Bayesian Optimization-Based Tunable Explicit MPC on a Pocket-Sized Embedded Platform

Peter Bakaráč – Erika Pavlovičová – Martin Klaučo – Juraj Oravec

Abstract— **The paper presents a pocket-sized embedded platform designed for the validation of advanced control methods. The platform is based on the ESP32-S3 microcontroller and is equipped with a miniaturized heat exchange device, making it suitable for temperature control experiments. The platform enables the implementation of a real-time tunable explicit Model Predictive Control (MPC) algorithm, which allows for online tuning of the weighting parameter in the MPC cost function. The paper also introduces a novel application of a method based on Bayesian optimization, which efficiently explores the parameter space to find an optimal performance metric value. The performance metric is a weighted overall parameter of the controller's performance, in which actuator power consumption and control signal fluctuation are evaluated. Experimental results demonstrate the effectiveness of the proposed approach, while the exploration and exploitation approaches have been tested. Both the explicit MPC controller and the Bayesian optimization method are implemented on the embedded platform, showcasing its capabilities for real-time control applications. The results highlight the potential of the platform for further research and development in advanced control strategies.**

I. INTRODUCTION

Advanced control strategies, particularly Model Predictive Control (MPC), have been extensively studied due to their ability to control multivariable systems under constraints, providing optimal real-time control performance [1]. Despite these advantages, MPC typically requires solving computationally intensive optimization problems at every sampling instant, posing significant implementation challenges on resource-constrained embedded systems [2]. Consequently, simpler control techniques such as proportional-integral-derivative (PID) controllers continue to dominate embedded applications, underlining a substantial gap between theoretical advancements and practical applications [3].

To mitigate computational demands, explicit MPC is introduced in [4], precomputing control actions offline to enable rapid online evaluations. However, the inflexibility of explicit MPC limits its adaptability in changing real-time conditions [5]. To overcome this limitation, interpolation-based methods for explicit MPC, facilitating real-time parameter adjustments, are presented in [6]. Moreover, in [7], this approach is enhanced, introducing real-time tunable approximated explicit MPC with improved dynamic adaptability.

Authors are with Institute of Information Engineering, Automation, and Mathematics, Faculty of Chemical and Food Technology, Slovak University of Technology in Bratislava, Radlinskeho 9, SK-812 37, Bratislava {peter.bakarac, erika.pavlovicova, martin.klauco, juraj.oravec}@stuba.sk. M. Klaučo is also with Czech Technical University in Prague, Department of Control Engineering.

Further advancements were presented in [8]. The authors proposed lattice piecewise affine approximations of explicit MPC controllers, significantly reducing computational complexity while preserving control performance. In [9], there is demonstrated a self-tunable explicit MPC on a heat exchanger, highlighting real-time adaptive capabilities. Additionally, in [10], the authors explored neural network approximations to explicit MPC laws, offering a balance between computational efficiency and control performance.

Despite these advancements, practical MPC implementation faces ongoing challenges in efficiently determining optimal tuning parameters on embedded devices. Bayesian Optimization (BO) has emerged as a promising approach, efficiently exploring parameter spaces through probabilistic models [11]. In [11], authors demonstrated BO's effectiveness in optimizing computationally expensive machine learning algorithms. [12] presents the application of BO to automatic MPC tuning under uncertainty, demonstrating improved robustness and efficiency. Additionally, BO is employed in [13] for robust MPC parameter tuning, showing enhanced performance under parameter uncertainty. Integration of BO into MPC-based shared controllers in [14], substantially outperforming manual tuning approaches. Moreover, the combination of BO with deep learning-enhanced MPC controllers in quadrotor applications in [15], achieving notable improvements in real-time control agility and accuracy.

In this paper, we introduce a compact, cost-effective embedded platform specifically designed to bridge the gap between theoretical MPC advancements and practical implementations. Building upon the real-time tunable approximated explicit MPC approach in [7], we develop a novel embedded implementation that integrates Bayesian optimization for real-time parameter tuning. Our platform, based on the ESP32-S3 microcontroller [16], provides the necessary computational capabilities for sophisticated control techniques while maintaining accessibility and affordability. We experimentally validate the embedded platform, demonstrate the implementation of a real-time tunable approximated explicit MPC algorithm with online interpolation, and show the integration of Bayesian optimization for automated, real-time MPC parameter tuning.

The paper proceeds as follows: Section II outlines the hardware and software design of the embedded platform. Section III describes the MPC formulation and Bayesian optimization-based tuning method and presents experimental validation through a temperature control case study. Section IV concludes with a summary of findings and directions for future research.

II. POCKET-SIZED EMBEDDED PLATFORM

Development kits based on the ESP32-S3 microcontroller from Espressif Systems[1] are widely used in process control applications, serving as control units in both commercial and non-commercial/open-source projects. Given their popularity, it is a natural consequence that the developer and academic communities have created control projects specifically designed to be compatible with these devkits. Various shields and platforms have been developed for educational and research purposes, enabling hands-on learning and experimentation.

This paper considers a project Sensoric Shi-3-ld, developed by Optimal Control Labs[2]. This platform integrates a diverse set of sensors, actuators, and essential components tailored for educational and research applications. It provides students with an opportunity to program data acquisition from sensors and control devices such as motors, buzzers, and advanced programmable RGB LEDs. For researchers, the platform includes an integrated dynamic system, which we utilized as a testbed for implementing the advanced control method of the real-time tunable explicit MPC driven by the novel tuning policy based on the Bayesian optimization.

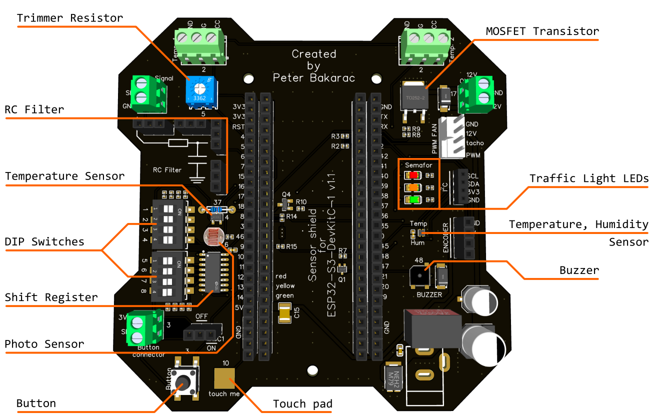

Fig. 1. The Sensoric Shi-3-ld platform developed by Optimal Control Labs.

Figure 1 above showcases the Sensoric Shi-3-ld device, with labeled positions of its sensors, actuators, and connectors. Additional details about this platform can be found on the Optimal Control Labs website. The pin layout of the Sensoric Shi-3-ld is designed for compatibility with the ESP32-S3-DevKitC-1, and in the considered setup, we specifically used the ESP32-S3-DEVKITC-1-N32R8V. This microcontroller features a 32-bit Xtensa LX7 processor running at 240 MHz, along with 32 MB of Flash storage, 512 kB of SRAM, and 8 MB of PSRAM, making it well-suited for demanding real-time control applications.

A. Embedded Dynamic System

The integrated dynamic system of the Sensoric Shi-3-ld platform is designed based on a miniaturized heat exchange device, making it a suitable benchmark for temperature control experiments analyzing the closed-loop performance of the investigated control policy. This system consists of an analog temperature sensor positioned beneath a resistor, which serves as an actuator. The ESP32-S3 microcontroller regulates the voltage across the resistor via an NPN Darlington transistor, enabling precise temperature control in the vicinity of the sensor. A thermally conductive paste is applied between the resistor and the sensor to optimize heat transfer, ensuring efficient thermal coupling. This setup is well-suited for validating various control algorithms related to temperature control. Figure 2 below illustrates the exact location of an embedded dynamic system of the Sensoric Shi-3-ld platform.

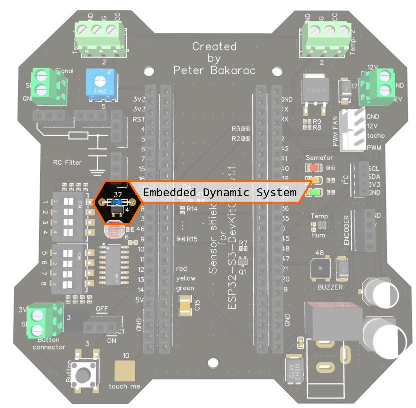

Fig. 2. The integrated dynamic system of the Sensoric Shi-3-ld platform.

The system features a TC1047A analog temperature sensor, which operates within a temperature range of $-40\,°\mathrm{C}$ to $125\,°\mathrm{C}$. Its output voltage varies between $100\,\mathrm{mV}$ and $1.75\,\mathrm{V}$, corresponding to the range of measured temperature. The sensor's supply voltage range is $2.5\,\mathrm{V}$ to $5.5\,\mathrm{V}$. Since the ESP32-S3 microcontroller's ADC has a sensing range of $0\,\mathrm{V}$ to $3.3\,\mathrm{V}$, the $5\,\mathrm{V}$ supply voltage on the Sensoric Shi-3-ld platform benefits from direct interfacing without the necessity for additional components.

The heating element consists of a standard $47\,\Omega$ carbon resistor, controlled by the ESP32-S3 via PWM. At maximum PWM duty cycle, the resistor can reach a temperature of approximately $45\,°\mathrm{C}$, providing a controlled and repeatable heat source for real-time control experiments.

III. CASE STUDY

The application range of the designed embedded platform is demonstrated using the extensive case study on temperature control. The ability to implement the advanced control strategy is investigated considering the novel method of the real-time tunable approximated explicit MPC according to [7]. First, the theoretical backgrounds are recalled. Next, the novel method of real-time tuning based on Bayesian optimization is proposed, followed by the implementation details and a discussion of the generated experimental results.

[1] https://www.espressif.com//
[2] https://www.ocl.sk/

A. Real-Time Tunable Approximated Explicit MPC

The real-time tunable implementation method for the explicit MPC policy designed subject to the challenging 2-norm penalization was originally presented in [6] and later improved by introducing the guarantees on the closed-loop system stability and recursive feasibility in [7]. Although the evaluated control action leads to a suboptimal solution, the considered control strategy enables a real-time tuning of the MPC design penalty function within a given interval without the need to repeatedly solve the optimization problem.

The evaluation of the control action is based on the solution of the following two boundary MPC design problems:

$$
\begin{aligned}
\min_{u_0,u_1,\ldots,u_{N-1}} \quad & \sum_{k=0}^{N-1} \left(\|y_{\mathrm{ref}} - y_k\|_Q^2 + \|\Delta u_k\|_{R_{\mathrm{L}}}^2\right) && \text{(1a)}\\
\text{s.t.:} \quad & x_{k+1} = A\,x_k + B\,u_k, && \text{(1b)}\\
& y_k = Cx_k + Du_k, && \text{(1c)}\\
& u_k \in \mathcal{U}, \quad y_k \in \mathcal{Y}, && \text{(1d)}\\
& \Delta u_k = u_k - u_{k-1}, && \text{(1e)}\\
& x_0 = \theta(t), && \text{(1f)}\\
& u_{-1} = u(t - T_s), && \text{(1g)}\\
& k = 0, 1, \ldots, N-1, && \text{(1h)}
\end{aligned}
$$

and the second MPC design problem has a different tuning of the penalty function:

$$
\begin{aligned}
\min_{u_0,u_1,\ldots,u_{N-1}} \quad & \sum_{k=0}^{N-1} \left(\|y_{\mathrm{ref}} - y_k\|_Q^2 + \|\Delta u_k\|_{R_{\mathrm{H}}}^2\right) && \text{(2a)}\\
\text{s.t.:} \quad & \text{(1b)} - \text{(1h)}, && \text{(2b)}
\end{aligned}
$$

where N is prediction horizon, matrices $Q \in \mathbb{R}^{p\times p}$, $R_{\mathrm{L}}, R_{\mathrm{H}} \in \mathbb{R}^{m\times m}$, are state, and input penalties, respectively. Prediction model in (1b) has the form of linear time invariant (LTI) system for state matrix $A \in \mathbb{R}^{n\times n}$, input matrix $B \in \mathbb{R}^{n\times m}$, output matrix $C \in \mathbb{R}^{p\times n}$, and the feedforward matrix $D \in \mathbb{R}^{p\times m}$. Vectors $y \in \mathbb{R}^p$, $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $d \in \mathbb{R}^n$ are vectors of system outputs, states, and control inputs, respectively. $\mathcal{U} \subseteq \mathbb{R}^m$, $\mathcal{Y} \subseteq \mathbb{R}^p$ are sets of input and output constraints, respectively, see [1]. $\theta(t) \in \Omega$ in (1f) is vector of initial conditions for the set of the feasible initial conditions.

Consider MPC problems (1), (2) are asymptotically stable and recursively feasible. Assume, in (1), (2) hold:

1) sets $\mathcal{U}$, $\mathcal{X}$, Ω are closed convex polyhedra containing origin in their strict interiors,
2) matrices Q, $R_{\mathrm{L}}, R_{\mathrm{H}}$ are positive definite.

According to the initial condition in (1f) having a parametric form, the MPC problems in (1), (2) lead to the problems of the multiparametric quadratic programming (mpQP), see [4].

Then, for the given value of tuning input penalty $\widetilde{R}$, the evaluation of the control action is addressed using the solution of the MPC problem in the form:

$$
\begin{aligned}
\min_{u_0,u_1,\ldots,u_{N-1}} \quad & \sum_{k=0}^{N-1} \left(\|y_{\mathrm{ref}} - y_k\|_Q^2 + \|\Delta u_k\|_{\widetilde{R}}^2\right) && \text{(3a)}\\
\text{s.t.:} \quad & \text{(1b)} - \text{(1h)}. && \text{(3b)}
\end{aligned}
$$

For input penalty matrix in (3a) holds:

$$
\widetilde{R} \;=\; (1-\rho)R_{\mathrm{L}} + \rho\, R_{\mathrm{H}}, \quad 0 \le \rho \le 1. \tag{4a}
$$

Then, in the real-time control, the control action is evaluated for the given system state vector $\theta(t) \in \Omega$. The control input $\widetilde{u} \in \mathbb{R}^m$ approximating the solution of MPC problem in (3) is evaluated for given explicit solutions of MPC problems in (1), (2) using their convex combination:

$$
\widetilde{u} \;=\; (1-\rho)u_{\mathrm{L}} + \rho\, u_{\mathrm{H}}, \qquad 0 \le \rho \le 1, \tag{5}
$$

where u_{L}, u_{H}, respectively, are optimal control inputs of MPC problems (1), (2) for $\theta(t)$. Therefore, the advantage of the tunable explicit MPC is that for any $\rho \in [0,1]$, the real-time evaluation of the approximated control action is optimization-free and handles just a simple evaluation of linear function. The further technical details, including a discussion on the control scenario based on the state penalty tuning, are summarized in [7].

B. Bayesian Optimization

In this work, we propose utilizing the real-time tunability of explicit MPC to identify the optimal value of the parameter $\widetilde{R}$ in the MPC problem (3a). To achieve this, we employ Bayesian optimization, an efficient method for optimizing expensive objective functions. Bayesian optimization is an iterative strategy that builds a probabilistic model of the objective function and uses an acquisition function to determine the most promising points for evaluation. This approach is particularly advantageous in scenarios where function evaluations are costly or when only a limited number of measurements can be obtained.

Bayesian optimization typically relies on a probabilistic surrogate model, most commonly a Gaussian Process (GP), which approximates the behavior of the objective function. A Gaussian Process deeply described in [11] defines a prior distribution over functions based on a chosen kernel, encoding assumptions about smoothness and correlation. Upon observing data, this prior is updated to form the posterior distribution:

$$
f(\mathbf{x}^*) \mid \mathcal{D} \sim \mathcal{N}\left(\mu(\mathbf{x}^*),\, \sigma^2(\mathbf{x}^*)\right), \tag{6}
$$

which provides both the predictive mean $\mu(\mathbf{x}^*)$ and variance $\sigma^2(\mathbf{x}^*)$ at new input points, capturing both the expected value and uncertainty of the function.

Given training or measured data $\mathcal{D} = \{\mathbf{X}, \mathbf{Y}\}$ over h number of points, the posterior distribution at a new point $\mathbf{x}^*$ is given by:

$$
\begin{aligned}
\mu(\mathbf{x}^*) =& k(\mathbf{x}^*, \mathbf{X}) \left[K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I\right]^{-1} \mathbf{Y}, && \text{(7)}\\
\sigma^2(\mathbf{x}^*) =& k(\mathbf{x}^*, \mathbf{x}^*) - && \text{(8)}\\
& k(\mathbf{x}^*, \mathbf{X}) \left[K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I\right]^{-1} k(\mathbf{X}, \mathbf{x}^*), && \text{(9)}
\end{aligned}
$$

where $k(\cdot,\cdot) \in \mathbb{R}^h$ is the kernel (covariance) function, $K(\mathbf{X}, \mathbf{X}) \in \mathbb{R}^{h\times h}$ is the kernel matrix over the measured inputs, σ_n^2 is the observation noise variance and $I \in \mathbb{R}^{h \times h}$ is the identity matrix.

The matrix K and the vectors $k(\mathbf{x}^*, \mathbf{X})$ and $k(\mathbf{X}, \mathbf{x}_*)$ are constructed using a predefined kernel function $k(\cdot, \cdot)$. Specifically, the covariance matrix $K(\mathbf{X}, \mathbf{X})$ is defined as:

$$K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j), \quad \text{for } i, j = 1, \ldots, h, \tag{10}$$

and the covariance vector between the test point $\mathbf{x}^*$ and the training inputs is:

$$k(\mathbf{x}^*, \mathbf{X}) = \begin{bmatrix} k(\mathbf{x}^*, \mathbf{x}_1) & \cdots & k(\mathbf{x}^*, \mathbf{x}_h) \end{bmatrix}. \tag{11}$$

The choice of the kernel function $k(\cdot, \cdot)$ plays a crucial role in shaping the prior assumptions about the function's behavior, such as smoothness, periodicity, or stationarity. In this work, we adopt the commonly used *squared exponential kernel* [17] defined as:

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2\ell^2}\|\mathbf{x} - \mathbf{x}'\|^2\right), \tag{12}$$

where σ_f^2 is the signal variance, controlling the vertical scale of variation, and ℓ is the length-scale parameter, which determines how quickly the correlation between function values decays with distance.

Once the Gaussian Process model is established and the posterior distribution is computed, the next step in Bayesian optimization is to determine where to evaluate the objective function next. This is guided by an acquisition function, which is composed of both the mean function $\mu(\mathbf{x}^*)$ and variance $\sigma^2(\mathbf{x}^*)$ from the GP model to balance exploration of uncertain regions and exploitation of known promising areas.

In this work, we use the *Upper Confidence Bound (UCB)* acquisition function adopted from [18], which selects the next point by maximizing the following criterion:

$$\mathrm{UCB}(\mathbf{x}^*) = \mu(\mathbf{x}^*) + \alpha_{\mathrm{UCB}}\, \sigma(\mathbf{x}^*), \tag{13}$$

where $\alpha > 0$ is a user-defined parameter that controls the trade-off between exploration (higher uncertainty) and exploitation (higher mean function $\mu(\mathbf{x}^*)$).

A larger value of α encourages exploration by favoring regions with higher uncertainty, while a smaller α focuses the search on areas with expected higher value of $f(\mathbf{x}_{\text{next}})$. The next query point is chosen as:

$$\mathbf{x}_{\text{next}} = \arg\max_{\mathbf{x}^*} \mathrm{UCB}(\mathbf{x}^*). \tag{14}$$

In our work, we consider the $\mathbf{X}$ to be a set of sampled values of the parameter $\widetilde{R}$, and the corresponding $\mathbf{Y}$ to be the measured closed-loop performance metrics more described in section IV-B. The Bayesian optimization process iteratively refines the surrogate model and selects new points to evaluate, gradually converging towards the optimal value of $\widetilde{R}$.

IV. Control Setup

The objective of this experiment is to utilize the Bayesian optimization method in combination with the tunable MPC framework to determine the optimal value of the parameter $\widetilde{R}$ in equation (3a). Since this parameter significantly influences overall control performance, finding its optimal value is essential for achieving the desired closed-loop behavior.

The experiment follows a structured approach, consisting of the following key steps:

1) Synthesis of two boundary explicit MPC controllers based on the optimization problems (1) and (2), corresponding to the given values of R_{L}, R_{H}.
2) Determine the performance metrics that will be used to assess the effectiveness of the control strategy.
3) Specify the Bayesian optimization method's tuning parameters, including acquisition function, exploration-exploitation trade-off, and kernel function for the Gaussian process model.
4) Deploy the Bayesian optimization algorithm and the two boundary explicit MPC controllers on an embedded system based on the ESP32-S3 microcontroller, which governs the integrated dynamic system of the heat exchanger.
5) Analyze the results and confirm the effectiveness of the Bayesian optimization method in tuning the MPC controller's parameter $\widetilde{R}$.

A. System Dynamics and MPC Design

As part of the data-driven identification process for the controlled system, we derived a linear time-invariant (LTI) model represented in state-space form:

$$A = [0.9934], \qquad B = [-3.5105 \times 10^{-4}], \tag{15a}$$
$$C = [-17.6075], \quad D = [0], \tag{15b}$$

where the LTI system is formulated in the discrete-time domain with a sampling period of $T_s = 0.1\,\mathrm{s}$. For identification purposes, the data was normalized according to the following transformations:

$$\mu_y = \frac{1}{N_{\text{data}}} \sum_{i=1}^{N_{\text{data}}} y_i, \tag{16a}$$
$$\sigma_y = \sqrt{\frac{1}{N_{\text{data}}} \sum_{i=1}^{N_{\text{data}}} (y_i - \mu_y)^2}, \tag{16b}$$
$$y_{\mathrm{norm}} = \frac{y - \mu_y}{\sigma_y}, \tag{16c}$$

where N_{data} represents the number of samples, y_i denotes the i-th output sample. The output vector is represented as y. We also performed the analogical scaling process for the input vector u. The normalized input and output vectors are denoted as u_{norm} and y_{norm}, respectively.

For the MPC controller design, we considered a prediction horizon of $N = 20$ and an output penalty weight of $Q = 100$.

For both boundary explicit MPC controllers, we imposed the identical non-symmetric input and output constraints:

$$\mathcal{U} = \{u \in \mathbb{R} \,|\, -5.6652 \leq u \leq 2.8326\}, \tag{17a}$$
$$\mathcal{Y} = \{y \in \mathbb{R} \,|\, -5.4293 \leq y \leq 4.0812\}. \tag{17b}$$

Corresponding physical constraints after inverse normalization are $u_{\mathrm{REAL}} \in [0, 255]$, because it is a range of 1 Byte value as a resolution of PWM signal. Output value is constrained within an interval of $y_{\mathrm{REAL}} \in [20\,^\circ\mathrm{C}, 40\,^\circ\mathrm{C}]$.

The MPC configurations differed in the selection of $R_{\mathrm{L}} = 1$ and $R_{\mathrm{H}} = 100$. Both explicit MPC controllers were designed in the MATLAB environment using the MPT3 Toolbox [19]. The resulting MPC controller parameters are summarized in Table I, which includes the number of critical regions N_{R}, the offline construction time of the explicit MPC controllers (explicit maps), along with the conversion time to C++ code. Additionally, the table provides information on the memory footprint of the controllers.

TABLE I
PARAMETERS OF MPC CONTROLLERS

MPC	R_{L}	R_{H}	N_{R}	**Synthesis Time [s]**
MPC$_{\mathrm{L}}$	1	-	427	11.14
MPC$_{\mathrm{H}}$	-	100	383	8.06

B. Performance Metrics

To determine the overall control quality metric, both designed controllers are implemented on the ESP32-S3 platform, where temperature control experiments are conducted. Based on the experimental results, the impact of the parameters R_{L} and R_{H} on the controller's performance is evaluated. The control objective is to regulate the temperature around multiple reference values. Several reference signals are generated and sequentially applied to the controlled system. Three reference values were selected:

1) $y_{\mathrm{ref1}} = 28\,°\mathrm{C}$, in the time span $[0, 24.9]$ seconds,
2) $y_{\mathrm{ref2}} = 31\,°\mathrm{C}$, in the time span $[25, 49.9]$ seconds,
3) $y_{\mathrm{ref3}} = 33\,°\mathrm{C}$, in the time span $[50, 75]$ seconds.

The figure below shows the control performance of the designed MPC controllers.

Experimental results indicate that MPC$_{\mathrm{L}}$ achieves faster convergence to the reference temperature compared to MPC$_{\mathrm{H}}$. However, MPC$_{\mathrm{H}}$ exhibits greater stability of control action signal.

For the observed properties of MPC$_{\mathrm{H}}$ and MPC$_{\mathrm{L}}$, it is clear that the overall control quality metric must be chosen as a compromise between the fluctuation of control signal and controller energy consumption. Therefore, the equation for calculating such a metric is as follows:

$$\text{Performance} = -(\alpha \cdot \mathrm{F} + \beta \cdot \mathrm{E}), \tag{18}$$

where F is defined as:

$$\mathrm{F} = \int_0^{75} \Delta u(t)^2 \, \mathrm{d}t, \tag{19}$$

and Energy Consumption E is defined as:

$$E = \int_0^{75} u(t) \, \mathrm{d}t. \tag{20}$$

Here, $\Delta u(t)$ is the difference between two successive control signal values. The more stable the control signal is, the smaller the value of $\Delta u(t)$ is. The term F represents the fluctuation of the control signal, while E represents the energy consumption of the actuator. The parameters α and β are weighting coefficients that determine the importance of individual metrics and balance the numerical difference in orders of magnitude. In the considered case, we select $\alpha = 1$ and $\beta = 18$.

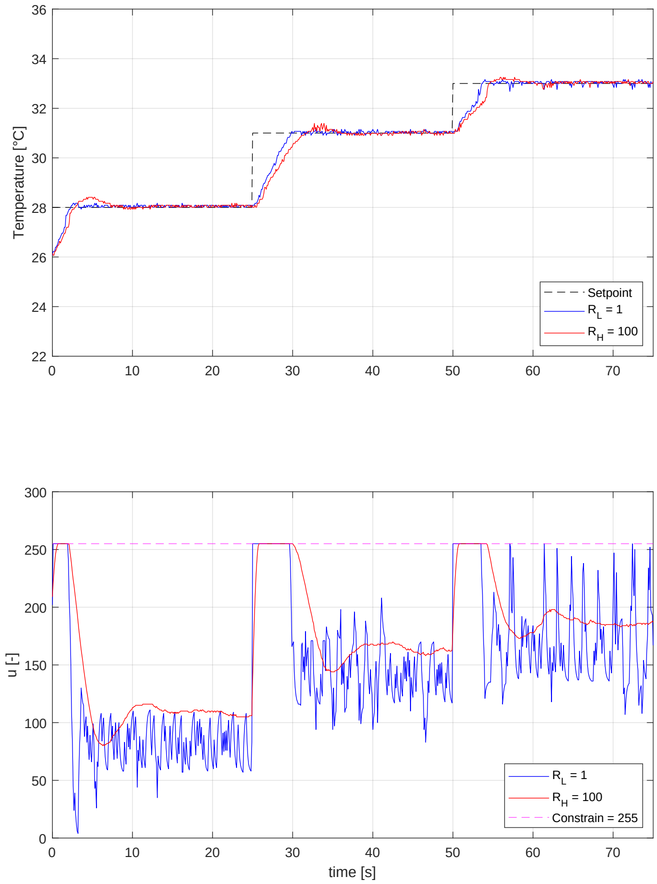

Fig. 3. Control performance of the designed MPC controllers.

In the context of temperature control of any system, it is important to consider that excessive fluctuation of the control signal can lead to premature depreciation of the heating element. Therefore, we have chosen this parameter as one of the metrics.

It should be noted that the term Energy Consumption does not directly represent the actuator's electrical energy consumption in Joules, as accurately quantifying this value would be challenging. Instead, this term is defined as the integral of the control input $u(t)$, which is proportional to the consumed energy and, therefore, relevant.

When evaluating the performance of MPC$_{\mathrm{H}}$ and MPC$_{\mathrm{L}}$ with respect to the individual components of the metrics F and E, it becomes evident that MPC$_{\mathrm{H}}$ provides superior stability of the control signal, whereas MPC$_{\mathrm{L}}$ achieves lower overall energy consumption. The responsibility of the control engineer is to determine an optimal value of the parameter $\widetilde{R}$ that achieves the best trade-off between these two competing objectives. In this paper, we propose that this parameter can be effectively tuned using the Bayesian optimization method.

C. Bayesian Optimization Setup

Bayesian optimization is used to determine the optimal value of the parameter $\widetilde{R}$ in the MPC controller. The effectiveness of this approach depends heavily on the proper configuration of key parameters, including the choice of the acquisition function, the exploration-exploitation trade-off, and the selection of the Gaussian process kernel.

The explored domain of $\widetilde{R}$ is defined as *domainMin* = 1 and *domainMax* = 100, since these values represent parameters R_{L} and R_{H} in explicit MPC solutions. The domain step size has been chosen as *domainStep* $= 0.5$. The signal variance parameter for the squared exponential kernel in (12) is set to $\sigma_f = 10^5$, under the premise that although the function may exhibit moderate fluctuations and noise, it is unlikely to undergo significant, abrupt changes in amplitude over short intervals. The length-scale parameter $\ell = 30$.

Since the evaluation of the performance metric is time and energy expensive, we limit the number of iterations to $N_{\text{iter}} = 20$. This value is sufficient to obtain a reliable estimate of the optimal $\widetilde{R}$ value while maintaining a reasonable computational cost.

In considered setup, we employ the Upper Confidence Bound (UCB) acquisition function with a key parameter α_{UCB}, which governs the balance between exploration and exploitation.

D. Deployment on the Pocket-Sized Embedded Platform

As discussed in IV-A, the MPT3 Toolbox supports exporting the generated explicit MPC solution to the C programming language. Converting the exported code to C++ and integrating it as a library into the development environment is relatively straightforward. Consequently, both explicit MPC controllers, corresponding to $R_{\mathrm{L}} = 1$ and $R_{\mathrm{H}} = 100$, were adapted and successfully implemented on the ESP32-S3 microcontroller unit (MCU). The implementation was performed using the Arduino IDE[3], which is a widely used development environment for programming microcontrollers. The explicit MPC controllers were integrated into the Arduino IDE as custom libraries, allowing for easy access and utilization within the development environment.

To the best of the authors' knowledge, there is no available C++ library for Bayesian optimization tailored for the ESP32-S3 microcontroller. Consequently, we implemented this algorithm as a considered custom library, which was then registered into the official Arduino Library Manager.

The library is publicly available and can be used for other applications. Users can download it directly in the Arduino IDE software using the Library Manager or download it from the GitHub repository here[4].

The procedure for experimentally obtaining results is as follows:

1) We perform temperature control using both explicit MPC controllers according to the scenario in IV-B and evaluate the control quality metric.
2) We also perform temperature control and evaluate the control quality metric, where the parameter $\widetilde{R} = 50$, which represents an approximate midpoint between the values of R_{L} and R_{H}.
3) These three measurements can be provided to the Bayesian optimization process as input data for finding the optimal value of the parameter $\widetilde{R}$.
4) Based on the results obtained so far, Bayesian optimization determines the next value of the parameter $\widetilde{R}$, and ρ in equation (5) is calculated for this parameter. Using $\mathrm{MPC_L}$ and $\mathrm{MPC_H}$, the values of u_{L} and u_{H} are calculated. The resulting signal $\widetilde{u}$ is calculated as a convex combination of u_{L} and u_{H} according to equation (5) and applied to control the temperature. The resulting control performance is evaluated and provided to Bayesian optimization as the next data point.
5) The previous step is repeated until the number of iterations $N_{\text{iter}} = 20$ is reached. The final result is the optimal value of the parameter $\widetilde{R}$, which provides the best compromise between the fluctuation of the control signal and the energy consumption.

Finally, the corresponding pseudo-code describing the procedure is summarized in Algoritm 1.

Algorithm 1 Bayesian Optimization-Based Tuning of Explicit MPC Controller.

```
Perform temperature control using MPC_L and MPC_H according to the scenario in IV-B.
Evaluate the control quality metric for the parameter R~ = 50.
Initialize the Bayesian optimization algorithm with the parameters described in IV-C.
Provide the results to Bayesian optimization as input data.
for i = 4 to N_iter do
    Determine the next value of the parameter R~ using Bayesian optimization and the UCB acquisition function.
    Calculate ρ according to equation (5).
    Evaluate the control performance by executing the temperature control process described in IV-B and provide the results to Bayesian optimization.
end for
Output: Optimal value of the tuning parameter R~.
```

E. Analysis of the Case Study

The results of the case study provide us with information about each iteration of the Bayesian optimization algorithm and its convergence to the value of the parameter $\widetilde{R}$ considered as optimal. Note that this value is not necessarily the absolute optimum because measurements from the temperature control are affected by noise and also depend on varying ambient conditions. Moreover, we performed only 20 iterations of the Bayesian optimization algorithm, which may not be sufficient to reach the global optimum. However, the results provide valuable insights into the effectiveness of the proposed control strategy and the potential for further improvements.

[3] `https://www.arduino.cc/`

[4] `https://github.com/PeterBakarac/BayesianOptimization/`

1) Exploitation Setup: In a first scenario we set $\alpha_{\text{UCB}} = 1$ in (13), which lead the optimization process to epxloit the most promising area of the search space. This choice is based on the assumption that the function is relatively smooth and that the optimal value of $\widetilde{R}$ is likely to be located in a region with lower uncertainty. The UCB acquisition function will prioritize points with high expected performance, leading to a more focused search around the current best estimate.

Below are listed the results of the Bayesian optimization algorithm for each iteration, including the value of the parameter $\widetilde{R}$, the corresponding control performance metric.

TABLE II
RESULTS OF THE BAYESIAN OPTIMIZATION ALGORITHM

Iteration #	**Tuning** $\widetilde{R}$	**Performance** $J^\star$
1	1.0	-246 627
2	100.0	-216 981
3	50.0	-206 620
4	70.0	-200 398
5	61.5	-195 824
6	62.0	-194 215
7	63.0	-192 941
8	61.5	-191 282
9	60.5	-189 580
10	63.5	-188 216
11	60.5	-187 885
12	61.5	-186 652
13	60.0	-187 611
14	61.0	-186 446
15	62.5	-186 166
16	61.5	-185 559
17	60.5	-184 856
18	58.5	-184 616
19	60.0	-183 704
20	62.0	-183 761

At the end of the experiment, we invoked the method that scans the parameter domain (from *domainMin* to *domainMax* in increments of *domainStep*) and, at each candidate point $\widetilde{R}$, computes the posterior mean (7) predicted by the Gaussian Process. It returns the $\widetilde{R}$ value that achieves the highest posterior mean value. As a result, the optimal value of the parameter $\widetilde{R}$ was determined to be $\widetilde{R} = 60$.

These repeated trials provide sufficient data for thorough visualization and analysis, confirming that the value of $\widetilde{R}$ identified by Bayesian optimization lies in the region that best balances control signal fluctuation against energy consumption.

The Figure 4 presents the outcomes of the Bayesian optimization iterations dataset $\mathcal{D} = \{\mathbf{X}, \mathbf{Y}\}$, where $\mathbf{X}$ is the set of sampled values of the parameter $\widetilde{R}$, and $\mathbf{Y}$ is the corresponding measured closed-loop performance metrics. The figure illustrates the posterior mean (7), and the vertical black line indicates the optimal value of the parameter $\widetilde{R}$.

2) Exploration Setup: In the second scenario, we set $\alpha_{\text{UCB}} = 5$ in (13), which leads the optimization process to explore the search space more extensively. This choice is based on the assumption that the function may exhibit more complex behavior, and a broader exploration of the parameter space is necessary to identify the optimal value of $\widetilde{R}$. The UCB acquisition function will prioritize points with higher uncertainty, allowing for a more comprehensive search across the entire domain.

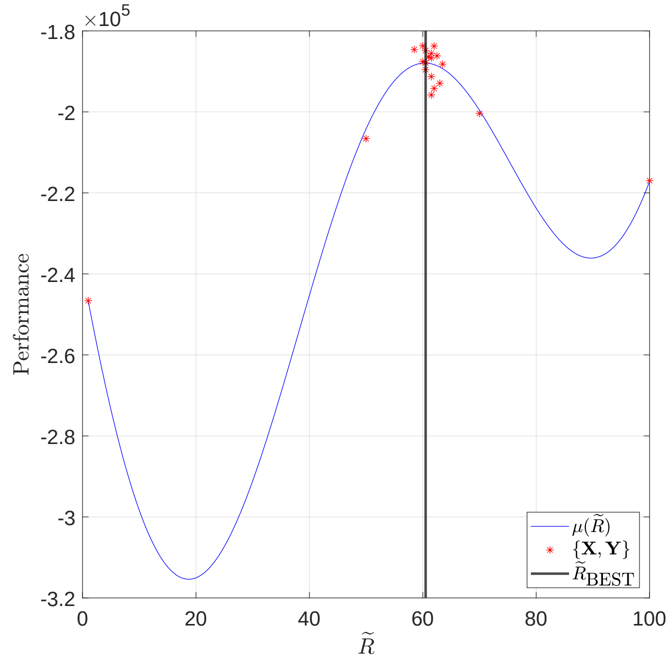

Fig. 4. Results of the Bayesian optimization algorithm set up for exploitation.

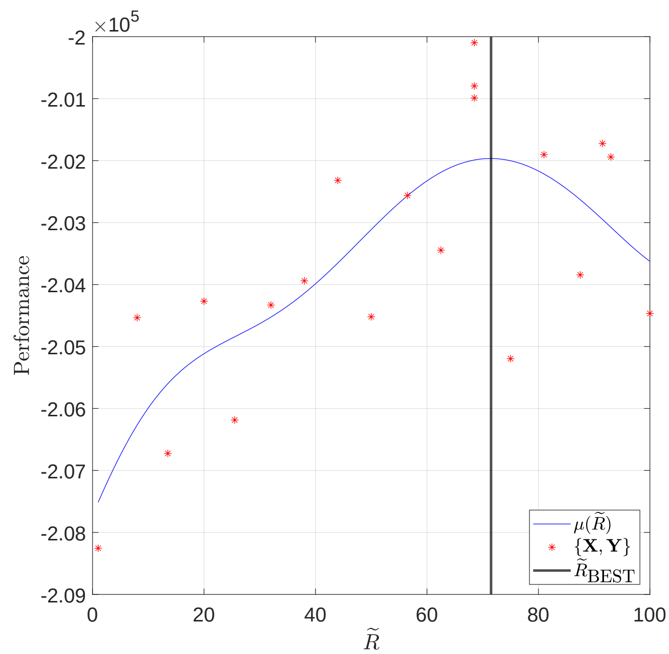

Fig. 5. Results of the Bayesian Optimization algorithm set up for exploration.

The results of the Bayesian optimization process for the exploration setup are depicted in Figure 5. The optimized parameter $\widetilde{R}$ was determined to be $\widetilde{R} = 68.5$, which is slightly higher than the value obtained in the exploitation setup. However, both values are in close proximity, indicating that the optimization process is converging towards a similar region of the parameter space. This suggests that the Bayesian optimization algorithm is effectively applied to the problem of real-time tuning the explicit MPC controller.

F. Results and Discussion

The results presented in the previous section show that the use of Bayesian optimization for tuning the parameter $\widetilde{R}$ for the explicit MPC controller was able to determine a narrow region of its optimal value. The optimization algorithm, as well as both explicit MPC solutions, were successfully implemented in the MCU control unit. Since the measurement of the performance metric was burdened with noise and also depended on slightly changing ambient conditions, it is clear that the value of $\widetilde{R}$ determined by Bayesian optimization may differ from the absolute optimum. However, the value of the parameter $\widetilde{R}$ determined in this way is sufficiently accurate to achieve satisfactory results in real-time temperature control while achieving a compromise defined by the performance metric in (18). The scenario using Bayesian optimization was set up for exploitation and focused on a narrow region with the promise of the highest value of the performance metric. The disadvantage of this approach is that if there is a local optimum in this area, the algorithm may get stuck and fail to find the global optimum. On the other hand, setting up Bayesian optimization for exploration allows for a broader exploration of the search space, which may lead to discovering better values of the parameter $\widetilde{R}$. However, this approach may be more time-consuming and less efficient in the case of smooth functions. Therefore, there is still room for exploring the impact of hyperparameters of Bayesian optimization on its results in the area of tuning the weight coefficients of the MPC controller.

V. CONCLUSIONS

In this paper, we presented a novel real-time tunable approximated explicit MPC strategy driven by Bayesian optimization. The integration of Bayesian optimization enabled efficient, automatic tuning of MPC parameters, substantially improving control accuracy and actuator efficiency. The benefits of the proposed method were validated by its practical implementation on an innovative, pocket-sized embedded platform. Through our temperature control case study, we demonstrated that the Bayesian-optimization-based approach converges rapidly to optimal control settings, effectively balancing performance trade-offs, including closed-loop system stability, overall energy consumption, and computational demands. The embedded implementation of this control framework on the ESP32-S3 microcontroller highlighted its potential for real-world deployment in resource-constrained environments. Future research will focus on extending the proposed tunable MPC approach towards complex multivariable systems, further refining the optimization algorithm to strengthen its real-time applicability, and investigating additional performance criteria relevant to practical-oriented applications.

ACKNOWLEDGMENT

The authors gratefully acknowledge the contribution of the Scientific Grant Agency of the Slovak Republic under the grants 1/0239/24, 1/0297/22, and the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under the project No. 09I01-03-V04-00024. This research is funded by the European Union's Horizon Europe under grant no. 101079342 (Fostering Opportunities Towards Slovak Excellence in Advanced Control for Smart Industries). Erika Pavlovičová thanks for a contribution from the STU Grant scheme for Support of Young Researchers. M. Klaučo is also supported by the European Union project ROBOPROX (Reg. No. CZ.02.01.01/00/22_008/0004590).

REFERENCES

[1] J. Maciejowski, *Predictive Control with Constraints*. London: Prentice Hall, 2000.

[2] A. Alessio and A. Bemporad, "A survey on explicit model predictive control," *Lecture Notes in Control and Information Sciences*, vol. 384, pp. 345–369, 2009.

[3] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.

[4] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, pp. 3–20, 2002.

[5] M. Kvasnica, P. Bakaráč, and M. Klaučo, "Complexity reduction in explicit mpc: A reachability approach," *Systems & Control Letters*, vol. 124, pp. 19–26, 2019.

[6] M. Klaučo and M. Kvasnica, "Towards on-line tunable explicit MPC using interpolation," in *Preprints of the 6th IFAC Conference on Nonlinear Model Predictive Control*, (Madison, Wisconsin, USA), 19-22.8 2018.

[7] J. Oravec and M. Klaučo, "Real-time tunable approximated explicit MPC," *Automatica*, vol. 142, p. 110315, 2022.

[8] J. Xu, "Lattice piecewise affine approximation of explicit linear model predictive control," in *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 2545–2550, 2021.

[9] L. Galčíková and J. Oravec, "Self-tunable approximated explicit mpc: Heat exchanger implementation and analysis," *Journal of Process Control*, vol. 140, p. 103260, 2024.

[10] S. Chen, K. Saulnier, N. Atanasov, D. D. Lee, V. Kumar, G. J. Pappas, and M. Morari, "Approximating explicit model predictive control using constrained neural networks," in *2018 Annual American Control Conference (ACC)*, pp. 1520–1527, 2018.

[11] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Advances in Neural Information Processing Systems*, vol. 25, pp. 2951–2959, 2012.

[12] F. Sorourifar, G. Makrygirgos, A. Mesbah, and J. A. Paulson, "A data-driven automatic tuning method for mpc under uncertainty using constrained bayesian optimization," *IFAC-PapersOnLine*, vol. 54, no. 3, pp. 243–250, 2021. 16th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2021.

[13] R. Guzman, R. Oliveira, and F. Ramos, "Bayesian optimisation for robust model predictive control under model parameter uncertainty," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 5539–5545, 2022.

[14] A. e. a. van der Horst, "A bayesian optimization framework for the automatic tuning of MPC-based shared controllers," *arXiv preprint arXiv:2311.01133*, 2023.

[15] 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 Robotics and Automation Letters*, vol. 8, no. 4, pp. 2397–2404, 2023.

[16] Optimal Control Labs, "Sensoric shi-3-ld platform," 2025. Available online: https://www.optimalcontrollabs.com (accessed March 2025).

[17] P. I. Frazier, "A tutorial on bayesian optimization," 2018.

[18] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger, "Information-theoretic regret bounds for gaussian process optimization in the bandit setting," *IEEE Transactions on Information Theory*, vol. 58, p. 3250–3265, May 2012.

[19] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, "Multi-Parametric Toolbox 3.0," in *Proc. of the European Control Conference*, (Zürich, Switzerland), pp. 502–510, July 17–19 2013.