k) \leq u_{max} \quad (11)$$

IMPLEMENTATION OF THE EDGE CONTROL SYSTEM

The implementation of the edge system was based on Siemens' solution – industrial edge (IE). This system was launched in 2019 as an extension of the Industry 4.0 concept, aiming to enable data processing directly at the network edge. It offers a wide range of applications for data exchange, processing, and visualization, but its key advantage lies in the ability to create and publish custom solutions. This made it possible to adapt the network to the adopted design requirements.

The control system operating within the edge computing architecture is divided into two parts. The first part of the industrial computer IPC 127E, serving as an edge device, is responsible for calculating the parameters of the DMC controller based on (3–7) and storing them in a database for future use. The algorithm was implemented in Python using the paho-mqtt library for communication with the internal IE Databus network. For communication with the PLC, the S7-Connector application was utilized – a ready-made solution within the system for data exchange with Siemens controllers. The algorithm was deployed to the edge system as a fully-fledged application using Docker and the Industrial Edge App Publisher tools.

The edge device continuously monitors the setpoint change signal with a selected sampling time τ_s. Upon detecting such an event, the algorithm searches the database for the expected controller parameters. If available, these parameters are transmitted to the PLC for control execution. If not, the process variable is brought close to the setpoint, which is appropriately verified. Subsequently, a control step Δu is applied in such a way that the process variable adequately covers the area around the setpoint. If the experiment is successful, the algorithm on the edge device retrieves the normalized discrete step response vector. Calculations are then performed sequentially to determine the controller parameters K^e and K^u, which are subsequently stored in the PLC memory.

The second part, related to the PLC S7 1500, is responsible for executing the control based on the determined controller parameters. In the TIA Portal v19 environment, the linear control law (8, 9) has been implemented along with the limitation to the allowable range (10, 11).

THERMAL PLANT

The verification of the proposed control system was carried out using a thermal plant (Figure 3) that is a part of a heat generation and distribution installation (Figure 2). The system includes an electric flow heater with a volume of $V = 1.6\ l$ and a nominal power $P_{nom} = 12\ kW$.

Municipal water flows into the heater at an inlet temperature T_{in}, °C, and a flow rate F, l/min, which is controlled by a valve with a Proportional-Integral (PI) controller. The heater operated

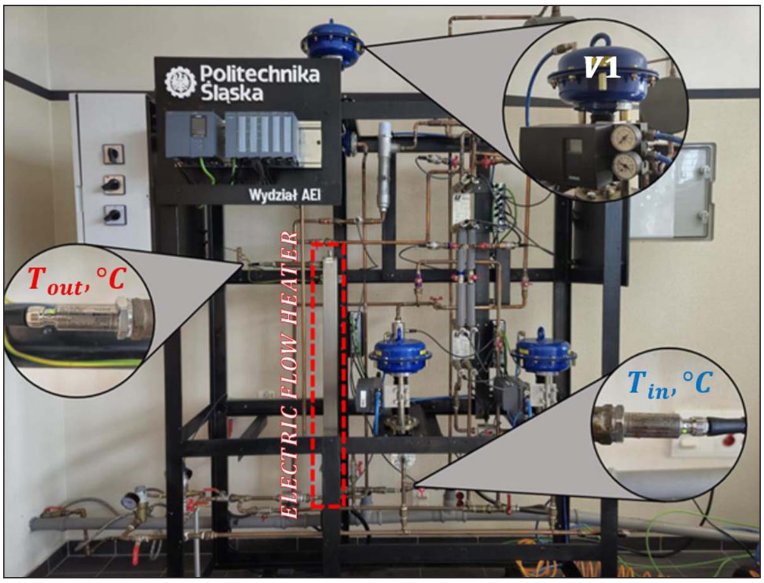

Figure 2. Laboratory thermal plant

in an ON/OFF mode, which requires the use of Pulse-Width Modulation (PWM), where the duty cycle determines the desired heating power P_h in the range of 0–100%.

Preliminary testing was also carried out using a model of the studied installation. Since the system was simulated, the flow control subsystem was omitted, and it was assumed that $F = F_{sp}$.

The plant model was adopted from [26]. This specific model was chosen because it effectively captures the fundamental nonlinear dynamics of thermal processes, such as flow-dependent gain and time constants, while maintaining a mathematical structure suitable for control system simulations. However, as with any simplified mathematical representation, this model has certain limitations compared to the real physical plant. It assumes ideal fluid mixing, simplifies the unknown phenomena into a basic transfer function, and does not explicitly account for varying sensor measurement noise present in every data acquisition system.

The first part of the model is based on a first-order nonlinear dynamic equation:

$$\frac{dT^*_{out}}{dt} = \frac{F}{60V}(T_{in} - T^*_{out}) + \frac{P_h P_{nom}}{100 c_w \rho_w V} \quad (12)$$

where: T^*_{out}, °C – unmeasurable water temperature inside the heater, $c_w = 4200 \frac{J}{kg \cdot °C}$ – specific heat capacity of water, and $\rho_w = 1$ kg/l – density of water.

The second part of the model, describing the dynamics of the sensors, the electric heater, and other unidentified phenomena occurring in the real system, is simplified to the form of a transfer function:

$$K(s) = \frac{T_{out}}{T^*_{out}} = \frac{k(P_h)}{(1 + s\tau_1(F))(1 + s\tau_2(F))} e^{-s\tau_0} \quad (13)$$

where: T_{out}, °C – measurable outlet temperature of the heater, k – gain dependent on the power P_h, τ_1, s and τ_2, s – time constants representing additional dynamics depending on the flow rate F, τ_0, s – transport delay representing the system's physical dead time.

Transfer function parameters are defined by the following relations:

$$k(P_h) = -0.0002347 \cdot P_h + 1.012 \quad (14)$$

$$\tau_1 = \tau_2 = 19.08 \cdot F^{-0.4293} - 4.042 \quad (15)$$

$$\tau_0 = 11.93 \cdot F^{-0.78} + 2.73 \quad (16)$$

Implementation of the model using the fourth-order Runge-Kutta method presented in [27] in the TIA Portal v19 environment, along with the experimental validation, confirmed its accuracy. The validation was conducted in two district scenarios to ensure the model's reliability under different operating conditions. The first scenario (Figure 4a, Figure 4c) evaluated the model's response to varying heating power at a constant flow rate of $F = 3$ l/min. The second scenario (Figure 4b, Figure 4d) verified the nonlinear flow-dependent dynamics by applying step changes to

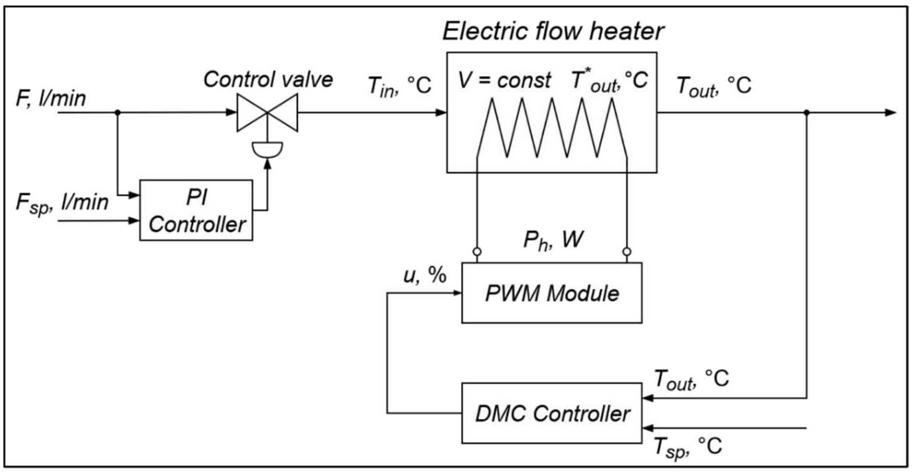

Figure 3. Simplified scheme of the temperature control system

the flow rate while maintaining a constant heating power of $P_h = 50\%$.

While the model closely follows the real plant's trajectory in both scenarios, minor modeling errors are observable. These differences come from the aforementioned limitations. Based on the validation data across both experiments, the maximum absolute modeling error during heating power changes was $\Delta T_{P_h} = 1.65\ °C$, and during flow rate changes it was $\Delta T_F = 1.98\ °C$. These maximum deviations appeared mostly in higher temperature states.

RESULTS AND DISCUSSION

The verification of the proposed EDMC algorithm structure was divided into two main stages. The first stage evaluated the edge system's ability to automatically conduct a step test and calculate new tuning parameters. The second stage contained a study evaluating setpoint tracking, disturbance rejection, and robustness to operating point variations against conventionally tuned PI controllers.

To compare control performance across all experiments, the following performance indices were calculated: *IAE* – Integral Absolute Error, t_s – settling time with a 5% assumption, M_p – maximum overshoot, max u – maximum controller output, max Δu – maximum change in controller output. The *IAE* index was normalized to the [0;1] range and expressed as IAE / IAE_{max}.

1. Automatic parameterization

The first experiment of determining the controller parameters for a new operating point was carried out both on the real plant and its model. The key to the performance of the DMC algorithm is the controller tuning, carried out based on the Klopot-Metzger strategy [23], which assumes approximating the process response using a FOPDT model. For a nominal flow rate of $F = 3\ l/min$ and an inlet temperature $T_{in} = 17\ °C$, the process was approximated using the two-point method around the operating point,

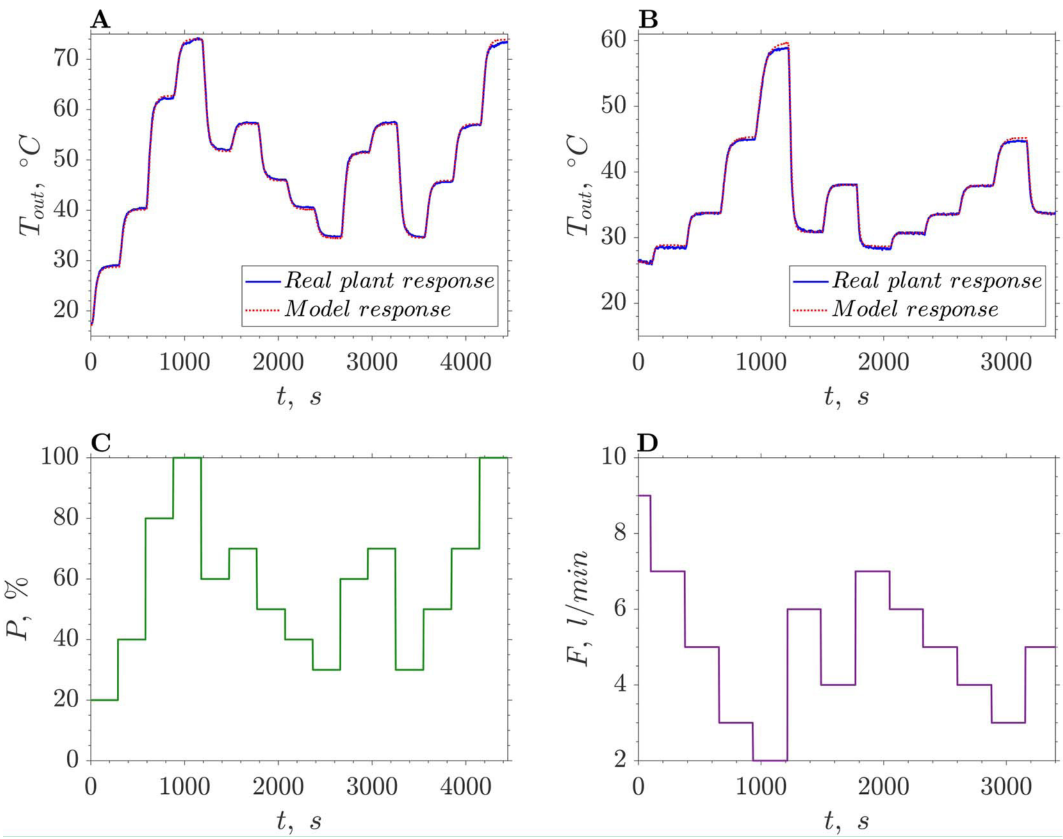

Figure 4. Validation experiments of the model representing the tested thermal plant:
a) plant and model outlet temperature during power changes experiment;
b) plant and model outlet temperature during flow rate changes experiment;
c) corresponding power changes; d) corresponding flow rate changes

i.e., between 45 °C and 55 °C with the following parameters:

$$K(s) = \frac{0.5588}{37.05s + 1} e^{-21.55s} \quad (17)$$

The controller parameters determined for (12), with a sampling time set to $\tau_s = 1s$ and a tuning parameter $x = 0.1$, for which the test were conducted, are as follows:

$$H_D = 133, H_P = 59, H_C = \\ = 2, H_w = 22, \lambda = 1.84. \quad (18)$$

The change of the operating point defined as (T_{op}, F_{sp}), from (50 °C, 3 l/min) to (45 °C, 3 l/min) initiates the process of bringing the process variable close to the new operating point (approximately 96.5% · T_{op}), during which the system operates in closed-loop tracking mode using previously determined parameters (Stage 1). Once the steady stat is reached, and experiment is conducted involving a control variable change Δu to obtain the step response around the operating point necessary for controller parameterization (Stage 2). Successful completion of this parameterization process allows the system to switch to closed-loop control (Stage 3). The algorithm accurately computed the new controller parameters and effectively brought the process variable to the new operating point (Stage 3).

2. Comparative study

To evaluate the control quality of the EDMC system against traditional methods, a comparative study with a PI controller was conducted. To ensure a fair benchmark, the PI controller was tuned using four well-established empirical methods: Ziegler-Nichols (Z-N), Cohen-Coon (C-C), and the Chien-Hrones-Reswick (CHR) method, configured for both 0% and 20% maximum overshoot [28]. Correspondingly, the EDMC controller was evaluated using two distinct heuristic tuning strategies: the Klopot-Metzger method [23] and the Shridhar-Cooper method [29].

The evaluation was divided into two scenarios to test the system's robustness across different operating conditions, accounting for both setpoint tracking and disturbance rejection.

The first scenario (Figure 5a, Figure 5c) was conducted at a nominal flow rate of $F = 3$ l/min. The experiment began with a setpoint change from 45 °C to 50 °C. Subsequently, at $t = 400$ s, a simulated load disturbance was introduced by dropping the inlet temperature T_{in} to 15 °C.

The second scenario (Figure 5b, Figure 5d) replicated the exact same setpoint and disturbance

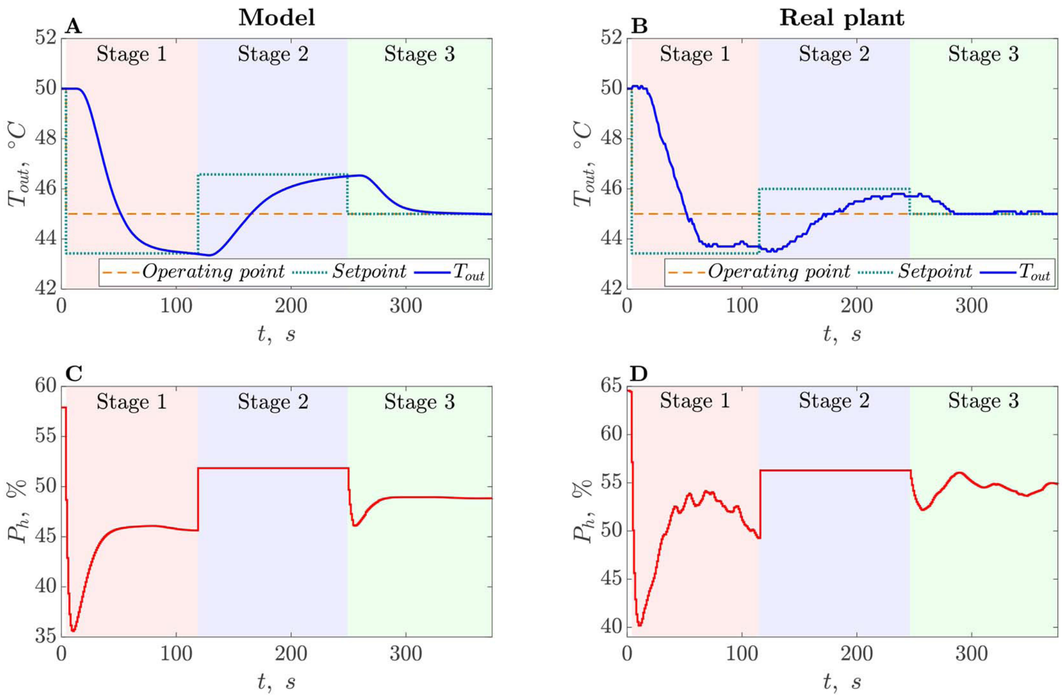

Figure 5. Dynamic behavior of the system during experiment of determining controller parameters: outlet temperature and corresponding controller output signal; a), c) for model experiment; b), d) for real plant experiment

sequence, but at a reduced flow rate of $F = 2.5$ *l/min*. According to the plant model Equations 14 and 15, changing the flow rate significantly alters the object's gain, time constants, and transport delay. This serves as a parametric disturbance, designed to test the system's robustness.

The calculated performance indices for both tested scenarios are summarized in Table 1.

In the nominal operating conditions, the predictive EDMC controllers demonstrated a clear superiority over classical PI controllers, particularly during the setpoint tracking phase. The EDMC tuned with the S-C method achieved the lowest normalized *IAE* index (IAE / IAE_{max} = 0.43), fastest settling time (t_s = 63 *s*) with almost no overshoot (Mp = 0.31%) . Among the classical approaches, the PI tuned with CHR 20% provided the most similar setpoint tracking, although being nearly twice as slow as the S-C EDMC. It is worth noting, however, that aggressive performance of the S-C method comes at the cost of high control effort (max Δu = 34.05%). In contrast, the K-M method implemented in the edge autotuning algorithm provided a highly balanced response: it outperformed the PI controllers while maintaining very conservative control increments (max Δu = 4.93%). During the regulatory phase, the EDMC again provided the best disturbance rejection, limiting the maximum temperature overshoot to M_p = 0.825 °C and minimizing the error integral (IAE / IAE_{max} = 0.423) (Figure 6).

The addition of a parametric disturbance in the form of a flow rate change significantly affected the control quality results. As the plant's gain and delay increased, most PI controllers experienced significant performance degradation. Conversely, the predictive controllers exhibited robustness against the model-plant mismatch. Although the altered dynamics caused a slight performance drop, the EDMC controllers maintained stability and high control quality. The S-C EDMC achieved the lowest IAE / IAE_{max} in both,

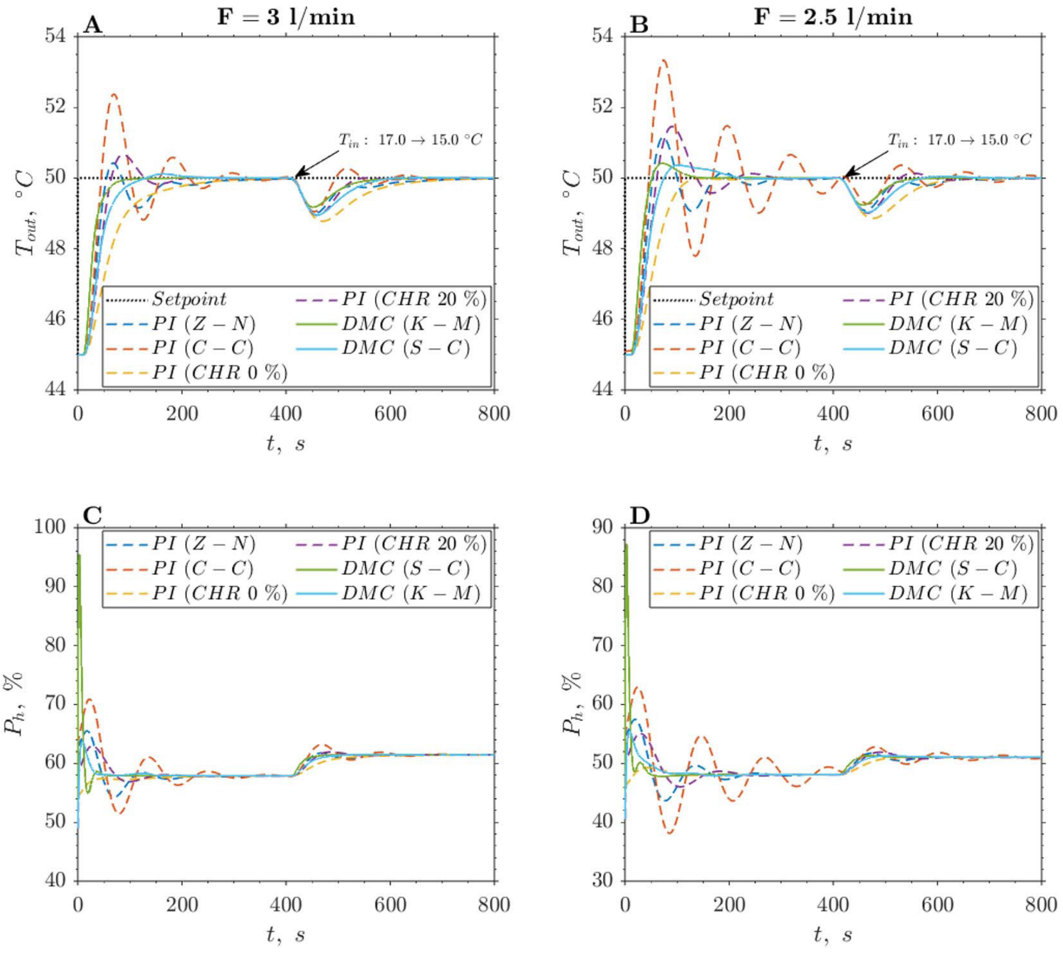

Figure 6. Comparison of PI and EDMC control: a) outlet temperature at nominal operating point; b) outlet temperature at parametric disturbance; c) corresponding controller output signal at nominal operating point; d) corresponding controller output at parametric disturbance

Table 1. Performance indices for the PI and EDMC comparative research

Disturbance rejection phase										
Parameter	$\frac{IAE}{IAE_{max}}$	t_s, s	$M_p, \%$	$\max u$	$\max \Delta u$	$\frac{IAE}{IAE_{max}}$	t_s, s	$M_p, °C$	$\max u$	$\max \Delta u$
$F = 3\ l/min$										
$PI\ (Z-N)$	0.686	156.1	8.43	65.51	14.04	0.647	166.8	0.996	61.75	0.112
$PI\ (C-C)$	0.899	248.7	47.44	70.91	15.03	0.449	122.9	0.951	63.19	0.149
$PI\ (CHR\ 0\,\%)$	1.0	192.3	0.0	57.88	5.51	1.0	192.3	1.223	61.45	0.052
$PI\ (CHR\ 20\,\%)$	0.652	114.7	13.35	62.95	9.48	0.499	112.6	1.062	61.94	0.092
$DMC\ (S-C)$	0.43	63.0	0.31	95.46	34.05	0.423	113.3	0.825	61.47	0.154
$DMC\ (K-M)$	0.641	93.0	2.33	63.81	4.93	0.611	139.3	1.043	61.55	0.091
$F = 2.5\ l/min$										
$PI\ (Z-N)$	0.526	160.9	23.08	57.48	14.04	0.642	106.7	0.917	51.73	0.098
$PI\ (C-C)$	1.0	–	66.69	62.95	15.01	0.619	197.3	0.732	52.79	0.111
$PI\ (CHR\ 0\,\%)$	0.583	107.1	0.0	49.29	5.51	1.0	169.4	1.14	51.04	0.046
$PI\ (CHR\ 20\,\%)$	0.583	191.1	29.21	55.08	9.48	0.607	112.3	0.988	51.91	0.081
$DMC\ (S-C)$	0.327	96.9	8.31	95.46	34.05	0.426	106.0	0.766	61.47	0.154
$DMC\ (K-M)$	0.471	154.8	7.42	63.81	4.93	0.658	129.9	0.972	61.55	0.09

the setpoint tracking phase and disturbance rejection phase, maintaining the lowest overshoot caused by the inlet temperature change. Again, the K-M tuning method for the EDMC provided a balanced response slightly affected by the parametric disturbance.

The conducted experiments validate not only the performance of the EDMC algorithm itself but, more importantly, the correctness and reliability of the proposed automatic parametrization architecture.

As demonstrated in the first experimental stage, the edge device successfully executed the entire autotuning sequence autonomously. The algorithm appropriately applied the control step and accurately acquired the step response. The DMC algorithm parameters derived within the Python-based Docker application proved highly reliable, ensuring robust control when transferred to the cooperative PLC. This drastically lowers the barriers to adopting predictive control in industry.

Furthermore, from a real-time systems perspective, a critical advantage of the EDMC architecture is its immunity to communication latency. In typical computation offloading strategies, the control variable is calculated remotely at each sampling instant, meaning network delays and jitter directly degrade closed-loop stability. In contrast, the proposed edge device transmits the calculated controller parameters using MQTT only once, strictly after a tuning procedure. Consequently, transmission delays do not influence the ongoing control stability.

From an algorithmic perspective, the integration of the EDMC controller provided significantly faster, and more precise setpoint tracking than classical PI control. More importantly, the robustness analysis demonstrated that the EDMC algorithm effectively compensates for parametric disturbance. Furthermore, the obtained results highlight the practical trade-offs in controller tuning. While the S-C method provided the fastest response, it resulted in aggressive control increments that could lead to premature wear of physical actuators. In contrast, the K-M method proved to be an industrial compromise. It delivered robust disturbance rejection and rapid setpoint tracking while maintaining conservative control increments.

Addressing the critical aspect of energy efficiency, the EDMC system also demonstrated a clear advantage over classical PI algorithms. To evaluate this, a control effort index I_u was introduced. This index is defined as the integral of the squared control signal over the settling time horizon $[0; t_s]$:

$$I_u = \int_0^{t_s} u^2(t)\, dt \tag{19}$$

Table 2. Control effort index during the setpoint tracking experiment

Parameter	$I_u/I_{u\,max}$
$PI\ (Z-N)$	0.615
$PI\ (C-C)$	1.0
$PI\ (CHR\ 0\,\%)$	0.718
$PI\ (CHR\ 20\,\%)$	0.464
$DMC\ (S-C)$	0.278
$DMC\ (K-M)$	0.369

In the context of the investigated thermal plant, where the control signal is equal to the PWM duty cycle of the electric heater, minimizing this integral directly correlates to reduced electrical energy consumption. The normalized $I_u/I_{u\,max}$ values, calculated for the nominal setpoint tracking phase and shown in Table 2, reveal significant differences between the evaluated systems. The EDMC algorithms reduced the energy index, outperforming classical PI controllers. This energy-efficient profile further emphasizes the practical usage of the proposed EDMC architecture.

CONCLUSIONS

The EDMC method proposed in the publication shows how the combination of an edge system with a distributed DMC algorithm partially implemented in a PLC controller allows for the use of a highly efficient control algorithm in the direct control layer. It is worth emphasizing that predictive algorithms are almost always implemented in the upper layer of the control system. The proposed approach extends the field of application of advanced control techniques, which significantly improve the quality indicators obtained in the system. On the other hand, this network architecture provides faster system response and more effective monitoring of transmitted data compared to cloud-based solutions, where information is transmitted over long distances to datacenters and is more vulnerable to attacks. The use of an industrial computer ensured the reliability, flexibility, and scalability of the solution. These factors determine the possibility of implementing innovative solutions, such as those presented in the publication, in industrial environments.

While the proposed solution proved highly effective, certain limitations exist. The current edge-based autotuning sequence relies on a step-response experiment, which temporarily disrupts the nominal operation of the plant. Moreover, the heuristic tuning methods provide a suboptimal solution, which might require further fine-tuning for highly nonlinear processes.

Consequently, future research will focus on developing the closed-loop identification methods that can continuously update the plant model without requiring explicit control signal step changes. Additionally, the integration of adaptive and learning-based MPC algorithms into the edge device will be explored, aiming to further enhance the system's performance.

REFERENCES

1. Jelali M., Control Performance Management in Industrial Automation: Assessment, Diagnosis and Improvement of Control Loop Performance. Springer-Verlag, London, 2013.
2. Sestino A., et al., Internet of Things and Big Data as enablers for business digitalization strategies, Technovation, 2020; 98. https://doi.org/10.1016/j.technovation.2020.102173
3. Cupek R., et al., Performance evaluation of redundant OPC UA architecture for process control, Transactions of the Institute of Measurement and Control, 2017; 39(3): 334–343. https://doi.org/10.1177/0142331215603792
4. Kolevski D., Michael K., Edge computing and IoT data breaches: Security, privacy, trust, and regulation, IEEE Technology and Society Magazine, 2024; 43(1): 22–32. https://doi.org/10.1109/MTS.2024.3372605
5. Shi W., et al., Edge computing: Vision and challenges, IEEE Internet of Things Journal, 2016; 3(5): 637–646. https://doi.org/10.1109/JIOT.2016.2579198
6. Varghese B., et al., Challenges and Opportunities in Edge Computing, 2016 IEEE Intenational Conference on Smart Cloud (SmartCloud), 2016, https://doi.org/10.1109/SmartCloud.2016.18
7. Seisa A. S., et al., An Edge-Based Architecture for Offloading Model Predictive Control for UAVs, Robotics, 2022; 11(4): 80. https://doi.org/10.3390/robotics11040080
8. Zhang P., et al., Adaptive transmission interval-based self-triggered model predictive control for autonomous underwater vehicles with additional disturbances, Journal of Marine Science and Engineering, 2024; 12(9): 1489. https://doi.org/10.3390/jmse12091489

9. Yang S., et al., A machine-learning-based event-triggered model predictive control for building energy management, Building and Environment, 2023; 233. https://doi.org/10.1016/j.buildenv.2023.110101
10. Zhang K., et al., A survey on learning-based model predictive control: Toward path tracking control of mobile platforms, Applied Science, 2022; 12(4): 1995. https://doi.org/10.3390/app12041995
11. Su Y., An intelligent heating system based on the Internet of Things and STM32 microcontroller, Energy Informatics, 2024; 7(22). https://doi.org/10.1186/s42162-024-00326-2
12. Currie J., Wilson D. I., Lightweight Model Predictive Control intended for embedded applications, IFAC-PapersOnLine, 2010; 43(5): 278–283. https://doi.org/10.3182/20100705-3-BE-2011.00046
13. McInerney I., et al., A Survey of the Implementation of Linear Model Predictive Control on FPGAs, IFAC-PapersOnLine, 2018; 51(20): 381–387. https://doi.org/10.1016/j.ifacol.2018.11.063
14. Bustos-Pulluquitin S., et al., Constrained dynamic matrix control under international electrotechnical commission standard 61499 and the open platform communications unified architecture, Sensors, 2023; 23(15): 6919. https://doi.org/10.3390/s23156919
15. Klopot T., et al., Function block practical implementation of Balance-Based Adaptive Control for pH process, IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society, 2013; 3549–3554. https://doi.org/10.1109/IECON.2013.6699699
16. Taheri S., et al., Real-world implementation of a cloud-based MPC for HVAC control in educational buildings, Energy Conversion and Management, 2024; 305. https://doi.org/10.1016/j.enconman.2024.118270
17. Dai L., et al., Cloud-based computational model predictive control using a parallel multiblock ADMM Approach, IEEE Internet of Things Journal, 2023; 10(12): 10326–10343. https://doi.org/10.1109/JIOT.2023.3238508
18. Krupa P., et al., PLC implementation of a real-time embedded MPC algorithm based on linear input/output models, IFAC-PapersOnLine, 2020; 53(2): 6987–6992. https://doi.org/10.1016/j.ifacol.2020.12.424
19. Horvath D., et al., Integrated Testing of Model Predictive Control on Industrial PLC, 2025 Cybernetics & Informatics (K&I), 2025; 1–6. https://doi.org/10.1109/KI64036.2025.10916464
20. Skogestad S., Adams II T. A., Origins of Dynamic Matrix Control: The Early Writings of Charles R. Cutler. PSE Press: Hamilton, 2025. https://doi.org/10.69997/pse.105161
21. Tatjewski P., Advanced control of industrial processes, structures and algorithms. Springer-Verlag, London, 2007.
22. Reverter C. M., et al., Tuning rules for a quick start up in Dynamic Matrix Control, ISA Transactions, 2014; 53(2): 612–627. https://doi.org/10.1016/j.isatra.2013.12.012
23. Kłopot T., et al., Tuning strategy for dynamic matrix control with reduced horizons, ISA Transactions, 2018; 76: 145−154. https://doi.org/10.1016/j.isatra.2018.03.003
24. Malanowski G., Klopot T., Genetic Algorithm-Based Optimization of DMC Controller in Thermal-Fluid Flow Processes, Advances in Artificial Intelligence Research. PP-RAI 2026. Lecture Notes in Networks and Systems, to be published.
25. de Almeida G. M., et al., Optimal tuning parameters of the dynamic matrix predictive controller with ant colony optimization, 2014 11th IEEE/IAS International Conference on Industry Applications, 2014; 1–6. https://doi.org/10.1109/INDUSCON.2014.7059396
26. Nowak P., et al., ADRC-based habituating control of double-heater heat source, Energies, 2022; 15(14): 5241. https://doi.org/10.3390/en15145241
27. Kiusalaas J., Numerical Methods in Engineering with Python 3. Cambridge University Press, 2013.
28. Daful A. G., Comparative study of PID tuning methods for processes with large & small delay times, 2018 Advances in Science and Engineering Technology International Conferences (ASET), 2018; 1–7, https://doi.org/10.1109/ICASET.2018.8376915
29. Shridhar R., Cooper D. J., A Tuning Strategy for Unconstrained SISO Model Predictive Control, Industrial & Engineering Chemistry Research, 1997; 36, 729–746.