

# Graphical Abstract

**Self-tunable approximated explicit MPC: Heat exchanger implementation and analysys**

Lenka Galčíková, Juraj Oravec, Petronela Belková

## Highlights

### **Self-tunable approximated explicit MPC: Heat exchanger implementation and analysys**

Lenka Galčíková, Juraj Oravec, Petronela Belková

- Self-tunable control technique does not need any intervention during control
- Tuning based on the operating conditions compensates the nonlinear plant behavior
- Implementation of the proposed tuning method on a laboratory heat exchanger
- Control performance improved compared to utilizing only one explicit MPC

# Self-tunable approximated explicit MPC: Heat exchanger implementation and analysys

Lenka Galčíková, Juraj Oravec, Petronela Belková

<sup>a</sup>Slovak University of Technology in Bratislava, Faculty of Chemical and Food Technology, Institute of Information Engineering, Automation, and Mathematics,, Radlinského 9, Bratislava, 81237, Slovakia

---

## Abstract

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 self-tunable technique that does not require any further tuning or interventions of the control engineer during operation. Based on the current operating conditions, the tuning parameter adjusts itself and scales the control input using linear interpolation. The adjustment of the tuning parameter depends on the 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 non-symmetric behavior. To ensure offset-free control, the built-in integrator was included in the explicit MPC. The asymmetric behavior of the plant was compensated by tuning the controller's aggressivity, as the negativity or positivity of reference change was considered in the tuning process. The control performance decreased maximal overshoots/undershoots, integral square error, and settling time compared to control with only one controller.

*Keywords:* tunable explicit MPC, self-tunable technique, tuning parameter, heat exchanger, control performance

---

## 1. Introduction

(len skopirovany text z jadra - hodil sa mi viac do Introduction, neskor možno pouzijem) The benefit in form of lower computational complexity in

the control phase comes hand in hand with a drawback. The size of the parametric solution may be so large, that it becomes impractical to utilize for two reasons: (i) memory footprint is higher than the available memory size of the control unit, (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.

## 2. Preliminaries

In this section, the theoretical background necessary for this work is summarized. First, the explicit model predictive control is recalled. Next, the tunable technique of the approximated explicit model predictive control is introduced. Last but not least, the ideas of self-tunable technique of the approximated explicit MPC are presented.

### 2.1. Explicit model predictive control

Explicit model predictive control [1] represents a parametric solution of the model predictive control which makes it suitable for online implementation. As the explicit solution is available, it is not necessary to solve the optimization problem in every control step. As this work deals with practical implementation, let us consider the optimization problem in the following form:

$$\min_{u_0, u_1, \dots, u_{N-1}} \sum_{k=0}^{N-1} ((y_k - y_{\text{ref}})^T Q_y (y_k - y_{\text{ref}}) + u_k^T R u_k + x_{I,k}^T Q_x x_{I,k}) \quad (1a)$$

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

$$y_k = \tilde{C} \tilde{x}_k, \quad (1c)$$

$$u_k \in \mathcal{U}, \quad (1d)$$

$$y_k \in \mathcal{Y}, \quad (1e)$$

$$\tilde{x}_0 = \theta, \quad (1f)$$

$$k = 0, 1, \dots, N-1, \quad (1g)$$

where  $k$  denotes the step of the prediction horizon  $N$ . The parameter  $\theta \in \Theta$  in Eq. (1f) represents the initial condition of the optimization problem for which it is parametrically precomputed. Variables  $\tilde{x} \in \mathbb{R}^{n_{\tilde{x}}}$ ,  $u \in \mathbb{R}^{n_u}$ ,  $y \in \mathbb{R}^{n_y}$  are vectors of corresponding extended system states, control inputs

and system outputs, respectively. The prediction model in (1b)–(1c) has the form of extended linear time-invariant (LTI) system for given extended state matrix  $\tilde{A} \in \mathbb{R}^{n_{\bar{x}} \times n_{\bar{x}}}$ , extended input matrix  $\tilde{B} \in \mathbb{R}^{n_{\bar{x}} \times n_u}$  and extended output matrix  $\tilde{C} \in \mathbb{R}^{n_y \times n_{\bar{x}}}$ . The sets  $\mathcal{U} \subseteq \mathbb{R}^{n_u}$ ,  $\mathcal{Y} \subseteq \mathbb{R}^{n_y}$  are sets of physical constraints on inputs and outputs, respectively. The penalty matrix  $Q_y \in \mathbb{R}^{n_y \times n_y}$  penalizes the squared control error, i.e., the deviation between the output and output reference value  $y_{\text{ref}}$ . The matrix  $R \in \mathbb{R}^{n_u \times n_u}$  penalizes the squared value of control inputs. To obtain the offset-free control results, the built-in integrator was included in the state-space model, see e.g. [2]. The value of integrator is also penalized in the cost function with the penalty matrix  $Q_x \in \mathbb{R}^{n_y \times n_y}$ .

The model extended with the built-in integrator in (1b)–(1c) can be rewritten as:

$$\tilde{x}_{k+1} = \begin{bmatrix} x_{k+1} \\ x_{I,k+1} \end{bmatrix} = \begin{bmatrix} A & 0 \\ -T_s C & I \end{bmatrix} \begin{bmatrix} x_k \\ x_{I,k} \end{bmatrix} + \begin{bmatrix} B \\ I \end{bmatrix} u_k, \quad (2a)$$

$$y_k = [C \ 0] \begin{bmatrix} x_k \\ x_{I,k} \end{bmatrix}, \quad (2b)$$

where  $x_I \in \mathbb{R}^{n_y}$  is the integral of the control error,  $T_s$  denotes the sampling time and matrices  $A$ ,  $B$ ,  $C$  are well-known linear time-invariant state-space matrices that form the extended model. Thanks to 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 similar results as incorporating integral part in PID controller.

Solving the MPC quadratic optimization problem in Eq. (1) parametrically leads to explicit solution in the form of piece-wise affine control law consisting of  $r$  critical regions:

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

where  $F_i$  and  $g_i$  respectively are the slope and affine section of the corresponding control law. The piece-wise affine function from Eq. (3) is stored and used in the online, i.e., control phase. Based on identifying the critical region where the parameter  $\theta$  belongs, the optimal control input is calculated based on the associated control law.

Note, many other formulations of the optimization problems are elaborated, mainly in the terms of cost functions. Also, the incremental (velocity) formulation of the state-space model is common, but leads to further extension of the vector of parameters  $\theta$  and therefore the explicit MPC complexity increases. Another option for offset-free tracking is also disturbance observer. For such an overview see e.g. ...TODO. Klauso 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 can be influenced by tuning the penalty matrices in the optimization problem (1). When the optimization problem is precomputed offline to obtain the parametric solution, it is not possible to tune the controller afterwards. As the operation conditions and demands on 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 matrices are diagonal square matrices such that  $\lambda_{i,L} \leq \lambda_{i,U}$ ,  $\forall i = 1, \dots, s$ , where  $\lambda$  denotes the vector of penalty matrix eigenvalues,  $L$  and  $U$  denote the lower and upper boundary setup respectively, and  $s$  is the rank of the tuned penalty matrix.

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

$$R = (1 - \rho)R_L + \rho R_U, \quad (4a)$$

$$Q_x = (1 - \rho)Q_{x,L} + \rho Q_{x,U}, \quad (4b)$$

$$Q_y = (1 - \rho)Q_{y,L} + \rho Q_{y,U}, \quad (4c)$$

where  $\rho$  represents the tuning parameter and  $0 \leq \rho \leq 1$  holds. Based on (4), it is possible to choose online any controller setup from the lower to the upper boundary of the tuned matrix. Note, it is practical to tune only one matrix, i.e., to store only two controllers corresponding to the boundary matrices. To determine which penalty matrix should be tuned, it is suggested to examine the control results subject to the specific system and to try systematic tuning of all the penalty matrices.

When the tuning parameter  $\rho$  is determined based on the current control conditions, the 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:

$$u = (1 - \rho)u_L + \rho u_U, \quad (5a)$$

where  $u_L$  and  $u_U$  denote the optimal control actions from the lower and upper boundary controller respectively. The 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.

Note, the concept of explicit MPC tuning can be applied also to other MPC formulations, based on the current specific needs. In this paper, the penalty matrices of cost function (1a) are considered, as it is necessary to satisfy offset-free reference tracking.

**Remark 2.1.** *If stability guarantee and recursive feasibility is required, it is suggested to follow the instructions from [4]. In order to satisfy the requirements, the study introduces a procedure of choosing the terminal penalty and terminal set in the two boundary controllers.*

### 2.3. Self-tunable explicit model predictive control

The advantage of tunable controller brings a question of how to design the logic of setting the tuning parameter  $\rho$ . 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 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 lies in compensation of the nonlinear behavior of the system.

The procedure of tuning the controller is based on evaluating the difference between the reference and the steady state, and using this deviation to scale the value of control action. First, the maximal absolute value of reference which can be set during control is defined. This value can be determined

based on the general knowledge of future reference values. Another suggestion is to set the maximal deviation  $d_{\max}$  simply based on the constraints on system outputs:

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

where  $y_{\min}$  and  $y_{\max}$  are respectively lower and upper bound on the output variable. Using the information about the maximal possible deviation  $d_{\max}$ , the tuning parameter  $\rho$  can be calculated as the ratio between the current reference value and the maximal deviation:

$$\rho = \frac{|y_{\text{ref}}|}{d_{\max}}. \quad (7)$$

Based on Eq. (7), the property  $0 \leq \rho \leq 1$  holds, as  $|y_{\text{ref}}| \leq d_{\max}$ . As a result, the parameter  $\rho$  represents a way how to normalize the deviation from steady state and is exploited to scale the penalty matrix or at the end of the day, the control action.

When considering tuning the control action based on Eq. (5a), a higher value of tuning parameter  $\rho$  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  $\rho$  would lead to more aggressive control actions. When controlling to the reference values closer to the steady state, the parameter  $\rho$  decreases and the control 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  $\rho$  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:*

$$\rho = \max \left( \frac{|y_{\text{ref}}|}{d_{\max}} \right). \quad (8)$$

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, i.e., whether it changed upwards or downwards. As many plants have non-symmetric 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 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 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 another way of the tuning parameter calculation 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 include 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 set during control operation:

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

where  $\Delta_{\text{ref}} = y_{\text{ref},k} - y_{\text{ref},k-1}$  is the size of the reference 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. Analogously to the original approach, the maximal reference step can be set based on the general knowledge of 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  $\Delta_{\text{ref}}$  and  $d_{\max}$  are considered in this procedure to ensure  $\rho \geq 0$ .

**Remark 3.1.** *The tuning parameter  $\rho$  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  $\rho$  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: 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 non-symmetric behavior, but can be used in any application, where „simple” tuning in the whole range of tuning parameter  $\rho$  is not sufficient.

The tuning method is based on splitting the interval of the tuning parameter  $\rho$  in order to utilize different parts of the interval in different operating conditions. The adjusted tuning parameter  $\tilde{\rho}$  is then utilized to scale the control input according to Eq. (5a) instead of the original value of tuning parameter  $\rho$ .

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

$$\tilde{\rho} = \rho_s \rho. \quad (10)$$

**Corollary 3.1.1.** *The following outcomes result from Eq. (10). If  $\rho = 0$ , then  $\tilde{\rho} = 0$ . When considering the other extreme, i.e.,  $\rho = 1$ , then  $\tilde{\rho} = \rho_s$ . In such way, the parameter  $\rho$  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  $\tilde{\rho}$  it is impossible to acquire other values than in the given interval of parameter  $\rho$ .*

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

$$\tilde{\rho} = (1 - \rho_s) \rho + \rho_s. \quad (11)$$

**Corollary 3.1.2.** *The following outcomes result from Eq. (11). When  $\rho = 0$ ,  $\tilde{\rho} = \rho_s$ . When considering the other extreme, i.e.,  $\rho = 1$ , then  $\tilde{\rho} = 1$ . In such way, the parameter  $\rho$  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  $\tilde{\rho}$  it is impossible to acquire other values than in the given interval of parameter  $\rho$ .*

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 determine the splitting value of the tuning parameter  $\rho_s$  and assign the parts of the interval to the associated operating conditions.

**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 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 tuning method are presented on an experimental implementation. The 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 is a laboratory liquid-to-liquid plate heat exchanger Armfield Process Plant Trainer PCT23 [6], see Fig. 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 is constant, while the aim is to track the reference value of its temperature. Therefore, the controlled output is the feed temperature  $T$  at the outlet of the heat exchanger and the control input is the voltage  $U$  corresponding to the power of the heating medium pump. The voltage is set in percentage, while it is constrained from 20% to 100%. As the heat exchange is a nonlinear and non-symmetric 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, discretized with sampling time  $T_s = 1$  s, are

$$A = [0.839], \quad (12a)$$

$$B = [0.039], \quad (12b)$$

$$C = [1], \quad (12c)$$

$$D = [0]. \quad (12d)$$

The constraints are considered in the terms of control inputs, i.e.

$$-15\% \leq u \leq 65\%. \quad (13)$$

Note, 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  $T^s = 35^\circ\text{C}$  and  $U^s = 35\%$ .

The penalty matrices of the problem Eq. (1) were systematically tuned and the corresponding control was implemented on the laboratory heat exchanger for every MPC setup. The aim was to determine, which penalty matrix will be the tuned one. Based on observations, 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 setpoint. The boundary values of matrix  $Q_y$  were tuned as  $Q_{y,L} = 100$  and  $Q_{y,U} = 1000$ . The built-in integrator was penalized with  $Q_x = 1$  and the control input with  $R = 10$ . The prediction horizon  $N$  was 20 steps long. The explicit model predictive controllers were constructed in MATLAB R2020b using the Multi-Parametric Toolbox 3 [8]. The controllers were implemented to track a time-varying reference. For the first 200 seconds, the system was in the steady state. 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 results for both control setups are compared in Fig. 2 for the controlled outputs, and in Fig. 3 for the control inputs.

The trajectories in Fig. 2 show the non-symmetric nature of controlling the process of heat exchange mainly when observing the overshoots and undershoots. When applying the control associated with the lower bound  $Q_{y,L}$ , 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  $Q_{y,U}$ , the undershoots are negligible, but significant overshoots can be seen when tracking the reference upwards.

These observations form the base for the strategy of self-tuning the penalty matrix  $Q_y$ . The strategy follows the ideas summarized in Section ???. Utilizing the boundary controller with the penalty matrix  $Q_{y,L}$  is preferred when the reference changes upwards. On the contrary, the controller associated with  $Q_{y,U}$  is preferred in control with negative reference changes. The splitting value of the tuning parameter was chosen in the middle of the interval,

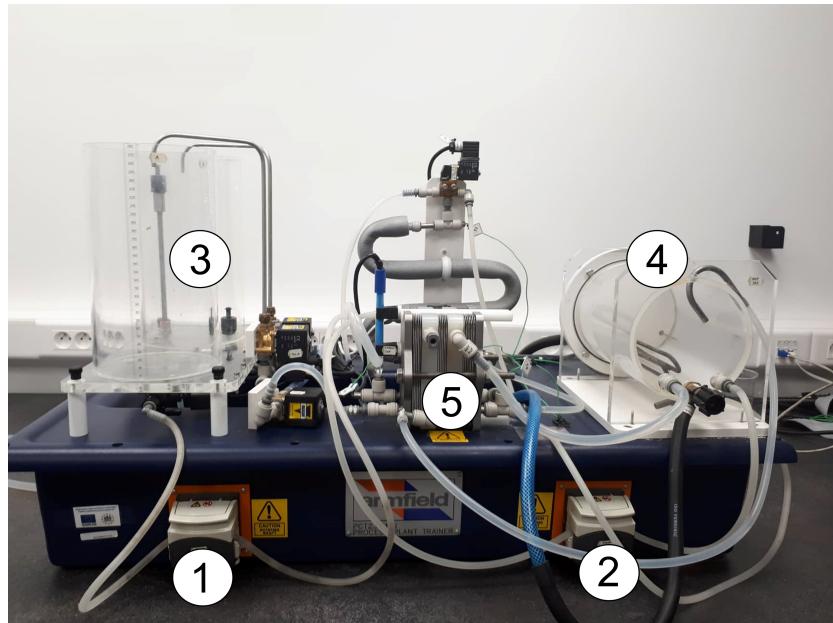


Figure 1: Laboratory heat exchanger Armfield Process Plant Trainer PCT23: feed pump (1), heating medium pump (2), feed tanks (3), heater for heating medium (4), heat exchanger (5).

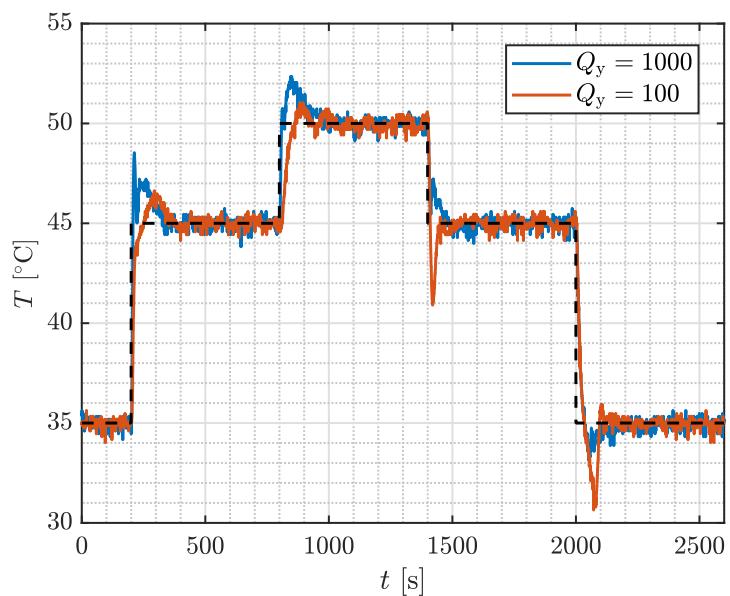


Figure 2: Controlled output trajectory for two boundary controllers. The solid lines represent the controlled temperature and the dashed line represents the reference value.

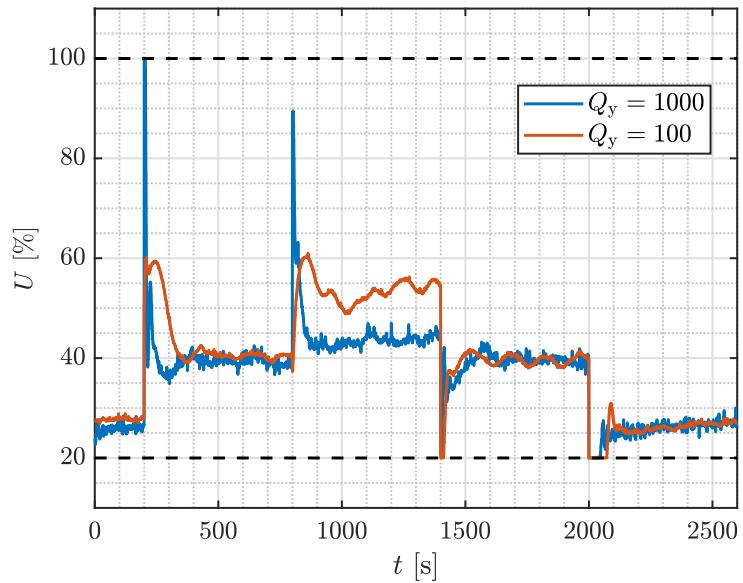


Figure 3: Control input trajectory for two boundary controllers. The solid lines represent the voltage and the dashed lines represent the constraints.

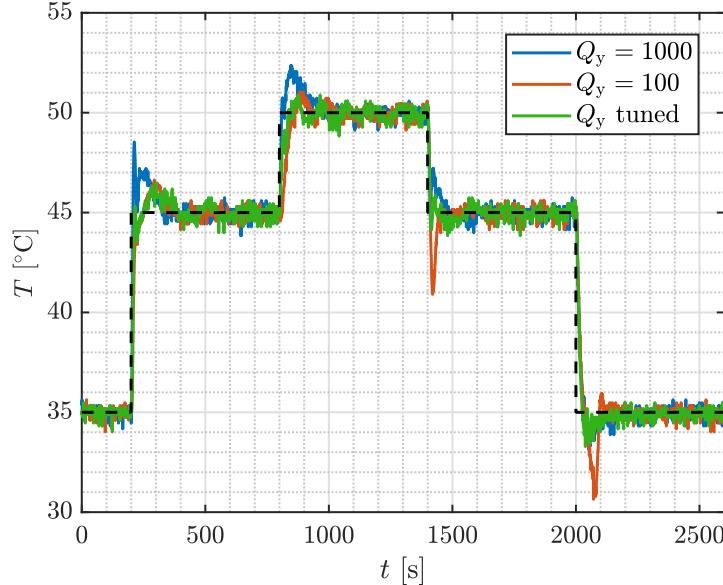


Figure 4: Controlled output trajectory for two boundary controllers and the tuned one. The solid lines represent the controlled temperature and the dashed line represents the reference value.

i.e.,  $\rho_s = 0.5$ . The last parameter that needed to be set was the maximal possible size of the reference step change  $d_{\max}$ , which was set to  $15^\circ\text{C}$ .

The control results of the self-tunable technique compared to the boundary controllers can be seen in Fig. 4 for the controlled outputs, and in Fig. 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 change, the penalty matrix acquired the values from the second half of the interval, i.e., closer to the upper bound  $Q_{y,U}$ . The similarity with the boundary controllers can be seen also on the control inputs trajectory. 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 constraints.

TODO: what about state constraints?

The control performance was also investigated quantitatively. The Ta-

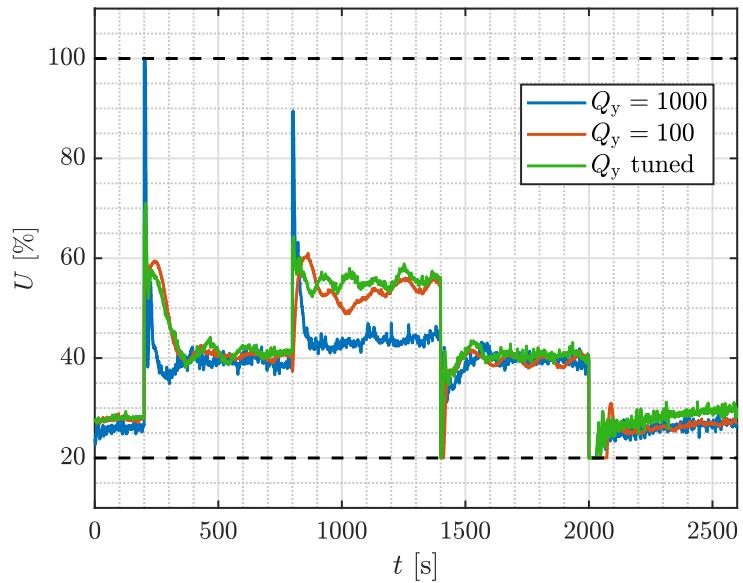


Figure 5: Control input trajectory for two boundary controllers and the tuned one. The solid lines represent the voltage and the dashed lines represent the constraints.

ble 1 summarizes the evaluated control performance criteria for the tuned and two boundary controllers. The control performance is evaluated for every reference step change respectively. The criteria are integral square error  $ISE$ , maximal overshoot/undershoot  $\sigma_{\max}$ , settling time  $t_\epsilon$  for 5%-neighbourhood of the reference value, and the volume of the heating medium utilized for the corresponding control. To provide better readability of the results, the best (i.e. minimal) values are written in bold.

Table 1: Control performance criteria.

Reference step change	$Q_y$	$ISE$ [ $^{\circ}\text{C}^2 \text{s}$ ]	$\sigma_{\max}$ [%]	$t_\epsilon$ [s]	$V$ [l]
1	1000	714	33.50	16.5	<b>2.12</b>
	100	867	16.65	12.5	2.36
	tuned	<b>678</b>	<b>15.19</b>	<b>9.5</b>	2.38
2	1000	365	47.20	<b>5</b>	<b>2.49</b>
	100	606	23.25	26.5	3.19
	tuned	<b>248</b>	<b>19.13</b>	9.5	3.35
3	1000	245	<b>18.92</b>	<b>6.5</b>	<b>2.00</b>
	100	398	79.64	31	<b>2.00</b>
	tuned	<b>186</b>	24.59	<b>6.5</b>	2.10
4	1000	1024	18.43	22.5	0.94
	100	1402	41.87	90	<b>0.93</b>
	tuned	<b>967</b>	<b>16.49</b>	<b>18.5</b>	1.10

It can be seen in Table 1, that real-time tuning of the 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 approximately 17%.

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

It can be seen that implementing a tunable controller leads to improved control performance in many criteria. Utilizing a controller with a scalable aggressivity according to the operating conditions leads in general to higher

Table 2: Improvement or deterioration of the control performance in percentage.

Reference step change	$ISE$	$\sigma_{\max}$	$t_\epsilon$	$V$
1	5.04	8.77	24.00	-12.26
2	32.05	17.72	-90.00	-34.54
3	24.08	-29.97	0	-4.48
4	5.57	10.53	17.78	-18.28

accuracy, lower overshoot and faster achieving the reference value. Although the volume of the heating medium is not saved in this control scenario, the improved control performance may lead to other benefits.

TODO: krajsie skomentovat výhody zlepšenia kvality riadenia

## 5. Conclusion

This paper deals with the experimental implementation of tunable approximated explicit model predictive control and provides a technique for self-tuning controller design. Based on the current reference value, the tuning parameter is scaled using linear interpolation. The previously published paper related to the self-tunable explicit MPC suggested tuning based on the distance of the reference value from the steady state. This paper presents a novel idea of tuning based on the size of reference step change. The tuning procedures aim to compensate for the nonlinear behavior of the controlled system. The 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 comes with splitting the interval of tuning parameter into two ranges, while both ranges are assigned to different operating conditions. The proposed method is implemented on a laboratory heat exchanger with nonlinear and non-symmetric 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 quantitatively. Compared to control with only one controller, the self-tunable technique helped to decrease the maximal overshoots/undershoots, integral square error, and settling time.

## Acknowledgements

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). 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.

## Nomenclature

### 5.1. Abbreviations

*MPC*: model predictive control

*ISE*: integral square error

*LTI*: linear time-invariant

### 5.2. Symbols

$A$ : system state matrix

$\tilde{A}$ : extended system state matrix

$B$ : system input matrix

$\tilde{B}$ : extended system input matrix

$C$ : system output matrix

$\tilde{C}$ : extended 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

$n_{\tilde{x}}$ : size of extended system states  
 $Q_x$ : penalty matrix of the built-in integrator  
 $Q_{x,L}$ : lower bound on the penalty matrix of the built-in integrator  
 $Q_{x,U}$ : upper bound on the penalty matrix of the built-in integrator  
 $Q_y$ : penalty matrix of the control error  
 $Q_{y,L}$ : lower bound on the penalty matrix of the control error  
 $Q_{y,U}$ : 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  
 $t_\epsilon$ : settling time, s  
 $T$ : temperature, °C  
 $T_s$ : sampling time, s  
 $T^s$ : steady state of temperature, °C  
 $u$ : control inputs  
 $u_L$ : control inputs associated with the lower boundary controller  
 $u_U$ : control inputs associated with the upper boundary controller  
 $U$ : voltage, %  
 $U^s$ : steady state of voltage, %  
 $\mathcal{U}$ : set of control inputs  
 $V$ : volume of heating medium, l  
 $x$ : system states  
 $\tilde{x}$ : extended 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  
 $y_{\text{ref}}$ : reference value of system outputs  
 $\mathcal{Y}$ : set of system outputs  
 $\theta$ : zero matrix

### 5.3. Greek letters

$\Delta_{\text{ref}}$ : size of the reference change, °C  
 $\rho$ : tuning factor  
 $\tilde{\rho}$ : scaled tuning factor

$\rho_s$ : splitting value of the tuning factor

$\sigma_{\max}$ : maximal overshoot, %

$\theta$ : parameter of optimization problem

$\Theta$ : set of parameter values

## 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.