[TODO: Please, prevent using “When...”-formulation to often.]

[TODO: Please, make consistent notation of equations: Eq-based vs. non-Eq notation.]

[TODO: Please, checkthe word limit.]

The tunable approximated explicit model predictive control (MPC) comes with the benefits of real-time tunability without the necessity of solving the optimization problem online. This paper provides a novel self-tunable control policy that does not require any further tuning or interventions of the control engineer during operation. Based on the current operating conditions, the autonomous tuning parameter scales the control input using linear interpolation between the boundary optimal control actions. The adjustment of the tuning parameter depends on the current reference value, which makes this strategy suitable for reference tracking problems. Furthermore, a novel technique for scaling the tuning parameter is proposed. It provides to exploit different ranges of the tuning parameter assigned to specified operating conditions. The self-tunable explicit MPC was implemented on a laboratory heat exchanger with nonlinear and asymmetric behavior. To ensure offset-free reference tracking control, the built-in integrator was included in the explicit MPC design. The asymmetric behavior of the plant was compensated by tuning the controller’s aggressiveness, as the negative or positive sign of reference change was considered in the tuning process. The designed self-tunable controller improved control performance by decreasing integral square error, maximal overshoots/undershoots, and settling time compared to the conventional control strategy based on a single (non-self-tunable) controller.

1. Introduction

(len skopirovany text z jadra - hodil sa mi viac do Introduction, neskor mozno pouzijem)

[TODO: We need a well-elaborated introduction/review to strongly motivate our control approach. Please, take into account the industrial-oriented reviewers, mostly experts in the field of the chemical engineering (heat exchangers and energy balances).]

The main benefit in form of lower computational complexity in the control phase comes hand in hand with a non-negligible drawback. The size of the parametric solution may increase to the value that prevents its real-time implementation for two reasons: (i) memory footprint is higher than the available memory size of the control unit, and (ii) the computational time associated with finding the optimal control action is higher than the available time period for control action implementation. Although this control strategy has its challenges, it is still very beneficial for practical usage for its benefits.

[TODO: Please, clearly point out the main contributions.]

This work directly extends our results presented in [5], where the basic principles of the self-tunable approximated explicit MPC were introduced. Compared to [5], we...

[TODO: Please, write a paper structure.]

The paper is organized as follows...

2. Theoretical backgrounds

In this section, the theoretical backgrounds necessary for the proposed method is summarized. First, the explicit model predictive control is briefly recalled. Next, the tunable technique of the approximated explicit model predictive control is introduced. Finally, the ideas of self-tunable technique of the approximated explicit MPC are presented.

2.1. Explicit model predictive control

Explicit model predictive control [1] utilizes a parametric solution of the model predictive control introducing its application range towards the systems with the fast dynamics. Moreover, the explicit solution enables providing a rigorous analysis and certification of the closed-loop system stability, constraints satisfaction, etc. As the explicit solution is available, real-time solving of the optimization problem in every control step is omitted. As this work deals with the industrial-oriented implementation, consider the optimization problem in the following form:

$$\min_{u_0,u_1,\ldots,u_{N-1}} \quad \sum_{k=0}^{N-1}\Big((y_k - y_{\mathrm{ref}})^\intercal Q_{\mathrm{y}}(y_k - y_{\mathrm{ref}}) + u_k^\intercal R u_k + x_{\mathrm{I},k}^\intercal Q_{\mathrm{x}} x_{\mathrm{I},k}\Big) \tag{1a}$$

$$\text{s.t.:} \quad \widetilde{x}_{k+1} = \widetilde{A}\,\widetilde{x}_k + \widetilde{B}\,u_k, \tag{1b}$$

$$y_k = \widetilde{C}\,\widetilde{x}_k, \tag{1c}$$

$$u_k \in \mathcal{U}, \tag{1d}$$

$$y_k \in \mathcal{Y}, \tag{1e}$$

$$\widetilde{x}_0 = \theta, \tag{1f}$$

$$k = 0, 1, \ldots, N-1, \tag{1g}$$

where k denotes the step of the prediction horizon N. To obtain the offset-free control results, the built-in integrator was included in the state-space model, e.g., see [2]. The prediction model in (1b)–(1c) has the form of augmented linear time-invariant (LTI) system for given augmented state matrix $\widetilde{A} \in \mathbb{R}^{n_{\widetilde{\mathrm{x}}} \times n_{\widetilde{\mathrm{x}}}}$, augmented input matrix $\widetilde{B} \in \mathbb{R}^{n_{\widetilde{\mathrm{x}}} \times n_{\mathrm{u}}}$ and augmented output matrix $\widetilde{C} \in \mathbb{R}^{n_{\mathrm{y}} \times n_{\widetilde{\mathrm{x}}}}$. Variables $\widetilde{x} \in \mathbb{R}^{n_{\widetilde{\mathrm{x}}}}$, $u \in \mathbb{R}^{n_{\mathrm{u}}}$, $y \in \mathbb{R}^{n_{\mathrm{y}}}$ are vectors of corresponding augmented system states, control inputs and system outputs, respectively. The sets $\mathcal{U} \subseteq \mathbb{R}^{n_{\mathrm{u}}}$, $\mathcal{Y} \subseteq \mathbb{R}^{n_{\mathrm{y}}}$ are convex polytopic sets of physical constraints on inputs and outputs, respectively. The penalty matrix $Q_{\mathrm{y}} \in \mathbb{R}^{n_{\mathrm{y}} \times n_{\mathrm{y}}}$ penalizes the squarred control error, i.e., the deviation between the output and output reference value y_{ref}. The matrix $R \in \mathbb{R}^{n_{\mathrm{u}} \times n_{\mathrm{u}}}$ penalizes the squarred value of control inputs. The value of integrator is also penalized in the cost function with the penalty matrix $Q_{\mathrm{x}} \in \mathbb{R}^{n_{\mathrm{y}} \times n_{\mathrm{y}}}$. The parameter $\theta \in \Theta$ in Eq. (1f) represents the initial condition of the optimization problem for which it is parametrically pre-computed.

The augmented model of the controlled system with the built-in integrator in (1b)–(1c) us rewritten as follows:

$$\widetilde{x}_{k+1} = \begin{bmatrix} x_{k+1} \\ x_{\mathrm{I},k+1} \end{bmatrix} = \begin{bmatrix} A & 0 \\ -T_{\mathrm{s}} C & I \end{bmatrix} \begin{bmatrix} x_k \\ x_{\mathrm{I},k} \end{bmatrix} + \begin{bmatrix} B \\ I \end{bmatrix} u_k, \tag{2a}$$

$$y_k = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x_k \\ x_{\mathrm{I},k} \end{bmatrix}, \tag{2b}$$

where $x_{\mathrm{I}} \in \mathbb{R}^{n_{\mathrm{y}}}$ is the integral of the control error, T_{s} denotes the sampling time and matrices A, B, C are well-known the state-space matrices that form the augmented LTI model. As the consequence of this extension and penalization in the cost function in Eq. (1a), not only the control error are penalized, but also its integral, which leads to analogous offset-free reference tracking results as incorporating integral part in PID controller.

Parametric solution of the optimization problem of the quadratic programming (QP) in (1), leads to the explicit solution in the form of piece-wise affine (PWA) control law defined above the domain consisting of r critical regions:

$$u(\theta) = \begin{cases} F_1\,\theta + g_1 & \text{if} \qquad \theta \in \mathcal{R}_1, \\ F_2\,\theta + g_2 & \text{else if} \qquad \theta \in \mathcal{R}_2, \\ \quad\vdots & \\ F_r\,\theta + g_r & \text{else if} \qquad \theta \in \mathcal{R}_r, \end{cases} \tag{3}$$

where $F_i \in \mathbb{R}^{n_\mathrm{u} \times n_\mathrm{x}}$ and $g_i \in \mathbb{R}^{n_\mathrm{u}}$ respectively are the slope and affine section of the corresponding control law. The PWA function defined in Eq. (3) is stored and recalled in the online phase, i.e., during the real-time control. Based on identifying the specific critical region $\mathcal{R}_i$, where the parameter θ belongs, the optimal control input is calculated based on the associated control law in (3).

Note, many other formulations of the optimization problems for the explicit MPC design, were mainly in the terms of the definition of the cost functions in (1a). Also, the incremental (velocity) formulation of the state-space model is common, but leads to further extension of the vector of parameters θ and therefore also the complexity of the explicit MPC controller increases. Another option for offset-free tracking is introducing the disturbance estimation. For such an overview see e.g.

[TODO: Please, fix.]

...TODO. Klauco reference governors? minimovka zdroje + dizertacka str.33

2.2. Tunable explicit model predictive control

The aggressivity of the controller and whole nature of the control is influenced by appropriate fine tuning of the penalty matrices in the optimization problem (1). When the multi-parametric QP (mp-QP) problem is precomputed offline to obtain the corresponding parametric solution, it is not possible to tune the controller afterwards without trading off a significant increase of the controller complexity or the performance loss. As the operation conditions and requirementson controller setup may differ throughout control, the ability to adjust the controller aggressivity can be very beneficial.

The idea of approximated tunable explicit MPC comes from the work [3], where the control action is calculated based on linear interpolation between two boundary control actions. These control actions result from evaluating two boundary explicit MPCs. The boundary explicit controllers have the same structure and setup, except one of the penalty matrices – the tuned one. The boundary penalty matrices are diagonal square matrices such that $\lambda_{i,\mathrm{L}} \leq \lambda_{i,\mathrm{U}}$, $\forall i = 1, \ldots, s$, where λ denotes the vector of eigenvalues of the penalty matrix, s is the rank of the tuned penalty matrix, and L, U denote the lower and upper boundary setup respectively.

When considering the penalty matrices in cost function (1a), these penalty matrices are scaled in the following way:

$$R(k) = (1-\rho(k))\, R_{\mathrm{L}} + \rho(k)\, R_{\mathrm{U}}, \tag{4a}$$
$$Q_{\mathrm{x}}(k) = (1-\rho(k))\, Q_{\mathrm{x,L}} + \rho(k)\, Q_{\mathrm{x,U}}, \tag{4b}$$
$$Q_{\mathrm{y}}(k) = (1-\rho(k))\, Q_{\mathrm{y,L}} + \rho(k)\, Q_{\mathrm{y,U}}, \tag{4c}$$

where ρ represents the tuning parameter such that $0 \leq \rho \leq 1$ holds. Based on the rules in (4), it is possible to choose online any controller setup from the lower to the upper boundary of the tuned matrix. From the implementation point of view, it is preferred to tune just a single penalty matrix, i.e., to store only two controllers corresponding to the boundary values of the selected penalty matrix. To determine which penalty matrix in (4) should be tuned, it is suggested to judge the control performance subject to the specific system and systematically tune all of the penalty matrices.

When the tuning parameter ρ is determined based on the current control conditions, the approximated optimal control action is evaluated using the two optimal controllers. Based on the boundary control actions, the interpolated, i.e., tuned control action is calculated using the convex combination:

$$u(k) = (1-\rho(k))\, u_{\mathrm{L}} + \rho(k)\, u_{\mathrm{U}}, \tag{5}$$

where u_{L} and u_{U} denote the optimal control actions from the lower and upper boundary controller respectively. The online tuning of the controller online comes with a cost of storing and evaluating two explicit controllers. Nevertheless, the ability to tune the controller may be more important in many practical applications.

We point out, the concept of explicit MPC tuning is applicable for a wide class of MPC design formulations, based on the current specific needs. Without loss of generality, hereafter, we consider the penalty matrices of cost function (1a), as it is necessary to satisfy offset-free reference tracking.

Remark 2.1. *If the asymptotic stability and recursive feasibility guarantees are required, we refer the reader to follow the instructions from [4]. In order to satisfy these requirements, the study introduces a procedure of computing the common terminal penalty and terminal set for the two boundary controllers.*

2.3. Self-tunable explicit model predictive control

[TODO: How about “upgrading” this subsection into the Section?]

The advantage of tunable controller brings a question of how to design the logic of setting the tuning parameter ρ. In this section, the idea of online self-tuning is summarized [5]. The concept of self-tuning provides the possibility to adjust the aggressiveness of the controller without the necessity to intervene and tune the penalty matrices during control.

The need for real-time controller tuning often arises from tracking a time-varying piece-wise constant (PWC) reference. The work [5] focuses on adjusting the penalty matrix when the reference value is changed. The further the reference value is from the steady state, the more aggressive controller is tuned. The idea behind the suggested scaling liest in compensation of the nonlinear behavior of the system.

Consider a single-input and single-output (SISO) system or a multiple-inputs and multiple-outputs (MIMO) system with a completely decoupled pairs of the control inputs and the system outputs. Then, the procedure of tuning the controller is based on evaluating the different operating points between the current value of the reference and the system steady-state value. This deviation is considered to scale the value of control action. First, the maximal admissible absolute value of reference is defined. Analogous to the reference trajectory preview concept of MPC design, this value can be determined based on the general knowledge of the expected future reference values. Another suggestion is to fit the maximal deviation $d_{\max}$ based on the constraints on system outputs:

$$d_{\max} = \max(|y_{\min}|, y_{\max}), \tag{6}$$

where $y_{\min}$ and $y_{\max}$ are respectively lower and upper bound on the output variable in the deviation form, ie., zero corresponds to the system steady-state value. Using the information about the maximal possible deviation $d_{\max}$, the tuning parameter ρ can be calculated as the ratio between the current reference value and the maximal deviation:

$$\rho = \frac{|y_{\mathrm{ref}}|}{d_{\max}}. \tag{7}$$

Based on Eq. (7), the property $0 \leq \rho \leq 1$ holds, as $|y_{\mathrm{ref}}| \leq d_{\max}$. As a consequence, the parameter ρ represents a way how to normalize the deviation from the steady-state value and is exploited to scale the penalty matrix or, implicitly, to scale the aggressiveness of the control action.

When considering tuning the control action based on Eq. (5), a higher value of tuning parameter ρ leads to approaching the upper boundary controller and vice versa. When tuning, e.g., the matrix Q_{y} penalizing the control error, a higher ratio ρ would lead to more aggressive control actions. When operating with the reference value close to the system steady-state value, the parameter ρ decreases and the control profiles becomes sluggish.

Remark 2.2. *The parameter $d_{\max}$ is vector in general, as it depends on the size of system outputs. If $d_{\max}$ is scalar, the parameter ρ is scalar as well and can be directly utilized to scale the control action. If multiple outputs are controlled, it is suggested to calculate the tuning parameter based on the maximal ratio:*

$$\rho(k) = \max\left(\frac{|y_{\mathrm{ref}}(k)|}{d_{\max}}\right). \tag{8}$$

Note, the relations (7) and (8) operate with the absolute value of the reference. It is not taken into account, whether the reference change was positive or negative with respect to the system steady-state value placed in the origin, i.e., whether it changed upwards or downwards. As many plants have asymmetric behavior, the positivity or negativity of the reference change could be considered in the controller tuning to improve the control performance.

3. Methodology

This section extends the ideas of self-tunable explicit MPC in order to improve the control performance. First, a different way of tuning parameter calculation is introduced. Furthermore, an extended self-tunable technique is presented to scale the tuning parameter for industrial-oriented applications, when it is beneficial to exploit a specific range of the tuning parameter in different operating conditions.

3.1. Tuning parameter based on the size of reference change

The approach of self-tunable explicit MPC in [5] suggested tuning based on the current reference value distance from the steady state. The aim is to compensate the nonlinear behavior of system, when using a simple linear prediction model. This work provides also another useful way of the real-time evaluation of the tuning parameter ρ based on the size of reference change.

When different sizes of reference step changes are made and the behavior of the closed-loop system is varying, it can be beneficial to inclue the size of the reference step change into the tuning procedure.

In this approach, the aggressivity is adjusted based on the ratio between the reference step change and the maximal reference step change that can be realized during control operation:

$$\rho(k) = \frac{|\Delta_{\text{ref}}(k)|}{d_{\max}}, \tag{9}$$

where $\Delta_{\text{ref}}(k) = y_{\text{ref}}(k) - y_{\text{ref}}(k-1)$ is the size of the reference step change. The denominator of Eq. (9) is changed as well. In contrast to the maximal deviation from the steady state in Section 2.3, this approach introduces $d_{\max}$ as the maximal possible reference step change given by: $d_{\max}(k) = \|\Delta_{\text{ref}}(k)\|_{\infty}$. Analogously to the original approach, the maximal reference step can be set based on the general knowledge of the expected future reference values. Another option is to exploit the information about the system constraints and set the parameter $d_{\max}$ according to Eq. (6).

Note, only absolute value of Δ_{ref} and $d_{\max}$ are considered in this procedure to ensure $\rho \geq 0$.

Remark 3.1. *The tuning parameter ρ should be updated only when the reference changes. Updating the tuning parameter in the control steps when $\Delta_{\text{ref}} = 0$ would lead to using tuning parameter ρ with zero value, i.e., the control input would correspond to one boundary controller and would not be scaled.*

3.2. Self-tunable technique for systems with asymmetric behavior

[TODO: Please, fix.]

TODO: co ak nepriama umera?
TODO: remark na nelinearnu interpolaciu
TODO: zovseobecnit na viac usekov intervalu?

This paper provides further extension of the self-tuning method proposed in [5]. The suggested technique of tuning is suitable, e.g., for systems with asymmetric behavior, but can be used in any application, where "simple" tuning in the whole range of tuning parameter ρ is not sufficient.

The proposed self-tuning method is based on splitting the interval of the tuning parameter ρ in order to utilize different parts of the interval in different operating conditions. Instead of the original value of tuning parameter ρ, the adjusted tuning parameter $\widetilde{\rho}$ is then utilized to scale the control input according to Eq. (5).

[TODO: Please, consider rearranging the following objects. Missing some formal requirements, such as: "Given..."]

Definition 3.1. *If the tuning parameter is scaled in the first part of the interval, i.e.,* $\langle 0, \rho_\mathrm{s} \rangle$*, where* ρ_s *is the splitting value of the tuning parameter, the following relation is used to adjust the tuning parameter:*

$$\widetilde{\rho} = \rho_\mathrm{s} \rho. \tag{10}$$

[TODO: Please, consider spliting the Corollaries into the Lemma and Proof.]

Corollary 3.1.1. *The following outcomes result from Eq. (10). If* $\rho = 0$*, then* $\widetilde{\rho} = 0$*. When considering the other extreme, i.e.,* $\rho = 1$*, then* $\widetilde{\rho} = \rho_\mathrm{s}$*. In such way, the parameter* ρ *can be scaled in the first part of the interval up to the splitting value. Note, as the relation in Eq. (10) is linear and* $0 \leq \rho \leq 1$ *holds, for* $\widetilde{\rho}$ *it is impossible to acquire other values than in the given interval of parameter* ρ*.*

Definition 3.2. *If the tuning parameter is scaled in the second part of the interval, i.e.,* $\langle \rho_\mathrm{s}, 1 \rangle$*, the following relation is used to adjust the tuning parameter:*

$$\widetilde{\rho} = (1 - \rho_\mathrm{s})\rho + \rho_\mathrm{s}. \tag{11}$$

Corollary 3.1.2. *The following outcomes result from Eq. (11). When* $\rho = 0$*,* $\widetilde{\rho} = \rho_\mathrm{s}$*. When considering the other extreme, i.e.,* $\rho = 1$*, then* $\widetilde{\rho} = 1$*. In such way, the parameter* ρ *can be scaled in the second part of the interval from to the splitting value up to 1. Note, as the relation in Eq. (11) is linear and* $0 \leq \rho \leq 1$ *holds, for* $\widetilde{\rho}$ *it is impossible to acquire other values than in the given interval of parameter* ρ*.*

The advantage of the proposed method remains in the self-tuning of the controller as in the approach from Section 2.3. Nevertheless, it is required to appropriately determine the splitting value of the tuning parameter ρ_s and assign the parts of the interval to the associated operating conditions.

[TODO: Please, consider rearranging the confusing Remark, as the task of the tuning is transformed into the Hessian updates.]

Remark 3.2. *Note, the suggested scaling method is suitable also for online MPC, as the optimization problem is solved in every control step. Therefore, it is possible to include the controller tuning in the procedure of computing the optimal control input. It should be also noted that with real-time updating penalty matrices, the optimization problem has to be rebuilt into a form suitable for a solver, so sufficient computational time is required.*

4. Results and discussion

In this section, the results of the proposed self-tuning method are analyzed using an experimental implementation. The self-tuning strategy utilizes tuning parameter based on the size of reference change (Section 3.1) and the scaling of tuning parameter based on splitting the interval of the parameter and assigning the interval parts to specific operating conditions (Section 3.2).

The plant on which the control was implemented and analyzed is a laboratory-scaled counter-current liquid-to-liquid plate heat exchanger Armfield Process Plant Trainer PCT23 [6], see Figure 1. The cold feed as well as heating medium are transported to the heat exchanger by two peristaltic pumps. The flow rate of the feed of the cold medium (water) is constant, while the aim of control is to track the reference value of the cold medium temperature. Therefore, the controlled variable is the feed temperature T at the outlet of the heat exchanger. The associated manipulated variable is the voltage U corresponding to the power of the pump feeding the heat exchanger by the hot medium (water). The voltage is within the range of $[0-5]\,\mathrm{V}$ normalized into the relative values in percentage. The manipulated variable is constrained from $20\,\%$ to $100\,\%$ to respect the physical limitations of the operating conditions. As the heat exchange is a nonlinear and asymmetric process [7], this heat exchanger represents a suitable candidate for the presented controller tuning strategy.

The matrices of the linear state-space model of the plant are

$$A = \begin{bmatrix}0.839\end{bmatrix}, \quad B = \begin{bmatrix}0.039\end{bmatrix}, \quad C = \begin{bmatrix}1\end{bmatrix}, \tag{12a}$$

considering the sampling time $T_\mathrm{s} = 1\,\mathrm{s}$. The constraints are considered in the terms of control inputs

$$-15\% \leq u \leq 65\% \tag{13}$$

where the variable u represents the control inputs in the deviation form. The values of feed temperature and voltage of the heating medium pump corresponding to zero steady state are respectively $T^{\mathrm{s}} = 35$ °C and $U^{\mathrm{s}} = 35\%$. To fully investigate the control performance of the pure self-tuning method, the constraints on the system states (1e) were omitted to prevent shaping the computed control inputs by taking into account the bounds on the controlled variable.

The penalty matrices of the problem Eq. (1) were systematically tuned and the corresponding control was implemented on the laboratory heat exchanger for each of the considered explicit MPC setup. First, the aim of tuning was to determine, which penalty matrix is the most suitable for real-time tuning. Based on the set of experimentally collected data, the most significant effect on the control trajectories had tuning the penalty matrix Q_{y}, while still preserving a satisfactory control performance, i.e., without steady-state control error and significant oscillations around the reference value. Next, the boundary values of the tunable matrix Q_{y} were tuned as $Q_{\mathrm{y,L}} = 100$ and $Q_{\mathrm{y,U}} = 1000$. The built-in integrator was penalized with the fixed penalty matrix $Q_{\mathrm{x}} = 1$ and the control input with the fixed penalty matrix $R = 10$. The prediction horizon N was set to 20 control steps. The explicit MPC controllers were constructed in MATLAB R2020b using the Multi-Parametric Toolbox 3 [8]. The controllers were implemented to track a time-varying PWC reference.

[TODO: Please, consider introducing the sequence of the reference values/control scenarios here.]

For the initial 200 seconds, the system was in its steady-state value. After that, the reference changed its value twice upwards and twice downwards. The reference changes also acquired different sizes in order to examine the proposed tuning method as it is dependent on the size of the reference step change. The control profiles generated for the both considered boundary control setups are compared in Figure 2 for the controlled outputs, and in Figure 3 for the control inputs. Note, the constructed explicit MPC controller computed control inputs to respect the constraints on the control inputs and they need not to be saturated/truncated afterwards.

The trajectories in Figure 2 show the asymmetric nature of controlling the plant of plate heat exchange mainly when observing the overshoots and undershoots. When applying the control inputs associated with the lower bound $Q_{\mathrm{y,L}}$ in (1a), significant undershoots are present when tracking the reference downwards, i.e., when the reference change is negative. On the contrary, when implementing the controller associated with $Q_{\mathrm{y,U}}$ in (1a), the undershoots are negligible, but significant overshoots can be seen when

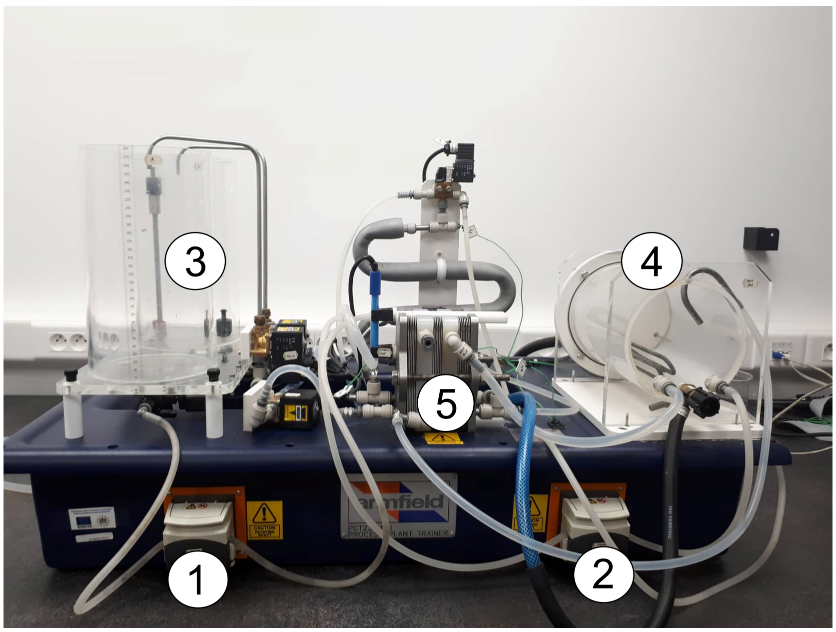

Figure 1: Laboratory heat exchanger Armfield Process Plant Trainer PCT23: feed pump (1), heating medium pump (2), feed tanks (3), heater for heating medium (4), heat exchanger (5).

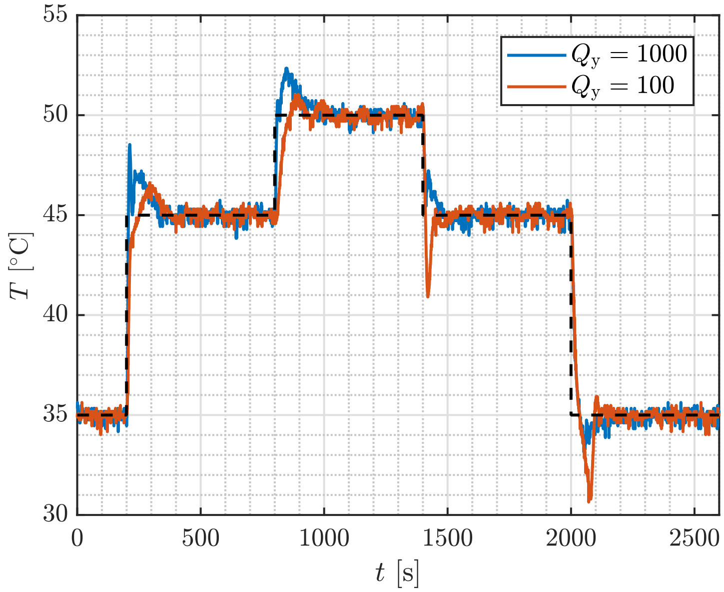

Figure 2: Controlled variable: output cold medium temperature for two boundary controllers. The solid lines represent the controlled temperature T and the dashed line represents the reference value T_{ref}.

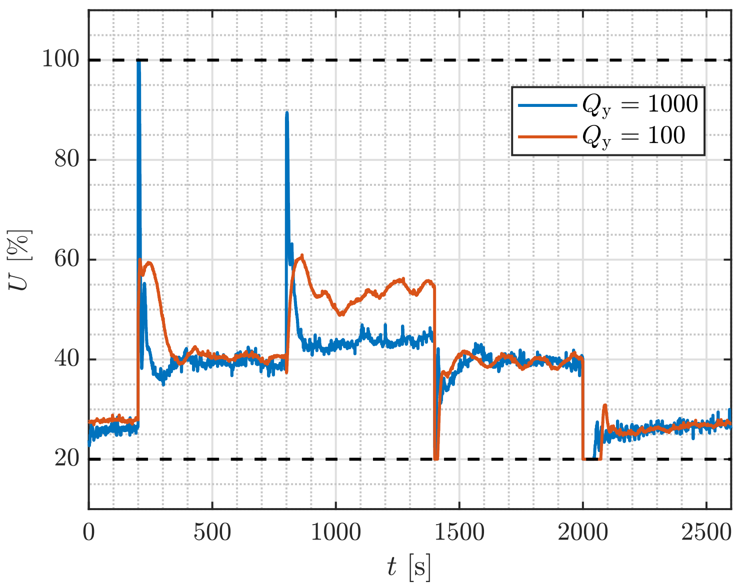

Figure 3: Manipulated variables: input hot medium flow-rate for two boundary controllers. The solid lines represent the voltage U and the dashed lines represent the constraints U_{bound}.

tracking the reference upwards.

These main experimental observations established the base for the strategy of self-tuning the penalty matrix Q_{y}. The strategy follows the ideas summarized in Section 2.3. Therefore, utilizing the boundary controller with the penalty matrix $Q_{\mathrm{y,L}}$ is preferred when the reference changes upwards. On the contrary, the controller associated with $Q_{\mathrm{y,U}}$ is preferred for negative reference step changes. The splitting value of the tuning parameter was chosen in the middle of the interval, i.e., $\rho_{\mathrm{s}} = 0.5$. The remaining parameter that needed to be set was the maximal admissible size of the reference step change $d_{\max}$, which was experimentally determined to 15 °C.

The control results of the self-tunable technique compared to the boundary controllers can be seen in Figure 4 for the controlled outputs, and in Figure 5 for the control inputs. It can be seen that the tuned controller combined the benefits of the two boundary controllers. The overshoots and undershoots were reduced, as in the first half of control the penalty matrix Q_{y} acquired value from the first half of the penalty interval. When tracking the reference with negative step change, the penalty matrix acquired the values from the second half of the interval, i.e., closer to the upper bound $Q_{\mathrm{y,U}}$. The similarity with the boundary controllers can be seen also on the control inputs profiles. Note, the constraints on the input variable were satisfied as they were scaled using linear interpolation based on the boundary controllers which are constructed considering the input constraints.

The control performance was also investigated quantitatively. The Table 1 summarizes the evaluated control performance criteria computed for the two boundary controllers and for the self-tuned controller. The control performance is evaluated for each reference step change separately. The considered quality criteria are: sum-of-squared based criterion – integral squared error ISE, maximal overshoot/undershoot $\sigma_{\max}$, settling time t_{ϵ} for 5 %-neighbourhood of the reference temperature T_{ref}, and the volume of the hot medium V consumed for the corresponding control. To provide a better readability of the computed results in Table 1, the best values, i.e., the minimum values, are emphasized using a bold font style.

As can be seen in Table 1, that real-time self-tuning of the explicit MPC controller helped to improve two to three criteria when tracking each reference value. The cost for this is the energy associated with control. Although the integral square error, maximal overshoot/undershoot and settling time decreased, the volume of consumed heating medium did not. Nevertheless, the average deterioration in the terms of consumed heating medium is ap-

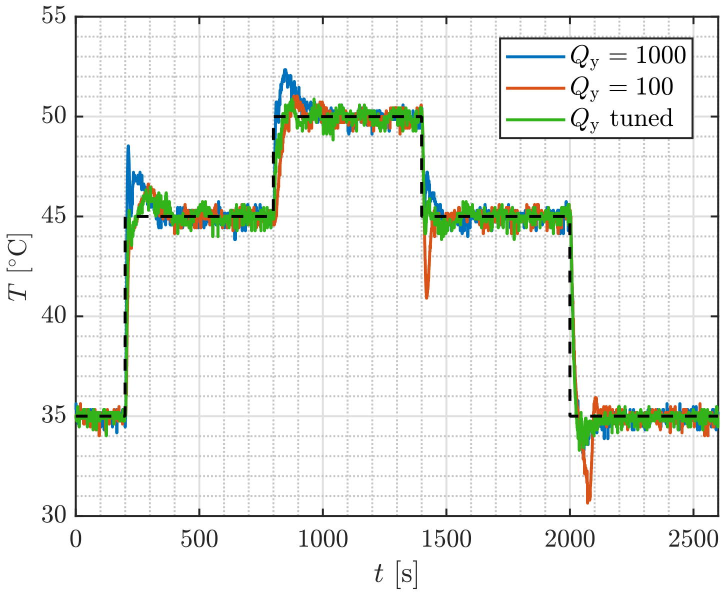

Figure 4: Controlled output trajectory for two boundary controllers and the tuned one. The solid lines represent the controlled temperature and the dashed line represents the reference value.

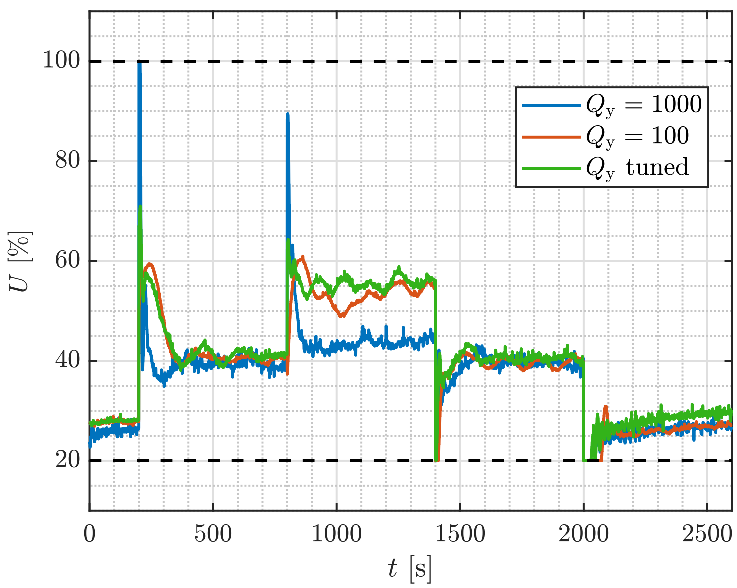

Figure 5: Control input trajectory for two boundary controllers and the tuned one. The solid lines represent the voltage and the dashed lines represent the constraints.

Table 1: Control performance criteria.

Reference step change	Q_{y}	ISE [°C^2 s]	$\sigma_{\max}$ [%]	t_{ϵ} [s]	V [l]
35 °C → 45 °C	1000	714	33.50	16.5	**2.12**
	100	867	16.65	12.5	2.36
	self-tuned	**678**	**15.19**	**9.5**	2.38
45 °C → 50 °C	1000	365	47.20	**5**	**2.49**
	100	606	23.25	26.5	3.19
	self-tuned	**248**	**19.13**	9.5	3.35
50 °C → 45 °C	1000	245	**18.92**	**6.5**	**2.00**
	100	398	79.64	31	**2.00**
	self-tuned	**186**	24.59	**6.5**	2.10
45 °C → 35 °C	1000	1024	18.43	22.5	0.94
	100	1402	41.87	90	**0.93**
	self-tuned	**967**	**16.49**	**18.5**	1.10

proximately by 17 %.

The relative improvement in percentage of using the self-tunable controller computed subject to the second best setup is summarized in Table 2 for each reference step change separately. The negative numbers represent deterioration compared to the best controller setup in the corresponding reference tracking.

Table 2: Relative improvement of the control performance using the self-tunable explicit MPC controller.

Reference step change	ΔISE [%]	$\Delta\sigma_{\max}$ [%]	Δt_{ϵ} [%]	ΔV [%]
35 °C → 45 °C	5	9	24	−12
45 °C → 50 °C	32	18	−90	−35
50 °C → 45 °C	24	−30	0	−4
45 °C → 35 °C	6	11	18	−18

Implementing a self-tunable explicit MPC controller leads to improved control performance in the most analyzed quality criteria, see Table 1. In general, utilizing a proposed controller with a scalable aggressiveness according to the operating conditions leads to higher accuracy (ISE), lower value of the overshoots ($\sigma_{\max}$), and faster achieving the reference value (t_{ϵ}). Although the volume of the hot medium V is not reduced in the considered control scenario, the improved control performance may lead to other benefits.

TODO: krajsie skomentovat vyhody zlepsenia kvality riadenia

[TODO: How about to simply omit the evaluation of the volume V and write something around these lines:]

Compared to the considered non-self-tunable controllers, the generated control trajectories and the computed quality criteria confirmed the improved control performance for the reference tracking control problem of the heat exchanger with the non-linear and asymmetric behaviour. For various operating conditions, the proposed self-tunable controller outperformed the conventional controllers in the means of the squared-error-based criterion (ISE) of the output temperature T within a range $[5 - 30]\,\%$. Together with the reduced overshoots and settling times, the achieved improvements of the control performance enables non-negligible energy savings when the precision and the operating time to ensure the product with the required properties. Although the analyzed extensive case study was realized considering the laboratory scaled heat exhcnager plant, the introduced self-tunable control policy enables significant reduction of the corresponding carbon footprint emissions in the industrial applications.

5. Conclusions

This paper deals with the experimental implementation of the self-tunable approximated explicit model predictive control and provides a strategy for an effective self-tuning controller design. Based on the current reference value, the tuning parameter is scaled using linear interpolation. The previously published work related to the self-tunable explicit MPC suggested tuning just based on the distance of the reference value from the system steady-state value. This paper presents a novel idea of tuning based on the size of reference step change. The self-tuning algorithm aims to compensate for the nonlinear behavior of the controlled system. The self-tuning parameter is updated when the reference changes. It is calculated as the ratio between the size of the reference change and the maximal possible size of the reference change, which is specified before operation. Another novel contribution addresses the challenging control problem of the asymmetric system behaviour by splitting the interval of tuning parameter into two ranges, while both intervals are assigned to different operating conditions. The proposed method is implemented on a laboratory heat exchanger with nonlinear and asymmetric behavior. The non-symmetry makes the plant a suitable candidate for splitting the interval of the tuning parameter. The decision criterium is negativity or positivity of reference change. When the reference changed upwards, the control input was tuned in the first part of the interval and approached the boundary controller associated with the lower bound on the penalty matrix. On the contrary, when the reference changed downwards, the control input was tuned to approach the control input from the boundary controller with the upper bound on the penalty matrix. To investigate the control results properly, the control performance was evaluated also quantitatively. Compared to the conventional control strategy handling just a single controller, the self-tunable control approach decreased the sum-of-squared control error, maximal overshoots/undershoots, and settling time. As the consqquence, the self-tunable controllers enables significant energy savings and the corresponding reduction of the carbon footprint emissions in the industrial applications.

Acknowledgments

The authors gratefully acknowledge the contribution of the Scientific Grant Agency of the Slovak Republic under the grants 1/0545/20, 1/0297/22, the Slovak Research and Development Agency under the project APVV-20-0261. 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).

Nomenclature

Symbols

A	system state matrix
$\widetilde{A}$	augmented system state matrix
B	system input matrix
$\widetilde{B}$	augmented system input matrix
C	system output matrix
$\widetilde{C}$	augmented system output matrix
$d_{\max}$	maximal deviation from the steady state
TODO	odlisit max size of reference step change
F	slope of the affine control law
g	section of the affine control law
I	identity matrix
k	step of the prediction horizon
N	prediction horizon
n_{u}	size of system inputs
n_{y}	size of system outputs
$n_{\widetilde{\mathrm{x}}}$	size of augmented system states
Q_{x}	penalty matrix of the built-in integrator
$Q_{\mathrm{x,L}}$	lower bound on the penalty matrix of the built-in integrator
$Q_{\mathrm{x,U}}$	upper bound on the penalty matrix of the built-in integrator
Q_{y}	penalty matrix of the control error
$Q_{\mathrm{y,L}}$	lower bound on the penalty matrix of the control error
$Q_{\mathrm{y,U}}$	upper bound on the penalty matrix of the control error
R	penalty matrix of system inputs
R_{L}	lower bound on the penalty matrix of system inputs
R_{U}	upper bound on the penalty matrix of system inputs
$\mathcal{R}$	critical region
$\mathbb{R}$	Euclidean space of real numbers
t	time, s
t_{ϵ}	settling time, s
T	temperature, °C
T_{ref}	sampling time, s
T_{s}	sampling time, s
T^{s}	steady state of temperature, °C
u	control inputs

u_{L}	control inputs asscociated with the lower boundary controller
u_{U}	control inputs asscociated with the upper boundary controller
U	voltage, %
U_{bound}	steady state of voltage, %
U^{s}	steady state of voltage, %
$\mathcal{U}$	set of control inputs
V	volume of heating medium, l
x	system states
$\widetilde{x}$	augmented system states
x_{I}	system states corresponding to the built-in integrator
y	system outputs
$y_{\max}$	maximal value of system outputs
$y_{\min}$	minimal value of system outputs
y_{ref}	reference value of system outputs
$\mathcal{Y}$	set of system outputs
$\mathit{0}$	zero matrix

5.1. Greek letters

Δ_{ref}	size of the reference change, °C
ρ: tuning factor	
$\widetilde{\rho}$	scaled tuning factor
ρ_{s}	splitting value of the tuning factor
$\sigma_{\max}$	maximal overshoot, %
θ	parameter of optimization problem
Θ	set of parameter values

Abbreviations

MIMO	multiple-inputs and multiple-outputs (system)
MPC	model predictive control
ISE	integral squared error
LTI	linear time-invariant (system)
mp-QP	multi-parametric quadratic programming (problem)
PWA	piece-wise affine (function)
PWC	piece-wise constant (function)
QP	quadratic programming (problem)
SISO	single-input and single-output (system)

References

[1] A. Bemporad, M. Morari, V. Dua, E. N. Pistikopoulos, The explicit linear quadratic regulator for constrained systems, Automatica 38 (2002) 3–20.

[2] D. D. Ruscio, Model predictive control with integral action: A simple mpc algorithm, Modeling, Identification and Control 34 (3) (2013) 119–129.

[3] M. Klaučo, M. Kvasnica, Towards on-line tunable explicit mpc using interpolation, in: Preprints of the 6th IFAC Conference on Nonlinear Model Predictive Controle, IFAC, Madison, Wisconsin, USA, 2018.

[4] J. Oravec, M. Klaučo, Real-time tunable approximated explicit mpc, Automatica 142 (2022) 110315.

[5] L. Galčíková, M. Horváthová, J. Oravec, M. Bakošová, Self-tunable approximated explicit model predictive control of a heat exchanger, Chemical Engineering Transactions, 2022, Vol. 94 (94) (2022) 1015–1020.

[6] Introduction Manual, PCT23-MkII Process Plant Trainer, Armfield, 2007.

[7] B. G. Liptak, Instrument Engineers' Handbook, 4th Edition, Vol. 2: Process Control and Optimization, CRC Press, London, 2005.

[8] M. Herceg, M. Kvasnica, C. Jones, M. Morari, Multi-parametric toolbox 3.0, in: European Control Conference, 2013, pp. 502–510.