

Slovenská technická univerzita v Bratislave
Fakulta chemickej a potravinárskej technológie
Ústav informatizácie, automatizácie a matematiky





**SÚBOR 10 VYBRANÝCH NAJVÝZNAMNEJŠÍCH
VEDECKÝCH PRÁC**

doc. Ing. Juraj Oravec, PhD.

Bratislava, 2024

1. Jiang, Y. [25%] – Oravec, J. [25%] – Houska, B. [25%] – Kvasnica, M. [25%]: Parallel MPC for Linear Systems with Input Constraints. **IEEE Transactions on Automatic Control (IF: 6,2)**, č. 7, zv. 66, str. 3401–3408, ISSN: 0018-9286, 2021.
2. Oravec, J. [60%] – Klaučo, M. [40%]: Real-time tunable approximated explicit MPC. **Automatica (IF: 4,8)**, zv. 142, str. 110315, ISSN: 1873-2836, 2022.
3. Drgoňa, J. [20%] – Bastida, J. A. [10%] – Figueroa, I. C. [10%] – Blum, D. [10%] – Arendt, K. [10%] – Kim, D. [10%] – Ollé, E. P. [5%] – Oravec, J. [10%] – Wetter, M. [5%] – Vrabie, D. [5%] – Helsen, L. [5%]: All you need to know about model predictive control for buildings. **Annual Reviews in Control (IF: 7,3)**, zv. 50, str. 190–232, ISSN: 1367-5788, 2020.
4. Oravec, J. [34%] – Horváthová, M. [33%] – Bakošová, M. [33%]: Energy efficient convex-lifting-based robust control of a heat exchanger. **Energy (IF: 9,0)**, č. 201, str. 117566, ISSN: 0360-5442, 2020.
5. Dyrská, R. [30%] – Horváthová, M. [30%] – Bakaráč, P. [20%] – Mönnigmann, M. [5%] – Oravec, J. [15%]: Heat exchanger control using model predictive control with constraint removal. **Applied Thermal Engineering (IF: 6,1)**, zv. 227, ISSN: 1359-4311, 2023.
6. Galčíková, L. [60%] – Oravec, J. [40%]: Self-tunable approximated explicit MPC: Heat exchanger implementation and analysis. **Journal of Process Control (IF: 3,3)**, zv. 140, str. 103260, ISSN: 0959-1524, 2024.
7. Galčíková, L. [60%] – Oravec, J. [40%]: Fixed complexity solution of partial explicit MPC. **Computers & Chemical Engineering (IF: 3,9)**, zv. 157, ISSN: 0098-1354, 2022.
8. Oravec, J. [40%] – Bakošová, M. [30%] – Galčíková, L. [15%] – Slávik, M. [5%] – Horváthová, M. [5%] – Mészáros, A. [5%]: Soft-constrained robust model predictive control of a plate heat exchanger: Experimental analysis. **Energy (IF: 5,0)**, zv. 180, str. 303–314, ISSN: 0360-5442, 2019.
9. Bakaráč, P. [25%] – Horváthová, M. [25%] – Galčíková, L. [25%] – Oravec, J. [20%] – Bakošová, M. [5%]: Approximated MPC for embedded hardware: Recursive random shooting approach. **Computers & Chemical Engineering (IF: 3,9)**, zv. 165, ISSN: 0098-1354, 2022.
10. Oravec, J. [45%] – Bakošová, M. [35%] – Trafczynski, M. [3%] – Vasičkaninová, A. [10%] – Mészáros, A. [5%] – Markowski, M. [2%]: Robust model predictive control and PID control of shell-and-tube heat exchangers. **Energy (IF: 5,0)**, zv. 159, str. 1–10, ISSN: 0360-5442, 2018.

Parallel MPC for Linear Systems With Input Constraints

Yuning Jiang , Member, IEEE, Juraj Oravec , Boris Houska , and Michal Kvasnica , Member, IEEE

Abstract—This article is about a real-time model predictive control algorithm for large-scale, structured linear systems with polytopic control constraints. The proposed controller receives the current state measurement as an input, and computes a suboptimal control reaction by evaluating a finite number of piecewise affine functions that correspond to the explicit solution maps of small-scale parametric quadratic programming (QP) problems. We provide asymptotic stability guarantees, which can be verified offline. The feedback controller is computing approximations of the optimal input, because we are enforcing real-time requirements assuming that it is not possible to solve the given large-scale QP in the given amount of time. Here, a key contribution of this article is that we provide a bound on the suboptimality of the controller. The approach is illustrated by benchmark case studies.

Index Terms—Model predictive control (MPC), parametric optimization.

I. INTRODUCTION

The advances of numerical optimization methods over the last decades [1], in particular, the development of efficient quadratic programming problem (QP) solvers [2], have enabled numerous industrial applications of model predictive control (MPC) [3]. Modern real-time optimization and control software packages [4], [5] achieve run-times in the milli- and microsecond range by generating efficient and reliable C-code [6], [7]. However, as much as these algorithms perform well on desktop computers or other devices with comparable computation power, the number of successful implementations of MPC on embedded industrial hardware, such as programmable logic controllers and field-programmable gate arrays, remains limited [8]. Here, the main question is what can be done if an embedded device has not enough computational power or storage space to solve the exact MPC problem in real-time.

Manuscript received February 17, 2020; revised June 22, 2020; accepted August 24, 2020. Date of publication September 1, 2020; date of current version June 29, 2021. This work was supported by bilateral Grant agreement between China and Slovakia SK-CN-2015-PROJECT-6558, and APVV SK-CN-2015-0016. The work of Yuning Jiang and Boris Houska was supported by the National Science Foundation China (NSFC), under Grant 61473185, as well as ShanghaiTech University, under Grant F-0203-14-012. The work of Juraj Oravec and Michal Kvasnica was supported by contribution of the Scientific Grant Agency of the Slovak Republic under Grant 1/0585/19 and Grant 1/0545/20, and the Slovak Research and Development Agency under the Project APVV-15-0007. Recommended by Associate Editor A. Giua. (Corresponding author: Juraj Oravec.)

Yuning Jiang and Boris Houska are with the School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China (e-mail: jiangyn@shanghaitech.edu.cn; boris.houska@gmx.de).

Juraj Oravec and Michal Kvasnica are with the Institute of Information Engineering, Automation, and Mathematics, Faculty of Chemical, and Food Technology, Slovak University of Technology in Bratislava, 81237 Bratislava, Slovakia (e-mail: juraj.oravec@stuba.sk; michal.kvasnica@stuba.sk).

Color versions of one or more of the figures in this article are available online at <https://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TAC.2020.3020827

Many researchers have attempted to address this question. For example, the development of Explicit MPC [9] aims at reducing both the online run-time and the memory footprint of MPC by optimizing precomputed solution maps of multiparametric optimization problems. However, Explicit MPC has the disadvantage that the number of polytopic regions over which the piecewise affine solution map of a parametric quadratic program is defined, grows, in the worst case, exponentially with the number of constraints. Some authors [10] have suggested addressing this issue by simplifying the MPC problem formulation by using move-blocking [11], but the associated control reactions can be suboptimal by a large margin. Other authors [12] have worked on reducing the memory footprint of Explicit MPC—certainly making considerable progress yet failing to meet the requirement of many practical systems with more than just a few states. In fact, despite all these developments in Explicit MPC, these methods are often applicable to problems of modest size only. As soon as one attempts to scale up to larger systems, Explicit MPC is often outperformed by iterative online solvers such as active set [2] or interior-point methods [5].

A recent trend in optimization-based control is to solve large MPC problems by breaking them into smaller ones. This trend has been initiated by research on distributed optimization [13]. For example, dual decomposition [14], alternating direction method of multipliers (ADMM) [13], and alternating direction inexact newton (ALADIN) [15] have been applied to MPC in various contexts and by many authors [16]–[20]. Additionally, applications of accelerated variants of ADMM to MPC can be found in [21], [22]. However, modern distributed optimization methods, such as ADMM or ALADIN, typically converge to an optimal solution in the limit, if the number of iterations tends to infinity. Thus, if real-time constraints are present, one could at most implement a finite number of such ADMM or ALADIN iterations returning a control input that may be infeasible or suboptimal by a large margin.

Therefore, this article asks the question whether it is possible to approximate MPC feedback laws by evaluating a constant, finite number of precomputed, explicit solution maps that are associated to MPC problems of a smaller scale. Here, a key requirement is that uniform asymptotic stability and performance guarantees of the implemented closed-loop controller have to be verifiable offline. The contribution of this article is the development of a controller, which meets this requirement under the restricting assumption that the original MPC problem is a strongly convex QP, as introduced in Section II. The control scheme itself is presented in the form of Algorithm 1 in Section III. This algorithm alternates between solving explicit solution maps that are associated with small-scale decoupled QPs and solving a linear equation system of a larger scale. However, in contrast to ALADIN, ADMM or other existing distributed optimization algorithms, Algorithm 1 performs a constant number of iterations per sampling time.

The stability and performance properties of Algorithm 1, which represent the main added value compared to our preliminary work [23], are summarized in Sections III-C, III-D, and III-E, respectively. Instead of relying on existing analysis concepts from the field of distributed optimization, the mathematical developments in this article rely on

results that find their origin in Explicit MPC theory [24]. In particular, the technical developments around Theorem 1 make use of the solution properties of multiparametric QPs in order to derive convergence rate estimates for Algorithm 1. Moreover, Theorem 2 establishes an asymptotic stability guarantee of the presented real-time closed-loop scheme. This result is complemented by Corollary 1, which provides bounds on the suboptimality of the presented control scheme. Finally, Section IV-A discusses implementation details with a particular emphasis on computational and storage complexity exploiting the fact that the presented scheme can be realized by using static memory only while ensuring a constant run-time, as illustrated by numerical case studies.

II. LINEAR-QUADRATIC MPC

This article concerns discrete-time MPC problems

$$J(x_0) = \min_{x,u} \mathcal{M}(x_N) + \sum_{k=0}^{N-1} \ell(x_k, u_k) \quad (1)$$

$$\text{s.t.} \quad \begin{cases} \forall k \in \{0, \dots, N-1\} \\ x_{k+1} = Ax_k + Bu_k \\ u_k \in \mathbb{U} \end{cases}$$

with strictly convex quadratic stage and terminal cost

$$\ell(x, u) = x^\top Qx + u^\top Ru \quad \text{and} \quad \mathcal{M}(x) = x^\top Px.$$

Here, $x_k \in \mathbb{R}^{n_x}$ denotes the state at time k and $u_k \in \mathbb{R}^{n_u}$ the associated control input assuming that the current time of the MPC controller is set to 0. The matrices $A, P, Q \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$, $R \in \mathbb{R}^{n_u \times n_u}$ are given and constant. Notice that (1) is a parametric optimization problem with respect to the current state measurement x_0 . The optimization variable $x = [x_1^\top, x_2^\top, \dots, x_N^\top]^\top$ includes all but the first element of the state sequence and the control sequence $u = [u_0^\top, u_1^\top, \dots, u_{N-1}^\top]^\top$ is defined accordingly.

Assumption 1: We assume that

- the control constraint set $\mathbb{U} \subseteq \mathbb{R}^{n_u}$ is a closed and convex polyhedron satisfying $0 \in \mathbb{U}$;
- the matrices Q , R , and P are all symmetric and positive definite.

Assumption 1 a) and b) implies strong convexity such that the primal solution of (1) is unique whenever it exists.

A. Asymptotic Stability

Notice that the stability properties of MPC controllers have been analyzed exhaustively [25]. In this context, a standard assumption can be formulated as follows.

Assumption 2: The terminal cost \mathcal{M} in (1) admits a control law $\mu : \mathbb{R}^{n_x} \rightarrow \mathbb{U}$ such that for all $x \in \mathbb{R}^{n_x}$

$$\ell(x, \mu(x)) + \mathcal{M}(Ax + B\mu(x)) \leq \mathcal{M}(x).$$

The MPC controller (1) is asymptotically stable if Assumptions 1 and 2 hold [25].

III. SUBOPTIMAL REAL-TIME MPC

In this section, we propose and analyze a real-time algorithm for finding approximate solutions of (1).

A. Preliminaries

Let us introduce the vectors $y_0 = u_0$, $y_k = [x_k^\top \ u_k^\top]^\top$, $y_N = x_N$, and their associated constraint sets

$$\mathbb{Y}_0 = \mathbb{U} \quad \text{and} \quad \mathbb{Y}_k = \{y \in \mathbb{R}^{n_u+n_x} \mid [0 \ I]y \in \mathbb{U}\} \quad (2)$$

for all $k \in \{1, \dots, N-1\}$. Moreover, we introduce

$$F_k(y_k) = \ell(x_k, u_k), \quad F_N(y_N) = \mathcal{M}(x_N) \quad (3)$$

for $k \in \{1, \dots, N-1\}$ and matrices

$$H_0 = B, \quad H_k = \begin{bmatrix} A & B \end{bmatrix}, \quad G_k = \begin{bmatrix} I & 0 \end{bmatrix}, \quad G_N = I$$

as well as $h_0 = Ax_0$, $h_k = 0$ for all $k \in \{1, \dots, N-1\}$. Now, (1) can be written in the form

$$J(x_0) = \min_y \sum_{k=0}^N F_k(y_k) \quad (4)$$

$$\text{s.t.} \quad \begin{cases} \forall k \in \{0, \dots, N-1\}, \\ G_{k+1}y_{k+1} = H_k y_k + h_k \mid \lambda_k \\ y_k \in \mathbb{Y}_k. \end{cases}$$

The notation “ λ_k ” behind the affine constraints in the abovementioned optimization problems indicates that λ_k denotes their associated multipliers. It is helpful to keep in mind that both the function F_0 and the vector h_0 depend on x_0 . In addition, we introduce a shorthand for the objective in (4) and its convex conjugate function

$$F(y) = \sum_{k=0}^N F_k(y_k), \quad F^*(\lambda) = \max_y \{-F(y) + \langle \lambda, y \rangle\}$$

where the shorthand notation

$$\langle \lambda, y \rangle = -(H_0^\top \lambda_0)^\top y_0 + \sum_{k=1}^N (G_k^\top \lambda_{k-1} - H_k^\top \lambda_k)^\top y_k + \lambda_{N-1}^\top G_N^\top y_N$$

is used to denote a weighted (nonsymmetric) scalar product of primal and dual variables. Notice that the functions F and F^* are strongly convex quadratic forms with $F(0) = 0$ and $F^*(0) = 0$ as long as Assumption 1 is satisfied. The optimal primal and dual solutions of (4) are denoted by x^* and λ^* , respectively. It is well-known that x^* and λ^* are continuous and piecewise affine functions of x_0 (see [26]).

B. Algorithm

The main idea for solving (4) approximately and in real time is to consider the auxiliary optimization problem

$$J(x_0) = \min_y \sum_{k=0}^N F_k(y_k - y_k^{\text{ref}}) \quad (5)$$

$$\text{s.t.} \quad \begin{cases} \forall k \in \{0, \dots, N-1\} \\ G_{k+1}y_{k+1} = H_k y_k + h_k \mid \lambda_k \end{cases}$$

with reference trajectory y^{ref} . If $y^{\text{ref}} = y^*$ is equal to the minimizer of (4), then y^* is a minimizer of (5). Notice that the main motivation for introducing the coupled QP (5) is that this problem approximates (4) without needing inequality constraints. Thus, this problem can be solved by using a sparse linear algebra solver.

Let us assume that y^m and λ^m are the current approximations of the primal and dual solution of (4). Algorithm 1 constructs the next iterate

Algorithm 1: Parallel Real-Time MPC.**Initialization:**

- Choose $y^1 = [y_0^{1\top}, \dots, y_N^{1\top}]^\top$, $\lambda^1 = [\lambda_0^{1\top}, \dots, \lambda_{N-1}^{1\top}]^\top$, a constant $\gamma > 0$, and a maximum number \bar{m} of iterations per sampling time.

Online:

- 1) Wait for the state measurement x_0 and compute the constant

$$f^1 = F(y^1) + F^*(\lambda^1).$$

If $f^1 \geq \gamma^2 x_0^\top Q x_0$, rescale

$$y^1 \leftarrow y^1 \sqrt{\frac{\gamma^2 \|x_0\|_Q^2}{f^1}} \quad \text{and} \quad \lambda^1 \leftarrow \lambda^1 \sqrt{\frac{\gamma^2 \|x_0\|_Q^2}{f^1}}$$

where $\|x_0\|_Q^2 \triangleq x_0^\top Q x_0$.

- 2) For $m = 1 \rightarrow \bar{m}$

- a) solve the small-scale decoupled QPs in parallel

$$\begin{aligned} \min_{\xi_0^m \in \mathbb{Y}_0} & F_0(\xi_0^m) - (H_0^\top \lambda_0^m)^\top \xi_0^m + F_0(\xi_0^m - y_0^m) \\ \min_{\xi_k^m \in \mathbb{Y}_k} & F_k(\xi_k^m) + (G_k^\top \lambda_{k-1}^m - H_k^\top \lambda_k^m)^\top \xi_k^m + F_k(\xi_k^m - y_k^m) \\ \min_{\xi_N^m} & F_N(\xi_N^m) + (G_N^\top \lambda_{N-1}^m)^\top \xi_N^m + F_N(\xi_N^m - y_N^m) \end{aligned}$$

for all $k \in \{1, \dots, N-1\}$ and denote solutions by $\xi^m = [\xi_0^m, \xi_1^m, \dots, \xi_N^m]$.

- b) Solve the coupled QP

$$\begin{aligned} \min_{y^{m+1}} & \sum_{k=0}^N F_k(y_k^{m+1} - 2\xi_k^m + y_k^m) \\ \text{s.t.} & \begin{cases} \forall k \in \{0, \dots, N-1\}, \\ G_{k+1} y_{k+1}^{m+1} = H_k y_k^{m+1} + h_k \mid \delta_k^m \end{cases} \end{aligned} \quad (7)$$

and set $\lambda^{m+1} = \lambda^m + \delta^m$.

End

- 3) Send $u_0 = \xi_0^{\bar{m}}$ to the real process.

- 4) Set $y^1 = [y_0^{\bar{m}\top}, \dots, y_N^{\bar{m}\top}, 0]^\top$, $\lambda^1 = [\lambda_1^{\bar{m}\top}, \dots, \lambda_{N-1}^{\bar{m}\top}, 0]^\top$, go to Step 1.

y^{m+1} and λ^{m+1} by performing two main operations. First, we solve augmented Lagrangian optimization problems of the form

$$\min_{\xi^m \in \mathbb{Y}} F(\xi^m) + \langle \lambda, y \rangle + F(\xi^m - y^m). \quad (6)$$

with $\mathbb{Y} = \mathbb{Y}_0 \times \dots \times \mathbb{Y}_{N-1} \times \mathbb{R}^{n_x}$. Problem (6) can be solved in parallel, see Step 2.a) of Algorithm 1. In the following, we set $\mathcal{Q} = \frac{1}{2} \nabla^2 F(0)$ such that $\|\xi^m - y^m\|_{\mathcal{Q}}^2 = F(\xi^m - y^m)$ recalling that F is a centered positive-definite quadratic form. And second, we solve QP (5) for the reference point $y^{\text{ref}} = 2\xi^m - y^m$ without considering the input constraints. These two main steps correspond to Step 2a) and b) in Algorithm 1.

Algorithm 1 is initialized with guesses

$$y^1 = [y_0^{1\top}, \dots, y_N^{1\top}]^\top \quad \text{and} \quad \lambda^1 = [\lambda_0^{1\top}, \dots, \lambda_{N-1}^{1\top}]^\top$$

for the primal and dual solution of (4) offline. Notice that Algorithm 1 receives a state measurement x_0 in every iteration (Step 1) and returns a control input to the real process (Step 3). Similar to the classical real-time MPC scheme [6], or related warm-start techniques [27], Step

4) shifts primal and dual variables y^1 and λ^1 , which are, however, rescaled in Step 1), based on a tuning parameter $\gamma > 0$.

Assumption 3: The constant γ in Algorithm 1 is such that

$$F(y^*) + F^*(\lambda^*) \leq \gamma^2 x_0^\top Q x_0.$$

Notice that such a bound γ exists and can be computed offline, because y^* and λ^* are Lipschitz continuous and piecewise affine functions of x_0 [26]. Notice that the choice $\gamma = \infty$ would mean that the variables are never rescaled. In this case, Algorithm 1 is unstable in general. In order to see this, consider the scenario that a user initializes the algorithm with an arbitrary $(y^1, \lambda^1) \neq 0$. Now, if the first measurement happens to be at $x_0 = 0$, the optimal control input is at $u^* = 0$. But, if we run Algorithm 1 with $\bar{m} < \infty$, it returns an approximation $u_0 \approx u^* = 0$, which will introduce an excitation as we have $u_0 \neq 0$ in general. Thus, if we would not rescale the initialization in Step 1), it would be impossible to establish stability.

C. Convergence Properties of Algorithm 1

This section provides a concise overview of the theoretical convergence properties of Algorithm 1. Here, we initially focus on establishing conditions for convergence of the iterates of this algorithm (Lemma 1), which are then, in a second step, used to establish a linear convergence rate estimate (Theorem 1).

Lemma 1: Let Assumption 1 be satisfied and let (4) be feasible, such that a unique minimizer y^* and an associated dual solution λ^* exist. Then, the iterates of Algorithm 1 satisfy

$$\sum_{m=\hat{m}}^{\bar{m}} F(\xi^m - y^*) \leq \frac{F(y^{\hat{m}} - y^*) + F^*(\lambda^{\hat{m}} - \lambda^*)}{4}$$

for all $\bar{m} \geq \hat{m}$ and all $\hat{m} \geq 2$.

Notice that the statement of Lemma 1 is useful in the sense that an immediate consequence of this statement is that the iterates of Algorithm 1 would converge to the exact solution of (4), if we would set $\bar{m} = \infty$, i.e.,

$$\lim_{m \rightarrow \infty} \xi^m = y^* \quad \text{and} \quad \lim_{m \rightarrow \infty} \lambda^m = \lambda^*.$$

The proof of the abovementioned lemma is technical but important for the developments in this article

Proof: Let us introduce the auxiliary functions

$$\mathcal{F}_0(\phi_0) = F_0(\phi_0) - (H_0^\top \lambda_0^m)^\top \phi_0^m + \nabla F_0(\xi_0^m - y_0^m)^\top \phi_0$$

$$\begin{aligned} \mathcal{F}_k(\phi_k) &= F_k(\phi_k) + (G_k^\top \lambda_{k-1}^m - H_k^\top \lambda_k^m)^\top \phi_k^m \\ &\quad + \nabla F_k(\xi_k^m - y_k^m)^\top \phi_k \end{aligned}$$

$$\mathcal{F}_N(\phi_N) = F_N(\phi_N) + G_N^\top \lambda_{N-1}^m \phi_N^m + \nabla F_N(\xi_N^m - y_N^m)^\top \phi_N.$$

Because ξ_k^m is a minimizer of the k th decoupled QP in Step 2a) of Algorithm 1, it must also be a minimizer of \mathcal{F}_k on \mathbb{Y}_k . Thus, because \mathcal{F}_k is strongly convex with Hessian $\nabla^2 \mathcal{F}_k$, we must have

$$\sum_{k=0}^N \mathcal{F}_k(\xi_k^m) + \sum_{k=0}^N F_k(\xi_k^m - y_k^*) \leq \sum_{k=0}^N \mathcal{F}_k(y_k^*).$$

On the other hand, due to duality, we have

$$\begin{aligned} & \sum_{k=0}^N F_k(y_k^*) + \langle \lambda^*, y^* \rangle + \sum_{k=0}^N F_k(\xi_k^m - y_k^*) \\ & \leq \sum_{k=0}^N F_k(\xi_k^m) + \langle \lambda^*, \xi^m \rangle. \end{aligned}$$

Adding both inequalities and collecting terms yields

$$\begin{aligned}
0 &\geq \sum_{k=0}^N \nabla F_k(\xi_k^m - y_k^m)^\top (\xi_k^m - y_k^*) + 2 \sum_{k=0}^N F_k(\xi_k^m - y_k^*) \\
&\quad + \langle \lambda^m - \lambda^*, \xi^m - y^* \rangle \\
&= (\xi^m - y^m)^\top \mathcal{Q}(\xi^m - y^*) + 2 \sum_{k=0}^N F_k(\xi_k^m - y_k^*) \\
&\quad + (\lambda^m - \lambda^*)^\top \mathcal{A}(\xi^m - y^*) \tag{8}
\end{aligned}$$

with $\mathcal{A} = \nabla_{\lambda, x} \langle \lambda, x \rangle$. Similarly, the stationarity condition QP (7) can be written as follows:

$$\mathcal{Q}(y^{m+1} - 2\xi^m + y^m) + \mathcal{A}^\top \delta^m = 0.$$

Because \mathcal{Q} is positive definite, we solve this equation with respect to ξ^m finding

$$\xi^m = \frac{1}{2} \mathcal{Q}^{-1} \mathcal{A}^\top (\lambda^{m+1} - \lambda^m) + \frac{y^m + y^{m+1}}{2}. \tag{9}$$

Here, we have additionally substituted the relation

$$\delta^m = \lambda^{m+1} - \lambda^m.$$

Notice that we have $\mathcal{A}y^m = \mathcal{A}y^{m+1} = \mathcal{A}y^*$ for all $m \geq 2$, because the solutions of the QP (7) must satisfy the equality constraints in (4). If we substitute this equation and the expression for ξ^m in (8), we find that

$$\begin{aligned}
&-2F(\xi^m - y^*) \\
&\geq (\xi^m - y^m)^\top \mathcal{Q}(\xi^m - y^*) + (\lambda^m - \lambda^*)^\top \mathcal{A}(\xi^m - y^*) \\
&= \frac{1}{4} (\lambda^{m+1} - \lambda^m)^\top \mathcal{A} \mathcal{Q}^{-1} \mathcal{A}^\top (\lambda^{m+1} - \lambda^m) \\
&\quad + \frac{1}{4} (y^{m+1} - y^m) \mathcal{Q} (y^m - 2y^* + y^{m+1}) \\
&\quad + \frac{1}{2} (\lambda^m - \lambda^*)^\top \mathcal{A} \mathcal{Q}^{-1} \mathcal{A}^\top (\lambda^{m+1} - \lambda^m) \\
&= \frac{1}{2} (F(y^{m+1} - y^*) - F(y^m - y^*)) \\
&\quad + \frac{1}{2} (F^*(\lambda^{m+1} - \lambda^*) - F^*(\lambda^m - \lambda^*)) \tag{10}
\end{aligned}$$

for all $m \geq 2$. Now, the statement of Lemma 1 follows by summing up the abovementioned inequalities for $m = \hat{m}$ to $m = \bar{m}$ and using that the last element in the telescoping sum on the right hand

$$\frac{F(y^{\bar{m}+1} - y^*) + F^*(\lambda^{\bar{m}+1} - \lambda^*)}{2} \geq 0$$

is nonnegative. \blacksquare

The following theorem uses the abovementioned result in order to derive a convergence rate estimate of Algorithm 1.

Theorem 1: Let Assumption 1 be satisfied and let (4) be feasible, such that a unique minimizer y^* and an associated dual solution λ^* exist. Then, there exists a positive constant $\kappa < 1$ such that

$$\begin{aligned}
&F(y^{m+1} - y^*) + F^*(\lambda^{m+1} - \lambda^*) \\
&\leq \kappa (F(y^m - y^*) + F^*(\lambda^m - \lambda^*)) \tag{11}
\end{aligned}$$

for all $m \geq 2$.

Proof: Let $\hat{\mathbb{Y}}_k$ denote the intersection of all active supporting hyperplanes at the solutions of the small scale QPs of Step 2a) in Algorithm 1

for $k \in \{0, \dots, N-1\}$. We construct the auxiliary optimization problem

$$\begin{aligned}
&\min_{\hat{y}} \sum_{k=0}^N F_k(\hat{y}_k) \\
&\text{s.t.} \begin{cases} \forall k \in \{0, \dots, N-1\} \\ G_{k+1} \hat{y}_{k+1} = H_k \hat{y}_k + h_k & | \hat{\lambda}_k \\ 0 = H_N \hat{y}_N & | \hat{\lambda}_N \\ \hat{y}_k \in \hat{\mathbb{Y}}_k \end{cases} \tag{12}
\end{aligned}$$

and denote optimal primal and dual solutions of this problem by \hat{y}^* and $\hat{\lambda}^*$. Next, we also construct the auxiliary QPs

$$\min_{\xi_0^m \in \hat{\mathbb{Y}}_0} F_0(\xi_0^m) - (H_0^\top \lambda_0^m)^\top \xi_0^m + F_0(\xi_0^m - y_0^m)$$

$$\min_{\xi_k^m \in \hat{\mathbb{Y}}_k} F_k(\xi_k^m) + (G_k^\top \lambda_{k-1}^m - H_k^\top \lambda_k^m)^\top \xi_k^m + F_k(\xi_k^m - y_k^m)$$

$$\min_{\xi_N^m} F_N(\xi_N^m) + (G_N^\top \lambda_{N-1}^m)^\top \xi_N^m + F_N(\xi_N^m - y_N^m).$$

Because these QPs have equality constraints only, their parametric solutions must be affine. Thus, there exists a matrix T_1 such that

$$\xi^m - \hat{y}^* = T_1 \begin{pmatrix} y^m - \hat{y}^* \\ \lambda^m - \hat{\lambda}^* \end{pmatrix}.$$

Similarly, the coupled QP (7) has equality constraints only; that is, there exists a matrix T_2 such that

$$\begin{pmatrix} y^{m+1} - \hat{y}^* \\ \delta^m \end{pmatrix} = T_2 \begin{pmatrix} \xi^m - \hat{y}^* \\ y^m - \hat{y}^* \end{pmatrix}.$$

Now, we use the equation $\lambda^{m+1} - \lambda^* = \lambda^m - \lambda^* + \delta$ and substitute the above equations finding that

$$\begin{pmatrix} y^{m+1} - \hat{y}^* \\ \lambda^{m+1} - \hat{\lambda}^* \end{pmatrix} = T \begin{pmatrix} y^m - \hat{y}^* \\ \lambda^m - \hat{\lambda}^* \end{pmatrix} \tag{13}$$

with

$$T = \begin{pmatrix} T_1 \\ T_2 \end{pmatrix} \begin{pmatrix} T_1 \\ (I \ 0) \end{pmatrix} + \begin{pmatrix} 0 \\ I \end{pmatrix}.$$

Next, we know from Lemma 1 that if we would apply Algorithm 1 to the auxiliary problem (12), the corresponding primal and dual iterates would converge to \hat{y}^* and $\hat{\lambda}^*$. In particular, inequality (10) yields

$$\begin{aligned}
&(y^{m+1} - \hat{y}^*)^\top \mathcal{Q} (y^{m+1} - \hat{y}^*) \\
&\quad + (\lambda^{m+1} - \hat{\lambda}^*)^\top \mathcal{Q}^{-1} \mathcal{A}^\top (\lambda^{m+1} - \hat{\lambda}^*) \\
&< (y^m - \hat{y}^*)^\top \mathcal{Q} (y^m - \hat{y}^*) \\
&\quad + (\lambda^m - \hat{\lambda}^*)^\top \mathcal{A} \mathcal{Q}^{-1} \mathcal{A}^\top (\lambda^m - \hat{\lambda}^*) \tag{14}
\end{aligned}$$

whenever $\begin{pmatrix} y^m - \hat{y}^* \\ \lambda^m - \hat{\lambda}^* \end{pmatrix} \neq 0$. By substituting the linear (13), we find that this is only possible if

$$T^\top \begin{pmatrix} \mathcal{Q} & 0 \\ 0 & \mathcal{A} \mathcal{Q}^{-1} \mathcal{A}^\top \end{pmatrix} T \preceq \kappa_{\mathbb{A}} I \tag{15}$$

for a constant $\kappa_{\mathbb{A}} < 1$. Now, one remaining difficulty is that the constant $\kappa_{\mathbb{A}}$ (as well as the matrix T) depends on the particular set \mathbb{A} of active supporting hyperplanes in the small-scale QPs. Nevertheless, because

there exists only a finite number of possible active sets, the maximum $\kappa = \max_{\mathcal{A}} \kappa_{\mathcal{A}}$ must exist and satisfy $\kappa < 1$. Now, the equation

$$\begin{pmatrix} y^{m+1} - y^* \\ \lambda^{m+1} - \lambda^* \end{pmatrix} = T \begin{pmatrix} y^m - y^* \\ \lambda^m - \lambda^* \end{pmatrix} \quad (16)$$

holds only for our fixed m and the associated matrix T for a particular active set, but the associated decent condition

$$\begin{aligned} & (y^{m+1} - y^*)^\top \mathcal{Q} (y^{m+1} - y^*) \\ & + (\lambda^{m+1} - \lambda^*)^\top \mathcal{A} \mathcal{Q}^{-1} \mathcal{A}^\top (\lambda^{m+1} - \lambda^*) \\ & \leq \kappa [(y^m - y^*)^\top \mathcal{Q} (y^m - y^*) \\ & + (\lambda^m - \lambda^*)^\top \mathcal{A} \mathcal{Q}^{-1} \mathcal{A}^\top (\lambda^m - \lambda^*)] \end{aligned} \quad (17)$$

holds independently of the active set of the QPs in the m th iteration and is indeed valid for all m . A resubstitution of F and F^* yields the statement of the theorem. ■

D. Asymptotic Stability of Algorithm 1

The goal of this section is to establish asymptotic stability of Algorithm 1. Because we send the control input $u_0 = \xi_0^{\bar{m}}$ to the real process, the next measurement will be at $x_0^+ = Ax_0 + B\xi_0^{\bar{m}}$. Notice that, in general, we may have $x_0^+ \neq x_1^* = Ax_0 + By_0^*$, since we run Algorithm 1 with a finite $\bar{m} < \infty$.

Theorem 2: Let Assumptions 1, 2, and 3 be satisfied. Let the constant $\sigma > 0$ be such that the semidefinite inequality $B^\top \mathcal{Q} B \preceq \sigma R$ holds and let the constants $\eta, \tau > 0$ be such that

$$|J(x_0^+) - J(x_1^*)| \leq \eta \|x_0^+ - x_1^*\|_Q + \frac{\tau}{2} \|x_0^+ - x_1^*\|_Q^2 \quad (18)$$

If the constant $\bar{m} \in \mathbb{N}$ satisfies

$$\bar{m} > \frac{2 \log \left(2\eta\gamma \sqrt{\frac{\sigma(1+\kappa)}{\kappa}} + 2\tau\sigma\gamma^2 \frac{1+\kappa}{\kappa} \right)}{\log(1/\kappa)} \quad (19)$$

then the controller in Algorithm 1 is asymptotically stable.

Proof: Because we have $x_0^+ - x_1^* = B(\xi_0^{\bar{m}} - y_0^*) = \mathcal{P}(\xi^{\bar{m}} - y^*)$ with $\mathcal{P} = [B, 0, \dots, 0]$, we can substitute (9) to find

$$\begin{aligned} x_0^+ - x_1^* &= \mathcal{P} \left[\frac{\mathcal{Q}^{-1} \mathcal{A}^\top (\lambda^{\bar{m}+1} - \lambda^{\bar{m}})}{2} + \frac{y^{\bar{m}+1} + y^{\bar{m}}}{2} - y^* \right] \\ &= \frac{1}{2} \mathcal{P} \mathcal{Q}^{-1} \mathcal{A}^\top (\lambda^{\bar{m}+1} - \lambda^*) + \frac{1}{2} \mathcal{P} \mathcal{Q}^{-1} \mathcal{A}^\top (\lambda^* - \lambda^{\bar{m}}) \\ &\quad + \frac{1}{2} \mathcal{P} (y^{\bar{m}+1} - y^*) + \frac{1}{2} \mathcal{P} (y^{\bar{m}} - y^*). \end{aligned}$$

The particular definition of σ implies $\mathcal{P}^\top \mathcal{Q} \mathcal{P} \preceq \sigma \mathcal{Q}$ and

$$\begin{aligned} & 4(x_0^+ - x_1^*)^\top \mathcal{Q} (x_0^+ - x_1^*) \\ & \leq 4(\lambda^{\bar{m}+1} - \lambda^*)^\top \mathcal{A} \mathcal{Q}^{-1} \mathcal{P}^\top \mathcal{Q} \mathcal{P} \mathcal{Q}^{-1} \mathcal{A}^\top (\lambda^{\bar{m}+1} - \lambda^*) \\ & \quad + 4(\lambda^{\bar{m}} - \lambda^*)^\top \mathcal{A} \mathcal{Q}^{-1} \mathcal{P}^\top \mathcal{Q} \mathcal{P} \mathcal{Q}^{-1} \mathcal{A}^\top (\lambda^{\bar{m}} - \lambda^*) \\ & \quad + 4(y^{\bar{m}+1} - y^*)^\top \mathcal{P}^\top \mathcal{Q} \mathcal{P} (y^{\bar{m}+1} - y^*) \\ & \quad + 4(y^{\bar{m}} - y^*)^\top \mathcal{P}^\top \mathcal{Q} \mathcal{P} (y^{\bar{m}} - y^*) \\ & \leq 4\sigma (F(y^{\bar{m}+1} - y^*) + F^*(\lambda^{\bar{m}+1} - \lambda^*)) \\ & \quad + 4\sigma (F(y^{\bar{m}} - y^*) + F^*(\lambda^{\bar{m}} - \lambda^*)) \end{aligned}$$

$$\begin{aligned} & \leq 4\sigma(1+\kappa)\kappa^{\bar{m}-1} (F(y^1 - y^*) + F^*(\lambda^1 - \lambda^*)) \\ & \leq 16\sigma(1+\kappa)\kappa^{\bar{m}-1}\gamma^2 F_0(y_0^*). \end{aligned}$$

The last inequality holds based on the inequalities

$$\begin{aligned} F(y^*) + F^*(\lambda^*) &\leq \gamma^2 x_0^\top \mathcal{Q} x_0 \leq \gamma^2 F_0(y_0^*), \\ F(y^1) + F^*(\lambda^1) &\leq \gamma^2 x_0^\top \mathcal{Q} x_0 \leq \gamma^2 F_0(y_0^*) \end{aligned}$$

which hold due to Assumption 3 and the particular construction in Step 1 of Algorithm 1. Now, a division by four yields

$$\|x_0^+ - x_1^*\|_Q^2 \leq 4\sigma\gamma^2 \left(\frac{1+\kappa}{\kappa} \right) \kappa^{\bar{m}} F_0(y_0^*). \quad (20)$$

By combining this inequality with (18) we find

$$\begin{aligned} & |J(x_0^+) - J(x_1^*)| \\ & \leq 2 \left[\eta\gamma \sqrt{\frac{\sigma(1+\kappa)}{\kappa}} + \tau\sigma\gamma^2 \frac{1+\kappa}{\kappa} \right] \kappa^{\frac{\bar{m}}{2}} F_0(y_0^*). \end{aligned} \quad (21)$$

Thus, if we set

$$\alpha = 1 - 2 \left[\eta\gamma \sqrt{\frac{\sigma(1+\kappa)}{\kappa}} + \tau\sigma\gamma^2 \frac{1+\kappa}{\kappa} \right] \kappa^{\frac{\bar{m}}{2}} > 0$$

we have

$$\begin{aligned} J(x_0^+) &\leq J(x_0) - (F_0(y_0^*) - J(x_0^+) + J(x_1^*)) \\ &\leq J(x_0) - \alpha F_0(y_0^*) \end{aligned} \quad (22)$$

which is sufficient to establish asymptotic stability [28]. ■

E. Performance of Algorithm 1

The result of Theorem 2 can be extended in order to derive an upper bound on the suboptimality of Algorithm 1.

Corollary 1: Let the assumption of Theorem 2 holds with

$$\alpha = 1 - 2 \left[\eta\gamma \sqrt{\frac{\sigma(1+\kappa)}{\kappa}} + \tau\sigma\gamma^2 \frac{1+\kappa}{\kappa} \right] \kappa^{\frac{\bar{m}}{2}}.$$

If $y_i^{\text{cl}} = (x_i^{\text{cl}}, u_i^{\text{cl}})$ denotes the sequence of closed-loop states and controls that are generated by the controller in Algorithm 1, an a-priori bound on the associated infinite-horizon closed-loop performance is given by

$$\sum_{i=0}^{\infty} \ell(x_i^{\text{cl}}, u_i^{\text{cl}}) \leq \frac{J(x_0)}{\alpha}.$$

Proof: Because (22) holds, we have

$$J(x_{i+1}^{\text{cl}}) \leq J(x_i^{\text{cl}}) - \alpha F_0(y_i^{\text{cl}})$$

which yields the inequality

$$\sum_{i=0}^{\infty} F_0(y_i^{\text{cl}}) \leq \frac{1}{\alpha} \sum_{i=0}^{\infty} (J(x_i^{\text{cl}}) - J(x_{i+1}^{\text{cl}})).$$

The statement of the corollary follows after simplifying the telescoping sum on the right and substituting the equation $F_0(y_i^{\text{cl}}) = \ell(x_i^{\text{cl}}, u_i^{\text{cl}})$. ■

Remark 1 (MPC with state constraints): Notice that (1) admits control constraints only. A complete discussion of how to extend the presented algorithm and analysis for MPC problems with state constraint would go beyond the scope of this article. However, one method for taking such state constraints into account can be obtained by adding

TABLE I
COMPLEXITY OF STEPS 2A) AND B) OF ALGORITHM 1

Step	Offline CPU time	Online CPU time	Memory Requirement
2a')	$\mathcal{O}(N_R^2)$	$\mathcal{O}(N \log_2(N_R))$	$\mathcal{O}(N_R)$
2b)	$\mathcal{O}(N n_x^3)$	$\mathcal{O}(N n_x^2)$	$\mathcal{O}(N n_x^2)$

L_1 -penalty functions to the stage cost ℓ . Our stability and convergence proofs can be extended for this case because adding L_1 -penalties does not change the fact that the cost-to-go function J is piecewise quadratic.

IV. IMPLEMENTATION DETAILS AND CASE STUDIES

This section applies Algorithm 1 to benchmark case studies.

A. Implementation on Hardware With Limited Memory

Algorithm 1 has two main steps, Step 2a) and b). In Step 2a) decoupled QPs have to be solved online. We solve these QPs offline using multiparametric programming by precomputing the solution maps

$$\begin{aligned}\xi_0^*(\theta_0, x_0) &= \arg \min_{\xi_0 \in \mathbb{Y}_0} 2F_0(\xi_0) + \theta_0^\top \xi_0 \\ \xi_1^*(\theta_1) &= \arg \min_{\xi_1 \in \mathbb{Y}_1} 2F_1(\xi_1) + \theta_1^\top \xi_1 \\ \xi_N^*(\theta_N) &= \arg \min_{\xi_N} 2F_N(\xi_N) + \theta_N^\top \xi_N\end{aligned}\quad (23)$$

with parameters $\theta_0 \in \mathbb{R}^{n_u+n_z}$, $\theta_1 \in \mathbb{R}^{n_x+n_u}$, and $\theta_N \in \mathbb{R}^{n_x}$. Here, ξ_0^* depends on x_0 recalling that this dependency had been hidden in our definition of F_0 and \mathbb{Y}_0 . We use multi-parametric toolbox (MPT) [29] to precompute and store the maps ξ_0^* , ξ_1^* and ξ_N^* . Consequently, Step 2a) in Algorithm 1 can be replaced by

- **Step 2a')** Compute the parameters

$$\theta_0^m = -H_0^\top \lambda_0^m - 2\Sigma_0 y_0^m \quad (24a)$$

$$\theta_k^m = G_k^\top \lambda_{k-1}^m - H_k^\top \lambda_k^m - 2\Sigma_k y_k^m \quad (24b)$$

$$\theta_N^m = G_N^\top \lambda_{N-1}^m - 2\Sigma_N y_N^m \quad (24c)$$

with $\Sigma_0 = R$, $\Sigma_k = \text{blkdiag}\{Q, R\}$, $k \in \{1, \dots, N-1\}$, $\Sigma_N = P$ and set

$$\xi_0^m = \xi_0^*(\theta_0^m, x_0), \quad \xi_k^m = \xi_1^*(\theta_k^m)$$

for all $k \in \{1, \dots, N\}$ by evaluating the respective explicit solution maps (23). In this article, we use the enumeration-based multiparametric QP algorithm from [30] for generating these maps.

Notice that the complexity of preprocessing the small-scale QPs (23) depends on the number $N_R = \max\{N_{R,0}, N_{R,1}\}$ of critical regions over which the piecewise affine (PWA) optimizers ξ_0^* , ξ_1^* , and ξ_N^* are defined [31], but N_R is independent of the prediction horizon N as summarized in the first row in Table I. For a derivation of the associated run-time and memory complexity results we refer to [23], [32], [33].

In Step 2b) coupled QP (7) must be solved. Because this QP has equality constraints only, (7) is equivalent to a large but sparse system of equations. Moreover, all matrices in (7) are given and constant during the online iterations. This means that all linear algebra decompositions

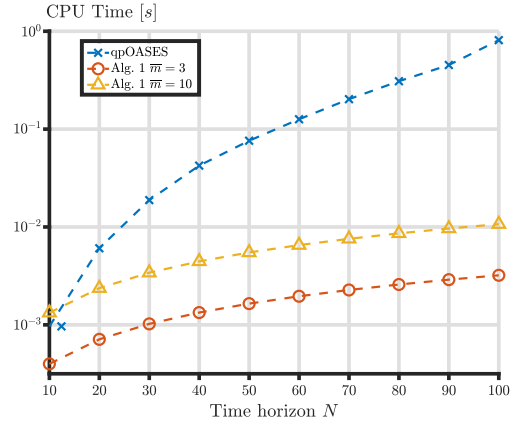


Fig. 1. CPU time comparison: Algorithm 1 versus traditional MPC (Condensing + qpOASES) both run in MATLAB R2018a interfacing C/C++ code.

can be precomputed offline. If one uses standard Riccati recursions for exploiting the band-structure of (7), the computational complexity for all offline computations is at most of order $\mathcal{O}(N n_x^3)$, while the online implementation has complexity $\mathcal{O}(N n_x^2)$ [34] as summarized in the second row in Table I.

B. Parallel MPC With Long Horizons

The first benchmark considers a linear dynamic system with

$$A = \begin{bmatrix} 0.9993 & -3.0083 & -0.1131 & -1.6081 \\ 0 & 0.9862 & 0.0478 & 0 \\ 0 & 2.0833 & 1.0089 & 0 \\ 0 & 0.0526 & 0.0498 & 1 \end{bmatrix}$$

$$\text{and } B = \begin{bmatrix} -0.0804 & -0.0291 & -0.8679 & -0.0216 \\ -0.6347 & -0.0143 & -0.0917 & -0.0022 \end{bmatrix}^\top$$

The states of this system can be interpreted as the yaw, pitch, roll, and the attack angles of an aircraft while the controls are given by the elevator and the flaperon angles [35]. The state constraint and control constraint are given by

$$\mathbb{X} = \left\{ x \in \mathbb{R}^4 \left[\begin{array}{c} -0.5 \\ -100 \end{array} \right] \leq \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x \leq \begin{bmatrix} 0.5 \\ 100 \end{bmatrix} \right\}$$

$$\mathbb{U} = [-25, 25] \times [-25, 25]$$

the stage cost weights are set to

$$Q = \text{diag}(0.1, 100, 0.1, 100), \quad R = \text{diag}(10, 10)$$

and the initial state is given by $x_0 = [20 \ 0 \ 20 \ 20]$. The matrix P is computed by solving an algebraic Riccati equation such that the terminal cost is locally equal to the unconstrained infinite horizon cost [25]. Moreover, the parameter $\gamma = 10$ is fixed in our implementations.

Fig. 1 shows a CPU time comparison of Algorithm 1 (with $\bar{m} = 3$ and $\bar{m} = 10$) and traditional MPC in dependence on the prediction horizon. The implementations of Algorithm 1 uses MATLAB R2018a with YALMIP [36] and MPT 3.1.5 [29] but the comparison is based on qpOASES [2]. Algorithm 1 is faster for large N , but this speed-up comes along with a loss of control performance (see Fig. 2). For $\bar{m} = 10$ the suboptimal closed-loop state trajectory is, however, almost indistinguishably close to the optimal trajectory.

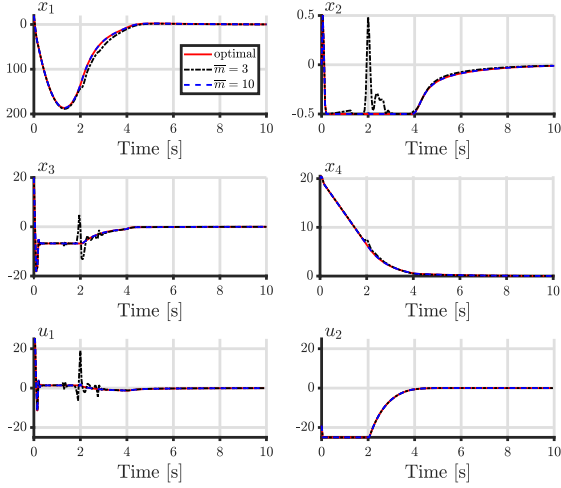


Fig. 2. Closed-loop state and control trajectories: $\bar{m} \in \{3, 10, \infty\}$, $N = 40$.

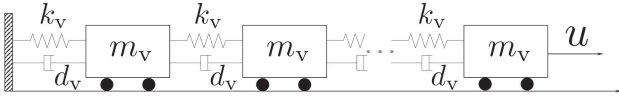


Fig. 3. Sketch of a spring-vehicle-damper system.

Our implementation of Algorithm 1 requires 81 kB memory corresponding to 92 regions (independent of N). These numbers can be compared with the following results for a standard Explicit MPC implementation using the geometric parametric linear complementarity problem (LCP) solver of MPT 3.1.5 [29]:

N	# of regions	memory [kB]	CPU time [μ s]
3	427	233	23
5	3 649	2 566	89
10	64 556	70 609	304

For $N > 10$ our implementation of Explicit MPC ran out of memory.

C. Spring-Vehicle-Damper System

Our second case study considers a spring-vehicle-damper system with \bar{I} vehicles with mass $m_v = 1$ kg, as visualized below.

The nonzero blocks of the system matrices are given by

$$A_{i,i} = \mathbb{I} + T_s \begin{pmatrix} 0 & 1 \\ -2 \frac{k_v}{m_v} & -2 \frac{d_v}{m_v} \end{pmatrix}, \quad B_i = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$A_{\bar{I},\bar{I}} = \mathbb{I} + T_s \begin{pmatrix} 0 & 1 \\ -\frac{k_v}{m_v} & -\frac{d_v}{m_v} \end{pmatrix}, \quad B_{\bar{I}} = \begin{pmatrix} 0 \\ T_s \\ \frac{1}{m_v} \end{pmatrix}$$

$$A_{i-1,i} = A_{i,i+1} = T_s \begin{pmatrix} 0 & 0 \\ \frac{k_v}{m_v} & \frac{d_v}{m_v} \end{pmatrix}$$

for $i \in \{1, \dots, \bar{I} - 1\}$. Here, $k_v = 3N/m$ denotes the spring constant, $d_v = 3Ns/m$ a damping coefficient, and $T_s = 0.1$ s the step-size of an

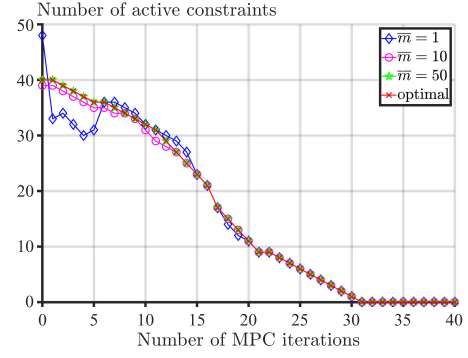


Fig. 4. Total number of active constraints of all distributed QP solvers during the MPC iterations for different choices of \bar{m} .

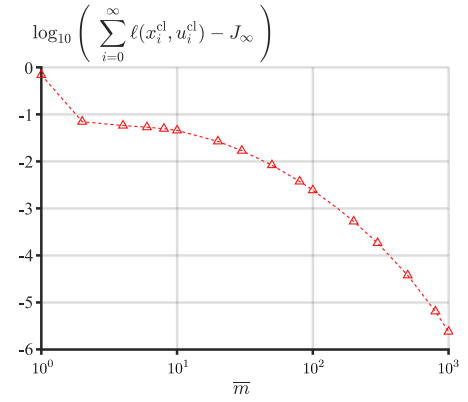


Fig. 5. Closed-loop performance degradation (log scale) with respect to the optimal objective function J_∞ as a function of \bar{m} .

Euler discretization. The state and control constraints are set to

$$\mathbb{X} = \mathbb{X}_1 \times \dots \times \mathbb{X}_{\bar{I}}, \quad \mathbb{U} = [-2, 0.5]$$

$$\text{where } \mathbb{X}_1 = \dots = \mathbb{X}_{\bar{I}} = [-0.5, 1.5] \times [-0.5, 1].$$

The weighting matrices of the stage cost are set to $Q = 10I$ and $R = I$.

In this example, an implementation of Algorithm 1 requires 287 kB corresponding to 432 critical regions. This memory requirement is independent of the number of vehicles \bar{I} and the prediction horizon N . In contrast to this, the number of regions for standard explicit MPC depends on both \bar{I} and N

(\bar{I}, N)	# of regions	memory [kB]
(1, 10)	58	14
(1, 50)	144	169
(2, 10)	2 244	877
(3, 10)	4 247	2 324

Fig. 4 shows the total number of active constraints of all distributed QP solvers for different choices of \bar{m} . Here, the number of active constraints of optimal MPC (corresponding to $\bar{m} = \infty$) are shown in the form of red crosses in Fig. 4. If we compare these optimal red crosses with the blue diamonds ($\bar{m} = 1$), we can see that the choice $\bar{m} = 1$ still leads to many wrongly chosen active sets. However, for $\bar{m} \geq 10$

a reasonably accurate approximation of the optimal number of active constraints is maintained during all iterations. Finally, Fig. 5 shows the suboptimality of Algorithm 1 in dependence on \bar{m} for a representative case study with $\bar{I} = 3$ and $N = 30$.

V. CONCLUSION

This article has introduced a parallelizable and real-time verifiable MPC scheme, presented in the form of Algorithm 1. This control algorithm evaluates at every sampling time a finite number of precomputed, explicit piecewise affine solution maps that are associated with parametric small-scale QPs. Theorem 2 and Corollary 1 provide both asymptotic stability guarantees as well as bounds on suboptimality. The presented explicit MPC approach can be used to reduce the storage and run-time of explicit MPC by orders of magnitude.

REFERENCES

- [1] M. Diehl, H. Ferreau, and N. Haverbeke, "Efficient numerical methods for nonlinear MPC and moving horizon estimation," in *Nonlinear Model Predictive Control*, (Series Lecture Notes in Control and Information Sciences), L. Magni, M. Raimondo, and F. Allgöwer, Eds, vol. 384. Berlin, Germany: Springer, 2009, pp. 391–417.
- [2] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "qpoc: A parametric active-set algorithm for quadratic programming," *Math. Program. Comput.*, vol. 6, no. 4, pp. 327–363, 2014.
- [3] S. Qin and T. Badgwell, "A survey of industrial model predictive control technology," *Control Eng. Pract.*, vol. 93, no. 316, pp. 733–764, 2003.
- [4] B. Houska, H. Ferreau, and M. Diehl, "An auto-generated real-time iteration algorithm for nonlinear MPC in the microsecond range," *Automatica*, vol. 47, pp. 2279–2285, 2011.
- [5] J. Mattingley and S. Boyd, "Automatic code generation for real-time convex optimization," *Convex Optim. Signal Process. Commun.*, D. Palomar and Y. Eldar, Eds., Cambridge University Press, pp. 1–41, 2009.
- [6] M. Diehl, H. Bock, J. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer, "Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations," *J. Process Control*, vol. 12, no. 4, pp. 577–585, 2002.
- [7] V. Zavala and L. Biegler, "The advanced-step NMPC controller: Optimality, stability and robustness," *Automatica*, vol. 45, no. 1, pp. 86–93, 2009.
- [8] D. Ingole and M. Kvasnica, "FPGA implementation of explicit model predictive control for closed loop control of depth of anesthesia" in *Proc. 5th IFAC Conf. Nonlinear Model Predictive Control*, 2015, pp. 484–489.
- [9] R. Oberdieck *et al.*, "On multi-parametric programming and its applications in process systems engineering," *Chem. Eng. Res. Des.*, vol. 116, pp. 61–82, 2016.
- [10] P. Grieder and M. Morari, "Complexity reduction of receding horizon control," in *Proc. 42nd IEEE Int. Conf. Decis. Control*, 2003, pp. 3179–3190.
- [11] R. Cagienard, P. Grieder, E. Kerrigan, and M. Morari, "Move blocking strategies in receding horizon control" *J. Process Control*, vol. 17, no. 6, pp. 563–570, 2007.
- [12] M. Kvasnica, B. Takács, J. Holaza, and S. Di Cairano, "On region-free explicit model predictive control" in *Proc. 54th IEEE Conf. Decis. Control*, 2015, pp. 3669–3674.
- [13] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.
- [14] H. Everett, "Generalized Lagrange multiplier method for solving problems of optimum allocation of resources," *Operations Res.*, vol. 11, no. 3, pp. 399–417, 1963.
- [15] B. Houska, J. Frasch, and M. Diehl, "An augmented Lagrangian based algorithm for distributed non-convex optimization," *SIAM J. Optim.*, vol. 26, no. 2, pp. 1101–1127, 2016.
- [16] P. Giselsson, M. Dang Doan, T. Keviczky, B. De Schutter, and A. Rantzer, "Accelerated gradient methods and dual decomposition in distributed model predictive control," *Automatica*, vol. 49, no. 3, pp. 829–833, 2013.
- [17] A. Kozma, J. Frasch, and M. Diehl, "A distributed method for convex quadratic programming problems arising in optimal control of distributed systems," in *Proc. 52nd IEEE Conf. Decis. Control*, 2013, pp. 1526–1531.
- [18] I. Necoara and J. Suykens, "Application of a smoothing technique to decomposition in convex optimization," *IEEE Trans. Autom. Control*, vol. 53, no. 11, pp. 2674–2679, Dec. 2008.
- [19] B. O'Donoghue, G. Stathopoulos, and S. Boyd, "A splitting method for optimal control," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 6, pp. 2432–2442, Nov. 2013.
- [20] S. Richter, M. Morari, and C. Jones, "Towards computational complexity certification for constrained MPC based on Lagrange relaxation and the fast gradient method," in *Proc. 50th IEEE Conf. Decis. Control Eur. Control Conf.*, 2011, pp. 5223–5229.
- [21] L. Ferranti, G. Stathopoulos, C. N. Jones, and T. Keviczky, "Constrained LQR using online decomposition techniques," in *Proc. IEEE Conf. Decis. Control*, 2016, pp. 2339–2344.
- [22] G. Stathopoulos, T. Keviczky, and Y. Wang, "A hierarchical time-splitting approach for solving finite-time optimal control problems," in *Proc. Eur. Control Conf.*, 2013, pp. 3089–3094.
- [23] J. Oravec, Y. Jiang, B. Houska, and M. Kvasnica, "Parallel explicit MPC for hardware with limited memory," in *Proc. 20th IFAC World Congress*, 2017, pp. 3356–3361.
- [24] F. Borrelli, *Constrained Optimal Control of Linear and Hybrid Systems* (Series Lecture Notes in Control and Information Sciences), vol. 290. Berlin, Germany: Springer, 2003.
- [25] J. Rawlings, D. Mayne, and M. Diehl, *Model Predictive Control: Theory and Design*. Madison, WI, USA: Nob Hill Publishing, 2018.
- [26] F. Borrelli, A. Bemporad, and M. Morari, "Geometric algorithm for multiparametric linear programming," *J. Optim. Theory Appl.*, vol. 118, no. 3, pp. 515–540, 2003.
- [27] P. Otta, O. Santin, and V. Havlena, "Measured-state driven warm-start strategy for linear MPC," in *Proc. Eur. Control Conf.*, 2015, pp. 3132–3136.
- [28] L. Grüne, "Analysis and design of unconstrained nonlinear MPC schemes for finite and infinite dimensional systems," *SIAM J. Control Optim.*, vol. 48, no. 2, pp. 1206–1228, 2009.
- [29] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, "Multi-parametric toolbox 3.0," in *Proc. Eur. Control Conf.*, 2013, pp. 502–510.
- [30] M. Herceg, C. N. Jones, M. Kvasnica, and M. Morari, "Enumeration-based approach to solving parametric linear complementarity problems," *Automatica*, no. 62, pp. 243–248, 2015.
- [31] A. Bemporad, M. Morari, V. Dua, and E. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.
- [32] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
- [33] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems*. Cambridge, U.K.: Cambridge Univ. Press, 2017.
- [34] D. Bertsekas, *Dynamic Programming and Optimal Control*, 3rd ed. Belmont, MA, USA: Athena Scientific, 2012.
- [35] A. Bemporad, A. Casavola, and E. Mosca, "Nonlinear control of constrained linear systems via predictive reference management," *IEEE Trans. Autom. Control*, vol. 42, no. 3, pp. 340–349, Mar. 1997.
- [36] J. Löfberg, "YALMIP" 2004, [Online]. Available: <http://users.isy.liu.se/johanl/yalmip/>



Technical communique

Real-time tunable approximated explicit MPC[☆]Juraj Oravec^{*}, Martin Klaučo

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, Slovakia

ARTICLE INFO

Article history:

Received 9 June 2021

Received in revised form 28 January 2022

Accepted 1 March 2022

Available online 26 April 2022

Keywords:

Model predictive control

Parametric optimization

Tunable controllers

ABSTRACT

Tunability is a major obstacle in the creation and subsequent application of the explicit model predictive control (MPC). The main bottleneck lies in the need to reconstruct the parametric solution each time weighting factors changes. Such an operation makes the implementation of the explicit MPC impractical. This manuscript addresses the problem of producing a suboptimal parametric solution to the optimal control problem, where the change of the weighting factor does not warrant the reconstruction of the explicit MPC. The solution is achieved by interpolating between two boundary explicit solutions for a range of values in weighing factors. Furthermore, we show that the suboptimal solution enforces the closed-loop stability and recursive feasibility. The stability and recursive feasibility are maintained by carefully choosing the terminal penalty and terminal set in those boundary explicit solutions.

© 2022 Elsevier Ltd. All rights reserved.

1. Introduction

In model predictive control (MPC) design, the tuning matrices serve a dual purpose. First is scaling individual components of the state and input vectors. For this purpose, the selection of penalty matrices is dictated by the physics of the plant. While in this scaling approach, we neglect the control-oriented objectives such as performance or comfort. The second purpose of tuning matrices is to manage the *aggressivity* of the controller. We can find many industrial applications where fixed values of tuning factors determined before control do not yield satisfactory performance over time, e.g., see Schutter, Zanon, and Diehl (2020), Sorourifar, Makrygirgos, Mesbah, and Paulson (2021) and Woznsnis, Gudaz, Blevins, and Mehta (2003), and references therein. Therefore, the tunability of the MPC is a highly demanded feature by control engineers. Implementation of optimal control problem with non-fixed tuning factors in explicit MPC framework (Bemporad, Morari, Dua, & Pistikopoulos, 2002) can be done in an optimal fashion in two ways. First, we can construct a parametric solution to the MPC problem subject to *all* parameters, which

now includes tuning factors, resulting in non-linear parametric optimization. The second approach lies in reconstructing the explicit solution each time instant if weighting factors change, which counteracts the effect of having an explicit MPC. So far, a limited number of scientific works have been published in this direction. However, by addressing the issue of effective real-time implementation of tunable explicit MPC strategies, we significantly increase their implementation potential in various fields of applications (Theunissen et al., 2020). Moreover, these controllers can, to some extent, replace gain-scheduling controllers with the added value of constraint satisfaction. The tunability properties of the explicit MPC solutions were previously addressed in Baric, Baotic, and Morari (2005), where a parametric solution with respect to the weighting factor has been derived. The drawback of the proposed solutions is its strict limitation to linear terms in the objective function, namely the $1/\infty$ -norm. Furthermore, the authors in Baric et al. (2005) provide a solution only for MPCs with scalar tuning factor multiplying the weighting factor related to the input penalty. Computing explicit MPC that can be re-tuned online will necessarily increase the complexity of the solution, as the number of parameters needs to be increased. Unfortunately, $1/\infty$ -norm-based penalization setup is rarely used in practice, as the controller exhibits more aggressive behavior around the origin than the quadratic (2-norm) counterparts. Moreover, $1/\infty$ -norm penalization induces a sort of a *dead-zone* behavior for the input penalty in the sense that despite the input penalty being tuned in some range, the optimal control input remains the same, i.e., the effect is discontinuous in contrast to the 2-norm case. A notable step forward in producing online tunable implementation of the explicit MPC with a 2-norm penalization

[☆] The authors gratefully acknowledge the contribution of the Scientific Grant Agency of the Slovak Republic under the grants 1/0585/19, 1/0297/22, and the Slovak Research and Development Agency under the project APVV-20-0261. The 6th IFAC Conference on Nonlinear Model Predictive Control, August 19–22, 2018, Madison, Wisconsin, USA. This paper was recommended for publication in revised form by Associate Editor A. Pedro Aguiar under the direction of Editor André L. Tits.

^{*} Corresponding author.

E-mail addresses: juraj.oravec@stuba.sk (J. Oravec), martin.klauco@stuba.sk (M. Klaučo).

was presented in Klaučo and Kvasnica (2018). Here the authors present an interpolation-based procedure, which allows for the online tuning of the input weighting factor leading to a suboptimal solution. The proposed approach relies on two boundary explicit MPCs, which are constructed for two specific input penalties. The presented approach allows a real-time change in the input penalty matrix within a given range and allows modifying the penalty matrix during the operation without the necessity to resolve optimization problems. The idea is to reconstruct the close-to-optimal control action by devising the convex combination of the available solutions of two boundary explicit MPCs. The approach presented in this paper significantly improves the propositions in Klaučo and Kvasnica (2018), as we generalize the original results and, under mild assumptions, we formulate a procedure where we guarantee the closed-loop system stability and recursive feasibility (Mayne, Rawlings, Rao, & Scokaert, 2000). The proposed control strategy is significantly less complex compared to the conventional approach in Baric et al. (2005), and the suboptimality level of the proposed scheme appears to be negligible.

2. Problem statement

Throughout the paper, we consider the following two MPC design problems implemented in the receding horizon control fashion (Maciejowski, 2000). The first boundary MPC formulation is stated as

$$\min_{u_0, u_1, \dots, u_{N-1}} x_N^T P_L x_N + \sum_{k=0}^{N-1} (x_k^T Q_L x_k + u_k^T R_L u_k) \quad (1a)$$

$$\text{s.t. : } x_{k+1} = A x_k + B u_k, \quad (1b)$$

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

$$x_k \in \mathcal{X}, \quad (1d)$$

$$x_N \in \mathcal{T}_L, \quad (1e)$$

$$x_0 = \theta(t), \quad (1f)$$

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

while the second one has a modification in the objective function and terminal set, namely,

$$\min_{u_0, u_1, \dots, u_{N-1}} x_N^T P_H x_N + \sum_{k=0}^{N-1} (x_k^T Q_H x_k + u_k^T R_H u_k) \quad (2a)$$

$$\text{s.t. : } (1b), (1c), (1d), (1f), (1g), \quad (2b)$$

$$x_N \in \mathcal{T}_H, \quad (2c)$$

where N is prediction horizon, $P_L, P_H \in \mathbb{R}^{n \times n}$, $Q_L, Q_H \in \mathbb{R}^{n \times n}$, $R_L, R_H \in \mathbb{R}^{m \times m}$, are terminal, state, and input pairs of the penalty matrices, respectively. Prediction model in (1b) has the form of linear time invariant (LTI) system for state matrix $A \in \mathbb{R}^{n \times n}$ and input matrix $B \in \mathbb{R}^{n \times m}$. Vectors $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$ are vectors of system states and control inputs, respectively. $\mathcal{U} \subseteq \mathbb{R}^m$, $\mathcal{X} \subseteq \mathbb{R}^n$ are sets of input and state constraints, respectively. $\mathcal{T}_L, \mathcal{T}_H \subset \mathbb{R}^n$, are sets of terminal constraints. $\theta(t) \in \Omega$ is vector of initial conditions and $\Omega \subseteq \mathcal{X}$ is set of feasible initial conditions.

Assumption 2.1. Let MPC problems (1), (2) be asymptotically stable and recursive feasible. Assume, in (1), (2) hold:

- (1) sets $\mathcal{U}, \mathcal{X}, \mathcal{T}_L, \mathcal{T}_H, \Omega$ are closed convex polyhedra containing origin in their strict interiors,
- (2) matrices $P_L, P_H, Q_L, Q_H, R_L, R_H$ are positive definite.

As the initial condition in (1f) has a parametric form, the MPC problems in (1), (2) are problems of the multiparametric quadratic programming (mpQP), see Bemporad et al. (2002).

Problem 2.1. Based on the parametric (explicit) solutions of the MPC problems in (1), (2), the task is to approximate the optimal solution \tilde{u}_0 for any MPC problem having input or state penalty matrix between the matrix pair (R_L, R_H) or (Q_L, Q_H) , respectively, without the necessity to solve the optimization problem.

3. Tunable explicit MPC

Problem 2.1 is addressed by approximated solution of the MPC problem having following form:

$$\min_{u_0, u_1, \dots, u_{N-1}} x_N^T \tilde{P} x_N + \sum_{k=0}^{N-1} (x_k^T \tilde{Q} x_k + u_k^T \tilde{R} u_k) \quad (3a)$$

$$\text{s.t. : } (1b), (1c), (1d), (1f), (1g), \quad (3b)$$

$$x_N \in \tilde{\mathcal{T}}, \quad (3c)$$

where $\tilde{P}, \tilde{\mathcal{T}}$ are appropriate terminal penalty and terminal constraint set, respectively. For input and state penalty matrices in (3a), respectively, hold:

$$\tilde{R} = (\rho - 1)R_L + \rho R_H, \quad 0 \leq \rho \leq 1, \quad (4a)$$

$$\tilde{Q} = (\phi - 1)Q_L + \phi Q_H, \quad 0 \leq \phi \leq 1. \quad (4b)$$

Remark 3.1 (Linear Tuning). We consider a linear tuning, i.e., either input penalty \tilde{R} or state penalty \tilde{Q} in cost (3a) is tuned. Therefore, if $0 < \rho < 1$ then $\tilde{Q} = Q_L = Q_H$, and, vice-versa, if $0 < \phi < 1$ then $\tilde{R} = R_L = R_H$ holds.

Remark 3.2 (Input-penalty-based Tuning). Without loss of generality, hereafter, we consider a tuning of input penalty \tilde{R} according to (4a). Analogous results hold for state penalty \tilde{Q} tuning according to (4b).

Remark 3.3 (Application Range). The application range of the proposed tunable explicit MPC is not limited just to MPC formulation in (3). For the sake of simplicity, we consider the general regulation problem, but various formulations satisfying stability and recursive feasibility could be applied, e.g., robust (Kvasnica, Takács, Holaza and Ingole, 2015), regionless (Kvasnica, Takács, Holaza and Di Cairano, 2015), approximated (Bakarác et al., 2018), convex-lifting-based (Nguyen, Gulan, Oлару, & Rodriguez-Ayerbe, 2018), etc.

The general structure of the MPC problems in (1), (2) is restricted by following mild assumptions.

Assumption 3.1. Consider following assumptions hold:

- (1) terminal penalties in (1a), (2a) are same and computed according to $\tilde{P} = P_L = P_H$ in (3a),
- (2) terminal sets in (1e), (2c) are same and computed according to $\tilde{\mathcal{T}} = \mathcal{T}_L = \mathcal{T}_H$ in (3c),
- (3) input penalties R_L, R_H are diagonal matrices such that $\lambda_i(R_L) \leq \lambda_i(R_H)$ holds $\forall i = 1, \dots, m$, where λ denotes vector of input penalty matrix eigenvalues.

Remark 3.4 (Sufficient Prediction Horizon). Assumption 3.1 is necessary to provide the guarantees on the closed-loop system stability and recursive feasibility. On the other hand, the sufficient length of the prediction horizon N lead to omitting Assumption 3.1(1), (2), see Mönnigmann (2019) to determine the minimum length prediction horizon providing the stability guarantees.

The main benefit of the real-time tunable explicit MPC is that for any $\rho \in [0, 1]$, the online evaluation of the approximated control action is optimization-free and boils down to a mere linear function evaluations.

Definition 3.1 (Approximated Control Input). Given the parametric solutions of MPC problems in (1), (2), and current system measurement $\theta(t) \in \Omega$. The vector of approximated control inputs $\tilde{u} \in \mathbb{R}^m$ for MPC problem in (3) is evaluated using the convex combination:

$$\tilde{u} = (\rho - 1)u_L + \rho u_H, \quad 0 \leq \rho \leq 1, \quad (5)$$

where u_L , u_H , respectively, are optimal control inputs of MPC problems (1), (2) for $\theta(t)$.

Although the \tilde{u}_0 is approximated solution, it provides primal feasibility of MPC problem in (3).

Lemma 3.5 (Primal Feasibility). Given parametric solution of MPC problems in (1)–(2), given MPC problem in (3), and state measurement $\theta(t) \in \Omega$. Approximated solution \tilde{u} evaluated by Definition 3.1 ensures primal feasibility of MPC problem in (3) for $\theta(t)$.

Proof. Let $u_{L,k}$, $u_{H,k}$, $k = 0, 1, \dots, N - 1$, be the optimal solutions of MPC problems in (1)–(2), respectively. According to Definition 3.1, \tilde{u} is evaluated using (5). As the consequence, for any $\theta(t) \in \Omega$ holds:

$$\text{if } u_{L,k} \leq u_{H,k} \Rightarrow u_{L,k} \leq \tilde{u}_k \leq u_{H,k}, \quad (6a)$$

$$\text{if } u_{L,k} \geq u_{H,k} \Rightarrow u_{H,k} \leq \tilde{u}_k \leq u_{L,k}, \quad (6b)$$

for $\forall k = 0, 1, \dots, N - 1$.

Optimal solutions $u_{L,k}$, $u_{H,k}$ are primal feasible, i.e., $u_{L,k}$, $u_{H,k} \in \mathcal{U}$. From the definition of the convex set holds true that convex combination of $u_{L,k}$, $u_{H,k}$ evaluated by (5) satisfies that $\tilde{u}_k \in \mathcal{U}$ for $\forall k = 0, 1, \dots, N - 1$.

Analogous hold true for state constraints in (1d), (1e), as they are, according to a linear prediction model in (1b), a linear combination of control inputs $u_{L,k}$, $u_{H,k}$ for given state measurement $\theta(t) \in \Omega$. \square

Assumption 3.2 (Terminal Penalty). Terminal penalty $\tilde{P} = \tilde{P}^\top > 0$ in (3a) is a matrix of a common quadratic Lyapunov function $V(x) = x^\top P x$, $V : \mathbb{R}^n \rightarrow \mathbb{R}$, constructed w.r.t. LTI prediction model in (1b) and tunable input penalty \tilde{R} in (3a) for $\forall \rho$ within the range of interval $[0, 1]$ in (4a).

In offline phase, there are various ways on how to construct and/or validate Lyapunov function candidates, and how to minimize their conservativeness. Inspired by the approach in Kothare, Balakrishnan, and Morari (1996), we evaluate the terminal penalty matrix \tilde{P} in (3a) by solving the following problem of semidefinite programming (SDP):

$$\min_{\gamma, X, Y} \quad \gamma + \text{tr}(X) \quad (7a)$$

$$\text{s.t. :} \quad \begin{bmatrix} X & \star & \star & \star \\ AX + BY & X & \star & \star \\ Q^{\frac{1}{2}}X & 0 & \gamma I & \star \\ R_j^{\frac{1}{2}}Y & 0 & 0 & \gamma I \end{bmatrix} \geq 0, \quad j = 1, 2, \quad (7b)$$

where decision variables are $X = X^\top \in \mathbb{R}^n$, $Y \in \mathbb{R}^{n \times m}$, and $\gamma \in \mathbb{R}$. $R_1 = R_L$, $R_2 = R_H$, and I , 0 are identity and zero matrices of appropriate dimensions, and symbol \star denotes Hermitian structure of LMIs in (7b). Note, $X > 0$ is weighted inverted Lyapunov matrix such that $\tilde{P} = \gamma X^{-1}$, and Y is well-known matrix of state feedback parametrization. Further technical details are introduced in Kothare et al. (1996). The other strategies determining the terminal penalty are reported, for example, in Bloemen, van den Boom, and Verbruggen (2002) and Lee, Hyun Kwon, and Choi (1998).

Assumption 3.3 (Terminal Constraint). Terminal constraint set $\tilde{\mathcal{T}}$ is a maximal control invariant set \mathcal{C}_∞ constructed in the intersection of the terminal penalties in (1e), (2c) w.r.t. input constraints in (1c), i.e., holds $\tilde{\mathcal{T}} = \mathcal{C}_\infty \subseteq (\mathcal{T}_L \cap \mathcal{T}_H) \subset \mathcal{X}$.

In offline phase, terminal sets \mathcal{T}_L , \mathcal{T}_H in (1e), (2c) are determined to be control invariant for given MPC problems. We consider evaluation of these terminal sets based on the solution of matrix Riccati equations in the LQR-based control framework (Borrelli, 2017). Next, following Assumption 3.3, the terminal set $\tilde{\mathcal{T}}$ in (3c) is evaluated to be a maximal control invariant set \mathcal{C}_∞ constructed in the intersection of $\mathcal{T}_L \cap \mathcal{T}_H$. \mathcal{C}_∞ is constructed w.r.t. input constraints in (1c), i.e., once system states x_k enter \mathcal{C}_∞ , there always exists such control input \tilde{u}_k that ensures $x_{k+1} \in \mathcal{C}_\infty$ hold true. Note, \mathcal{C}_∞ always exists, as there is a neighborhood of origin such that $0 \in \mathcal{T}_L$, $0 \in \mathcal{T}_H \Rightarrow \mathcal{T}_L \cap \mathcal{T}_H \neq \emptyset$. The iterative procedure constructing \mathcal{C}_∞ is introduced, e.g., in Borrelli (2017).

Having a suitable terminal penalty matrix \tilde{P} and terminal constraint set $\tilde{\mathcal{T}}$, we formulate the main results of this work.

Theorem 3.6 (Tunable Explicit MPC). Given parametric solutions of MPC problems in (1), (2), given MPC problem in (3), current system state measurement $\theta(t) \in \Omega$, and corresponding control input \tilde{u}_0 by Definition 3.1. If Assumptions 3.1–3.3 hold, then the closed-loop system is asymptotically stable and problem in (3) is recursively feasible for any ρ within the range of interval $[0, 1]$ in (4a).

Proof. According to Assumption 3.2, for $\forall \rho$ within the range of interval $[0, 1]$ in (4a), there exists a common quadratic Lyapunov function represented by positive definite matrix \tilde{P} for MPC problem in (3). According to Assumption 3.1, the terminal constraint set $\tilde{\mathcal{T}}$ and terminal penalty \tilde{P} of MPC problems in (1), (2) are the same. According to Assumption 3.3, terminal constraint set $\tilde{\mathcal{T}}$ is maximal control invariant set \mathcal{C}_∞ w.r.t. constraints on control inputs in (1c).

The pair of control inputs $u_{L,0}$, $u_{H,0}$ represent the optimal solutions of MPC problems in (1), (2), respectively. This pair of optimal control inputs $u_{L,0}$, $u_{H,0}$ was evaluated w.r.t. the same terminal penalty represented by \tilde{P} , and system states converge to the same terminal penalty set $\tilde{\mathcal{T}}$. As the consequence, they guarantee the closed-loop system stability and recursive feasibility for MPC problems in (1), (2), respectively.

According to Assumption 3.1, MPC problem in (3) shares the same terminal penalty and terminal constraint set. According to Definition 3.1 and Lemma 3.5, for any \tilde{u}_0 holds (6). As the consequence, \tilde{u}_0 leads to closed-loop system stability and recursive feasibility of MPC problem in (3). \square

Compared to the implementation of the conventional (non-tunable) explicit MPC, the proposed method increases effort in both: (i) offline phase by solving two parametric optimization problems and in (ii) online phase by evaluating two boundary explicit control laws. The evaluation of convex combination in (5) is negligible. Moreover, the memory footprint increases as the parametric solution of both MPC problems needs to be stored. On the other hand, the real-time tunable explicit MPC offers an infinite number of stabilizing sub-optimal control actions (i.e., MPC controllers) for input penalty tuning given by $\rho \in [0, 1]$ (or $\phi \in [0, 1]$ for state penalty tuning) in (4) without the necessity to solve any optimization problem. This valuable benefit enables convenient tuning, validation, and verification of the closed-loop control performance reducing the evaluation effort to a few linear algebra operations. The increased online effort could be reduced by introducing advanced techniques recalling some information from the computation of u_L to speed up evaluation of u_H in (5). This approach goes beyond the scope of this paper and is a subject of further research.

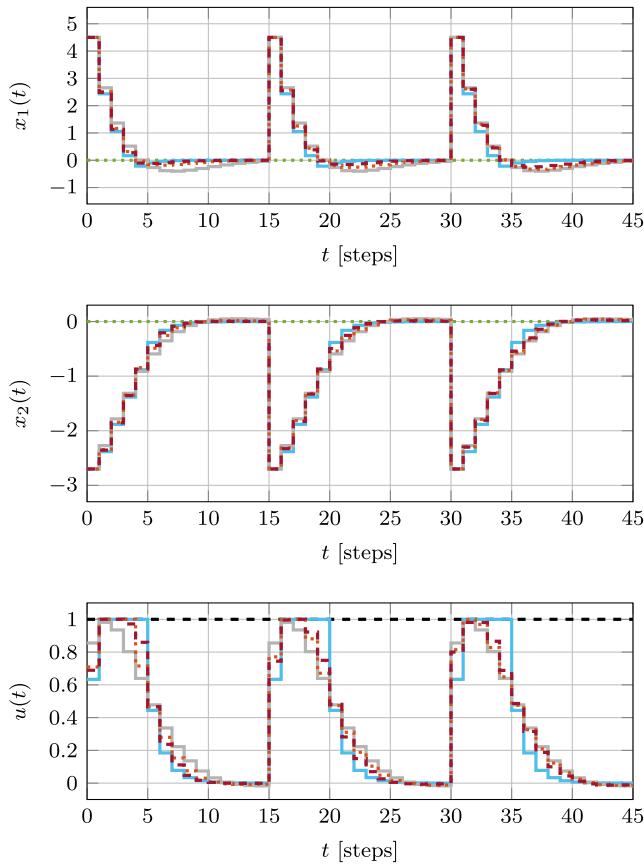


Fig. 1. Control performance of real-time tunable explicit MPC ensured by \tilde{u} (dark red, dashed), u_L (blue, solid), u_H (gray, solid), u_{opt} (orange, dotted), reference (green, dotted), constraints (black, dashed) for sequence of tuning parameter $\rho = 0.25, 0.50, 0.75$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Remark 3.7 (1-norm and ∞ -norm Cost). Cost functions in MPC problems (1), (2) have the form of (squared) 2-norm. Analogous results of Lemma 3.5 and Theorem 3.6 hold for 1-norm and ∞ -norm cost functions. Moreover, as 1-norm and ∞ -norm are of linear nature (piecewise affine), the suboptimality level of approximated control input in (5) converge much faster to the optimum value for given prediction horizon N . The closed-loop system stability and recursive feasibility issues of Assumptions 3.2, 3.3 can be addressed according to Remark 3.4.

4. Example

The simplified numerical example demonstrates the real-time tunable level of aggressiveness/energy losses of the closed-loop performance under different setups of (Q, R) proportion. This ability is illustrated considering a well-known example of a double integrator system with LTI prediction model in (1b) having $A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$, $B = \begin{bmatrix} 1.0 \\ 0.5 \end{bmatrix}$, and following setup of MPC problems in (1), (2): $Q_L = Q_H = I$, $R_L = 0.5$, $R_H = 10.0$, $\mathcal{U} = \{u : -1 \leq u \leq 1\}$, $\mathcal{X} = \{x : -5 \leq x \leq 5\}$, and $N = 2, 3, 5, 10$. Solving SDP in (7) and evaluation of c_∞ , return terminal penalty and terminal set in the form

$$\tilde{P} = \begin{bmatrix} 6.3743 & 0.5172 \\ 0.5172 & 15.5601 \end{bmatrix},$$

Table 1
Performance criteria.

N	Performance loss [%]			Offline [s]		Online [ms]
	$\rho = 0.25$	$\rho = 0.50$	$\rho = 0.75$	t_{con}	t_{opt}	t_{app}
2	0.23	0.13	0.11	0.34	0.52	1.07
3	0.36	0.27	0.10	0.45	0.55	1.12
5	0.26	0.28	0.16	0.80	0.57	1.13
10	0.21	0.22	0.13	1.26	0.63	1.27

$$\tilde{\tau} = \left\{ x : \begin{bmatrix} 0.0651 & -0.4463 \\ 0.1742 & -0.2388 \\ 0.4004 & 0.6479 \\ -0.0651 & 0.4463 \\ -0.1742 & 0.2388 \\ -0.4004 & -0.6479 \end{bmatrix} x \leq \begin{bmatrix} 0.8925 \\ 0.9553 \\ 0.6479 \\ 0.8925 \\ 0.9553 \\ 0.6479 \end{bmatrix} \right\}.$$

Based on value of N , solving¹ MPC problems (1), (2) lead to polytopic partitions having from 9 up to 21 critical regions for (1) and from 11 up to 15 critical regions for (2), after processing an optimal regions merging (Kvasnica, Holaza, Takács and Ingole, 2015). The series of explicit MPC controllers for input penalty \tilde{R} corresponding to particular setup of tuning parameter $\rho = 0.25, 0.50, 0.75$ in (5) was constructed to investigate the closed-loop performance. Fig. 1 depicts the closed-loop control trajectories excited by initial conditions $x_0 = x_{15} = x_{30} = [4.5, -2.7]^T$ for particular setup of $N = 10$. The control profiles generated by approximated control inputs \tilde{u} are compared to optimal performance u_{opt} . Table 1 summarizes construction time t_{con} of explicit controllers and the average real-time evaluation of optimal t_{opt} and approximated t_{app} control actions running non-optimized code on a non-industrial hardware. It can be observed that approximated control action increased a runtime, approximately, by factor 2. On the other hand, there is no need to solve any multiparametric optimization problem online. Moreover, the performance loss computed as a ratio of optimal and approximated closed-loop costs is negligible, as reported in Table 1.

Acknowledgment

The authors sincerely thank Michal Kvasnica, whose comments helped improve the manuscript.

References

Bakaráč, P., Holaza, J., Klaučo, M., Kalúz, M., Löfberg, J., & Kvasnica, M. (2018). Explicit MPC based on approximate dynamic programming. In *ECC 2018* (pp. 1172–1177). Limassol, Cyprus.

Baric, M., Baotic, M., & Morari, M. (2005). On-line tuning of controllers for systems with constraints. In *Proc. of the 44th IEEE CDC* (pp. 8288–8293). <http://dx.doi.org/10.1109/CDC.2005.1583504>.

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

Bloemen, H. H. J., van den Boom, T. J. J., & Verbruggen, H. B. (2002). Optimizing the end-point state-weighting matrix in model-based predictive control. *Automatica*, 38(6), 1061–1068.

Borrelli, F. (2017). *Constrained optimal control of linear and hybrid systems*. Springer Berlin Heidelberg.

Klaučo, M., & Kvasnica, M. (2018). Towards on-line tunable explicit MPC using interpolation. In *Preprints of the 6th IFAC conference on nonlinear model predictive control*. Madison, Wisconsin, USA.

Kothare, M. V., Balakrishnan, V., & Morari, M. (1996). Robust constrained model predictive control using linear matrix inequalities. *Automatica*, 32, 1361–1379.

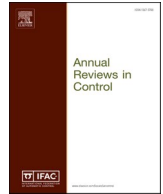
Kvasnica, M., Holaza, J., Takács, B., & Ingole, D. (2015). Design and verification of low-complexity explicit MPC controllers in MPT3. In *ECC 2015* (pp. 2600–2605). Linz, Austria.

¹ The results were generated using Intel(R) Core(TM) i7-1065G7 CPU 1.50 GHz, 16 GB RAM, MATLAB R2021b, YALMIP R20210331, MPT v3.2.1, MOSEK v9.3.6.

- Kvasnica, M., Takács, B., Holaza, J., & Cairano, S. D. (2015). On region-free explicit model predictive control. In *54th IEEE CDC* (pp. 3669–3674). Osaka, Japan.
- Kvasnica, M., Takács, B., Holaza, J., & Ingole, D. (2015). Reachability analysis and control synthesis for uncertain linear systems in MPT. In *Proc. of the 8th IFAC Rocond* (pp. 302–307). Bratislava, Slovakia.
- Lee, J. W., Hyun Kwon, W., & Choi, J. (1998). On stability of constrained receding horizon control with finite terminal weighting matrix. *Automatica*, 34(12), 1607–1612.
- Maciejowski, J. (2000). *Predictive control with constraints*. London: Prentice Hall.
- Mayne, D. Q., Rawlings, J. B., Rao, C. V., & Sokaert, P. O. M. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6), 789–814.
- Mönnigmann, M. (2019). On the structure of the set of active sets in constrained linear quadratic regulation. *Automatica*, 106, 61–69.
- Nguyen, N. A., Gulan, M., Oлару, S., & Rodriguez-Ayerbe, P. (2018). Convex lifting: Theory and control applications. *IEEE Transactions on Automatic Control*, 63(5), 1243–1258.
- Schutter, J. D., Zanon, M., & Diehl, M. (2020). TuneMPC—a tool for economic tuning of tracking (N)MPC problems. *IEEE Control Systems Letters*, 4(4), 910–915.
- Sorourifar, F., Makrygorgos, G., Mesbah, A., & Paulson, J. A. (2021). A data-driven automatic tuning method for MPC under uncertainty using constrained Bayesian optimization. *IFAC-PapersOnLine*, 54(3), 243–250, 16th IFAC Symposium on Advanced Control of Chemical Processes.
- Theunissen, J., Sornioti, A., Gruber, P., Fallah, S., Ricco, M., Kvasnica, M., et al. (2020). Regionless explicit model predictive control of active suspension systems with preview. *IEEE Transactions on Industrial Electronics*, 67(6), 4877–4888.
- Wojsznis, W., Gudaz, J., Blevins, T., & Mehta, A. (2003). Practical approach to tuning MPC. *ISA Transactions*, 42(1), 149–162.

Contents lists available at [ScienceDirect](https://www.sciencedirect.com)

Annual Reviews in Control

journal homepage: www.elsevier.com/locate/arcontrol

All you need to know about model predictive control for buildings

Ján Drgoňa ^{*,a,b}, Javier Arroyo ^{b,c,d}, Iago Cupeiro Figueroa ^{b,c}, David Blum ^e, Krzysztof Arendt ^f, Donghun Kim ^{e,g}, Enric Perarnau Ollé ^b, Juraj Oravec ^h, Michael Wetter ^e, Draguna L. Vrabie ^a, Lieve Helsen ^{b,c}

^a Pacific Northwest National Laboratory, Richland, WA, USA^b KU Leuven, Department of Mechanical Engineering, Leuven, Belgium^c EnergyVille, Thor Park, Waterschei, Belgium^d VITO NV, Boeretang Mol, 200, Belgium^e Lawrence Berkeley National Laboratory, Berkeley, CA, USA^f University of Southern Denmark, Center for Energy Informatics, Denmark^g Purdue University, School of Mechanical Engineering, West Lafayette, IN, USA^h Slovak University of Technology in Bratislava, Faculty of Chemical and Food Technology, Slovakia

ARTICLE INFO

Keywords:

Model predictive control
Building climate control
MPC formulation
MPC software tools
MPC implementation

ABSTRACT

It has been proven that advanced building control, like model predictive control (MPC), can notably reduce the energy use and mitigate greenhouse gas emissions. However, despite intensive research efforts, the practical applications are still in the early stages. There is a growing need for multidisciplinary education on advanced control methods in the built environment to be accessible for a broad range of researchers and practitioners with different engineering backgrounds. This paper provides a unified framework for model predictive building control technology with focus on the real-world applications. From a theoretical point of view, this paper presents an overview of MPC formulations for building control, modeling paradigms and model types, together with algorithms necessary for real-life implementation. The paper categorizes the most notable MPC problem classes, links them with corresponding solution techniques, and provides an overview of methods for mitigation of the uncertainties for increased performance and robustness of MPC. From a practical point of view, this paper delivers an elaborate classification of the most important modeling, co-simulation, optimal control design, and optimization techniques, tools, and solvers suitable to tackle the MPC problems in the context of building climate control. On top of this, the paper presents the essential components of a practical implementation of MPC such as different control architectures and nuances of communication infrastructures within supervisory control and data acquisition (SCADA) systems. The paper draws practical guidelines with a generic workflow for implementation of MPC in real buildings aimed for contemporary adopters of this technology. Finally, the importance of standardized performance assessment and methodology for comparison of different building control algorithms is discussed.

1. Introduction

Buildings today contribute to roughly 40% of the global energy use (approx. 64 PWh), of which a large portion is used for heating, cooling, ventilation, and air-conditioning (HVAC) (IEA International Energy Agency & International Partnership for Energy Efficiency Cooperation, 2015). Energy savings thus become a priority in the design and

operation of modern HVAC systems. Numerous studies reported that advanced HVAC control can notably reduce energy use and mitigate greenhouse gas emissions with average energy savings of 13% to 28% (Gyalistras et al., 2010; del Mar, Álvarez, de A., & Berenguel, 2014; Roth, Westphalen, Dieckmann, Hamilton, & Goetzler, 2002). This means that in the ideal case of full employment of this technology, annual final energy savings of roughly 8PM h to 18PM h can be projected. Based on

* Corresponding author at: Pacific Northwest National Laboratory, Richland, WA, USA.

E-mail addresses: jan.drgona@pnnl.gov (J. Drgoňa), javier.arroyo@kuleuven.be (J. Arroyo), iago.cupeirofigueroa@kuleuven.be (I. Cupeiro Figueroa), DHBlum@lbl.gov (D. Blum), krza@mmmi.sdu.dk (K. Arendt), donghunkim@lbl.gov (D. Kim), enric.perarnauolle@student.kuleuven.be (E.P. Ollé), juraj.oravec@stuba.sk (J. Oravec), mwetter@lbl.gov (M. Wetter), draguna.vrabie@pnnl.gov (D.L. Vrabie), lieve.helsen@kuleuven.be (L. Helsen).

<https://doi.org/10.1016/j.arcontrol.2020.09.001>

Received 4 May 2020; Received in revised form 30 August 2020; Accepted 2 September 2020

Available online 29 September 2020

1367-5788/Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

this potential, recently revised EU policy on the energy performance of buildings states that large buildings should be equipped with building automation and control systems by 2025 if economically and technically feasible (EUp, 2018).

However, the majority of buildings today still adopt simple rule-based control (RBC) techniques with only limited energy saving capabilities (Aghemo et al., 2013; Mechri, Capozzoli, & Corrado, 2010). The promise of a digital age comes with decreasing costs in computation and sensing, which is paving the way for the adoption of advanced control strategies, like model predictive control (MPC). In the last decade, MPC has become a dominant control strategy in research on intelligent building operation. The main benefit of MPC is a systematic thermal comfort improvement with simultaneous energy savings spanning from 15% up to 50% demonstrated on numerous simulation and several pilot case studies (Ma et al., 2012; Oldewurtel et al., 2012; Sturzenegger, Gyalistras, Morari, & Smith, 2016; Široký, Oldewurtel, Cigler, & Prívará, 2011), as well as grid flexibility services via price-responsiveness and active demand response capabilities (Bianchini, Casini, Pepe, Vicino, & Zanvettor, 2017; Borsche, Oldewurtel, & Andersson, 2014; Cutsem, Kayal, Blum, & Pritoni, 2019a; Esther & Kumar, 2016). The strength of MPC lies in the use of a mathematical model of the building to predict its future behavior. By using these predictions, MPC can optimally choose the control actions based on a given objective while taking into account the comfort and technological constraints, and weather forecasts in a systematic and flexible way.

Despite the abundance of research papers and several pilot installations, the transfer of this technology to the building market is still in its early stages. The difficulty of the building sector stems from the fact that building management systems (BMS) engineers do not have advanced education in modern optimal control methods and tools, as control engineers do in other fields that have utilized MPC successfully, such as the process industry. Moreover, in contrast to the production of cars or user electronics, design and production of building and their HVAC systems are not standardized. Every building is a unique system which requires tailored modeling and control design, hence imposing increased engineering time and cost, particularly for advanced control strategies. All of this emphasizes the requirement for extending the theoretical education and practical skill set of the building control practitioners to enable the installation, maintenance, and operation of advanced MPC applications. An additional limiting factor is the poor ICT infrastructure in pre-existing buildings. One of the emerging advanced building control solutions is cloud-based control as a service platform. Although, significant privacy and cyber-security challenges are linked with these remote control architectures. Based on the observations described above and reflections presented in Cigler, Gyalistras, Široký, Tiet, and Ferkl (2013a); Prívará et al. (2013), six main challenges for wide-spread application of MPC to buildings are defined:

1. Availability of appropriate hardware and software infrastructure with compatible communication interfaces.
2. User-friendly, control-oriented, accurate, and computationally efficient building modeling.
3. Automated design, tuning, and deployment of MPC.
4. Plug-and-play implementation, and robust operation of MPC.
5. Privacy and cyber-security issues and the user trust.
6. Trained personnel to handle commissioning, and maintenance of MPC in practice.

The first challenge does not fall in the scope of research anymore because it lies in the domain of market adaptation. To address the second challenge, a methodology for the automatic synthesis of building models based on Building Information Models (BIM) has been proposed (Andriamonjy, 2018). Different attempts in reducing the model development effort via available templates in Modelica libraries like Buildings (Wetter, Zuo, Nouidui, & Pang, 2014) and IDEAS (Baetens et al., 2015), or via physically inspired reduced-order automated system

identification toolchains (De Coninck, Magnusson, Åkesson, & Helsen, 2016). Dedicated tools are also emerging for automated MPC design for buildings (Blum & Wetter, 2017; Drgoña, 2019; Jorissen, Boydens, & Helsen, 2018a). Computationally lightweight approximations of MPC control laws (Drgoña, Picard, Kvasnica, & Helsen, 2018), and rule extraction algorithms based on machine learning (Domahidi, Ullmann, Morari, & Jones, 2014), or toolchains for generation of optimized C-code (Jorissen et al., 2018a) aim to tackle the fourth challenge of easy installation and robust operation. The privacy issues could be solved in two ways, first by employing local control solutions without the need for real-time remote communication, and second by the adoption of advanced cybersecurity measures (Cybersecurity in smart buildings inaction is not an option anymore, 2015).

The ambition of this paper is to deliver a comprehensive summary on the topic of MPC for buildings, which could help to tackle the last challenge from the list. The necessary theoretical base on MPC is first supported by a literature review of the most recent advances in the field. Then, an extensive overview and conceptual comparison of dedicated software tools is given, followed by practical guidelines for implementation and performance assessment of MPC in real buildings.

1.1. Previous reviews considering MPC for buildings

We would like to acknowledge a first attempt to provide a unified MPC framework, which was given in Serale, Fiorentini, Capozzoli, Bernardini, and Bemporad (2018). This review paper aims to build a bridge between control and building engineers with a common dictionary and taxonomy of classes to enhance the professional relationship between these two originally distinct engineering areas. The most recent review on MPC for buildings with the focus on demand-side flexibility compares the pros and cons of the current technology and highlights the requirement of expert knowledge as the main bottleneck (Zong et al., 2019). An overview on three major research topics in building control, in particular semantic interoperability, fault detection, and MPC, was presented in Benndorf, Wystrcil, and Réhault (2018). A paper with in-depth literature review and classification of building control methods with particular focus on MPC has been published by (Afram & Janabi-Sharifi, 2014b).

More specific reviews of the MPC technology focusing on particular aspects of building control are as follows. One of the earliest short reflections on MPC technology for buildings was given by (Henze, 2013) envisioning a large impact of MPC technology on intelligent building operation. A review paper focused on artificial neural network based MPC was given in Afram, Janabi-Sharifi, Fung, and Raahemifar (2017). A review on an important aspect of occupancy behavior focused MPC was introduced in Mirakhorli and Dong (2016), concluding that using occupancy measurement and models in combination with MPC can improve the comfort and decrease the energy use in contrast to a standard schedule based control strategy. Reviews (Hilliard, Kavgic, & Swan, 2015; Rockett & Hathway, 2017) focus on challenges, aspects and future trends of MPC for commercial buildings. Paper (Hilliard et al., 2015) provides recommendations for selecting a building response model, simulation timestep, prediction horizon, forecast resolution, and optimization algorithm, while (Rockett & Hathway, 2017) stresses the urgent need for research on the automated creation and updating of predictive models for MPC. Authors in Killian and Kozek (2016) ask ten questions about MPC for buildings and provide critical analysis of challenges, future trends, and potential of MPC for the general building market. The identified challenges are high modeling and parametrization effort, shortage of modeling and optimal control experts active in the building automation domain, and lack of commercial tools for expert-free building modeling. In Kavgic, Hilliard, and Swan (2015), the authors discussed the opportunities for implementation of MPC in commercial buildings together with the identification of specific building characteristics indicating increased potential for MPC, like large thermal mass, high solar gains, discrete occupancy periods, and the

Table 1
Nomenclature of terms and acronyms used in the paper.

Notation	Meaning	Notation	Meaning
Control terminology			
PID	proportional-integral-derivative	RBC	rule-based control
MPC	model predictive control	LMPC	linear MPC
NMPC	nonlinear MPC	HMPC	hybrid MPC
eMPC	explicit MPC	OSF-MPC	offset-free MPC
RMPC	robust MPC	SMPC	stochastic MPC
LBMPC	learning-absed MPC	RHC	reciding horizon control
DPC	data predictive control	OCF	optimal control problem
SSM	state-space model	TF	transfer function
KF	Kalman Filter	MHE	moving horizon estimation
UKF	Unscented Kalman Filter	EKF	extended Kalman Filter
TVKF	time-varying Kalman Filter	SKF	stationary Kalman Filter
ADP	approximate dynamic programming	DP	dynamic programming
HJB	Hamilton-Jacobi-Bellman equation	RL	reinforcement learning
Optimization terminology			
OP	optimization problem	ADMM	alternating direction method of multiliers
LP	linear programming	QP	quadratic programming
NLP	nonlinear programming	SQP	sequential quadratic programming
MIP	mixed integer programming	MINLP	mixed integer nonlinear programming
MILP	mixed integer linear programming	MIQP	mixed integer quadratic programming
GDP	generalized disjunctive programming	mpP	multi parametric programming
mpQP	multi parametric quadratic programming	mpLP	multi parametric linear programming
LMI	linear matrix inequality	CC	chance constraints
SDP	semidefinite programming	SOCP	second order cone programming
Modeling terminology			
ODE	ordinary differential equations	DAE	differential algebraic equations
AR	auto regressive	ARMA	auto regressive moving average
BJ	Box-Jenkins	ARMAX	auto regressive moving average with exogenous inputs
ANN	artificial neural network	DT	decision tree
SVM	support vector machines	RF	random forests
kNN	k-nearest neighbors	GP	gaussian processes
4SID	subspace state space system identification	OE	output error
MBE	mean biased error	RMSE	root mean square error
EEP	expected error percentage	CV	coefficient of variation
PRBS	pseudo random binary signal	CRPS	continuous ranked probability score
Building domain terminology			
HVAC	heating, ventilation, and air conditioning	AHU	air handling unit
VAV	variable air flow	BES	building energy simulation
FMI	functional mockup interface	BIM	building information modeling
PIR	passive infrared sensor	BaU	business as usual
iCRTF	inverse comprehensive room transfer functions	CFD	computational fluid dynamics
SCADA	supervisory control and data acquisition	BMS	building management system
HMI	human machine interface	CM	number of comfort violations minimization
CT	comfort tracking	PMV	predicted mean vote

opportunity to vary temperature setpoints.

Dounis and Caraiscos (2009); Naidu and Rieger (2011); Wang and Ma (2008) are more general building control reviews and classification studies covering advanced intelligent and optimal building control strategies. A detailed review in Shaikh, Nor, Nallagownden, Elamvazuthi, and Ibrahim (2014) summarizes the impact of smart control strategies on energy and comfort management in buildings focusing on aspects such as building sector, optimization objectives, energy source, control algorithm and simulation tools used. Optimal operation of energy management systems with a weather forecast is reviewed in Lazos, Sproul, and Kay (2014) concluding that weather has a significant influence on building energy operation and that the minimization of forecast uncertainty can lead to increased energy savings in the range of 15% to 30%. A most recent analysis of optimization-based building automation and control systems focusing on performance gap mitigation and uncertainty evaluation was given in Aste, Manfren, and Marenzi (2017). Different optimization methods applied to different energy domain areas are reviewed in Baños et al. (2011). A review of multi-criteria decision analysis (MCDA), which could be used to aid the selection of the objectives for MPC for buildings, was presented in Wang, Jing, Zhang, and Zhao (2009). Finally, deeper insights into MPC technology, in general, can be found, e.g. in Bemporad (2006); Mayne (2014).

1.2. Contributions and structure of the paper

The presented paper aims to provide a comprehensive up to date overview of MPC technology applied to buildings. Although there are several reviews on general intelligent building operation strategies and MPC, to the authors' best knowledge, a unifying overview integrating both theoretical and practical aspects is still missing in this field. The ambition of this paper is thus to fill this gap and provide the reader with a single publication capable of guiding the whole process of implementation of MPC in a real building. The paper is also aimed to act as a detailed introduction to the topic for control and mechanical engineers and researchers, facilitating the information exchange in the multidisciplinary domain of building control. In comparison to referenced literature reviews in Section 1.1, the purpose of this paper is not to redefine, but refine and extend given MPC frameworks from previous literature overviews with a particular focus on providing a detailed list of software resources for increased accessibility of the technology.

The first part of the paper emphasizes a theoretical perspective. Section 2 defines the general MPC framework with standard notation. Section 3 elaborates on building modeling. Section 4 gives a brief summary of algorithmic principles behind MPC. Sections 5 and 6 summarize different MPC problem classes and corresponding solution approaches, respectively. Section 7 compactly reviews methods for dealing

with uncertainties in MPC for buildings. Everything is supported by a comprehensive up-to-date literature review reporting successful real-world applications or in-depth simulation case studies of the presented concepts.

In the second part of the paper, the emphasis lies on the practical aspects of the technology. Section 8 provides a comprehensive list of available software tools for modeling, analysis, and solution of MPC problems. Section 9 delves deeper into practical aspects of MPC implementation, such as control configuration, communication infrastructure, and SCADA architecture, together with practical guidelines for implementation in real buildings. Section 10 introduces the need and methodology for performance assessment and comparison of MPC strategies for buildings. Finally, Section 11 concludes the paper.

1.3. Nomenclature

Table 1 summarizes the terminology and acronyms used in the paper with domain-specific classification.

2. Model predictive building control

The purpose of the following section is to compactly define and summarize the general MPC framework for building applications. We present here the fundamental building blocks and corresponding concepts of MPC, different problem formulations, and a notation based on standards used in the control engineering community. The general MPC framework compatible with the structure of this paper is presented in Fig. 1. The presented framework is the extension of the MPC framework given in Serale et al. (2018).

2.1. Model predictive control basics

MPC is a constrained optimal control strategy that calculates the optimal control inputs by minimizing a given objective function over a finite prediction horizon. The mathematical model of the system together with the current state measurements and weather forecast are used to predict and optimize the future behavior of the building.

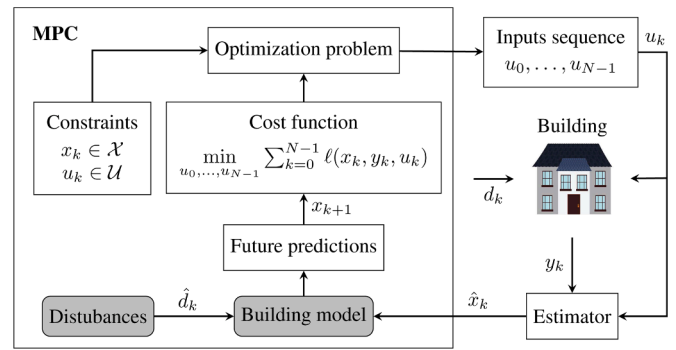


Fig. 2. Schematic representation of the standard closed-loop system for building control with MPC and state estimator.

2.1.1. Standard MPC scheme

Fig. 2 illustrates a typical abstract closed-loop MPC scheme which can describe most of the building control applications. The control loop consists of the building affected by disturbances d (e.g., weather conditions), predicted by weather forecast \hat{d} , the state estimator providing the state estimates \hat{x} and the MPC controller which optimally manipulates the control actions u (e.g., heat flows, valves opening, pump powers), e.g., such that it minimizes used energy and keeps the output vector y (e.g., room temperatures) within the given comfort bounds.

2.1.2. General MPC formulation

The general MPC formulation for buildings can be represented as the following optimal control problem (OCP) in discrete time:

$$\min_{u_0, \dots, u_{N-1}} \ell_N(x_N) + \sum_{k=0}^{N-1} \ell_k(x_k, y_k, r_k, u_k, s_k) \tag{1a}$$

$$\text{s.t. } x_{k+1} = f(x_k, u_k, d_k), k \in \mathbb{N}_0^{N-1} \tag{1b}$$

$$y_k = g(x_k, u_k, d_k), k \in \mathbb{N}_0^{N-1} \tag{1c}$$

$$u_k = f_{\text{HVAC}}(x_k, a_k, m_k), k \in \mathbb{N}_0^{N-1} \tag{1d}$$

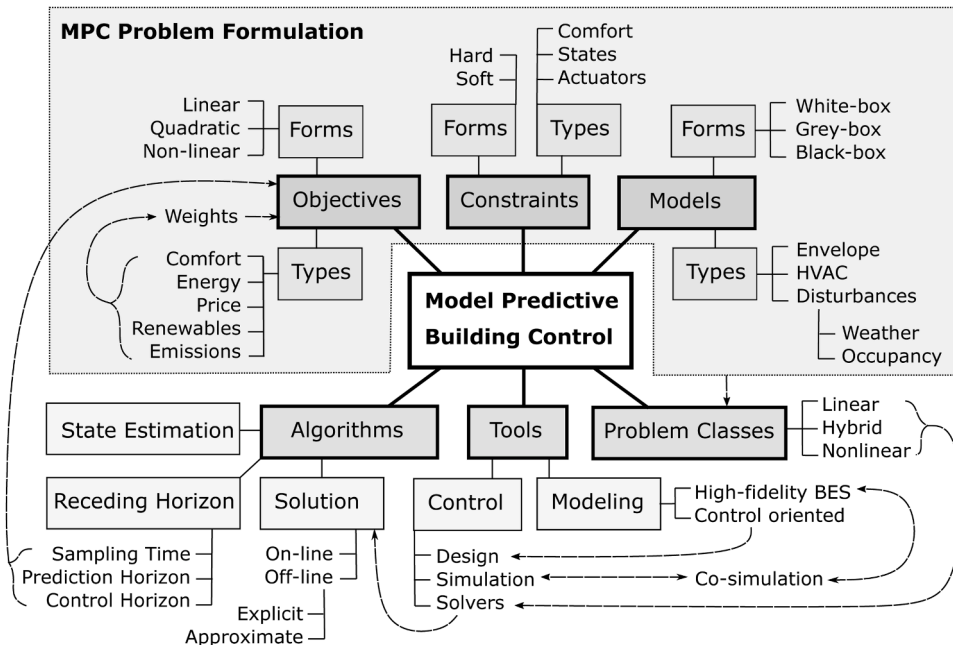


Fig. 1. Structure of the general MPC framework for building control applications compatible with the structure of this paper. Solid lines represent the sub-categories, while dashed lines with arrows depict causal dependencies and information flow during the design process.

$$s_k = h(x_k, y_k, u_k, r_k), k \in \mathbb{N}_0^{N-1} \quad (1e)$$

$$x_k \in \mathcal{X}, u_k \in \mathcal{U}, a_k \in \mathcal{A}, s_k \in \mathcal{S}, k \in \mathbb{N}_0^{N-1} \quad (1f)$$

$$d_k = d(t + kT_s), k \in \mathbb{N}_0^{N-1} \quad (1g)$$

$$r_k = r(t + kT_s), k \in \mathbb{N}_0^{N-1} \quad (1h)$$

$$x_0 = \hat{x}(t), \quad (1i)$$

where $x_k \in \mathbb{R}^{n_x}$ denotes the values of states, $y_k \in \mathbb{R}^{n_y}$ the outputs, $u_k \in \mathbb{R}^{n_u}$ the building envelope inputs, $a_k \in \mathbb{R}^{n_a}$ the HVAC actuators, $m_k \in \mathbb{R}^{n_m}$ the additional measured variables, $d_k \in \mathbb{R}^{n_d}$ the disturbances, $r_k \in \mathbb{R}^{n_r}$ the reference signals, and $s_k \in \mathbb{R}^{n_s}$ denote the slack variables, at the k th step of the prediction horizon N with a sampling time T_s , where n_x denotes the dimensionality of associated variable \star .

The objective function is given by (1a), where $\ell_N(x_N)$ represents the terminal penalty used to ensure the stability and convergence of the control. For most of the building control applications the terminal penalty is omitted. $\ell(r_k, y_k, u_k, s_k)$ is called a stage cost and its purpose is to assign a cost to a particular choice of x_k, y_k, r_k, u_k and s_k .

The predictions of the state values are obtained from the state update Eq. (1b), while the values of the predicted outputs are given by the output Eq. (1c). The building envelope inputs u_k are subject to the HVAC dynamics (1d). Slack variables usually represent the violations of additional algebraic constraints (1e), such as comfort zones. States, envelope inputs, actuators, and slack variables are often subject to bounding constraints (1f). The initial conditions of the state variables are given by (1i) which are either measured or estimated. A forecasts of the disturbances and reference signals are given by (1g) and (1h), respectively. For building control applications, disturbances usually represent weather conditions and occupancy behavioral patterns, while reference signals span from tracking a single reference signal to more common comfort ranges on controlled variables. For the sake of generality we denote by ξ the vector that encapsulates all time-varying parameters of (1), i.e., the current state estimates $\hat{x}(t)$, current and future disturbances $d(t), \dots, d(t + (N - 1)T_s)$, and reference signals $r(t), \dots, r(t + (N - 1)T_s)$. Compression of all parameters into single vector ξ is convenient for compact representation of MPC feedback law $A = f_{MPC}(\xi)$, where $A = [a_0, a_1, \dots, a_{N-1}]$ is the vector of computed optimal control actions.

2.1.3. Standard MPC notation

Table 2 summarizes the standard notation and meaning of the variables used in the control community together with most common

Table 2
Standard notation and most common physical representation of the variables used in MPC for buildings.

Notation	Controller	Building	Units
x	states	building structure temperatures	[K]
y	outputs	room operative temperatures	[K]
u	inputs	heat flows to the zones	[W]
a	actuators	valve and pump modulations	[%]
m	measurements	HVAC states	[K, W, %]
d	disturbances	ambient temperatures, solar radiation, and internal heat gains	[K, W]
r	references	comfort zones, setpoints	[K]
s	slack variables	discomfort measures	[K]
ξ	parameters	aggregate of the building states, references, and disturbances	[K]
Q	weighting factors	importance of particular objective	[-]
N	prediction horizon	predicted future time window	[-]
N_c	control horizon	optimized future time window	[-]

physical representations in buildings.

2.2. Objectives in building control

The objective, or also called cost function, represents the performance target to be minimized. When two or more targets are set, the problem is referred to as a multi-objective optimization. In such cases, the terms of the objective function are often conflicting and a trade-off among them has to be found. Common approaches for multi-objective optimization include *goal attainment*, *minimax*, and *Pareto front*.

Goal attainment In building control, the vast majority of MPC problems are using *goal attainment* formulations aiming to find a balance between weighted goals, such as energy use and thermal comfort of the occupants. This balance is typically adjusted by means of weighting terms to give priority to one of the targets. For example, Eq. (1a) can be re-written as (2). Where $\|Q_s s_k\|_2^2$ represents an arbitrary discomfort term in the form of the weighted squared 2-norm of the slack variables, and $Q_u u_k$ stands for the weighted linear energy term. The matrices Q_s and Q_u here represent the weighting factors, and κ_k is the time-varying factor representing, e.g. the weight associated with price or emission profiles. In human perspective, these weighting factors represent the “price” that the user is willing to pay to have more or less comfort. Besides standard weighting techniques, other methods to select the preferred objective have also been tested, such as lexicographic formulations which assume that the objectives can be ranked in order of importance (O’Dwyer, De Tommasi, Kouramas, Cychowski, & Lightbody, 2017).

$$\min_{u_0, \dots, u_{N-1}} \sum_{k=0}^{N-1} (\|Q_s s_k\|_2^2 + Q_u \kappa_k u_k) \quad (2)$$

Minimax Also called *Min-Max* formulations aim to minimize the worst-case values of a set of multivariate functions. *Minimax* objective functions are typically being used for finding conservative solutions to the optimization problems in the presence of uncertainties. More details on this class problems are provided in Section 7.2 dedicated to robust MPC. **Pareto front** Finds trade-off solutions in which an improvement in one objective requires a degradation in another. A generic review on MPC and PID design with Pareto front objectives was provided in Gambier (2008). Authors in Zhao, Shen, Li, and Bentsman (2017) demonstrated how to formulate and solve the preference adjustable multi-objective MPC for constrained nonlinear systems. The advantage of multi-objective MPC in the context of building control is that the resulting Pareto front solution space allows the user to choose the outcome according to his comfort preferences and economic constraints (Arendt et al., 2019; 2016; Ascione, Bianco, De Stasio, Mauro, & Vanoli, 2016; Ascione, Bianco, Mauro, Napolitano, & Vanoli, 2019; Li & Malakawi, 2016; Liu et al., 2013).

The formulation of the objective function is influenced by several factors, like building dynamics, type of the HVAC system, the level of detail of the controller model, observability and controllability of the system and user preference. For example, if only the building envelope is modeled, a classic approach is to minimize its heat inputs from the different heating and cooling systems, with each system having an associated cost (Picard & Helsen, 2018). In other approaches where the HVAC is explicitly modeled, setpoints of the components are usually manipulated to minimize the energy use (Jorissen, 2018). Although energy use and user comfort are the most frequently used objectives, it is possible and for some cases desired to have also different objectives like minimization of monetary costs, or greenhouse gases (GHG) emissions, maximization of the share of renewable energy sources (RES), and more. The following subsections elaborate more on different objectives used in the building control sector. Earlier reviews on MPC objective functions for building control can be found in Cigler, Široký, Korda, and Jones (2013b); Cupeiro Figueroa, Cigler, and Helsen (2018).

2.2.1. *Comfort satisfaction*

The main purpose of heating, cooling, and ventilation systems in buildings is to maximize the thermal comfort and indoor environmental quality (IEQ) for the occupants. Enhanced IEQ can improve occupants' productivity by 5 to 10% Olesen (2005), or satisfy the specific requirements of more demanding occupants like elderly people who in general prefer warmer thermal conditions (Schellen, van Marken Lichtenbelt, Loomans, Toftum, & De Wit, 2010).

In general, the main constituent of the IEQ is thermal comfort. The standard way to achieve thermal comfort is to maintain the zone temperatures of the building within a given temperature range or so-called comfort zone, e.g., as defined by the international standard ISO7730 (International Organization for Standardization, 2005). An advanced metric used to assess thermal comfort is the Predicted Mean Vote (PMV) indicator of Fanger (Fanger, 1973). PMV is used not only in the thermal comfort model of ISO7730 (International Organization for Standardization, 2005) but also in other standards like ASHRAE55 (American Society of Heating Refrigerating & Air Conditioning Engineers, 2013), EN15251 (Comite'Europe'en de Normalisation, 2007), and ISSO74 (van der Linden, Boerstra, Raue, Kurvers, & de Dear, 2006). PMV is a nonlinear model, which depends on various parameters like the metabolic rate, the clothing insulation, the indoor air temperature, the radiant temperature, the air velocity, the relative humidity, and on the outdoor meteorological conditions. However, its nonlinear nature makes it computationally more expensive for MPC applications (Castilla, Álvarez, Normey-Rico, & Rodriguez, 2014; Castilla et al., 2011), leading to the use of approximated versions of this model (Cigler, Prívarva, Vána, Žáčeková, & Ferkl, 2012; Klaučo & Kvasnica, 2014; Yang et al., 2018). The PMV value is moreover complicated to calculate in such a way that it fits the real observed mean vote (Humphreys & Nicol, 2002). On the other hand, some studies recommend an adaptive thermal model that involves acclimation of people, which could improve people's health by increasing their thermo-neutral zone (van Marken Lichtenbelt & Kingma, 2013). Standards including adaptive comfort bounds are defined by the thermal models in EN15251 (Comite'Europe'en de Normalisation, 2007), ASHRAE55 (American Society of Heating Refrigerating & Air Conditioning Engineers, 2013), and ISSO74 (van der Linden et al., 2006). A comprehensive comparison of adaptive thermal comfort models defined by different standards can be found in Sourbron and Helsen (2011). The main disadvantage of these personalized comfort metrics is that their parameters need to be properly measured or estimated, which often increases their cost and limits their applicability in control practice. For a more comprehensive overview of thermal comfort models, we refer the reader to Enescu (2017). Table 3 presents a compact summary of most common thermal comfort models used in MPC.

However, thermal comfort constitutes only a part of IEQ since it also depends on additional factors, such as indoor air quality (IAQ), lighting quality, visual and acoustic comfort. For example, evidence exists that mechanical ventilation systems lead to an overall improvement of the IAQ and reduction of reported comfort and health-related problems (Kephelopoulos, Geiss, Barrero-Moreno, D'Agostino, & Paci, 2016). To predict the air quality an occupancy model needs to be developed, e.g.,

Table 3
Selective summary of thermal comfort models used in MPC formulations.

Reference	Static model	Adaptive model	PMV	Others
Sturzenegger et al. (2013)	•	–	–	–
Oldewurtel et al. (2013)	•	–	–	•
Feng, Chuang, Borrelli, and Bauman (2015)	–	•	–	–
Maasoumy et al. (2014)	–	•	–	–
Castilla et al. (2014, 2011)	–	–	•	–
Freire, Oliveira, and Mendes (2008)	–	–	•	•

based on statistical data or available measurements (Jorissen, Boydens, & Helsen, 2017). The occupancy models can also be used to predict the thermal loads and thus improve thermal comfort, and when correctly implemented they can further save up to 30% of energy (Mirakhorli & Dong, 2016). Furthermore, ventilation units can have some degree of freedom with respect to the relative humidity of the supplied air, and therefore they can also be straightforwardly incorporated into MPC formulations via humidity models or additional constraints on temperatures (Freire, Oliveira, & Mendes, 2005). The lighting quality can be improved by utilizing blind control and electric lighting power control (Oldewurtel, Sturzenegger, & Morari, 2013). In general, based on the available sensors, the output vector y can include not only the temperature measurements but also CO₂ concentrations, humidities, illuminance, and others.

2.2.2. *Minimization of cost*

The minimization of the energy use in a building does not necessarily result in the minimization of the related operational costs. If for example, the energy prices are volatile, as it is the case for electricity, it may be worth to shift the load and store thermal energy during cheap periods for its later use when the energy prices are higher. This thermal energy can be stored in buffer tanks, geothermal borefields or by using the building's own thermal inertia. An economic objective can be formulated by transforming the energy use into monetary cost by means of a variable cost factor, (i.e., the term κ_k in Eq. (2)) which can be considered as a forecasted disturbance to the model.

The variability fuel prices (gas, oil, and wood) can be neglected because their dynamics is relatively slow, making the cost factor quasi-constant over the prediction horizon. These cost factors could be updated offline in the formulation when the price has a substantial change. Nonetheless, times are changing for electricity prices. The minimization of the monetary cost is equal to the minimization of energy in the cases where only electricity-based systems are used and the user has contracted a flat tariff. However, today, a wider variety of tariffs are being implemented with higher variability in both energy and peak demand prices. With the implementation of smart meters, even for the residential sector, it would be possible to access, e.g., hourly prices. Subsequently, using an economic objective has major potential if electricity-based supply systems such as heat pumps and chillers are used. The advantage of these objectives has been widely studied in the context of demand-response problems with real-time pricing (Avci, Erkoc, Rahmani, & Asfour, 2013; Bianchini, Casini, Vicino, & Zarrilli, 2016a). It has been shown that economic optimization could be used to reduce the peak electricity demand (Oldewurtel, Ulbig, Parisio, Andersson, & Morari, 2010b), or increase the stability, flexibility, and sustainability of the energy system, particularly in the face of growing intermittent renewable generation (Patteeuw, Henze, & Helsen, 2016; Qureshi & Jones, 2018). Examples of such a pricing-formulation are given by Bianchini, Casini, Vicino, and Zarrilli (2016b); Oldewurtel, Ulbig, Parisio, Andersson, and Morari (2010c); Vrettos, Lai, Oldewurtel, and Andersson (2013). A simulation study of different economic MPC formulations under commercial time-of-use tariffs concluded that multiple MPC formulations could offer the same value for the user (in terms of utility bill cost) but different grid service capabilities such as load shifting (Cutsem, Kayal, Blum, & Pritoni, 2019b).

2.2.3. *Minimization of greenhouse gas emissions*

This objective can be chosen if the user is motivated to reduce the carbon footprint of the building HVAC system. In contrast to the economic objective, the cost factor is replaced by an emission factor on the used energy amount. The minimization of GHG is equal to the minimization of energy in the cases where only conventional fossil energy sources are being used. The emissions for gas and oil boilers are proportional to the amount of combustible used. When electricity is supplied by a distributor who guarantees that it comes from the renewable electricity pool, the direct GHG emissions are zero. In this case,

minimization of GHG emissions is not possible. The emission factor differs from the cost factor when electricity-based components take the energy from a standard electricity supplier. The cost profiles usually do not coincide with the GHG emissions profile. The GHG emission factor varies with the distribution of the different generation system types active at the considered moment. These emission factors can be provided or estimated through generation schedules by the grid operators. Cases where this objective function is used, can be found in [Knudsen and Petersen \(2016\)](#); [Vogler-Finck, Wisniewski, and Popovski \(2018\)](#).

2.2.4. Maximization of the share of renewable energy use

In cases where the building has local RES, these terms can typically be added to the above formulations with a negative cost/emission factor, which would lead to their maximum usage. The formulation that maximizes the share of RES (or minimizes the share of fossil fuels) uses different weighting factors on different available energy sources. Moreover, when sufficiently large thermal energy storage capacity and accurate controller models are available, the MPC can harness the power of the predictions to maximize the use of intermittent renewable systems by storing the energy for later use into thermal mass or batteries. The abstract factor $\kappa_k = 1 - R_k$ in Eq. (2) is used as the time-varying factor, where R_k represents the share of renewable energy in the load at the moment k . Some examples that use this objective function are treated in [Vandermeulen, Vandeplas, Patteeuw, Sourbron, and Helsen \(2017\)](#); [Vogler-Finck, Pedersen, Popovski, and Wisniewski \(2017\)](#)

2.2.5. Optimization of multiple generation and storage components

Another prominent set of multi-term objectives is optimizing the use of multiple energy generation (eg., PV cells) and energy storage components. The objective here is to increase the energy efficiency and flexibility of the building stock by load shifting, the energy exchange between multiple buildings or storage units, and by prioritizing the use of cheapest, cleanest, or most efficient energy sources. For instance, MPC formulation increasing the flexibility of a commercial building with thermal energy storage (TES) in demand-side management (DSM) programs was evaluated in [Cao, Du, and Soleymanzadeh \(2019\)](#). Authors in [Tarragona, Fernández, and de Gracia \(2020\)](#) apply MPC in a heating system with TES, PV panels, and electricity grid supply and study the impact of different MPC settings on the energy cost performance. MPC formulation for extremely large central cooling systems with TES was introduced in [Shan, Fan, and Wang \(2019\)](#). [Kuboth, Heberle, König-Haagen, and Brüggemann \(2019\)](#) formulate economic MPC for a residential building with a coupling of thermal and electric supply by an air source heat pump. A simulation study of MPC formulation for a residential house leveraging local PV microgeneration was presented in [Godina, Rodrigues, Pouresmaeil, and Catalão \(2018\)](#). MPC formulation optimizing the coefficient of performance (COP) of a hybrid geothermal system with a borefield heat exchanger was presented in [Cupeiro Figueroa, Picard, and Helsen \(2020\)](#). in [Zhao, Lu, Yan, and Wang \(2015\)](#), an MPC formulation with multiple energy generation and storage components was tested on a real building.

2.2.6. Optimization of large-scale systems

In the case of large-scale commercial HVAC systems, the implementation of MPC as a single monolithic optimization problem is not practical nor desirable given real-time operating requirements ([Rawlings et al., 2018](#)). In these situations, decomposing the problem into a set of smaller problems presents a viable and practical alternative. A hierarchical decomposition of economic MPC in large-scale HVAC systems with district heating/cooling networks was applied and tested on a 500-zone campus in [Rawlings et al. \(2018\)](#). The energy hub concept allows optimizing a set of buildings in a cooperative manner, providing opportunities for load shifting, and sharing of costly energy generation and storage components, such as heat pumps, boilers, batteries ([Dariyanakis, Georghiou, Smith, & Lygeros, 2015](#)).

2.2.7. Long-term objectives

In general, it is difficult to incorporate the long-term dynamic effects of the system which exceed the defined prediction horizon N . Such problems arise, for example, in MPC applications with seasonal energy storage units, like underground thermal energy storage (UTES), large-scale storage tanks, etc. However, to avoid thermal depletion of these storage systems, a thermal balance should be ensured on the long-term. To this end, some authors ([Jorissen, 2018](#); [Verhelst, 2012](#)) have included a long-term cost in the objective function, that penalizes the use of the seasonal storage system at specific moments, thereby inviting the system to use the secondary production unit. However, this long-term cost could move from penalization objectives within the horizon towards shadow costs over a longer horizon. Since the accuracy of the predictions would decay over time, historical data may be needed to fit the predictions over longer horizons.

2.2.8. Design and tuning factors

In MPC there are several important setup and tuning factors, which can be considered as hyperparameters with a strong influence on the overall performance of the system. We summarize them in the following list:

Weighting factors Q : give the preferences to the multiple objectives to be penalized in the objective function.

Sampling period T_s : is the time interval in which the computed control actions remain constant, and the choice of it depends on the time constant of the controlled system.

Prediction horizon N : N are here the number of time steps and hence $N T_s$ defines the length of a time window for which MPC computes the predictions given by model (1b) and enforces the system behavior desired by the objective function.

Control horizon N_c : $N_c \leq N$ represents the number of time steps for which MPC computes the optimal control actions which minimize the given objective function. Hence the length of an optimized time window is given by $N_c T_s$.

[Fig. 3](#) provides a conceptual example of the characteristic MPC behavior for a building with a highlighted summary of design and tuning factors.

Weighting factors are usually determined based on magnitudes of the penalized signals, while the other parameters should be set up based on the dynamics of the controlled system. The practical rule is that T_s should be large enough for computing, communicating, and implementing the next control signal, though small enough to control the system in a stable way. The general rule in control theory is to choose T_s such that there are at least 10 to 20 samples in the rise time T_{90} of the process step response. Buildings are in principle slow dynamic systems with T_s usually spanning from 15min to 180min. In control theory, N should be large enough to cover the settling time of the process step response. N for building control applications usually spans between 5h to 48h ([Afram & Janabi-Sharifi, 2014b](#)). Typically, the control horizon is chosen such that $N_c \leq N$ and $N_c \geq 2$. For many practical applications, the rule of thumb is to set N_c roughly to 20% of N . The advantage of $N_c < N$ lies in reduced computational demands by having fewer decision variables in the resulting optimization problem ([Cagienard, Grieder, Kerrigan, & Morari, 2004](#)). The reason why $N_c \leq N$ is often used in practice is that the effect of the computed control actions decreases with each step in the future, which means that only the first few computed control actions have a major impact on the trajectory of the controlled variables. [Eq. \(3\)](#) serves as an example of such an objective function where $N_c < N$. The number of optimized variables is decreased from $n_{it}N$ to $n_{it}N_c$ which can significantly reduce the computational burden especially for problems with many control inputs.

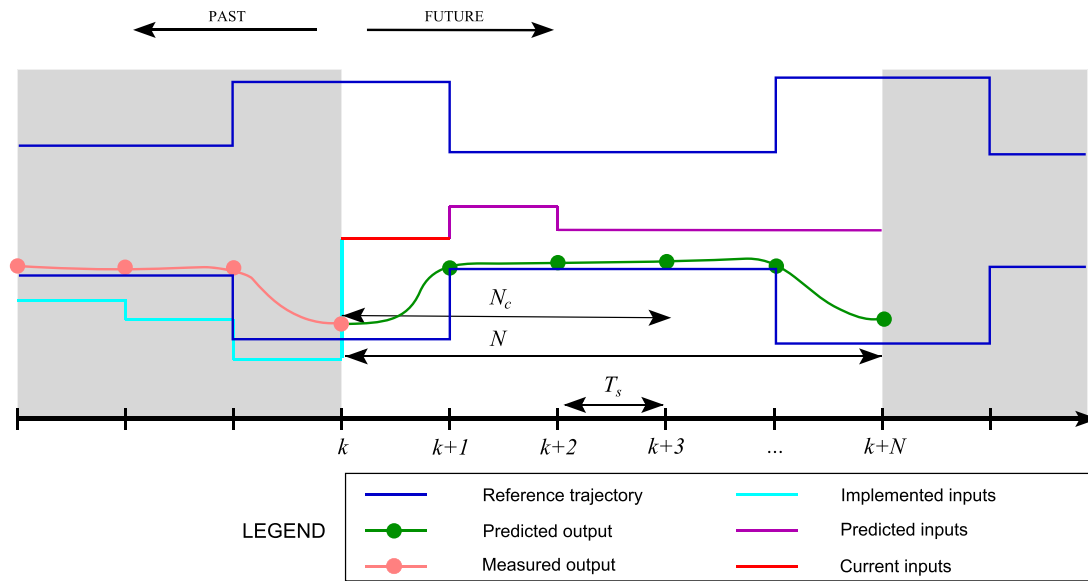


Fig. 3. Characteristic features and illustrated behavior of MPC for building temperature control.

$$\min_{u_0, \dots, u_{N_c-1}} \sum_{k=0}^{N-1} \|Q_s s_k\|_2^2 + \sum_{k=0}^{N_c-1} Q_u \kappa_k u_k \quad (3)$$

A common practical problem that can appear in poorly tuned MPC is an oscillatory behavior. If the weights are unbalanced and control constraints are not tight enough, the control actions can result in bang-bang control profiles, i.e., either idle (no energy) or deadbeat (full power) control actions. These oscillations can be corrected by properly balancing the weighting terms, e.g., based on the magnitudes and ranges of the penalized variables, or by introducing the rate of change or slew rate constraints on control inputs (Cigler et al., 2013b). Definitions and a discussion about different types of constraints that can be used for tuning the performance of MPC are given in Section 2.3.

Further reading with detailed overviews, comparisons, and strategies for selection of an appropriate objective function and tuning parameters for MPC-based building control can be found, e.g. in Afram and Janabi-Sharifi (2014b); O’Dwyer et al. (2017); Rincón, Santoro, and Mendoza (2016); Serale et al. (2018). Please note that different mathematical formulations of the objective function can lead to different MPC problem classes with varying solution complexity and computational demands, which is further discussed in Section 5.

2.3. Constraints used in building control

MPC can handle a wide variety of constraints on state, input or output variables (Maciejowski, 2002). In general, there are two types of constraints: inequality (control inputs range, comfort zones, etc.) and equality (building model dynamics, rate limits, etc.) constraints. *Hard constraints* are those for which satisfaction is mandatory. An example of such constraints is the state update equation given by the equality constraint (1b), or control action bounds (4), which need to be satisfied at every time instant for the whole prediction horizon.

$$u \leq u_k \leq \bar{u} \quad (4)$$

Soft constraints, on the other hand, are those for which violations can occur. They are usually relaxed by slack variables s_k that are added to and penalized in the objective function (1a). Soft constraints commonly used in building control are thermal comfort zone inequality constraints given by (5), defined by upper \bar{y}_k and lower bounds \underline{y}_k on the controlled variable y_k . For these types of constraints, the softening may be necessary to avoid infeasibility of the optimization problem during the time

periods when the comfort constraints will be violated, as can happen in practical implementation. In general, soft constraints are preferable due to numerical reasons that guarantee the feasibility of the resulting optimization problem.

$$\underline{y}_k - s_k \leq y_k \leq \bar{y}_k + s_k \quad (5)$$

Another type of constraints consist of *time-varying constraints*, which in contrast to constant constraints, vary in time. Eq. (5) represents such constraints because comfort boundaries are defined as time-varying parameters \bar{y}_k and \underline{y}_k . *Slew rate constraints* penalize the rate of change of certain variables, for example Eq. (6) limits the one-step difference of the control variable u_k . This type of constraint is useful for avoiding overshooting and peak behavior.

$$\Delta u_k = u_k - u_{k-1} \quad (6a)$$

$$\Delta u \leq \Delta u_k \leq \overline{\Delta u} \quad (6b)$$

Move blocking constraints represent a formulation strategy for decreasing the computational burden by reducing the number of decision variables of the resulting optimization problem, as discussed in the Section 2.2.8. The basic idea is based on reducing the degrees of freedom by fixing the control variables or its derivatives to be constant over a defined time-period (Cagienard et al., 2004). See for example Eq. (7).

$$u_k = \begin{cases} u_k & \text{if } k \leq N_c \\ u_{N_c} & \text{otherwise,} \end{cases} \quad k \in \mathbb{N}_0^{N-1} \quad (7)$$

Terminal constraints penalize the last predicted state to stay within a given terminal region: $x_N \in \mathcal{X}_N$. They are usually used for enforcing the

Table 4
Summary of constraints types used in MPC for buildings, inspired by Serale et al. (2018).

Form	Violations	Time	Math	Variables	Meaning
Inequality	Soft	Varying	Linear	States	Model dynamics
Equality	Hard	Constant	Nonlinear Mixed-integer	Outputs Inputs	Ranges Slew rate
					Move blocking Terminal

stability and recursive feasibility of the OP (1) with respect to the controller model.

From a practical perspective in building control applications, the constraints are most commonly used to enforce selected variables to stay within given ranges, e.g., heat fluxes and room temperatures (Picard, Drgoňa, Kvasnica, & Helsen, 2017), supply air temperatures (Rehrl & Horn, 2011), airflow rates (Huang, 2011), and other HVAC variables (Afram & Janabi-Sharifi, 2014b), or for tuning the MPC performance via, e.g., slew rate constraints on control variables (Cigler et al., 2013b). From a mathematical point of view, the constraints can be further classified as linear, nonlinear or mixed-integer. The latter two can lead to better performance but also result in an increased complexity of the optimization problem. Table 4 compactly summarizes the constraint types discussed in this section. The influence of the constraints on the type and complexity of the OP is discussed in Section 5 in more detail.

3. Building models for control

The main bottleneck in practical implementation of MPC is the controller model development (Cigler et al., 2013a). Naturally, the quality of the MPC solution relies on the model accuracy, but also the overall MPC implementation is affected by the chosen modeling approach in a number of ways. Efficient optimization algorithms utilize specific model characteristics, like linearity, continuity, or known derivatives. However, the phenomena and processes occurring in buildings are often nonlinear and discontinuous, and complex physical models or advanced data-driven models are required to model such processes accurately. On the other hand, more complex models increase the overall computational demand of MPC, not only by increased simulation time but also because they are not suitable for efficient optimization algorithms and gradient-free algorithms have to be used instead. Therefore, a sound trade-off between the model accuracy and simplicity is required. This section provides an overview of the building model types, three modeling paradigms used in building modeling, as well as practical aspects of building modeling.

3.1. Building model types

This section elaborates on individual components of a generic building model structure, as shown in Fig. 4, and summarized by hybrid differential algebraic equations (DAE) with continuous and discrete time dynamics (1b)–(1d). These components are the building envelope, HVAC system, sources of disturbances such as weather and occupancy, and the peripherals represented by sensors and actuators.

3.1.1. Envelope

A building envelope consists of the external and internal walls, roofs, windows, ground floors, and other partitions separating the indoor environment from the outdoor environment, or two indoor thermal zones. In general, the building envelope model should take into account

the heat transfer through conduction, radiation (especially solar gains), and convection (especially air infiltration). The conductive heat transfer depends on the thermal resistance of the partition and on its thermal mass. Heavier materials, e.g. brick or concrete, have higher thermal mass (inertia) and can absorb more energy, effectively working as a thermal buffer between the indoor and outdoor environments. This buffer can be utilized in MPC to shift energy use. Lighter materials, e.g. wood or thermal insulations, have low thermal mass resulting in a lower potential for accumulating energy. On the other hand, lighter materials have lower conductivity and therefore increase the thermal resistance of the partition. The radiative heat transfer from solar gains has to be taken into account in the case of transparent partitions (windows, curtain walls), but is often considered also for opaque partitions (heating building thermal mass). The transparent partitions have low thermal mass and are often modeled using the steady state heat equation.

The conduction in building envelopes is typically modeled using the 1D transient heat equation (Clarke, 2001; Hensen & Lamberts, 2019), converted to a system of ordinary differential equations using for example the method-of-lines, whereas the radiation and convection modeling approaches vary extensively. For example the radiative heat transfer can be modeled with anything from a simple solar heat gain coefficient to a complex dynamic shading model. In contrast, data-driven models typically do not consider the building envelope directly. Instead, they model indoor environment as a function of indoor and outdoor disturbances, and therefore the effect of building envelope is taken into account implicitly (Arendt, Jradi, Shaker, & Veje, 2018a).

3.1.2. HVAC

Building HVAC systems vary greatly in designs, however some of the commonly used components are as follows: boilers, heat pumps, chillers, fans, filters, pumps, dampers, valves, heat exchangers, diffusers, ducts, and pipes. There is a vast spectrum of controls regulating the fluid flow, supply temperatures, and indoor air conditions.

HVAC components and controls coupled to building envelope models are challenging to simulate while maintaining reasonable computational demands (Jorissen, Wetter, & Helsen, 2018d). Fans and pumps have nonlinear characteristics (Wetter, 2013), which are coupled to nonlinear relations of mass flow rates and pressure differences in the system caused by active components such as valves and dampers and static components such as ducts and pipes. Excluding computationally expensive modeling approaches such as Computational Fluid Dynamics (CFD) that may be prohibitive from an MPC point of view, the final model structure highly depends on phenomena and processes the model has to cover. For example, in some cases MPC does not control all HVAC components directly and instead controls high-level setpoints. In such cases the model may have to include some embedded controls, which also can be nonlinear or even discrete (e.g. on/off), or assume ideal, instantaneous control.

3.1.3. Disturbances

Disturbances refer to every non-controllable input that has an influence on the building system. Some examples are weather conditions (e.g. outdoor temperature, solar radiation), internal heat gains (e.g. occupancy, equipment), and electricity prices. The weather conditions are simply inputs to the building simulation and so just an accurate forecast is required (no feedback). The easiest way to obtain it is from some online weather forecast service (many free and commercial are available), typically providing forecasts based on advanced climate models. A potential drawback of this approach is that the online weather forecast is typically generated based on data from climate stations which can be far from the considered building, and may not represent the actual weather conditions for the building. Alternatively, machine learning models can be employed and trained on the data collected from the building site, if available. The machine learning models can be especially accurate for short-term predictions, up to several hours ahead (Wollsen & Jørgensen, 2015), which is the range typically relevant for

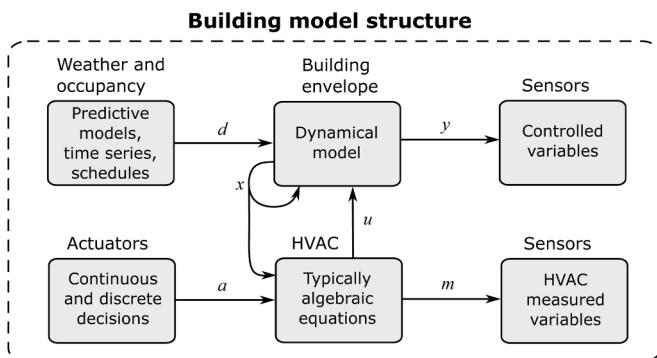


Fig. 4. Generic structure of a building model.

MPC in buildings.

The most straightforward approach for including weather forecast in the prediction model of MPC is based on a data-driven linear dynamics model of the weather variables (Oldewurtel et al., 2012; Průvara, Široký, Ferkl, & Cigler, 2011), representing a cost-effective alternative to sophisticated simulation models or costly weather stations (Hedegaard, Pedersen, Knudsen, & Petersen, 2018). However, in some cases, linear dynamics might not be sufficiently accurate and can result in hampering the performance of the predictive controller (Kim, Witmer, Brownson, & Braun, 2014). In case of inaccurate disturbance models, stochastic (Drgoňa, Kvasnica, Klaučo, & Fikar, 2013; Parisio, Fabietti, Molinari, Varagnolo, & Johansson, 2014) or adaptive (Mazar & Rezaeizadeh, 2020) data-driven methods have been applied for mitigating the uncertainties associated with the weather forecast errors. For instance, (Liu, Paritosh, Awalgaonkar, Bilonis, & Karava, 2018) use a probabilistic time-series autoregressive model to quantify solar irradiance uncertainty. However, the disadvantage of data-driven correction methods is that the underlying disturbance distributions are often poorly represented based purely on historical data. Authors in Darivianakis, Georghiou, Smith, and Lygeros (2019) address this issue by exploiting the historical data to construct families of distributions based on real weather data, and construct a first-order model for weather prediction error.

The indoor occupancy can be modeled either as the heat gain profile, presence (room empty vs. at least one person in the room), occupant count, or occupant count and behavior. The latter approach is the most accurate, since building occupants not only generate heat, but also interact with the building, sometimes taking actions to adjust the indoor environment (window opening, overriding default setpoints). However currently, the state-of-the-art occupancy behavior models (e.g. obFMU Hong, Sun, Chen, Taylor-Lange, & Yan, 2016 or StROBe Baetens & Saelens, 2016) are computationally too expensive to be included in MPC. Therefore, less computationally demanding approaches are typically adopted in the context of MPC, for example models based on heuristics (e.g. anticipated schedules) or machine learning. Reviews in Balvedi, Ghisi, and Lamberts (2018); Yan et al. (2015) provide in-depth discussion on current methods of monitoring and modeling occupant behavior suitable for real-time control applications. For more comprehensive and systematic literature review of models for occupant behavior we refer the reader to (Carlucci et al., 2020). One of the popular data-driven models for the occupancy behavior are Markov chains processes, providing systematic framework for evaluating accurate scenarios for human-building interaction suitable for integration in scenario-based MPC formulations (Johnson, Starke, Abdelaziz, Jackson, & Tolbert, 2014). Sangogboye et al. (2017) presented data-driven occupancy prediction methods with average errors of 7% and 3% for passive infrared (PIR) sensor and stereovision camera training data, respectively. Peng, Rysanek, Nagy, and Schlüter (2018) incorporated data-driven occupancy models to optimize rule-based control in a real building and reported 7–52% energy savings, depending on the room type. Capozzoli, Piscitelli, Gorrino, Ballarini, and Corrado (2017) reported 14% energy savings through a pattern-recognition analysis of occupants' displacement. The most accurate occupancy predictions are yielded by models trained on dedicated-sensor data (PIR, cameras), however occupancy can be also predicted from other sensors, such as CO₂ or plug power (De Coninck & Helsen, 2016; Jorissen et al., 2017; Sangogboye et al., 2017).

Purely theoretical studies of the building dynamics using detailed white-box models can often include dozens sometimes up to a hundred of disturbance signals (Picard et al., 2017). However, for most practical applications, it is sufficient to take into account only a small subset of dynamically dominant disturbances. Authors in Drgoňa et al. (2018) used feature extraction based on principal component analysis (PCA) to select the most significant disturbances for residential building control, selecting the ambient temperature and solar irradiation. Similarly, (Lambrechts, 2020) studied the impact of the weather and occupancy

uncertainties on MPC's performance, finding that uncertainties associated with the forecast of ambient temperature, solar irradiance, and internal heat gains have the largest impact on the performance of the predictive controller. Some intuition on the selection of dynamically dominant disturbances in specific cases can be derived from the studies above. However, systematic theoretical studies and practical guidelines for selecting dominant disturbances in a wide range of building model types, climate zones, and types of use are lacking in the current literature.

3.2. Modeling paradigms

This section provides an introduction to the three modeling paradigms used in building modeling and discusses their applicability to MPC. For a more extensive review on the modeling techniques used in HVAC control we refer to Afroz, Shafiullah, Urmee, and Higgins (2018). Additionally, a broad comparative study between the different modeling paradigms can be found in Boodi, Beddiar, Benamour, Amirat, and Benbouzid (2018).

3.2.1. White-box

White-box models describe the building dynamics from physical knowledge. They are based on the principles of heat transfer and conservation of energy and mass. The parameters of these models are physically meaningful and are obtained from the building technical documentation regarding geometry, material properties, and equipment specifications. For a detailed description of the equations that are considered in this modeling approach we refer to Jorissen et al. (2018c).

The main challenge in white-box modeling is the significant effort required to describe the building properties. Despite the advances in Building Information Modeling (BIM), this process is still largely manual and tedious (Gao, Koch, & Wu, 2019). The resulting model typically includes hundreds or, more likely, thousands of parameters. Hence, there are many potential sources of model inaccuracy, making the process of parameter setup difficult. With available measurement data, calibration may be used to tune the selected parameters. However, a model calibrated based on the overall yearly energy use might be still inaccurate for predicting performance on the individual zone level (Arendt et al., 2018a) or at smaller timescales. Moreover, the large number of equations and their nonlinear nature makes the implementation of white-box MPC more difficult.

On the other hand, when the parameters of the white-box models are accurate, their physical properties endorse them with highly reliable results. They can also track the evolution of physically meaningful variables. As a consequence, they are often considered suitable for fault detection (Henze, 2013) and building monitoring systems (Jradi et al., 2018). In addition, detailed building envelope and HVAC models can also enable control of the building at a more granular level, since the optimization variables may have a direct translation into the signals used in the actuators. This direct control skips the development of any sub-controller, which can be a cumbersome task, and it also increases the overall MPC performance due to direct assessment of HVAC efficiencies. For these reasons, research has been conducted to facilitate the implementation of these models into optimal control. As a result, tool-chains have been developed to define white-box models for buildings and couple them with or translate them into an optimization problem. Coupling has traditionally taken the form of using a building energy simulation program within iterations of a numerical optimization technique, such as a scheme that couples EnergyPlus and a particle swarm optimization algorithm (Corbin, Henze, & May-Ostendorp, 2012) or one that couples TRNSYS and a genetic algorithm (Coffey, Haghhighat, Morofsky, & Kutrowski, 2010). However, these schemes can be computationally expensive, especially as the number of optimization variables grows and complexity of the model increases, and prone to convergence issues (Wetter, 2004; Wetter & Wright, 2004). Wetter, Bonvini, and Nouidui (2016) argued that equation-based languages,

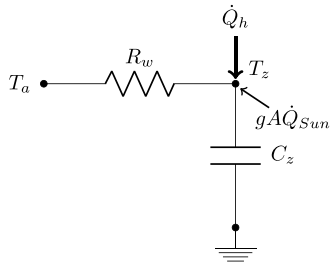


Fig. 5. Example of an RC building envelope model.

such as Modelica, can address some of the limitations of traditional building energy modeling software tools, such as EnergyPlus, specifically by (a) enabling symbolical manipulation of model equations and by (b) separating the model definition from the numerical solver. For instance, some of the most prominent Modelica libraries for building modeling are Buildings (Wetter et al., 2014), IDEAS (Baetens et al., 2015), AixLib Müller, Lauster, Constantin, Fuchs, and Remmen (2016), and BuildingSystems (Nyttsch-Geusen et al., 2016). Jorissen et al. (2018a) implemented and validated an automated toolbox for automatically parsing white-box models written in Modelica into MPC, showing the feasibility of this approach. A detailed overview of the available software tools is presented in Section 8.1.1.

3.2.2. Gray-box

The gray-box category represents a wide spectrum of models encompassing simplified physical relationships, but also requiring parameter estimation based on measured data. Usually, the physics in gray-box models is simplified by means of state space dimensionality reduction or linearization. A typical concept in gray-box modeling is the RC analogy that defines any model by its affinity with a resistor-capacitor electrical circuit as the one shown in Fig. 5. This very simple example represents the model of a building envelope where C_z is the thermal capacitance of the zone which can be seen as the capacity of a zone to store thermal energy. The thermal resistor R_w represents the building walls that separate the ambient temperature T_a from the zone temperature state T_z . Finally, \dot{Q}_h and $gA\dot{Q}_{sun}$ represent the thermal power from the building heating system and the solar irradiation, respectively. From this scheme it is possible to derive the equations that define the simplified physics of the system. For a one state RC model as the one shown in the example, the only equation defining the model is shown in Eq. 8. In this way, the model can be represented using state-space matrices by carefully grouping the parameters in the elements of the matrices for the specified inputs and outputs. An alternative formulation, called inverse comprehensive room transfer functions (iCRTF), is derived from discretization of the state space formulation and creation of linear transfer functions, whose coefficients can be identified based on regression (Armstrong, Leeb, & Norford, 2006). Such an approach has been used in simulation (Blum, Xu, & Norford, 2016; Zakula, Armstrong, & Norford, 2014) and experimental (Gayeski, Armstrong, & Norford, 2012) studies.

$$C_z \frac{dT_z}{dt} = \frac{T_a - T_z}{R_w} + \dot{Q}_h + gA\dot{Q}_{sun} \quad (8)$$

For buildings, model order reduction has proven to be able to maintain the same level of accuracy even when strong simplifications are carried out (Picard et al., 2017). This enables the use of more suitable models for optimization without any expected loss of controller model performance. It is often argued that the gray-box approach can address the limitations of both white- and black-box models. First, since some knowledge about the modeled system is already *hardcoded* in the model equations, gray-box models are more likely to stay reliable outside the calibration range than black-box models (Afroz et al., 2018), they require less data for calibration (Arendt et al., 2018a), and there is a

lower risk of overfitting than in black-box models. Second, the equations used in a gray-box model can be more easily adapted to the needs of MPC solvers, e.g. by ensuring continuity, linearity or differentiability. Finally, gray-box models are found to be easily portable between similar systems. For instance, (Reynders, Diriken, & Saelens, 2014) argued that only few model types are required to represent the majority of buildings. Verhelst (2012) showed low-order models provide similar accuracy to higher order models for both building and borehole heat exchanger modeling. It was concluded that the quality of the measured data has higher impact on the accuracy of the model than the model structure itself. A direct comparison of the gray- and white-box approaches for their application in MPC can be found in Picard et al. (2016). In this case, the white-box MPC resulted in a better thermal comfort and used only half of the energy used by the gray-box MPC.

The main challenge related to gray-box modeling is the need for a robust parameter estimation method. The approaches can be divided into batch and online estimation. The batch estimation is an offline process in which model parameters are found by minimization of the model error over a specific time period. The estimation can be performed only once or the models can be periodically recalibrated based on more recent data. Typically, the batch estimation leads to a non-convex optimization problem with many local and flat optima as shown by Arendt, Jradi, Wetter, and Veje (2018b). The complexity of the objective function can also bring the parameters to the constraint boundaries. Therefore, there is a need for a global optimization strategy, either by using evolutionary methods as in Arendt et al. (2018b) or multi-start methods as in De Coninck et al. (2016). In addition, an expert involvement and cross-validation of the parameter estimation results is advised (Verhelst, 2012). The online estimation is usually based on methods related to recursive Bayesian estimation, such as sequential Monte Carlo Rouchier, Jiménez, and Castaño (2019) or Kalman filtering (Shi & O'Brien, 2019). Online parameter estimation forms the basis of indirect adaptive MPC approaches, which are covered in more detail in Section 7.4.

Finally, unlike many data-driven models which usually perform better when trained on more data, gray-box models often require special care regarding the data chosen for training. For example, parameter estimation in an RC (resistor-capacitor) thermal network may lead to an overestimated thermal mass if the training period is too long and the gray-box model cannot find a good fit for the entire period (Arendt et al., 2018a). Blum et al. (2019b) found that the optimal training period length depends on the MPC horizon, suggesting that a periodic re-calibration is necessary.

3.2.3. Black-box

Black-box models learn the dynamics of the buildings from the measured data without making any prior assumptions regarding any physical relationships. The main advantages of the black-box approach compared to gray- and white-box are that they usually lead to lower development cost and that any signal can be used as an input or output, as there are no physics involved. On the other hand, black-box models require more training data than gray-box models (Afroz et al., 2018) and are not reliable outside the training range (Afram & Janabi-Sharifi, 2014a).

Linear models The simplest and most intuitive black-box models are the parametric linear models. The forecasts of these models are linear in the observations and the uncertainty increases with the prediction horizon. The models that belong to this group are Auto Regressive (AR), Auto Regressive with eXogenous inputs (ARX), Auto Regressive with Moving average or Box-Jenkins (ARMA or BJ), Auto Regressive with Moving Average and eXogenous inputs (ARMAX) and Output Error (OE). All these models can be transformed into the general state space formulation. A common alternative for estimating the state space parameters is the Subspace-based State Space System IDentification (4SID) (Van Overschee & De Moor, 1996). With this methodology, the state sequence and its order are first calculated, and later the state space

matrices are estimated just by solving a least squares problem. A comparison between subspace identification and an ARMAX model for their use within the MPC of a large building was made in Ferkl and Široký (2010). The authors concluded that subspace identification is faster, easier to implement and more accurate. This conclusion is corroborated by Prívarva et al. (2011). Finally, a number of MPC relevant identification methods exist, which aim to minimize multi-step ahead prediction error in relation to the MPC optimization horizon, such as the MRI+PLS method introduced by Prívarva, Cigler, Vaňa, Oldewurtel, and Žáčeková (2013b).

Parametric nonlinear models The parametric nonlinear models provide a nonlinear relation between the inputs and outputs of the model and have a non-monotonous increase of their uncertainty over the prediction period. Artificial Neural Networks (ANN) (Hagan, Demuth, & Beale, 1996; Jiang & Wang, 1999; Siegelmann & Sontag, 1995) are probably the most renowned models of this type. Some building implementation examples of these models can be found in Dodier and Henze (2004); Huang, Chen, and Hu (2015); Kusiak and Xu (2012); Ruano, Crispim, Conceicao, and Lucio (2006); Tang and Wang (2001). Some researchers have shown that these models perform better and more accurately than physical models (Arendt et al., 2018a; Ruano et al., 2006) and other forms of statistical models (Mustafaraj, Lowry, & Chen, 2011). However, Huang, Chen, and Hu (2014) state that the application of ANN for model predictive control on real commercial buildings is still challenging because it has a complicated structure, which results in non-convex optimization problems that are hard to solve. The latest advances in convexification of neural network modeling may provide a remedy. The use of convex ANN in optimal control of the building HVAC system demonstrated a performance improvement compared with classical linear models (Chen, Shi, & Zhang, 2018).

Nonparametric nonlinear models The nonparametric models, like k-Nearest Neighbors (kNN), Support Vector Machines (SVM), Decision trees (DT), and Random Forest (RF), do not make strong assumptions about the model structure. Therefore, these models can learn generic function mapping between inputs and outputs. The main drawbacks of these modeling approaches are the larger data requirements and the higher risk of overfitting. Control-oriented building models based on regression trees and random forests have been presented in Jain, Behl, and Mangharam (2017a); Jain, Smarra, and Mangharam (2017b); Smarra et al. (2018).

Gaussian Processes (GP) are particularly powerful nonparametric stochastic models, which has been recently used to model building dynamics. They capture the model uncertainty directly, providing a distribution of the predictions of the model, and enable the use of prior knowledge in the system identification process. Moreover, a comparison of four data-driven methods for building energy prediction concluded that GP are accurate and highly flexible (Zhang, O'Neill, Dong, & Augenbroe, 2015). Short- and long- term building energy consumption forecasts using GP were investigated in Noh and Rajagopal (2013). More examples of GP-based models in the building modeling context can be found in Abdel-Aziz and Koutsoukos (2017); Ahn, Kim, Kim, Park, and Kim (2015); Jain, Nghiem, Morari, and Mangharam (2018); Nghiem and Jones (2017).

3.3. Practical aspects of building modeling

3.3.1. Data acquisition and processing

Special care should be taken with data sets used for training data-driven models because poor data may not capture the main dynamics of the system. The data can be obtained from a detailed model or from actual measurements. The first approach is interesting for research purposes since different types of excitation signals can be applied at no cost. The drawback is that a reliable model is required. The second approach is more suitable for real applications. However, when using real measurements, the input excitations for obtaining rich training data are limited by the technical and operational constraints of the available

HVAC systems.

Design of Experiments assesses which excitations provide the most useful data. When the objective is to build a model suitable for control, the generated inputs do not need to cover the entire frequency domain, but rather some control-relevant selection. Therefore, the sampling time should be chosen based on the time constants of the building, with a typical range for building systems between 5min to 60min. In system identification of building systems, usually Pseudo Random Binary Signals (PRBS) and normal operation (business as usual) signals are used to generate the training data sets. The former is probably the most appropriate signal to provide rich data (Ljung, 1999), while the latter is used to avoid the extra costs of the experiments, as well as the risks of discomfort and the need for technical support. Case studies of building system identification using PRBS as input signals are (Bacher & Madsen, 2011; Hazyuk, Ghiaus, & Penhouet, 2012; Madsen & Holst, 1995; Prívarva et al., 2011; Royer, Thil, Talbert, & Polit, 2014), while examples of cases that used normal operation are (Berthou, Stabat, Salvazet, & Marchio, 2014; Ferkl & Široký, 2010; Reynders et al., 2014; Verhelst, 2012). Although a lot of system identification studies have already used data from normal operation, this data is usually insufficiently informative to reliably estimate a model (Prívarva et al., 2013). This is because during normal operation only a small part of the possible HVAC range is used. Consequently, the other operating conditions remain unexplored in the data and cannot be learned. Jain et al. (2018) proposed a method for optimal experiment design based on maximizing information gain or variance with a faster learning rate than using uniform random sampling or PRBS. This method reduced the required training period up to 50%, but was tailored for black-box Gaussian Processes.

There exist different indicators to check the quality of the obtained data. The most commonly used are the signal-to-noise ratio. This ratio is proportional to the amplitude of the response of the output to the excited input, and inversely proportional to the amplitude of the response to modeled disturbances and to measurement noise. The measurement length is also important and it should be at least larger than the largest time constant of the system. The minimum sampling time period should be defined by the Nyquist criterion, but in practice, a smaller sampling time is advised. Obviously, missing data-points should be avoided, although it is a common issue in building management systems. Filtering and re-sampling the data can not only overcome this threat, but can also help in the modeling process by smoothing the data to get rid of the measurement errors and other fast dynamics that may be blurring the main dynamics.

3.3.2. Model validation

The main purpose of the validation process is to ensure that the identified model is reliable not only within the training conditions, but also beyond. For this purpose, the data is normally divided into two sets: 1) a training set and 2) a validation or test set. The training set is used to tune the parameters of the model, while the test set simulates the trained model to check whether it captures the real behavior of the building when using different data than that used in the training.

There exist different statistical tests to validate a model. One example is the analysis of the residuals which are defined as the differences between the measurements and the outputs of the model given as $e_k = y_k - m_k$. Here, e_k , y_k and m_k are the residual, the model output and the measurement at time k , respectively. These residuals should be white-noise in the training data to ensure that all systematic dynamics are captured within the model. Any correlation in the residuals would indicate that the model can be improved further. Another option to test the performance of a model are the typical t-tests for checking the significance of the parameters, and the maximum likelihood tests for comparing the goodness of fit of two statistical models.

Many statistical indicators exist, such as the n-step ahead prediction error, the Root Mean Square Error (RMSE), the Continuous Ranked Probability Score (CRPS), the Expected Error Percentage (EEP), the Coefficient of Variation (CV), the Mean Biased Error (MBE) or the R^2

(also called coefficient of determination or fit). However, these indicators do not provide information about the control performance of the model, but instead about the simulation errors. Therefore, their interpretation has to be taken carefully. The statistical indicator choice depends on the desired highlights to put forward when analyzing the model. The RMSE, for instance, provides a symmetric and absolute score for model error over a period of time facilitating the comparison of different models. The CRPS is used for stochastic models and is defined as the mean root squared value of the difference between the cumulative distribution function of the forecast and the cumulative distribution function of the observations. The CRPS in probabilistic forecasting is the analogous key performance indicator to the RMSE in deterministic forecasting. In some cases, mainly for tuning purposes, it may be interesting to investigate the direction of the bias of the model. In such cases, metrics that indicate the direction of the bias like the MBE should be used. Alternatively, a box-plot with the n-step ahead prediction error can be used. Finally, the CV, EEP and R-squared indicators show relative values for the evaluation of the residuals.

3.4. Concluding remarks of building modeling

Modeling is one of the main bottlenecks for implementing MPC in buildings. White-, gray- and black-box modeling are three different paradigms used in practice. The choice of a particular paradigm mainly depends on the available resources and possibly on additional requirements, such as transferability between buildings and systems, high accuracy, smoothness (required by some optimization solvers), or reliability (generalization capabilities) (Fig. 6). If detailed technical documentation and physics-based modeling expertise are available, then it may be preferable to follow the white-box approach, as it leads to reliable and interpretable models with little requirements on the sensor data amount and quality (Afroz et al., 2018). On the other hand, if extensive reliable measurement data is available, the black-box approach provides models which are often more accurate and easily transferable to different buildings and systems, reducing the implementation time (Afroz et al., 2018). In industry, there is a trend towards data-driven modeling approaches as they can be more easily automated. Finally, if information about the building and HVAC design is available as well as some historical measurements, the gray-box approach may be the most convenient, as it shares many features of white- and black-box paradigms (Afroz et al., 2018). In any case, it is strongly recommended to carry out an exhaustive model validation to ensure good MPC performance.

Table 5 shows some examples of building modeling applications for optimal control that have been classified by the building system size, real implementation, modeling paradigm, the excitation input signal in the training data, the training data period, the modeling tool used to estimate the model parameters, and the model complexity regarding the number of thermal zones and the number of states in the model. BaU

stands for business as usual and refers to the standard operation of the building without any additional excitation. Finally, the hyphens indicate that the attribute does not apply to that type of model or that such characteristic is not specified in the reference. Notice that a more elaborate list of modeling tools is provided in Section 8.

4. MPC algorithms and methods

After building modeling and MPC problem formulation, designing and tuning the algorithmic implementation is the next step to take on the path towards real life operation of the building. This section summarizes key algorithmic principles and methodologies which are being used to implement and solve MPC problems in practice.

4.1. Receding horizon control

Typically, the MPC algorithms are being implemented in closed-loop using the principle of *receding horizon control* (RHC), defined by Algorithm 1. Here, the prediction horizon N keeps being shifted forward, with the controller implementing only the first step of the computed control strategy and discarding the rest, as described in Step 3. The algorithm introduces feedback into the system in the first step of Algorithm 1, where, at each time step, it corrects the deviations of the prediction from reality by updating the initial conditions of the system with measurements or estimates of the system parameters.

4.2. State estimation

Successful application of MPC relies on accurate information about the state variables to be used by the controller model for predictions. However, in most of the building control applications, measuring all state variables is not possible and state estimation algorithms need to be used as an integral part of the MPC system. By definition, the state estimator is an algorithm that provides an estimate of the internal states of a real system, from the measurements of its inputs and outputs. There are many distinct state estimation algorithms. The suitability and performance of each depends on the type of the observed system, nature of the disturbances, and availability and accuracy of the prediction model. A comprehensive review of the different state estimators in the context of process control can be found in Ali, Hoang, Hussain, and Dochain (2015).

The nature of the building’s dynamics allows us to use several assumptions to simplify the selection and design of the appropriate state estimator. First, the building envelope model can be accurately described by the linear dynamics (9):

$$x_{k+1} = Ax_k + Bu_k + Ed_k + w_k, \tag{9a}$$

$$y_k = Cx_k + Du_k + v_k, \tag{9b}$$

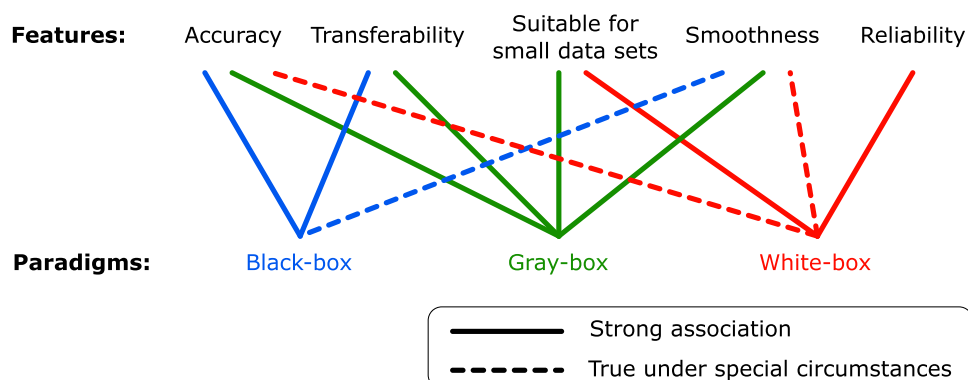


Fig. 6. Summary of the often cited features of the three modeling paradigms (based on Afram and Janabi-Sharifi (2014a); Afroz et al. (2018)).

Table 5

Sample of building modeling applications for optimal control categorized by building size, real implementation, modeling paradigm, modeling tool, input data, training period, number of zones and number of states.

Ref.	Building size [m ²]	Real impl.	Modeling paradigm	Modeling tool	Input data	Training period [days]	#Zones	#States
May-Ostendorp, Henze, Corbin, Rajagopalan, and Felsmann (2011)	1750	–	White	EnergyPlus (Crawley et al., 2001)	–	–	11	–
Corbin et al. (2012)	46,320	–	White	EnergyPlus (Crawley et al., 2001)	–	–	15	–
Drgoňa et al. (2020)	3760	–	White	Modelica Lin. (Picard et al., 2015)	–	–	12	700
Picard and Helsen (2018)	10,135	–	White	Modelica Lin. (Picard et al., 2015)	–	–	10–20	941
Jorissen and Helsen (2019)	150	–	White	Modelica (Baetens et al., 2015; Wetter et al., 2014)	–	–	9	330
Jorissen et al. (2018b)	2232	–	White	Modelica (Baetens et al., 2015; Wetter et al., 2014)	–	–	27	1262
Jorissen (2018)	10,000	–	White	Modelica (Baetens et al., 2015; Wetter et al., 2014)	–	–	32	1151
Li et al. (2015)	6982	–	White	TRNSYS (Beckman et al., 1994)	–	–	10	–
Bengea et al. (2011)	–	–	Gray	RLS MATLAB (Jiménez et al., 2008)	Monte-Carlo	2	5	15
Sourbron et al. (2013b)	24	–	Gray	TRNSYS (Beckman et al., 1994)	Step, BaU ^a	4	1	2–4
Bacher and Madsen (2011)	120	–	Gray	CTSM-R (Kristensen et al., 2004a)	PRBS	6	1	2–4
Reynders et al. (2014)	136	–	Gray	CTSM-R (Kristensen et al., 2004a)	BaU ^a	7–28	1	3–5
Madsen and Holst (1995)	60	–	Gray	CTSM-R (Kristensen et al., 2004a)	PRBS	4	1	2
De Coninck and Helsen (2016)	960	•	Gray	Modelica (Baetens et al., 2015), GB tbx (De Coninck et al., 2016)	BaU	18	1	4
Arroyo, van der Heijde, Spiessens, and Helsen (2018)	109	–	Gray	Modelica (Wetter et al., 2014), GB tbx (De Coninck et al., 2016)	BaU ^a	14	9	23
Blum and Wetter (2017)	37	–	Gray	Modelica (Wetter et al., 2014), MPCpy (Blum & Wetter, 2017)	BaU ^a	3	3	10
Blum et al. (2019b)	48	–	Gray	Modelica (Wetter et al., 2014), MPCpy (Blum & Wetter, 2017), ModestPy (Arendt et al., 2018b)	BaU ^a	1–21	1	1–4
Blum et al. (2016)	4982	–	Gray	MATLAB (The MathWorks, 2000)	Pulses ^a	5	18	72
Li et al. (2015)	6982	•	Black	MATLAB (The MathWorks, 2000)	BaU ^a	2	10	–
Hilliard, Swan, Kavgić, Qin, and Lingras (2016)	27,000	•	Black	Rand forest R (Liaw & Wiener, 2001)	BaU ^a	6570	32	–
Hilliard, Swan, and Qin (2017)	10,000	•	Black	Rand forest R (Liaw & Wiener, 2001)	Pulses, BaU ^a	–	40	–
Ma, Qin, and Salisbury (2014)	–	•	Black	MATLAB (The MathWorks, 2000)	PRBS ^a	9	5	–
Royer et al. (2014)	515	–	Black	MATLAB (The MathWorks, 2000)	PRBS ^a	24	5	–
Kusiak and Xu (2012)	–	•	Black	Neural network	–	22	4	–
Mustafaraj et al. (2011)	260	–	Black	MATLAB (The MathWorks, 2000)	BaU	5	1	–
Smarra et al. (2018)	210	–	Black	Rand forest MATLAB (The MathWorks, 2000)	BaU	46	4	–

^a Simulation model used to generate training data.

where x_k , u_k and d_k are states, inputs and disturbances at the k -th time step, respectively. The model is subject to uncertainties, where model uncertainty is represented by the process noise variable w_k and measurement uncertainty is defined by the measurement noise v_k . Second, the HVAC dynamics can be decoupled from the building envelope model. Third, the statistical properties of the measurement noise v_k can be induced from the data, and the nature of model uncertainty described by the process noise w_k can be induced from the model accuracy by verifying it with real data. Therefore, we focus only on the class of Bayesian estimators. They use the accurate mathematical model of the building and update its predictions by measurements in a feedback fashion with estimator gain L or by solving an online optimization problem. The probabilistic distributions of the process and measurement noise act as tuning factors, similarly to the weighting factors in the MPC objective function. The linear dynamics of the prediction model make linear Bayesian estimators the most straightforward choice. In particular, the Kalman filter (KF) family, for which, based on the nature of the estimator gain L , computation can come in stationary (SKF) or time-varying (TVKF) form. A more advanced optimization-based algorithm is moving horizon estimation (MHE), which is an extension of the KF framework capable of handling constraints over an arbitrary estimation horizon. When the prediction model is nonlinear, classical linear estimators do not guarantee satisfactory performance. In such a case, one can use nonlinear estimators, most notably extended (EKF) or unscented

(UKF) Kalman filters.

For the complete picture, we provide here the equations for SKF as a most straightforward example. In general, a KF consists of two stages executed at every sampling instant: update and prediction. The prediction stage, represented by Eq. (10b), predicts the state at the next time step $k + 1$ based on the current state and the mathematical model of the building. In the update stage represented by Eq. (10a), the measurement y_k^m is used to refine the predicted state estimate from the previous time step by introducing feedback into the system.

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + L \left(y_k^m - \hat{y}_{k|k-1} \right) \tag{10a}$$

$$\hat{x}_{k+1|k} = A\hat{x}_{k|k} + Bu_k + Ed_k \tag{10b}$$

A compact overview of selected works with a focus on state estimators applied to building control can be found in Table 6. For more technical details and performance comparison of the linear estimators using white-box building models, we refer the reader to Cupeiro, Drgoňa, Abdollahpouri, Picard, and Helsen (2018).

4.3. Optimal control solution methods

In general, optimal control problems (OCP) are traditionally solved via numerical methods which can be classified into three categories. For

1. At time t , measure, estimate, or forecast the plant's parameters ξ , i.e. states $\hat{x}(t)$, references $r(t), \dots, r(t + (N - 1)T_s)$ and disturbances $d(t), \dots, d(t + (N - 1)T_s)$.
2. Compute the optimal sequence of control inputs $U_{N_c}^*(\xi) = \{u_0^*, \dots, u_{N_c}^*\}$ by solving the problem (1).
3. Select only the first element of the control signals sequence, i.e., $u^*(t) = u_0^*$.
4. Implement the selected control signal over a pre-defined time interval, called sampling time T_s .
5. Time advances to the next interval $t + T_s$, and the procedure is repeated from step 1 with updated parameters ξ , using values of $\hat{x}(t + T_s), r(t + T_s), \dots, r(t + NT_s)$ and $d(t + T_s), \dots, d(t + NT_s)$.

Algorithm 1. Receding horizon control.

more details, see (Binder et al., 2001; Kelly, 2017; Rao, 2019): *Direct methods* These approaches are based on the translation of the OCP (1) to the corresponding optimization problem (OP) and solution via optimization algorithms. Their efficiency and versatility make direct methods most popular for the solution of the OCP in practice today. They are discussed into more detail in Section 4.4. *Indirect methods* These approaches are based on the calculus of variations and Pontryagin's maximum (minimum) principle. Here, the OCP (1) is reformulated as a boundary value problem and the optimal solution is obtained by maximization (minimization) of the control Hamiltonian, which is the function incorporating the stage cost and costate equations. This problem can be solved by several types of numerical methods, namely, gradient-based, multiple shooting and collocation methods. Indirect methods carry, however, several practical drawbacks: difficult formulation of the problem in a numerically suitable way, problems with handling the active constraints, the need for an accurate initial guess, and difficulties with including changes in the problem formulation, such as re-parameterization of the constraints. *Dynamic programming (DP) methods* These approaches provide a globally optimal control policy via recursive solution of the Hamilton-Jacobi-Bellman (HJB) equations as a single step optimization problem. The main disadvantage of this approach is the so-called *curse of dimensionality*, which restricts the solutions to very small state dimensions. However, this disadvantage is reduced with the concept of *approximate dynamic programming (ADP)*, which is also known as *reinforcement learning (RL)* in the machine learning community. These algorithms are based on simple principles of reward and punishment to facilitate the learning of approximate control policies and/or value functions by interacting with the controlled system. Advances in RL research in recent years may provide an interesting framework for solution of the building climate control tasks (Liu & Henze, 2007).

Recently, there has been given a substantial research effort into new optimal control (OC) solution methods emerging from various fields. To give the reader a broad overview of the complexity and various possibilities in solving OCP, we refer to Fig. 7, which captures an approximate taxonomy of the classical and alternative OC solution methods relevant for the field of building control and MPC in general.

Due to the before-mentioned claims on the dominance of direct methods in today's practice, the scope of this paper will focus on direct methods only. Further, in Section 5, we define basic MPC problem classes which differ in the type of the resulting OP, and in Section 6, we define three solution paradigms based on direct methods.

4.4. Direct methods

Direct methods are based on translation of the OCP into an OP and obtain its solution via numeric optimization methods. In general, there are two distinct strategies for the translation (Binder et al., 2001):

Sequential simulation and optimization: In every time step, the model equations (1b) are solved via numerical integration for the current control variables.

Simultaneous simulation and optimization: The model equations (1b) are represented in the OP as equality constraints that can be

violated during the optimization process and need to be satisfied at the solution.

The particular methods are: *Single shooting* This method is also called *dense formulation*, or *state condensing* method. It is a sequential approach, which solves a boundary value problem by reducing it to the solution of an initial value problem. It 'shoots' the candidate trajectories in different directions until it finds the one which satisfies the boundary conditions. The OCP is rewritten into a smaller, but denser OP form, eliminating the states from the vector of optimization variables. This approach is recommended for systems with computationally cheap numerical integration, such as linear systems. The underlying principle of this strategy is illustrated in Fig. 8a. *Multiple shooting* This method is also called *sparse formulation*. It is a hybrid method because it divides the solution interval into smaller intervals, for each of which an initial value problem is being solved with additional conditions that match the solution on the whole interval. In this formulation, each input u_k and each state x_k are considered as optimization variables, forming a large, but sparse OP form. The efficiency of many advanced optimization solvers tailored to solve OCP is based on exploiting the sparsity of the problem. This approach is usually faster than single shooting for systems with nonlinear dynamics. The underlying principle of this strategy is illustrated in Fig. 8b. *Collocation* This method is a simultaneous approach, which selects a finite-dimensional space of candidate solutions and set of collocation points in the parametric domain, and chooses the solution which satisfies the given equations at these collocation points. In this formulation, the set of optimization variables consists of all inputs u_k , states x_k , and collocation points $x_{k,j}$, where index j corresponds to the j -th collocation point for each state x_k . Therefore, the resulting OP is even larger, but also sparser, than in multiple shooting approach. Collocation may bring improved speed and performance for systems with highly nonlinear dynamics. The underlying principle of this strategy is illustrated in Fig. 8c.

5. MPC problem classes

In this section, we recall the most notable MPC problem classes which differ in the type and structure of the corresponding optimization problem to be solved via *direct methods*.

5.1. Linear MPC

We speak about linear MPC (LMPC) when the objective function (1a) is either linear or quadratic and the prediction model (1b) is linear as given by Eq. (9). Then, the OCP (1) can be translated to a Linear Programming (LP) or Quadratic Programming (QP) problem, depending on whether the objective function is linear or quadratic. The main advantage of linear systems is that they can be integrated in a straightforward manner via dense formulation by recursive substitution of consecutive state variables. The complexity of such dense LP becomes $\mathcal{O}(N^3 n_u^3)$, with N the control horizon, and n_u the number of inputs. Usually, because of a large number of states, the condensing method is appropriate for linear building control applications. On the other hand, the computation cost of sparse LP is $\mathcal{O}(N^3 (n_x + n_u)^3)$, where n_x is the number of states (Frison

Table 6
Selective summary of state estimators applied to building control.

Reference	SKF	TVKF	EKF	UKF	MHE
Picard et al. (2017)	•	–	–	–	–
Zong et al. (2017)	•	–	–	–	–
Cupeiro et al. (2018)	•	•	–	–	•
Li, O'Neill, and Braun (2013); Li et al. (2015)	–	•	–	–	–
Chandan and Alleyne (2014)	–	•	–	–	–
O'Neill, Narayanan, and Brahme (2010)	–	–	•	–	–
Fux et al. (2014)	–	–	•	–	–
Chen, Wang, and Srebric (2015)	–	–	•	–	–
Maasoumy et al. (2013, 2014)	–	–	•	•	–
Baldi, Yuan, Endel, and Holub (2016)	–	–	•	•	–
Radecki and Hencsey (2012)	–	–	•	•	–
Bonvini, Sohn, Granderson, Wetter, and Piette (2014)	–	–	–	•	–
Ferhatbegović, Zucker, and Palensky (2012)	–	–	–	•	–
Fielsch, Grunert, Stursberg, and Kummert (2017)	–	–	–	•	–
Vande Cavey et al. (2014)	–	–	–	•	•

& Jorgensen, 2013). If the solver makes use of the sparsity of the problem, the complexity of the problem becomes $\mathcal{O}(N(n_x + n_u)^3)$.

Today, LMPC is a well-studied and established technology in many industries, with efficient online implementation scalable even to problems with hundreds of thousands of parameters and optimization variables (Muske & Rawlings, 1993). Due to this fact, and also because the thermal dynamics of the building envelope can be linearized with high accuracy (Picard, Jorissen, & Helsen, 2015b), LMPC is considered to be a mature technique for building climate control (Rehrl & Horn, 2011; Sourbron, Verhelst, & Helsen, 2013b).

5.2. Nonlinear MPC

Nonlinear MPC (NMPC) emerges when either the objective function (1a) or the prediction model (1b) is nonlinear. Then, the translation of the OCP (1) yields a Nonlinear Programming (NLP) problem. In general, for nonlinear dynamic equations, multiple shooting and direct collocation methods are numerically more efficient. This is due to the available solvers' capabilities of exploiting the sparsity of the corresponding NLP. However, in general, nonlinearities in building models can be decoupled from the linear dynamics and represented by Hammerstein-Wiener models. These models are composed of linear dynamical equations representing the building envelope, and nonlinear static algebraic equations representing the HVAC and effects of disturbances. In this case, single shooting is more efficient than multiple shooting and collocation due to cheaper numerical integration of linear dynamic equations.

NLPs can be efficiently solved even on larger scales by using algorithms such as sequential quadratic programming algorithms (SQP) (Gill, Murray, & Saunders, 2005b), or newton-based methods (Wächter & Biegler, 2006). A more detailed discussion about solutions for NMPC

can be found in Binder et al. (2001). NMPC has a large potential in the building sector due to more accurate predictions of nonlinear models (HVAC models in particular) and higher flexibility in the formulation of the OCP (1). Several studies and real applications of NMPC for buildings have already been reported (Castilla et al., 2014; Jorissen, Picard, Cupeiro Figueroa, Boydens, & Helsen, 2018b; Santos, Zong, Sousa, Mendonca, & Thavlov, 2016; Touretzky & Baldea, 2014), and we can expect more of them to come in the years to come.

5.3. Hybrid MPC

When the dynamical model of the system (1b) employs switching dynamics, binary or integer control variables, logic states or constraints, then we speak about hybrid MPC (HMPC). If the hybrid dynamic model is piecewise linear

$$x_{k+1} = A_i x_k + B_i u_k + E_i d_k, \quad \text{if } (x_k, u_k, d_k) \in \mathcal{R}_i, \tag{11}$$

the corresponding optimization problem to be solved is either a Mixed-Integer Linear Programming (MILP) or Mixed-Integer Quadratic Programming (MIQP) problem, depending on the objective function being linear or quadratic. On the other hand, when the hybrid dynamical model incorporates nonlinearities, we end up with an extremely difficult Mixed-Integer Nonlinear Programming (MINLP) problem.

There exist three main frameworks for modeling of HMPC:

Mixed logical dynamical (MLD) systems: This framework incorporates both continuous and binary variables by means of mixed-integer linear equalities and inequalities and auxiliary binary variables (Bemporad & Morari, 1999a).

Big-M approach: This approach translates the hybrid model into a set of if-then-else conditions which are subsequently translated into corresponding mixed-integer equalities and inequalities by using auxiliary binary variables and large positive values of the constant parameters (Williams, 1993).

Generalized Disjunctive Programming (GDP): This method represents discrete decisions in the continuous space via logical disjunctions and uses logical propositions to denote algebraic constraints in the discrete space (Castro & Grossmann, 2012; Grossmann & Ruiz, 2012). Compared to traditional MIP, the inherent logic structure in GDP yields tighter relaxations that are exploited by the global branch and bound algorithms to improve solution quality (Bhattacharya, Ma, & Vrabie, 2020).

In general, Mixed-Integer Programming (MIP) problems are NP-complete problems and thus are hard to solve. However, there are several state-of-the-art optimization solvers capable of solving these problems even on larger scales (Bemporad, 2006). From a practical point of view, HMPC based on MIP optimization is a powerful tool for control of buildings employing discrete decision variables (e.g., shadings positions, on-off valves, etc.) (Le, Bourdais, & Gueguen, 2014), switching dynamics (e.g., operating modes of the heat pump) (Mayer, Killian, & Kozek, 2015), or for the formulation of supervisory HMPC optimizing

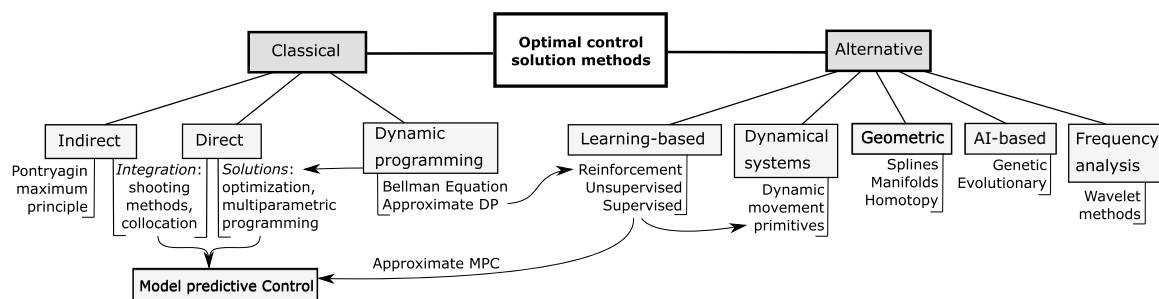


Fig. 7. Approximate taxonomy of optimal control solution methods.

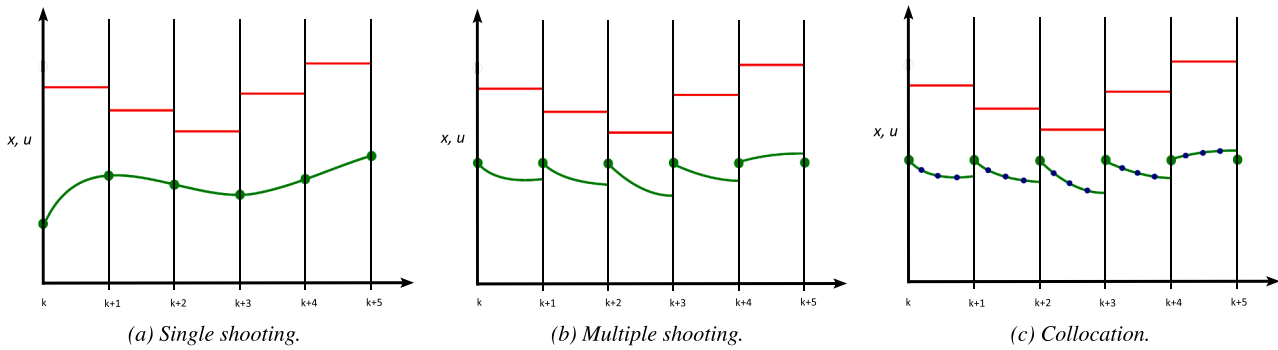


Fig. 8. Visual comparison of discretization principles behind different translation methods. Actions u_k (red) are discretized at each sampling interval to control the state trajectories x_k (green). Green dots represent the values of state variables, or their initial guess in the case of multiple shooting and collocation, while blue dots correspond to collocation points. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the performance of relay-based thermostats (Drgoňa, Klaučo, & Kvasnica, 2015). The first use of GDP in the context of building control and its comparison with classical MIP method was reported in (Bhattacharya et al., 2020).

6. MPC problem solutions

In this section, we recall three distinct solution paradigms based on *direct methods* which can be used to obtain solutions to the MPC problems described in the previous section.

6.1. Implicit MPC

In the case of implicit MPC, the optimal control sequence U_N^* for a particular choice of parameters ξ is obtained by solving online the corresponding optimization problem (1). Computational complexity of obtaining such a sequence depends on the type of the prediction model (1b) and the choice of the cost function (1a), as discussed in the previous section. Depending on the problem type and solver used, the solution of such OP usually requires a relatively powerful computing platform, and in practice it is performed most often via desktop or industrial computers. However, recent advances in dedicated solvers for fast MPC allow us to implement the online MPC algorithms also on embedded hardware with limited computing power and memory storage (Wang & Boyd, 2010). An overview of the most notable optimization solvers for each class of problems is provided in Section 8.3.

As mentioned in Section 2.2, buildings are inherently slow dynamic systems, which allow sufficiently large time windows for the solution of a large-scale OP. Such problems are emerging from MPC formulations with long prediction horizons and a larger number of parameters, which are typical for building control applications. Hence, there is no surprise that most of the building MPC applications reported in a survey (Afram & Janabi-Sharifi, 2014b) have been implemented in an online fashion via implicit MPC.

6.2. Explicit MPC

The development of explicit MPC was driven by the motivation to overcome the primary drawback of implicit MPC, which is the need to compute the optimal control law online at every sampling instant by solving the corresponding OP. Instead, explicit MPC employs *parametric programming* (Bemporad, M., Dua, & Pistikopoulos, 2002; Borrelli, 2003) to pre-calculate the optimal control law for *all* admissible values of parameters ξ . Hence, the explicit representation of the optimizer is constructed offline as a function of the MPC parameters given as $u = f_{\text{MPC}}(\xi)$. Then, the online identification of the optimal control action boils down to mere function evaluations for particular measurements. This significantly reduces computational requirements of the

implementation. From a mathematical point of view, the problems to be solved in the case of linear MPC are multi-parametric linear programs (mpLP) or multi-parametric quadratic programs (mpQP), respectively.

The fundamental limitation of explicit MPC solution is, however, that the complexity of the computed explicit control law grows exponentially with the dimensionality of the parametric space imposed by the number of constraints of the problem, which grow with higher prediction horizon and number of parameters. Therefore, it can only be used for small-scale systems with up to 10 parameters (Mayne, 2014). Also, the memory storage capacity of the hardware should be large enough to accommodate the pre-computed explicit control law (Bemporad, 2006). One possible remedy to overcome this large memory footprint drawback is to employ the recently introduced approach of so-called *region-free explicit MPC* (Kvasnica, Takács, Holaza, & Cairano, 2015b). The solution complexity of this approach no longer depends on the number of parameters ξ , but rather on the number of optimization variables u , for instance up to 20. This still limits the problem complexity mainly with respect to the prediction horizon length and number of decision variables. In both cases, these restrictions are usually not a realistic assumption for complex building control problems with several thousands of parameters and hundreds of optimization variables. Thus, only a few applications of explicit MPC with simplified building models have been reported (Drgoňa et al., 2013; Parisio et al., 2014).

6.3. Approximate MPC

The idea behind this solution approach is to train machine learning (ML) models such that they mimic the behavior of MPC. This concept is also known as imitation learning, where MPC is acting as a teacher and generates the training data for an ML model. The training data are generated in closed-loop simulations by implicit MPC, as defined in Section 6.1. Then, the ML model represents an approximation of the MPC control law, also called control policy.

The parametric solution of problem (1) represents a mapping of the parametric space to the space of control variables, i.e. $f_{\text{MPC}} : \mathbb{R}^{n_\xi} \rightarrow \mathbb{R}^{n_u}$. For this task, state-of-the-art supervised learning algorithms can be used to approximate MPC policies with an arbitrary type of cost functions and constraints. Regression algorithms can be used for problems with continuous control variables, while classification algorithms can be used for problems with discrete control variables. Consider a set of m training data $\{(\xi^{(1)}, u^{(1)}), \dots, (\xi^{(m)}, u^{(m)})\}$, with $\xi^{(i)} \in \mathbb{R}^{n_\xi}$ and $u^{(i)} \in \mathbb{R}^{n_u}$ generated by an implicit MPC approach acting as an expert teacher for an ML algorithm. The objective is to devise a regression/classification function $f_\theta : \mathbb{R}^{n_\xi} \rightarrow \mathbb{R}^{n_u}$, which predicts the values of control variables u (often called the *response* or *target variable*) that correspond to the parameters ξ

¹ Here, $\xi^{(i)} \in \mathbb{R}^{n_\xi}$ denotes the i th sample of a vector ξ .

(representing the *feature vector*) as accurately as possible. During online evaluation, the implicit MPC described in Section 6.1 is replaced by an approximate control policy $u = f_{\Theta}(\xi)$.

The advantage over implicit MPC is that the solution of the optimization is replaced by a computationally cheap function evaluation similar to the case of explicit MPC. The main advantage over the explicit MPC approach, however, is that the ML approach is not limited to lower-dimensional parametric spaces, which allows for construction of the approximated explicit control laws with a low memory footprint for large-scale problems with many parameters. The drawback of the ML approach is that the control policy is suboptimal with respect to the solution of the MPC problem (1), and that a larger amount of informative training data is needed to learn well-performing control policies. Additionally, in a standard imitation learning setup, the learned controller does not provide any guarantees on stability and constraints handling.

Generic approaches dealing with imitation learning of MPC control laws have been recently proposed in Chen, Wang, Atanasov, Kumar, and Morari (2019b); Hertneck, Köhler, Trimpe, and Allgöwer (2018); Lucia, Navarro, Karg, Sarnago, and Lucía (2018); Maddalena, Moraes, Waltrich, and Jones (2019); Zhang, Bujarbaruah, and Borrelli (2019). One of the first attempts to generate MPC laws for building control problems in the form of look-up tables was introduced by Coffey (2013). Other researchers used classification algorithms for extracting decision rules from hybrid MPC closed-loop behavior (Domahidi et al., 2014; Le, Bourdais, & Guéguen, 2014b; May-Ostendorp, Henze, Corbin, Rajagopalan, & Felsmann, 2011b). Approaches for the more challenging task of approximating the continuous control laws are also available. For example, they are based on a piecewise linear mixing architecture (Baldi, Michailidis, Ravanis, & Kosmatopoulos, 2015), regression trees with piecewise linear approximations (Klaučo, Drgoňa, Kvasnica, & Di Cairano, 2014), nonlinear regression (Žáčková et al., 2015), or deep learning models (Drgoňa et al., 2018).

7. Dealing with uncertainties in MPC

The real-world implementation of the model-based control strategies suffers from the plant-model mismatch and inaccurate or corrupted measurements. This section aims to present an overview of methods used to mitigate the effect of uncertainties on the performance and safety indicators of MPC, such as constraints handling and stability guarantees. In general, we face two classes of uncertainties modeled by following parameters:

Parametric uncertainty: $q \in \mathbb{R}^{n_q}$ originates directly in neglected dynamics of the plant, so-called plant-model mismatch.

Non-parametric uncertainty: also called additive is caused by external disturbances, in particular measurement noise $v_k \in \mathbb{R}^{m_y}$, and process noise $w_k \in \mathbb{R}^{m_x}$.

Lets consider following uncertain linear system:

$$x_{k+1} = A(q)x_k + B(q)u_k + E(q)d_k + w_k, \tag{12a}$$

$$y_k = C(q)x_k + D(q)u_k + v_k. \tag{12b}$$

From the building perspective, the most common parametric uncertainties arise from the modeling errors caused by unknown parameters, inaccurate equations, or components not working according to specifications. Most common non-parametric uncertainties are associated with measurements and predictions of ambient temperature, solar irradiation, temperature sensors inaccuracy, or by a limited number of sensors, and unmeasured disturbances, such as windows opening. In principle, implementation of MPC in RHC approach implicitly reduces the plant-model mismatch due to the presence of feedback. However, for higher degrees of uncertainties, it is often not sufficient by itself and more advanced techniques need to be adopted to ensure the desired

control performance.

7.1. Offset-free MPC

The purpose of this popular technique is to compensate the effect of uncertainties via prediction model augmentation by extra states p representing unmeasured disturbances (Muske & Badgwell, 2002). These disturbances p are estimated by Kalman Filters or moving horizon estimation (MHE), and their effect is subsequently compensated by the MPC via predictions. One extra state with a constant dynamic is added per each output or state of the prediction model (Pannocchia & Rawlings, 2003). This approach is also called active disturbance rejection control and allows us to consider a simpler controller model, because the modeling error is compensated for in real time (Picard et al., 2017). For a linear system, the disturbance augmented prediction model represented by matrices $\hat{A}, \hat{B}, \hat{E}, \hat{C}, \hat{D}$ is given as:

$$\begin{bmatrix} \hat{x}_{k+1} \\ \hat{p}_{k+1} \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \hat{x}_k \\ \hat{p}_k \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_k + \begin{bmatrix} E \\ 0 \end{bmatrix} d_k, \tag{13a}$$

$$\hat{y}_k = \begin{bmatrix} CF \\ \bar{c} \end{bmatrix} \begin{bmatrix} \hat{x}_k \\ \hat{p}_k \end{bmatrix} + \begin{bmatrix} D \\ 0 \end{bmatrix} u_k. \tag{13b}$$

where the output disturbance matrix F was chosen as a full column rank identity matrix and all other matrices are the same as in Eq. 9.

Variants of OSF-MPC A linear offset-free MPC (OSF-MPC) for reference tracking formulation was studied in Maeder, Borrelli, and Morari (2009). A comprehensive overview of OSF-MPC for the linear and nonlinear discrete-time system together with economic MPC formulation was presented in Pannocchia, Gabbicini, and Artoni (2015). A disturbance modeling and estimator design were systematically studied for different formulations of state-space process models in Tatjewski (2011). The design and tuning of OSF-MPC based on the black-box ARX model was discussed in Huusom, Poulsen, Jørgensen, and Jørgensen (2010). Authors in Huang, Biegler, and Patwardhan (2010b) presented an approach for reduction of the computational burden associated with the online computation of nonlinear OSF-MPC with MHE.

OSF-MPC for Buildings In the context of building control, the OSF-MPC formulation for a white-box heat pump model developed in Modelica was given (Wallace, Mhaskar, House, & Salsbury, 2014). A multi-zone heat pump model developed in Modelica was augmented with a disturbance offset of the measured outputs for the design of centralized linear OSF-MPC (Krupa et al., 2019). A simulation study of an OSF-MPC for energy-efficient operation of the hotel’s central chiller plant in a tropical climate was presented in Lara, Molina, Yanes, and Borroto (2016). Systematic analysis with varying order of the building envelope model for three variations of the residential houses showed that state augmentation can reduce the modeling errors and improve the overall control performance in terms of energy use and comfort constraints satisfaction (Picard et al., 2017).

7.2. Robust MPC

In case the impact of uncertainties significantly decreases the control performance, or even endangers the closed-loop system stability, we introduce the *robust* MPC policy, see (Bemporad & Morari, 1999b) and references therein. Robust MPC strategy is also suitable if we need to certify the designed MPC w.r.t. the impact of the bounded uncertainties. As the values of uncertain parameters vary, there are various scenarios of the future behavior of the plant. Therefore, the crucial task of robust MPC is to design a control law that guarantees the closed-loop system stability of the plant subject to all the admissible evolution scenarios of

the uncertain system. As a consequence, the robust MPC strategy is usually *conservative*. This means that the robust control policy ensures the constraints satisfaction by creating an energy buffer (in the case of energy minimization) to be able to mitigate the impact of some unexpected disturbances. More generally, the robust MPC creates reserves for potentially difficult times in the future, the quantity of which is determined based on the estimates of the worst-case scenario and robust control design method.

Complexity of RMPC The robust MPC assumes the impact of the bounded disturbance. Consider a linear state-space system in (12) affected by bounded uncertainty $q \in \mathcal{Q}^{n_q}$, where $\mathcal{Q}^{n_q} \subset \mathbb{R}^{n_q}$ is the n_q -dimensional set of uncertain parameters. Consider constraints given by (1f), where \mathcal{U}, \mathcal{X} are polytopes including origin in their strict interior. Then, the closed-loop system is robustly stable if and only if all the vertices of the constraints parametrized by uncertainty q are simultaneously stable. In other words, although the uncertainty set \mathcal{Q} includes an infinite number of a possible realization of q , the system is stable within all constraints on the feasible region under bounded uncertainty q by checking 2^{n_q} the system vertices. Total number of uncertain system vertices 2^{n_q} originates in the enumeration of hyper-box vertices defined in n_q -dimensional space, e.g., see (Kothare, Balakrishnan, & Morari, 1996). The main drawback is that the number of investigated vertices increases exponentially with a prediction horizon N , i.e., $2^{n_q} \times N$. The complexity is high because the controller evaluates all scenarios w.r.t. all combinations of uncertain parameters. Therefore, the dominant term of the complexity evaluation is determined by the number of uncertain parameters n_q .

Min-max RMPC In general, robust optimal values of the manipulated variables are computed either (i) directly as a *sequence* of the control actions, or (ii) by designing the linear/affine state-feedback *control laws*, see (Langson, Chrysochoos, Raković, & Mayne, 2004). Various approaches are considered to keep the optimization problem tractable, mostly considering the *worst-case*, i.e., so-called MIN-MAX optimization (Campo & Morari, 1987). In this approach, only the worst-case scenario is evaluated and used for the robust controller design.

LMI-based RMPC Another popular approach in tackling the exponential complexity is based on linear matrix inequalities (LMIs), see (Boyd, El Ghaoui, Feron, & Balakrishnan, 1994). The advantage of LMIs lies in the possibility of transforming non-convex optimization problem into the convex form. The original problem could also have infinity many decision variables, but introducing LMIs enables to transform it into a tractable optimization problem with modest complexity. The idea of LMIs is to optimize the control performance by minimizing the eigenvalues of the matrices. For instance, the aim is to optimize the trajectories of the controlled variables, e.g., zone temperatures, subject to the influence of uncertain parameters. The solution of the optimization problem shapes the set of admissible values of the controlled variables, i.e., defines their limit values. In the case of LMIs, the resulting optimal set has the shape of *ellipsoid* ε and contains the setpoint values in its center. The volume of this set is minimized in each control step to keep the controlled variables closer to the setpoint value. This strategy is pioneered by Kothare et al. (1996) and refined by many later works, see Oravec, Pakšiová, Bakošová, and Fikar (2017); Zhang, Wang, and Wang (2014) and references therein.

SDP in RMPC From a technical point of view, the problem is transformed into the form of *semidefinite programming* (SDP) (Vandenberghe & Boyd, 1996) that has a convex (usually linear) objective function and the constraints have the form of LMIs. For the class of SDP problems, various tailored solvers are available, for instance, SeDuMi, MOSEK, to list some, while a more extensive list is provided in Table 13. The online computational complexity of SDPs can be further reduced by replacing them by QPs w.r.t. the construction of the maximal robust positive *invariant sets* (Blanchini, 1999), forward and backward *reachable sets* (Borrelli, Bemporad, & Morari, 2017). Once the system state enters the invariant set, it is trapped inside this set also in the future. As a consequence, the states will not diverge into infinity/instability. For instance,

life sentence in a prison is an invariant set. Analogously, the reachable sets limit the future behavior of the states. In control theory, the reachability for a dynamical system means that a certain state is reachable from a given initial state within a given cost threshold (Allen, Clark, Starek, & Pavone, 2014). We can think of it as a formal reality check, answering questions of a type: "Can we reach the thermal comfort zone from a given room temperature within an hour by using a given amount of energy?". Therefore, these properties are crucial tools to guarantee/certify the closed-loop system stability and performance.

Explicit and tube-based RMPC The explicit solution of the robust MPC problem was proposed in Kvasnica, Takács, Holaza, and Ingole (2015). However, from a computational viewpoint, it is limited by the modest complexity of the optimization problem, i.e., a number of constraints. So-called *tube-based* robust MPC addresses the problem of the conservatives minimization of robust MPC policy by reducing the exponential evolution of the predicted states, see pioneer work (Langson et al., 2004), or more recent papers (Yadbantung & Bumroongsri, 2018; Zeilinger, Raimondo, Domahidi, Morari, & Jones, 2014), and references therein. The "tube" refers to the shape of the bounded set of admissible evolutions of the controlled variable.

RMPC for buildings The detail analysis of the sources of the uncertain parameters and the origins of the imperfect models in building energy assessment is provided in Tian et al. (2018). From building control perspective, the robust MPC based on offline precomputed LMIs for temperature control of variable-air-volume air-handling units was designed in Huang, Wang, and Xu (2010a), Xu, Wang, and Huang (2010). Simulation results show robust control performance and constraints satisfaction. A robust MPC framework based on the input disturbance feedback for building HVAC systems was proposed in Maasoumy, Razmara, Shahbakhti, and Vincentelli (2014); Maasoumy and Sangiovanni-Vincentelli (2012). A novel robust adaptive MPC strategy reducing the conservativeness of the uncertainty handling was presented in Tanaskovic, Sturzenegger, Smith, and Morari (2017). The simulation results show improved control performance in contrast to non-robust adaptive MPC. In Antonov and Helsen (2016), robustness analysis of the designed MPC was performed. Satisfaction of the robustness conditions subject to the uncertain prediction of the system states was investigated *a posteriori* to prevent evaluation of computationally demanding robust MPC design procedure. The classification of various Robust MPC approaches to building control is given in Table 7.

7.3. Stochastic MPC

Stochastic MPC (SMPC) is a framework for systems affected by probabilistic uncertainty. A key feature of SMPC are chance constraints (CC), which enable a systematic trade-off between control performance and probability of constraints violations (Heirung, Paulson, OLeary, & Mesbah, 2018). Chance constraints, for example on state variables, are given in the form:

$$\Pr(x_k \in \mathcal{X}) \geq 1 - \alpha, \quad k \in \mathbb{N}_0^{N-1} \quad (14)$$

where $\Pr(x_k \in \mathcal{X})$ denotes the probability of satisfaction of the constraint $x_k \in \mathcal{X}$, and $1 - \alpha$ specifies the value of that probability for $\alpha \in [0, 1]$. Unfortunately, these types of constraints are in general non-convex and extremely computationally demanding for optimization. Hence, for any practical implementation of SMPC, a computationally tractable reformulation of CC needs to be derived. For this task, there are several approaches which are based on solving convex realizations of chance constrained optimization problems.

An overview of linear SMPC with CC classifying alternative approaches in terms of the system model, the objective function, the meaning and management of the chance constraints, and their feasibility and convergence properties was given in Farina, Giulioni, and Scattolini (2016a). The connection to stochastic dynamic programming as well as Bayesian estimation of SMPC problem in the dual control paradigm was

reviewed (Mesbah, 2018). Authors in Lorenzen, Müller, and Allgöwer (2017d) provide assumptions that are sufficient to establish closed-loop stability for various approximations of CC used in SMPC methods. For the purposes of this paper, we classify the alternative SMPC methods into three principal groups based on Mesbah (2016), namely scenario-based approaches, chance constraints approximations, and disturbance feedback control law parametrizations. The conceptual difference of the latter two approaches compared to scenario approaches is that no samples need to be generated. Instead, some prior knowledge of the system or the past realization of the uncertainties is exploited to derive the accurate approximations of chance constraints.

Scenario-based approaches Sampling-based techniques replace the CC with a finite number of deterministic constraints generated by the various realizations of the stochastic variables. Sampling density is chosen as a trade-off between computational demands and violation probability. A larger number of samples decreases the violations but usually leads to increased computational burden (Zhang, Schildbach, Sturzenegger, & Morari, 2013). Another concern of SMPC is safety in terms of closed-loop stability and constraint handling capabilities. Stochastic stability and recursive feasibility can be enforced through linear matrix inequality (LMI) for linear problems (Bernardini & Bemporad, 2009). An alternative approach uses an offline sampling of probabilistic constraints realizations to guarantee recursive feasibility and asymptotic stability of linear SMPC (Lorenzen, Allgöwer, Dabbene, & Tempo, 2015). Additionally, it has been shown that bounds on closed-loop constraint violations can be provided for linear SMPC formulations (Schildbach, Fagiano, Frei, & Morari, 2014). Modern approaches involve machine learning methods in the design of SMPC, for instance, using Gaussian processes GP (Bradford, Imsland, Zhang, & del Rio Chanona, 2019), or Support Vector Clustering (SVC) for learning an uncertainty set directly from available data (Shang & You, 2019).

Chance constraints approximations Sometimes also referred to as stochastic tube approaches, these approximations are based on replacing CC with deterministic constraints by tightly bounding the disturbances. A convexity of chance-constrained SMPC for linear systems was studied in Cinquemani, Agarwal, Chatterjee, and Lygeros (2011). An extension of CC-based SMPC to nonlinear dynamics was presented in Xie, Li, and Wozny (2007). Of the latest approaches, CC defined as a discounted sum of violation probabilities on an infinite horizon guarantees the recursive feasibility without an assumption of boundedness of the disturbance (Yan, Goulart, & Cannon, 2018). Authors in Lorenzen, Dabbene, Tempo, and Allgöwer (2017c) propose a constraint tightening to non-conservatively guarantee recursive feasibility and stability of CC-based SMPC.

Control law parametrizations Set of techniques directly mapping the influence of the disturbances onto control actions, for instance by expressing the feedback control law as an affine function of the past disturbances. Authors in Oldewurtel, Jones, and Morari (2008) presented a tractable approximation of CC based on affine disturbance feedback for linear systems. An alternative approach with affine parametrization of the control law capable of handling possibly unbounded stochastic disturbances via solving convex second-order cone program (SOCP) was given in Paulson, Buehler, Braatz, and Mesbah (2017).

SMPC for buildings Table 8 summarizes numerous applications of SMPC in the building control context and classifies them based on the principal method used. Please note that the domain of SMPC is far more complex and used methods are more numerous and branched as those presented here. For more detailed overviews and fundamentals on SMPC we refer the interested reader to the following publications (Farina, Giullioni, & Scattolini, 2016b; Heirung et al., 2018; Mayne, 2016; Mesbah, 2016).

7.4. Adaptive MPC

The essential idea of adaptive control is online update of the

controller or the prediction model parameters, such that the systems with time-varying dynamics can be handled using the adaptive strategy, see (Åström & Wittenmark, 2008) and references therein. On the other hand, standard receding-horizon MPC addresses real-time computation of the optimal control actions subject to the fixed structure and parameters of the system model. The control law itself is static, but the control actions are parametrized by system states, references, and disturbances. *Adaptive MPC* merges the benefits of both, i.e., introduces the model updates in the context of MPC. The uncertainties are then corrected not only via feedback of the control law parameters, but also with updates of the model parameters. The parameters updates are typically obtained from autoregressive models, recursive least squares (RLS), Kalman Filters, or other maximum likelihood parameter estimation algorithms. Adaptive model updates allow the MPC to potentially cope with time-varying disturbances and correct plant-model mismatch over longer prediction horizon, as opposed to static disturbance correction terms of the offset-free MPC.

Challenges and approaches Except for the closed-loop system stability and recursive feasibility, the crucial challenges lie in (i) handling MIMO systems (Maniar, Shah, Fisher, & Muthas, 1997); (ii) design control action subject to constraints (Tanaskovic, F., Smith, & Morari, 2014); and (iii) considering the impact of the uncertain parameters (Lorenzen, Allgöwer, & Cannon, 2017; Tanaskovic et al., 2017). A compact overview of adaptive MPC challenges was given in Kim (2010). As pointed out in Qin and Badgwell (2003), only a few adaptive MPC algorithms have been used in practice, despite the strong market incentive for a self-tuning MPC controller. Moreover, due to the above-mentioned challenges, this status quo is likely to be maintained in the near future.

Adaptive MPC remains an active area of research, and it is out of the scope of this paper to provide a complete survey and classification of different approaches. Instead, we mention only a few occurring themes. For increased robustness, an adaptive MPC is often combined with stochastic and robust MPC principles such as set membership identification (Adetola, Guay, 2011; DeHaan, Adetola, & Guay, 2007; Fagiano, Schildbach, Tanaskovic, & Morari, 2015; Lorenzen, Allgöwer, & Cannon, 2017b). An adaptive strategy based on multiple linear models was introduced in Dougherty and Cooper (2003). A novel approach of *dual adaptive MPC* reformulates the original nonlinear deterministic problem into the tractable problem of convex optimization (Heirung, Ydstie, & Foss, 2017; Kumar, Heirung, Patwardhan, & Foss, 2015). The literature on simultaneous state and parameter estimation is complementarily focused on aspects such as estimation error, and signal excitation (Kamalapurkar, 2017; Rangegowda, Valluru, Patwardhan, & Mukhopadhyay, 2018). In recent years, the principles of adaptive MPC are being revised and combined with various machine learning-based methods and are often labeled as learning-based MPC, which is covered separately in the following section.

Adaptive MPC for buildings Adaptive MPC of the HVAC system based on self-adapting building models was investigated in Herzog, Atabay, Jungwirth, and Mikulovic (2013) using simulation. The self-adaptive model for buildings enabling correction of the prediction errors of pre-defined models using a dynamic Kalman filter-bank was proposed in Killian, Leitner, Goldgruber, and Kozek (2017). Robust adaptive MPC for building climate control was proposed in Tanaskovic et al. (2017), where the uncertainty set was recursively updated based on the system identification procedure. Authors in Lauro, Longobardi, and Panziri (2014) studied an adaptive distributed MPC scheme for multi-zone building temperature control and its comparison with a decentralized approach. Adaptive MPC based on multiple linear regression for the control of a low-temperature thermo-active building system was designed in Schmelas, Feldmann, and Bollin (2017). A self-adaptive MPC based on EKF improved the model prediction accuracy for a passive house (Fux, Ashouri, Benz, & Guzzella, 2014). An adaptive MPC mechanism proposing recursive estimation and updating approach for electronic expansion valves with adjustable setpoint for evaporator superheat minimization was addressed in Tesfay, Alsaleem, Arunasalam,

Table 7
Classification of the publications reporting Robust MPC for building control.

Reference	Robust constraints satisfaction	Min-Max approach	LMI-based approach	Offline optimization	Control law parametrization
Huang et al. (2010a); Xu et al. (2010)	•	•	•	•	--
Tanaskovic et al. (2017)	•	–	–	–	--
Ma et al. (2012b); Ma, Borrelli, Hancey, Packard, and Bortoff (2009)	•	–	–	–	--
Maasoumy et al. (2014); Maasoumy and Sangiovanni-Vincentelli (2012)	•	•	–	–	•
Yang, Wan, Chen, Ng, and Zhai (2019)	•	•	–	–	•
L. Chen and Hu (2016)	•	•	–	–	--
Antonov and Helsen (2016)	–	–	•	–	--

and Rao (2018). An online simultaneous state and parameter estimation for building predictive control using extended and unscented Kalman Filters have been proposed in Maasoumy, Moridian, Razmara, Shahbakhti, and Sangiovanni-Vincentelli (2013); Maasoumy et al. (2014).

7.5. Learning-based MPC

In recent years the intersection of the areas of control and learning has been rapidly expanding with the emerging concept of learning-based MPC (LBMPC). However, due to the ubiquitous use, the label LBMPC has an ambiguous meaning. Moreover, LBMPC is an active area of research with rapidly emerging new concepts and applications. The most recent review (Hewing, Wabersich, Menner, & Zeilinger, 2020) classifies LBMPC approaches into three categories, (i) learning of the prediction model from data with uncertainty quantification, (ii) learning the controller design, i.e., learning the constraints and cost function terms, (iii) MPC for safe learning to derive safety guarantees for learning-based controllers. The comprehensive overview of the method is beyond the scope of this paper. Instead, we refer the interested reader to Hewing et al. (2020) and references therein.

Uncertainty-aware LBMPC The first category of LBMPC approaches is the most numerous. The case of learning a static model with uncertainty quantification is directly related to some of the gray- and black-box modeling approaches, discussed in Section 3.2.3. The concept of LBMPC in the context of robust and safe control with data-driven models and online updates was first introduced by (Aswani, Gonzalez, Sastry, & Tomlin, 2013). The main insight of LBMPC is that performance and safety can be decoupled by using reachability analysis (Asarin, Bournez, Dang, & Maler, 2000; Rakovic, Kerrigan, Mayne, & Lygeros, 2006), making the approach tractable and practical. In general, LBMPC is considered to be a generalization of robust adaptive MPC, which is typically restricted to specific types of model structures and learning algorithms. Instead, LBMPC uses statistical learning methods to improve the model of the system dynamics, while using robust MPC techniques to ensure stability and constraints handling (Aswani, Bouffard, Zhang, & Tomlin, 2014). Alternative methods in this category, include, formulation of robust MPC with state-dependent uncertainty for data-driven

linear models (Soloperto, Müller, Trimpe, & Allgöwer, 2018), or an iterative model updates for linear systems with bounded additive uncertainty and robust guarantees on all feasible offsets (Bujarbaruah, Zhang, Rosolia, & Borrelli, 2018).

Learning-based controller design and updates Approaches falling in the second category are represented, e.g., by control methods updating time-varying dynamics, constraints, and stage cost based on closed-loop data for period tasks (Scianca, Rosolia, & Borrelli, 2019). An inverse optimization is a more challenging task dealing with an inference of unknown parameters of an optimization problem based on knowledge of its optimal solutions (Aswani, Shen, & Siddiq, 2015). In this context, pivotal research without performance guarantees on learning of the MPC parameters from available closed-loop data was recently proposed by differentiable MPC (Amos, Rodriguez, Sacks, Boots, & Kolter, 2018). It is important to mention that inverse optimal control approaches are closely linked with imitation learning and approximate MPC solutions discussed in Section 6.3. The difference is that approximate MPC deals with parameterizing an explicit control law based on given samples of closed-loop behavior of the expert controller, as opposed to finding parameters of a given MPC formulation matching the data.

MPC safety certificates for learning-based control The methods in the third category represent new research avenues and are primarily concerned with employing robust or stochastic MPC in conjunction with data-driven controllers for safety certificates (Muntwiler, Wabersich, Carron, & Zeilinger, 2019) or safe exploration (Koller, Berkenkamp, Turchetta, & Krause, 2018), aspects particularly important for reinforcement learning (RL) approaches.

LBMPC for buildings One of the first experimental results of LBMPC applied to the office building in the US with significant energy savings was reported in Aswani et al. (2012), where learning refers to model updates of the gray-box hybrid system model. In the building control literature there is a multitude of learning-based, data-driven, data-enabled, or data predictive approaches representing an ambiguous set of methods, which primary concern is learning of the prediction model. Those methods are often not dealing with uncertainty quantification in the sense of original LBMPC (Aswani et al., 2013). Hence some of them may not provide robust performance guarantees or uncertainty

Table 8
Classification of the publications reporting SMPC for building control based on their principal methods.

Reference	Offline optimization	Scenario-based approach	Chance constraints approximation	Control law parametrization
Oldewurtel et al. (2008); Oldewurtel, Jones, Parisio, and Morari (2014); Oldewurtel et al. (2010)	–	–	•	•
Ma, Matusko, and Borrelli (2015); Ma, Vichik, and Borrelli (2012c)	–	–	•	–
Zhang, Grammatico, Schildbach, Goulart, and Lygeros (2014); Zhang et al. (2013)	–	•	–	–
Long, Liu, Xie, and Johansson (2014)	–	•	–	–
Tanner and Henze (2014)	–	•	–	–
Garifi, Baker, Touri, and Christensen (2018)	–	•	–	–
Kumar et al. (2020)	–	•	–	–
Drgoňa et al. (2013)	•	•	–	–
Parisio et al. (2014)	•	•	–	–

quantification. Of those methods, authors in [Jain et al. \(2017b\)](#); [Smarra et al. \(2018\)](#) successfully implemented random forest and regression trees for optimal buildings control in different scenarios. However, they showed that in some cases, these models suffered from limitations due to overfitting. These so-called data-predictive controllers (DPC) can achieve comparable performance to MPC while avoiding the cost and effort associated with constructing a gray/white-box model of the building ([Jain et al., 2017a](#)). An experimental validation of the DPC method based on random forests applied to the room temperature control reported significant energy savings and thermal comfort improvement compared to baseline rule-based controller ([Bünning, Huber, Heer, Aboudonia, & Lygeros, 2020](#)). Another popular approach is the use of gaussian process (GP) models for real-time receding horizon control with probabilistic guarantees on constraint satisfaction applied to closed-loop simulations of large-scale building models ([Jain et al., 2018](#)). The authors showed how this approach could provide the desired load curtailment in the context of demand response with high confidence. Data-driven MPC based on GP models of the building's power demand compensating the uncertainty was presented in [Nghiem and Jones \(2017\)](#). A preliminary experimental result on the use of differentiable linear MPC trained offline on the RBC data with online reinforcement learning-based updates was presented in [Chen, Cai, and Bergés \(2019a\)](#).

8. Software tools for building modeling, simulation and control

This section aims to provide an extensive overview and high-level comparison of tools for the modeling, simulation, and control of buildings in the context of MPC. The inspiration and some information were obtained from online directories listing available software tools for modeling, analysis, optimization, and simulation for buildings ([EUROSI, 2020](#); [Berkeley Lab, 2020](#); [Nghiem, 2011](#); [US Department of Energy, 2020](#)).

8.1. Building modeling and simulation tools

8.1.1. Building energy simulation tools

Building energy simulation (BES) programs are software tools that simulate energy, mass, and contaminant flows in buildings. This includes the interaction between the building envelope and its surroundings (i.e., weather, radiation heat losses, etc.), internal loads (i.e. occupants, lighting, equipment), and HVAC systems. A number of software modeling tools for buildings are available, which usually consider detailed models of building components. Typically, these tools are built and used for building design purposes. However, as discussed previously in [Section 3.2.1](#), these tools may also be used to implement white-box models for MPC. In addition, these tools are often used to develop digital twins of real buildings (also called emulators), which can be used in simulation case studies to assess the performance of MPC algorithms.

BES tools can be divided into two main subgroups ([Wetter et al., 2016](#)). First, traditional imperative languages which declare the sequence of commands to be executed and are usually defined in function-based format. In this approach, the modeling is interconnected with the solver with a primary purpose of building performance evaluation. An advantage here is that the execution of the simulation can be relatively fast. However, the main disadvantage is that these programs are difficult to extend to support new use cases, such as modeling of controls, reformulation of model equations into optimal control problems or integration with electric grid simulation tools. The second group represents equation-based, object-oriented, declarative languages such as Modelica. The principal difference of this paradigm of modeling in contrast to the imperative paradigm is that instead of giving the sequence of instructions which define the way how the program should behave, they provide a higher-level abstraction in the form of hybrid differential algebraic systems of equations. These equations can then be encapsulated into graphical components and organized into hierarchical

libraries in an object-oriented fashion, which makes them highly reusable and modular. In addition, this type of implementation allows for the explicit definition of state initial conditions as well as symbolic differentiation for efficient numerical integration. Finally, these equations, and their derivatives, can be used for generation of an optimal control problem for MPC, or more easily be integrated with modeling tools from other domains.

A compact summary of BES tools which have been used to replace real buildings for testing MPC algorithms using co-simulation is given in [Table 9](#). Besides programming language paradigm type, the last column indicates whether it is possible to model the control systems with these tools directly. The mentioned programs, however, represent only a subset of all BES tools. For a more comprehensive overview of building and HVAC system modeling and simulation tools, we refer to ([Clarke, 2001](#); [Hensen & Lamberts, 2019](#); [Trcka & Hensen, 2010](#); [Zhou, Hong, & Yan, 2013](#)). More comprehensive comparisons and discussions about BES programs can be found in ([Nageler et al., 2018](#); [Sousa, 2012](#); [Wetter, 2011](#); [Wetter et al., 2016](#); [Wetter & Haugstetter, 2006](#)).

8.1.2. Co-simulation tools and interfaces

BES programs are typically not directly suitable for design, synthesis, and testing of optimal controllers. To deal with this issue, middleware software and interface protocols were designed for making communication bridges between BES programs and control-oriented tools and programming languages like MATLAB or Python. [Table 10](#) provides a summary of selected interface tools and standards relevant to building simulation and control. FMI here stands for Functional Mock-up Interface, which is an interface standard for general modeling and simulation tools not only pertaining to buildings ([Blochwitz et al., 2011](#)). For an in-depth overview and comparison of co-simulation technology we refer to ([Trcka, Hensen, & Wetter, 2009](#)).

8.1.3. Control-oriented building modeling tools

Obtaining models that are accurate enough and at the same time not too complex for optimal control represents one of the bottlenecks which prevents wider adoption of MPC in practice. The main reasons are, first, that models generated by BES programs described in previous sections are often too complex for use in the subsequent optimization problems. Second, they compute numerical approximations to cost functions that are not differentiable ([Polak & Wetter, 2006](#); [Wetter & Polak, 2004](#)). Third, there is a substantial shortage of user-friendly and freely available tools for accurate control-oriented modeling of the buildings. Luckily, in recent years, there has been some progress in this direction, and several tools have emerged to help create the models for MPC. [Table 11](#) provides an overview of such tools. However, it is important to note that most of the tools in this list still either require substantial multi-disciplinary expertise or are only available as a research tool.

Tools exist for the linearization of Modelica models ([Picard, Jorissen, & Helsen, 2015](#)), returning a state space formulation of the model. This allows for direct integration within the optimal control problem. The linearization methodology has proven to have a high level of accuracy. Moreover, Modelica models can be exported as a Functional Mockup Unit, which allows accessing directional derivatives as needed to solve optimal control problems ([Blochwitz et al., 2011](#)). Another white-box control-oriented modeling approach for multi-zone buildings was developed based on the Simscape library in Matlab/Simulink environment ([Lapusan, Balan, Hancu, & Plesa, 2016](#)). The emphasis lies on easy modeling with a modular framework based on a set of pre-defined blocks. The popularity of gray-box models extends to toolboxes for parameter estimation and application of the derived models into MPC. The Grey-Box Toolbox ([De Coninck et al., 2016](#)), for instance, allows parameter estimation of Modelica models using the JModelica ([Modolon, 2017](#)) platform with a front end in Python. The toolbox has been extended for the direct application of the obtained models into MPC ([Vande Cavey, De Coninck, & Helsen, 2014](#)). MPCPy ([Blum & Wetter, 2017](#)) is another toolbox using reduced order grey-box models and

relying on JModelica (Modelon, 2017) for both parameter estimation and solving MPC problems, with the parameter estimation and optimization problems automatically generated based on specification of a Modelica model and high-level input parameters in Python. The modeling environment (ME) for MPC (Zakula et al., 2014) is based on TRNSYS and its coupling with MATLAB to obtain a simplified inverse thermal response model in the form of an inverse comprehensive room transfer functions (iCRTF). The Building Resistance-Capacitance Modeling (BRCM) toolbox (Sturzenegger, Gyalistras, Semeraro, Morari, & Smith, 2014) facilitates physical modeling of buildings for MPC via generation of control-oriented linear RC models from EnergyPlus models. OpenBuild (Gorecki, Qureshi, & Jones, 2015) provides an integrated simulation environment for building control in MATLAB. Like BRCM, it generates the RC models from EnergyPlus. In both tools, co-simulation of MATLAB with EnergyPlus is built on BCVTB (Wetter & Haves, 2008) and MLE+ (Bernal, Behl, Nghiem, & Mangharam, 2012). Another Matlab toolbox BLDG (Kircher & Zhang, 2016) provides users with a standalone building model based on simplified PDE equations with a small number of parameters, along with system identification and parameter estimation functionality. IDENT (Jiménez, Madsen, & Andersen, 2008) provides a graphical user interface in MATLAB to estimate the RC models of building envelopes from the measurement data. BASBenchmarks (Cauchi & Abate, 2018) represents a modular model library for building automation systems covering physical components as well as digital control strategies. The software package LORD (Gutschker, 2008) performs a combination of two different methods alternatively for parameter estimation. One is the Downhill Simplex Method, and the other is a specially adopted Monte Carlo procedure. LORD also offers a graphical user interface for creating the RC model structures based on nodes and connections. CTSM-R (Kristensen, Madsen, & Jørgensen, 2004a) and MoCaVa (Bohlin, 2003) feature maximum likelihood and maximum a posteriori estimation of stochastic grey-box models. The former is accessed through the programming language R, while the latter runs under Matlab. A comparison between MoCaVa and CTSM was studied in Kristensen, Madsen, and Jørgensen (2004b). It shows that CTSM has better performance in terms of quality of estimates for nonlinear systems with significant diffusion and in terms of reproducibility. In particular, CTSM provides more consistent estimates of the diffusion term parameters. Finally, there exist more generic tools that can be used to calibrate simulation models that do not make any assumptions regarding the model (language, paradigm) except the interface. For example, ModestPy (Arendt et al., 2018b) is a parameter estimation Python package for FMI-compliant models, mostly used with gray-box models as in Arendt et al. (2018a), while GenOpt (Wetter, 2001) is an optimization software that can be used for parameter estimation in any model that can be interfaced through text files, e.g. EnergyPlus, TRNSYS.

Table 9
Summary of the selected BES programs used to emulate the buildings for testing MPC in co-simulation.

BES Tool	Free	Equation-based	Imperative	Explicit controls modeling
ESP-r (Yahiaoui, Hensen, & Soethout, 2003)	•	–	•	•
EnergyPlus (Crawley et al., 2001)	•	–	•	–
TRNSYS (Beckman et al., 1994)	–	–	•	–
Modelica (Baetens et al., 2015; Wetter et al., 2014)	•	•	–	•

8.2. MPC design tools

Table 12 provides an overview of the most important software tools which can be used or are particularly dedicated to modeling, simulation, evaluation, deployment and code generation of MPC controllers. Most advanced and widely popular tools are based on MATLAB, Modelica or Python languages and come with a free license. These modeling languages allow for high-level modeling of the optimization problems and provide an interface to a wide variety of optimization solvers in an automated way. This reduces the engineering burden of error-prone and time-consuming manual translation of the OCP (1) to the OP form required by a particular solver.

OpenBuild (Gorecki et al., 2015) supports the design and simulation of the state observer and MPC using an RC model generated based on an EnergyPlus model. BRCM toolbox (Sturzenegger et al., 2014) offers the generation of the cost and constraint matrices for MPC based on the generated RC model from EnergyPlus. However, it does not provide the environment for simulation, tuning, and analysis of MPC. EHCM toolbox (Darivianakis, Georghiou, Smith, & Lygeros, 2020) is an extension of BRCM providing a framework for controlling the operation of the energy hub with multiple interconnected buildings in a cooperative manner. BLDG (Kircher & Zhang, 2016) provides functionality for state and parameter estimation, and MPC based on the identified simplified RC model. BeSim (Drgoña, 2019) provides functionality for fast development, tuning, and simulation of model order reduction, state estimation and MPC based on linearized white-box building models from Modelica (Picard et al., 2015) and optimization toolbox Yalmip (Löfberg, 2004). Modeling environment (ME) (Zakula et al., 2014) is a modular simulation tool for buildings that employs MPC based on TRNSYS for virtual building modeling and Matlab for MPC implementation. TACO (Jorissen et al., 2018a) automates the process of setting up an MPC from a white-box model in Modelica. The nonlinear MPC is formulated using the CasADi (Andersson, Gillis, Horn, Rawlings, & Diehl, 2018) framework and solved with the JModelica (Modelon, 2017) optimizer.

8.3. MPC solvers

Today, dozens of optimization solvers are available, both commercially and free, for a wide variety of problems. Tables 13 and 14 provide an overview of the most significant solvers suitable to solve MPC problems on desktop and embedded platforms, respectively. The used acronyms stand for Linear Programming (LP), Quadratic Programming (QP), Mixed-Integer Linear Programming (MILP), Mixed-Integer Quadratic Programming (MIQP), Mixed-Integer Nonlinear Programming (MINLP), Nonlinear Programming (NLP), Second Order Cone Programming (SOCP), Semi Definite Programming (SDP), Multi-Parametric Linear Programming (mplP), and Multi Parametric Quadratic Programming (mpQP), respectively.

Progress in the solution techniques and an increase in the computational power of the desktop platforms allow us to efficiently solve large-scale optimization problems with up to hundreds of thousands of variables. In the case of embedded platforms, several tools have automated and optimized code generation features supporting different languages (e.g., C, C++ or Python) for rapid development and deployment of the MPC controllers in real-world applications. These embedded applications are, however, mostly suitable for small, fast dynamic systems, which are different from the large and slow dynamics of the buildings. Nevertheless, their efficiency and cheap computational power could be harnessed in large buildings for local control loops, or small-scale residential applications of MPC.

In the case of data-driven approximate MPC, the machine learning models can be trained by solving a wide variety of optimization problems offline. The type of OP to be solved depends on the used models (e.g., neural networks, regression trees, etc.) and their specification. While dedicated algorithms also exist to train more complex and specific ML

Table 10

Summary of the co-simulation tools and interface standards to bridge BES programs with other simulation platforms and control-oriented programming languages.

Co-simulation tool or interface standard	Free	Interface for					
		ESP-r	EnergyPlus	TRNSYS	Modelica	MATLAB	Python
BCVTB (Wetter & Haves, 2008)	•	•	•	•	•	•	–
MLE+ (Bernal et al., 2012)	•	–	•	–	–	•	–
OpenBuild (Gorecki et al., 2015)	•	–	•	–	–	•	–
FMI (Broman et al., 2013; Pang et al., 2016)	•	–	•	•	•	•	•

Table 11

Summary of selected control-oriented building modeling tools. The acronyms are explained in the text.

Tool	Free	Language					Paradigm		
		Modelica	MATLAB	Python	TRNSYS	R	White	Grey	Black
Modelica Linearization (Picard et al., 2015)	•	•	–	–	–	•	–	–	
Simscape Library (Lapusan et al., 2016)	•	–	•	–	–	•	–	–	
ME for MPC (Zakula et al., 2014)	–	–	•	–	•	–	•	–	
OpenBuild (Gorecki et al., 2015)	•	–	•	–	–	–	•	–	
IDENT (Jiménez et al., 2008)	•	–	•	–	–	–	•	–	
BRCM Toolbox (Sturzenegger et al., 2014)	•	–	•	–	–	–	•	–	
BLDG (Kircher & Zhang, 2016)	•	–	•	–	–	–	•	–	
BASBenchmarks (Cauchi & Abate, 2018)	•	–	•	–	–	–	•	–	
Grey-box Toolbox (De Coninck et al., 2016)	•	•	–	•	–	–	•	–	
MPCPy (Blum & Wetter, 2017)	•	•	–	•	–	–	•	–	
LORD (Gutschker, 2008)	•	–	–	–	–	–	•	–	
CTSM-R (Kristensen et al., 2004a)	•	–	–	–	–	•	–	–	
MoCaVa (Bohlin, 2003)	–	–	•	–	–	–	–	•	
System Identification Toolbox (Ljung, 2006)	–	–	•	–	–	–	–	•	

models (Sra, Nowozin, & Wright, 2011), most of the problems in this domain are solved via gradient descent algorithms. However, they can also be solved by using general purpose solvers listed in Table 13.

9. Practical implementation of MPC in buildings

The ambition of this section is to provide a complete overview of the

key components and nuances of practical implementation of MPC in buildings. A schematic representation of the presented framework corresponding to the structure of this section is given in Fig. 9. The three key elements are: the control configuration discussed in Section 9.1, the SCADA architecture presented in Section 9.2, and the communication infrastructure described in Section 9.3. Section 9.4 concludes the topic and provides experience-based practical guidelines for MPC

Table 12

Overview of the modeling software for optimization problems suitable for formulating and solving MPC problems.

Tool	Free	MATLAB	Python	Julia	Modelica	C/C+	Java	Tool-specific language
Yalmip (Löfberg, 2004)	•	•	–	–	–	–	–	–
CVX (Grant & Boyd, 2014)	•	•	–	–	–	–	–	–
MPC Toolbox™ (Mathworks, 2020)	–	•	–	–	–	–	–	–
MPC Tools Package (Amrit, 2008)	•	•	–	–	–	–	–	–
Hybrid Toolbox (Bemporad, 2004)	•	•	–	–	–	–	–	–
MPT3 (Herceg, Kvasnica, Jones, & Morari, 2013)	•	•	–	–	–	–	–	–
NMPC Tools (Rawlings & Amrit, 2008)	•	•	–	–	–	–	–	–
ACADO (Houska, Ferreau, & Diehl, 2011)	•	•	•	–	–	•	–	–
ACADOS (Verschueren et al., 2019)	•	•	•	–	–	•	–	–
CasADi (Andersson et al., 2018)	•	•	•	–	–	•	–	–
APMonitor (Hedengren, Shishavan, Powell, & Edgar, 2014)	•	•	•	•	–	–	–	–
HPMPC (Frison, Sørensen, Dammann, & Jørgensen, 2014)	•	–	–	–	–	•	–	–
CVXPY (Diamond & Boyd, 2016)	•	–	•	–	–	–	–	–
Pyomo (Hart et al., 2017)	•	–	•	–	–	–	–	–
Picos (Sagnol & Stahlberg, 2018)	•	–	•	–	–	–	–	–
OpenModelica (Fritzson et al., 2018)	•	–	•	–	•	•	–	–
JModelica.org (Modelon, 2017)	•	–	•	–	•	•	•	–
JuMP (Dunning, Huchette, & Lubin, 2017)	•	–	–	•	–	–	–	–
AMPL (Fourer, Gay, & Kernighan, 2002)	–	–	–	–	–	–	–	•
GAMS (Rosenthal, 1988)	–	–	–	–	–	–	–	•
Building control oriented								
OpenBuild (Gorecki et al., 2015)	•	•	–	–	–	–	–	–
BRCM toolbox (Sturzenegger et al., 2014)	•	•	–	–	–	–	–	–
EHCM toolbox (Darivianakis, 2020)	•	•	–	–	–	–	–	–
BLDG (Kircher & Zhang, 2016)	•	•	–	–	–	–	–	–
BeSim (Drgoña, 2019)	•	•	–	–	•	–	–	–
FastSim (Arroyo et al., 2018)	•	–	•	–	•	–	–	–
MPCPy (Blum & Wetter, 2017)	•	–	•	–	•	–	–	–
GenOpt (Coffey et al., 2010)	•	–	–	–	–	–	•	–
ME for MPC (Zakula et al., 2014)	–	•	–	–	–	–	–	–
TACO (Jorissen et al., 2018a)	–	–	–	–	•	•	–	–

Table 13

Overview of the most notable optimization solvers suitable to solve MPC problems on a desktop platforms.

Solver	Free	LP	QP	MILP	MIQP	MINLP	NLP	SOCP	SDP
CPLEX (ILOG, 2007)	–	•	•	•	•	–	–	•	–
Gurobi (Gurobi Optimization, 2012)	–	•	•	•	•	–	–	•	–
MOSEK (Andersen & Andersen, 2000)	–	•	•	•	•	–	–	•	•
XPRESS (Berthold, Farmer, Heinz, & Perregaard, 2018)	–	•	•	•	•	–	–	•	•
SeDuMi (Sturm, 2003)	•	•	•	–	–	–	–	•	•
SDPT3 (Toh, Todd, & Tütüncü, 1999)	•	•	•	–	–	–	–	•	•
CVXOPT (Andersen & Vandenberghe, 2018)	•	•	•	–	–	–	–	•	•
GLPK (Makhorin, 2012)	•	•	–	•	–	–	–	–	–
IPOPT (Wächter & Biegler, 2006)	•	•	•	–	–	–	•	–	–
ALGLIB (Bochkanov, 2019)	•	•	•	–	–	–	•	–	–
Artelys Kitro (Byrd, Nocedal, & Waltz, 2006)	–	•	•	–	–	–	•	–	–
SNOPT (Gill, Murray, & Saunders, 2005a)	–	•	•	–	–	–	•	–	–
APOPT (APOPT, 2020)	–	•	•	•	•	•	•	–	–
BARON (Sahinidis, 2017)	–	•	•	•	•	•	•	–	–
Bonmin (Bonami et al., 2005)	–	•	•	•	•	•	•	–	–
WORHP (Büskens & Wassel, 2013)	•	•	•	•	•	•	•	–	–
GenOpt (Wetter, 2001)	•	–	–	–	–	•	•	–	–

implementation in real buildings.

9.1. Control configuration

The following terminology is used in this section for networked control systems. See Fig. 10 for conceptual diagrams.

- Centralized control: a centralized agent (or controller) regulates an entire system.
- Decentralized control: each agent controls its own subsystem without communicating with neighbors.
- Distributed control: multiple agents are distributed horizontally over a whole system. There is no central agent.
- Hierarchical control: multiple agents are arranged in a hierarchical tree to control an entire system.

Centralized MPC The centralized MPC scheme solves a plant-wise optimization problem in a central computer and has been the primary method in the building sector. However, for buildings which are composed of a large number of dynamic subsystems composing a complex topological network, applying centralized MPC could be challenging due to increased computational complexity and reliability issues (Jamshidi, 1996). In this case, it is favorable to decompose a large centralized optimization problem into smaller multiple subproblems, which motivates configurations of decentralized, distributed and hierarchical MPCs.

Decentralized MPC In the decentralized MPC scheme, each local controller is designed as MPC and optimizes its own performance index without considering costs and dynamic influences of the others. Therefore, overall performance could be quite poor, especially for strongly coupled systems, though the communication overhead is minimal (Rawlings & Mayne, 2009).

Distributed MPC In the distributed MPC approach, each local controller which regulates its own subsystem solves a subproblem and communicates with others in order to improve the entire system performance. The information exchange consist of predicted state or control inputs so that any local controller can predict better by considering influences of neighboring systems. The communication can occur only once at each sampling time (non-iterative algorithms), i.e. only after each local optimization problem is solved, or many times within the sampling time (iterative algorithms) (Scattolini, 2009). Iterative algorithms could show better performance in terms of convergence and closed loop stability, but have higher communication burdens, causing concerns about communication delays, overloads and transmission package losses (Camponogara, Jia, Krogh, & Talukdar, 2002).

In the literature of process control, the non-cooperative and coop-

erative MPCs (Rawlings & Mayne, 2009; Venkat, Rawlings, & Wright, 2005) are the most popular distributed MPC methods. Both of them optimize control inputs to minimize a global index in the form of $\sum_{k=0}^{N-1} \sum_{i=1}^S \ell_i(x_k^i, u_k^i) + \sum_{i=1}^S \ell_i(x_N^i)$ where ℓ_i is the stage cost for the i th subsystem. Note that the cost function is not separable² when control inputs are the only optimization variables (because of dynamic couplings between subsystems), although separable with respect to the state and control inputs. During the optimization phase, each MPC predicts the state evolution considering dynamic couplings to neighbors assuming that input profiles received from neighbors are fixed. The key difference between the non-cooperative and cooperative MPCs is that local controllers of the cooperative MPC tend to minimize the same global cost function, while those of non-cooperative MPC tend to minimize individual cost functions, i.e. the i th MPC minimizes $\sum_{k=0}^{N-1} \ell_i(x_k^i, u_k^i) + \ell_i(x_N^i)$ (Moroşan, Bourdais, Dumur, & Buisson, 2011). In cooperative control, the distributed optimization problems are equivalent to the centralized MPC problem and are solved iteratively. Therefore, the cooperative control guarantees global performance, such as constraint feasibility, convergence, optimality and closed loop stability. See (Stewart, Venkat, Rawlings, Wright, & Pannocchia, 2010) for detailed proofs.

Hierarchical MPC The hierarchical control configuration is particularly useful when coordination between local controllers is needed in order to improve overall performance, or control actions for different time scales need to be decided (Scattolini, 2009), e.g. an upper layer computes optimal temperature setpoints in an economic sense while lower layers focus on setpoint tracking.

To design a coordinator (upper-level) and local MPCs (lower-level), the dual decomposition method (Jamshidi, 1996) or the Alternating Direction Method of Multipliers (ADMM) (Boyd et al., 2011) can be employed. Both methods solve a global optimization problem indirectly by solving the Lagrangian dual problem and adopt the dual ascent method (Bazaraa, Sherali, & Shetty, 2013). The key to decomposing the primal MPC problem is that coupled dynamic equations can be separated in the Lagrangian function when the primal objective function is separable. If the dynamics are linear and the objective function is convex on a convex compact set, the dual and primal problems are equivalent (the strong duality theorem (Boyd & Vandenberghe, 2004)). In the hierarchical control scheme, the upper-level controller represents the dual ascent step, and hence vertical communication between the upper and all lower-level controllers are necessary. For the ADMM,

² An objective function, $f(x^1, \dots, x^S)$, is called *separable*, if f can be expressed as a sum of functions of the individual variables of x^1, \dots, x^S , i.e. $f(x^1, \dots, x^S) = \sum_{i=1}^S f_i(x^i)$

Table 14
Overview of the most notable optimization software tools suitable to solve MPC problems on embedded platforms.

Solver	Free	Code generation	LP	QP	mpLP/mpQP	MILP/MIQP	NLP
OOQP (Gertz & Wright, 2003)	•	–	•	•	–	–	–
qpOASES (Ferreau, Kirches, Potschka, Bock, & Diehl, 2014)	•	–	•	•	–	–	–
ECOS (Domahidi, Chu, & Boyd, 2013)	•	–	•	•	–	–	–
CVXGEN (Mattingley & Boyd, 2012)	•	•	•	•	–	–	–
FiOrdOs (Ullmann, 2011)	•	•	•	•	–	–	–
FORCES PRO (Embotech, 2020)	–	•	•	•	–	–	–
Falcopt (Torrisci et al., 2017)	•	•	•	•	–	–	•
Toolbox							
ACADO (Houska et al., 2011)	•	•	•	•	–	–	•
Hybrid Toolbox (Bemporad, 2004)	•	•	•	•	•	•	–
MPT3 (Herceg et al., 2013)	•	•	•	•	•	•	•
MPC Toolbox™ (Mathworks, 2020)	–	•	•	•	•	•	•

communications between lower-level MPCs which represents the Gauss-Seidel algorithm is additionally required. Note that the upper-level controller is not MPC since it does not predict future behaviors.

When both short-term and long-term behaviors of a system are concerned, a hierarchical control system can be designed so that an upper layer regulator acts on lower frequencies and computes a control action concerning a long-term effect, while lower layer controller(s) act on higher frequencies and are responsible for short-term behavior(s) (Scattolini & Colaneri, 2007). This approach is related to cascade controls in which the inner and outer control loops are associated to faster and slow dynamics, respectively. One of the most significant advantages of this control approach is that it can substantially improve control performance under disturbances and nonlinearities associated with the inner loop, and that control designs can be separated when the upper layer works on a sufficiently low frequency range, say a factor of five or more in terms of inner close-loop system (Skogestad & Postlethwaite, 2007).

9.1.1. Review of applied MPC architectures for HVAC systems

Centralized MPC in buildings In the building control domain, the majority of the theoretical work and simulation-based case studies consider centralized MPC architecture. However, there are not many truly centralized MPC solutions that have been considered for application in practice. The main reason is non-standardized use of the communication protocols preventing straightforward access to the field layer of the SCADA architecture, which will be discussed in the following Section 9.2. Moreover, keeping the low-level RBC and PID loops intact may improve the operational robustness of the hierarchical MPC implementation by avoiding a single point of failure in the control system. Despite this fact, the paper (Jorissen et al., 2018b) presents an implementation strategy of centralized high-fidelity MPC for the real office building in Belgium.

Decentralized MPC in buildings The design of decentralized MPC for thermal control of buildings based on reduced order models and state observers was studied in Chandan and Alleyne (2014). A methodology determining an appropriate decentralized architectures, which provide a satisfactory trade-off between control performance and robustness for building control was proposed in Chandan and Alleyne (2013). An agent-based approach for distributed monitoring and model-based control of an office building was presented in Davidsson and Boman (2005). A graph theory-based approach and consensus protocols applied to thermal modeling of buildings was presented in Moore, Vincent, Lashhab, and Liu (2011). However, all of the aforementioned decentralized studies on building modeling and control remain in the simulation domain.

Distributed MPC in buildings An application of non-cooperative MPC can be found in Ferrarini, Mantovani, and Costanzo (2014); Moroşan, Bourdais, Dumur, and Buisson (2010) and those of cooperative MPC-like schemes³ are found in Moroşan et al. (2011); Putta, Zhu, Kim, Hu, and Braun (2012); Putta, Kim, Cai, Hu, and Braun (2014). For those studies, the objectives are to distribute multi-zone building loads to multiple units in an optimal way. The dynamic interactions are due to thermal couplings between zones through convective or conductive heat transfer. More precisely, in Putta et al. (2012), the case study building has two coupled zones and each zone is served by a separate air handling unit (AHU). Two local MPCs were designed to control the individual AHUs targeting to reduce operating costs for the entire system. in Putta et al. (2014), a nine-zone building served by one AHU is considered. Ten local MPCs were designed, where nine of them control air flow rates of individual variable air volume boxes (VAVs) to regulate nine zone air temperatures. The remaining MPC optimizes the supply air temperature setpoint. Similarly, in Moroşan et al. (2011), four distributed MPCs were designed where three of them manipulate their own local electric heaters while the remaining one controls a central heating system in order to optimally reduce the electricity cost while meeting individual zonal heating loads. For the last two cases, the control configurations are not purely distributed MPC and modifications of the cooperative MPC were necessary since there is a global variable which influences all subsystems, i.e. the supply air temperature of the central heating unit, resulting in a different cost structure compared to that of the cooperative MPC. The proposed MPCs have two-level pyramid structures where the upper-level controller optimizes the global variable based on information from lower-level controllers, while multiple lower-level controllers solve their own problems in a cooperative-MPC approach using the optimized global variable.

Hierarchical MPC in buildings In the literature of the building control, many applications adopt traditional hierarchical control architectures as discussed in previous section. Examples of such applications can be found in Abreu, Bourdais, and Guéguen (2018); Kim and Braun (2018); Ma, Anderson, and Borrelli (2011). in Ma et al. (2011), a three-zone building served by a VAV AHU system is considered. A standard dual decomposition method (Jamshidi, 1996) for decomposing MPC problems was adopted to design a hierarchical control system where lower-level MPCs regulate individual zone air temperatures in an economic way while the upper-level optimizer coordinates possible conflicts in local decisions. in Kim and Braun (2018), the hierarchical MPC system was designed for optimal demand response for a building served by multiple on-off stage HVAC units. The upper layer MPC predicts longer-term performance (about a day) and optimizes thermostat temperature setpoints to shift building loads in response to a utility price signal, while the lower layer MPC predicts short-term performance

³ It means that each local MPC has a shared objective function and considers influences of neighbors like the cooperative MPC, although decomposition methods and implementation details are different.

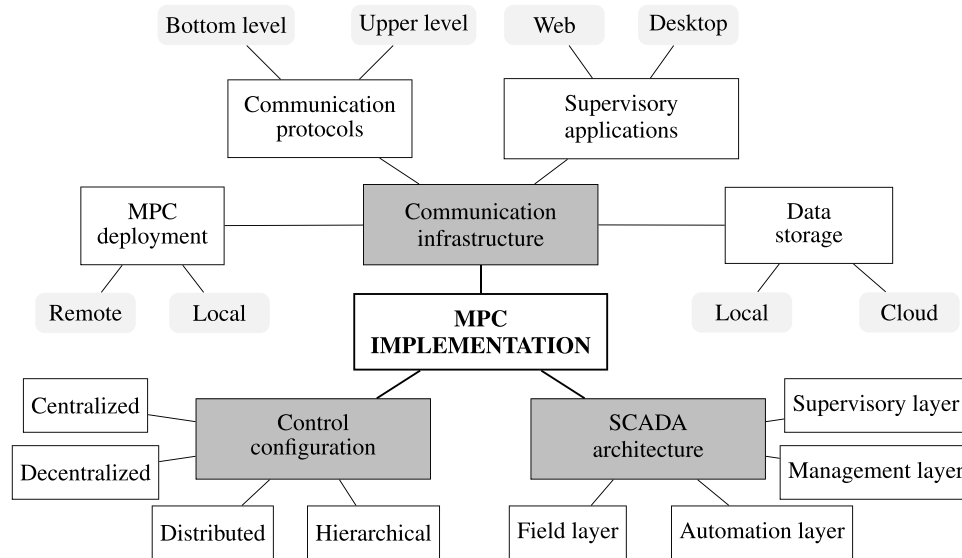


Fig. 9. General framework for the MPC implementation in buildings.

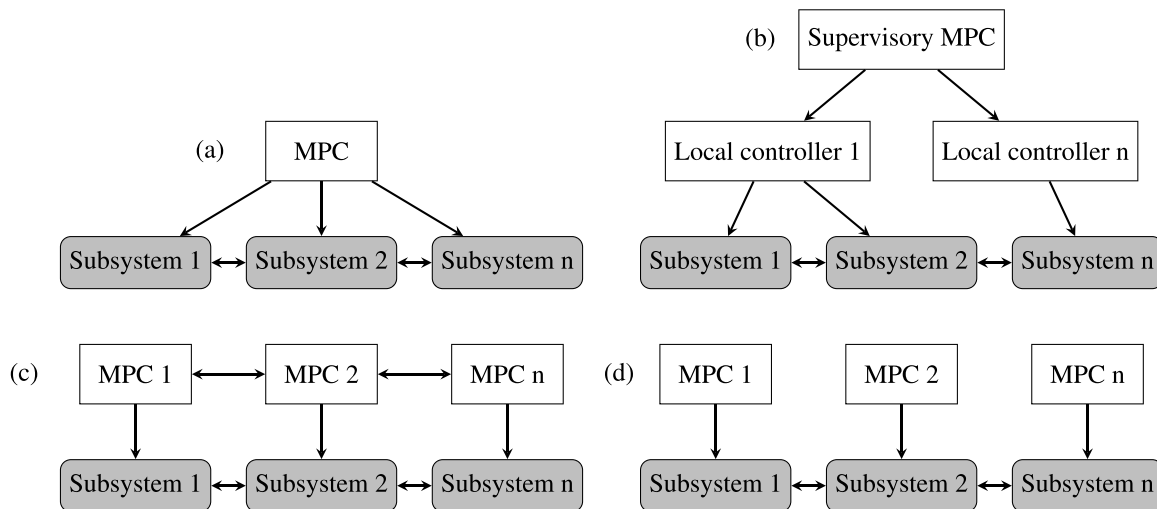


Fig. 10. Schematic of a centralized (a), hierarchical (b), distributed (c) and decentralized (d) MPC control configuration. Extension of figure given in Serale et al. (2018).

(about an hour) and supervises multiple units to prevent simultaneous unit activation during a precooling period, which could cause an unnecessarily higher demand charge. in Abreu et al. (2018), the upper layer MPC optimizes the setpoint while the lower layer MPCs track the setpoint. Recently, studies of applying ADMM to decompose MPC or general optimization problems for buildings become popular and are found in Cai, Braun, Kim, and Hu (2016a,b); Cai, Kim, Putta, Braun, and Hu (2015); Gupta, Kar, Mishra, and Wen (2015); Hou, Xiao, Cai, Hu, and Braun (2017); Moroşan et al. (2011); Xiao, Hou, Cai, and Hu (2018).

Concluding remarks on MPC architecture in buildings Despite a large number of MPC studies, distributed or hierarchical-distributed MPC schemes got relatively little attention from the building HVAC control field. This may be due to lack of practical needs of distributing computational loads. In other words, many MPC problems in building HVAC systems could be handled in a centralized way. In addition, the sufficient conditions for convergence, i.e. convex functions for objective and inequality constraints and a linear structure for equality constraints, make it difficult to use distributed algorithms for practical building controls where HVAC systems exhibit nonlinear and non-convex characteristics and constitute nonlinear equality constraints. However,

because building systems need to be integrated with renewable energy resources, energy storage systems and networks (electric, thermal, gas), and because the study of convex approximations is progressing, e.g. (Atam & Helsen, 2015), in the near future it is expected that there are more opportunities of applying distributed and/or hierarchical MPCs for building controls.

9.2. SCADA architecture

Supervisory control and data acquisition (SCADA) is a standard architecture to define the different layers of hierarchical control systems. SCADA systems are widely used in various fields, such as process control, energy, and power systems operation, and have recently gained a lot of importance for the control and data acquisition of the so-called Building Automation Systems (BAS). One of the main advantages of using a SCADA configuration is that the different layers of control and communication flows can be depicted sequentially, in a much more structured and organized way. Another advantage is that other automated systems used in the building can be integrated into one single platform (i.e. HVAC, security, lighting or gas automation systems),

which makes the management of the whole installation more effective (Figueiredo & Costa, 2012). A SCADA system for building control and operation typically consists of four different layers (see Fig. 11):

Supervisory layer: the highest layer of the control architecture, where MPC is normally executed. It also includes all *clients* that interact with the system for the purpose of top-management activities. For example, supervisory control or data-analysis by means of visual interfaces used to monitor the whole building’s performance.

Management layer: includes one or several *servers* that allow the interaction between the higher and the lower layers of the control architecture. It is also used to conduct preliminary monitoring and preprocessing of information, as well as to store data by means of local or online databases. This layer includes all Building Management Systems (BMS) that are normally used to manage and control modern building installations.

Automation layer: integrates all local controllers that allow the execution of primary plant control by using conventional control strategies, like PID and RBC. All different modules collecting the measurements from the building process downstream are also included in this layer.

Field layer: the lowest layer of the control architecture. It includes all physical components, sensors and actuators.

It is important to outline that the division between layers of control can be apparent in software, hardware or a combination of both. This will depend on the communication infrastructure which is tackled in the next section.

9.3. Communication infrastructure

Communication is yet another crucial element of any practical control implementation. The importance can be emphasized if we would put the whole building control concept into a human body analogy. The building envelope would then be the torso, heating/cooling capacities the digestive system, air handling units (AHU) the respiratory system, piping the blood vessels, and pumps the heart. The SCADA infrastructure would be the nervous system, control configuration the wiring of the brain, and the MPC formulation its mental program. The communication infrastructure would represent the electrochemical signals traveling throughout the pathways of the nervous system, carrying the information from the subconscious level of low-level control to the conscious level of supervisory applications, while storing the data in the memory represented by a database.

9.3.1. MPC deployment

In SCADA-based control systems, the interaction between MPC and the building is implemented in a *client-server* model. A *client* can be defined as a device or computer program that executes the MPC formulation and accesses the building by means of a *server*; which can be seen as a device or computer program that acts as a bridge of communication between MPC and the rest of the building installation. There are two main configurations and networking typologies for the implementation of MPC in a client-server model: local and remote configuration.

Local: The MPC algorithm is executed in the same building installation where the control is performed. Hence, the division between client and server is only apparent in software (Afram & Janabi-Sharifi, 2017; Skeledzija et al., 2014). Local configurations, however, lack flexibility, since MPC developers need to be present in the building during the commissioning phase. Moreover, any modifications in the controller’s formulation will have to be applied locally to the building, which might result in a quite tedious and ineffective process.

Remote: The MPC algorithm is executed remotely from the building installation where the control is performed. The division between client and server is apparent in software but also in hardware, as two separate devices are normally implemented. Internet or other wireless communications are used to interact with the building, see e.g. (Gwerder, Gyalistras, Sagerschnig, Smith, & Sturzenegger, 2013; Ma, 2012). Remote configurations have several advantages, such as increased flexibility and interoperability from multiple platforms and devices. However, the disadvantage is the need for secure and stable communication channels.

Regarding MPC solvers, for practical installations, they are being deployed using several programming languages, such as C++, Python, Julia, or even JavaScript. In the research domain, however, the MPC algorithm can often run in MATLAB, with limited industrial applicability due to the associated software costs.

9.3.2. Communication protocols

In general we can differentiate between two levels of communication and their corresponding protocols, bottom and upper-level:

Bottom level: communication on the lowest layers of control, for example local controllers and HVAC actuators.

Upper level: communication on the highest layers between MPC and the local controllers of the building by means of a server. Servers can be understood as ‘interpreters’ that translate all inputs coming from the MPC into a language that local controllers can understand, and vice-versa (Nyvit, 2011).

In recent years, plenty of communication protocols have been developed for the purpose of Building Automation Systems (BAS). They can be grouped into two main categories: closed and open protocols.

Closed protocols: based on proprietary communication structures developed by each manufacturer separately, usually tailored to particular applications, hence they often lack versatility and flexibility (Bovet, Ridi, & Hennebert, 2014).

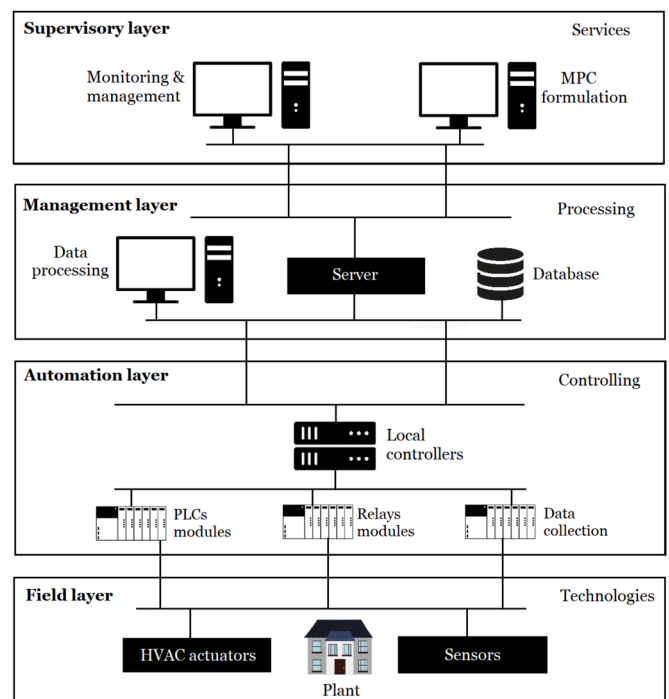


Fig. 11. SCADA-based control architecture for building control and operation using MPC.

Open protocols: based on standard specifications which leads to clear advantages, such as greater flexibility of implementation and interoperability of devices from hundreds of different vendors (Nyvlt, 2011).

Table 15 provides a compact overview and classification of selected communication protocols specifically designed or reported to be used in building control applications.

The communication challenges in real buildings proliferate with the scale, use of multi-vendor devices, different protocols, and geographical distribution of the units. In recent years, modern communication platforms for distributed sensing and control systems have been under development to mitigate those challenges. Examples of such platforms are the commercial Niagara Framework® (Tridium, 2019) developed by Tridium’s Inc. belonging to the Honeywell group, or open-source Volttron™ (Akyol et al., 2016) developed by Pacific Northwest National Laboratory.

9.3.3. Supervisory applications

Supervisory applications are implemented by means of Human Machine Interfaces (HMIs), which allow monitoring the MPC performance using visual and graphical interfaces. In a SCADA-based architecture, HMIs act as clients that connect to the building server. They can be divided into two groups: desktop, and web-based applications.

Desktop applications: stand-alone software tailored to one particular computer, which only can be accessed by a restricted number of users. They offer more privacy, security, and usually also better performance than web-applications, but they lack the portability, scalability, and flexibility of implementation (Pop, 2008), which are crucial for integration with other automation systems, using the same BMS.

Web-based applications: accessed through the network by multiple clients and devices simultaneously, exploiting the use of internet and web-services. They are much more flexible because they are platform-independent and are not tailored to one specific device. Moreover, they are more scalable and can be easily integrated into the whole BMS of the building. For obvious reasons, they are more suitable in a remote configuration. Some disadvantages of using web-applications are slower performance, internet-dependency, or security risks compared to a desktop application (Pop, 2008).

9.3.4. Data storage

The storage of data has significant importance for the implementation of MPC in buildings. MPC developers make use of historical data for three main purposes: (i) to develop and calibrate the building model used by MPC; (ii) to keep track of variables that are used as parameters in the MPC formulation (i.e. weather-data, electricity prices, etc.); (iii) and to analyze the performance of the controller. Regarding their implementation, databases can be classified into local and cloud databased.

Local databases: stored in a dedicated device or computer and can only be accessed by a limited number of applications.

Cloud databases: make use of a web-server to store data, which is connected to the Internet and can be accessed remotely by multiple applications.

For MPC implementations cloud databases are usually preferred above local databases due to their flexibility of operation and less tedious setting-up phase. Moreover, a common practice today is to outsource the storage of data using an external server from a third-party, normally referred to as a cloud provider. As a result, cloud services provide a reduction in the creation and maintenance costs of the database, better scalability, and more safety towards losing backups (Li, Li, Vrabie, Bengea, & Mijanovic, 2014). The downsides of the cloud-based

solutions are potential cyber-security issues, which may often impose more secure local implementation.

Regarding the model they implement, databases can be classified into relational and non-relational databases (Gyorodi, Gyorodi, & Sotoc, 2015).

Relational databases: are based on a Structure Query Language (SQL) to store and retrieve data from the database in a really organized way using tables. They count on rigid schemes that need to be designed before data is stored and are quite difficult to change afterward.

Non-relational databases: do not use relational management systems, hence data is not stored using tables, nor rigid schemes. They offer big advantages compared to relational databases, such as superior performance, better scalability and more flexibility of implementation.

Relational databases are widely implemented for all kinds of applications showing a pretty good performance. However, recent studies have proven that they present some limitations, especially when dealing with large amounts of data and transaction (Gyorodi et al., 2015). Thus, big-data organizations (e.g. Google, Amazon or Facebook) are starting to use non-relational databases to store their data. However, this is still yet a relatively new direction and the majority of MPC implementations reported in the literature opted for the relational databases, see. e.g. (Fabietti, 2014; Skeledzija et al., 2014). However, for the future implementation of MPC, non-relational databases seem to be a better candidate, since the controller is expected to deal with big volumes of data, especially in large-scale buildings where a central database might be used for the whole installation.

9.4. Practical guidelines

This section summarizes practical aspects discussed in detailed in previous sections and extracts step by step guidelines for developing and implementing a successful MPC application for a real building. A general methodology is systematically shown in Fig. 12 covering the high-level workflow, starting with setting up the communication infrastructure, followed by control-oriented modeling, control configuration with MPC design and tuning, finalized by MPC deployment as a supervisory application in modern SCADA systems and closing the loop with communication setup in case of necessary modifications.

A more detailed and practically oriented flowchart is presented in Fig. 13. It encompasses the necessary actions and decisions of the whole MPC workflow from scratch to implementation in a real building. The preliminary phase starts with a feasibility analysis which should be based on controllability and measurability of the building via the

Table 15 Summary and classification of selected communication protocols used in building control.

Protocol	Standard	Bottom level	Upper level	Closed	Open	BAS oriented
Nikobus		•	–	•	–	–
iNels		•	–	•	–	•
BACnet	ISO 16484-5	•	–	–	•	•
KNX	ISO/IEC 14,543	•	–	–	•	•
Modbus		•	–	–	•	•
LonWorks	ANSI/CEA-709.1	•	–	–	•	•
M-bus	EN 13,757	•	–	–	•	•
OPC		•	•	–	•	–
TCP/IP	IETF	–	•	–	•	–
UDP	RFC 768	–	•	–	•	–
FTP	RFC 2428	–	•	–	•	–
HTTP/HTTPS	RFC 7230	–	•	–	•	–

installed building automation system (BAS). The second step is to evaluate the economic potential for a building of interest via return on investment analysis.

The design phase starts with the third step of the flowchart by setting up the real-time communication between the BAS and the supervisory computer for automated data logging and storage, as summarized in [Section 9.3](#). This automated communication functionality is a must for any real-time dynamic optimization scheme such as MPC, while historical data stored in databases serve for modeling and tuning purposes. Necessary data points need to be selected based on the design of the model and control architecture. However, today, such functionality still represents a bottleneck due to the large variety of used protocols, interfaces and BAS vendors with closed solutions.

The fourth step consists of modeling, as elaborated in [Section 3](#). First, engineers need to define the objectives, constraints and key performance indicators (KPIs) for performance assessment of the models and control strategies. Subsequently, a control-oriented model needs to be developed via dedicated software tools partially listed in [Section 8.1.3](#) and evaluated with respect to selected performance measures.

After construction of a sufficiently accurate model, a control configuration needs to be defined in the fifth step. If the selected configuration is realizable within the current communication infrastructure, then we proceed to the next step, else either control configuration is modified, or a list of available data points is extended.

In the sixth step, a control engineer initiates MPC design by formulating the optimal control problem, identifying the problem class and selecting the solution paradigm as described in [Sections 2, 5, and 6](#), respectively. Subsequently, appropriate design tools and solvers are selected, e.g. based on the lists given in [Sections 8.2, and 8.3](#), respectively. The implementation of MPC algorithms presented in [Section 4](#) follows.

After tuning and performance evaluation in closed-loop simulation studies, controllers with acceptable quality are selected for deployment in the seventh step of the workflow. MPC solvers need to be installed either locally or on a remote computational platform and integrated in the SCADA system of the building (see [Section 9.2](#)). The deployment phase consists of functionality tests, and installation of the user interface and backup solutions, such as watchdog timers, alarms and automatic fallback controller for recovering operation after failures. Only after this, the operation phase can be initiated in the final eighth step. The installed applications continuously monitor MPC functionality and if error handling logic is triggered, the operation autonomously switches to the fallback control strategy, typically represented by simple RBC logic or PID loops. Each operational failure is typically accompanied with alarm messages to the building operators.

10. Comparison and performance assessment

Comparison and performance assessment of MPC approaches are important to identify the most promising approaches and guide transition of MPC strategies from research to industry. However, a number of challenges exist that make such comparisons difficult. Therefore, this section outlines these challenges, reviews the literature on studies that have compared MPC approaches, and suggests the needs of a more unifying framework for such assessment.

10.1. Challenges

An initial challenge of comparing MPC approaches is the large variation any implementation can take compared to another. As presented in this paper, there are a number of factors and methods to consider for each of the many components of the MPC, creating a very large solution space. In addition, each application, whether it be a single zone, building, campus, or neighborhood, presents its own set of design and operation characteristics that may promote the use of one method over another. These include architectural design and construction,

climate, HVAC and lighting system design, occupancy and usage, system controllability, available measurements, data management, and control objective.

A second challenge of comparing MPC approaches is the relatively small number of field tests available, compared to the solution space of available approaches and applications. In such field tests reported in literature, it is common to document the performance of a single implementation for a particular application to demonstrate performance advantages over a more conventional method of control. It is important to point out that the choice and tuning of the benchmark controller has a direct influence on the improvements calculated for MPC. It is uncommon to consider and compare a range of methods. In addition, the real implementations are often not long-term studies, lasting weeks to months and not years, limiting the insight on how MPC strategies perform during all seasons, holidays, and other specialty types of days, as well as how much maintenance is required over time. Moreover, few studies report on or discuss implementation costs and payback periods.

A final challenge is defining the grounds for comparison. Common metrics are used in the literature associated with energy savings, operating costs savings, and occupant comfort improvement. However, other important bases of comparison of implementation and performance include computer hardware and software requirements, computation time, robustness to changing conditions, sensitivity to model and forecast uncertainty, data requirements, implementation effort, and installer expertise. Such a broad range of factors makes objective comparison difficult.

10.2. Literature

Studies that have compared specific MPC formulations are summarized in [Table 16](#). All studies were performed using simulation and the baseline for comparison tended to be a centralized, linear, deterministic MPC implementation, except for one study that compared the use of two different nonlinear optimization solver algorithms. Each study utilized metrics related to energy use or cost and thermal comfort, while some other metrics included computational burden and setpoint tracking error. The results of each study were consistent with the hypotheses presented for each test implementation. For instance, stochastic and robust MPC can significantly improve the handling of disturbance or model uncertainty with respect to maintaining comfort compared to deterministic MPC, with only a small loss in energy savings potential. Another example is that distributed MPC can lessen the computational burden and communication requirements of a centralized MPC, with only small losses in energy savings potential and comfort. Differences in the studies, however, make it difficult to compare the implementations among each other and to evaluate the scalability of each technique in practice. First, each study considered a different building design, construction, climate, and HVAC system. In addition, each study considered different periods of operation, ranging from one hour to one year.

Other studies have focused on comparing various factors and techniques related to thermal envelope model development for the MPC. In ([Blum et al., 2019b](#)) seven factors affecting the accuracy of thermal envelope models were identified and their subsequent effect on MPC performance was tested, including building design, model structure, model order, identification data set, identification data quality, identification algorithm, and software tool-chain. The study showed that model order and initial parameter guesses during identification have strong influences. In ([Sourbron et al., 2013b](#)) the effects of model order and training data on final MPC performance for a concrete core activated HVAC system were studied. Studies in [Picard et al. \(2017, 2016\)](#) showed that linearizing detailed models, rather than building grey-box models, is a technique that works well. Other authors [Harb, Boyanov, Hernandez, Streblov, and Müller \(2016\)](#); [Reynders et al. \(2014\)](#) studied the effect of building and HVAC system type, training data, model order, noise, and measured inputs on parameter identification accuracy. Study in [Reynders et al. \(2014\)](#) found that a fourth order model was

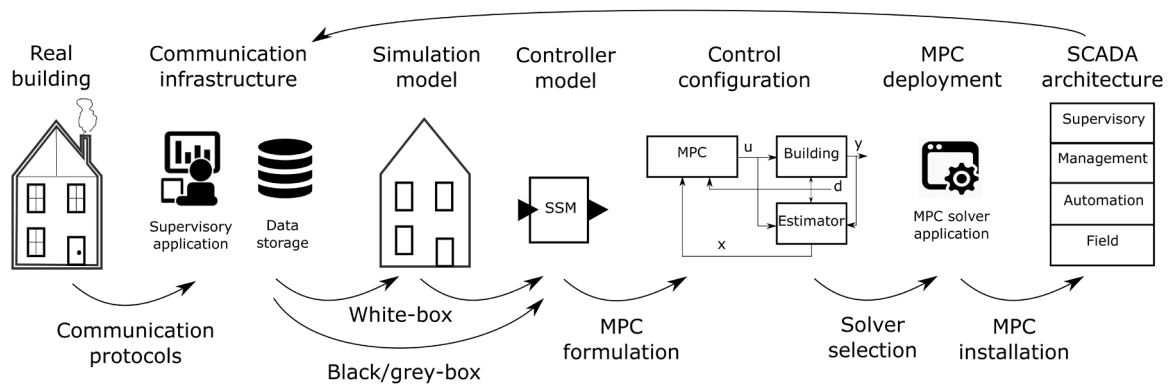


Fig. 12. A general methodology for modeling, design, and implementation of MPC in buildings based on (Drgoňa, Picard, & Helsen, 2020).

acceptable, while Harb et al. (2016) found that a second order model was acceptable, though neither tested these models in an MPC controller. Finally, (Vande Cavey et al., 2014) compared MPC performance with and without proper state estimation, showing the importance of using a well-tuned state estimator. Similar to the studies comparing specific MPC formulations, these studies suffer from not utilizing the same building cases or evaluation periods, making inter-study comparison difficult.

In addition to academic studies, Zurich (2020) and Cigler, Tomáško, and Široký (2013c) present tools developed to assess the performance of MPC. The BACTool (Zurich, 2020) represents a web-based tool that utilizes a large number of pre-calculated, yearly building energy simulation results to display performance indicators using MPC and one of two rule-based controllers. Users can build cases to compare from a number of inputs, including among them building construction, orientation, climate, HVAC system, and control type. Performance metrics that can be compared include energy use [kWh/m²], comfort [K-h], and peak demand [W/m²], as well as timeseries of indoor temperature, illuminance, blind position, and power demand of HVAC and lighting system components. In this way, users can evaluate the potential benefits of using MPC over rule-based control in a similar building project. Cigler et al. (2013c) presents BuildingLAB, a tool for illustrative and educational purposes related to MPC control in buildings. Users can change parameters such as prediction horizon, initial conditions, constraints, and objective function weights (e.g. of operating cost and discomfort) and execute simulations of building control using MPC, with optimal control results calculated upon execution using the given parameters. In this way, users can see the differences that result from changing parameters and gain intuition on expected performance.

10.3. Framework development

While the literature review presents a number of studies that compare two or more MPC formulation and modeling methods as well as tools that were designed to compare MPC performance among various conditions and parameter settings, performance and assessment of MPC lacks a unified framework designed to tackle the challenges outlined in the previous section. The literature studies are limited to the specific implementations and conditions under which they were compared, while the tools are limited to the building models and MPC approaches implemented by the tool designers. Instead, the framework needs to provide representative, yet bounded, testing conditions and scenarios which any control developer can use to test his/her individual approach. Such a framework is similar to the BESTEST (American Society of Heating Refrigerating & Air Conditioning Engineers, 2008), a set of building specifications and operating scenarios developed for benchmarking and comparison of building energy simulation tools. This can be implemented in the form of reference building models and simulation scenarios that represent a range of building and system types, are

implemented with the necessary dynamics for controls design and testing represented, are available for use by all MPC researchers and control developers regardless of expertise in building simulation modeling, can be simulated within a controlled, yet distributable, computing environment, and are independent from the control implementation. In addition, the framework needs a reference set of performance indicators to objectively compare MPC controllers with respect to all or a chosen subset of these metrics. The metrics should include operational performance, such as energy, cost, and comfort, as well as implementation metrics, such as computational requirements and data needs. In this way, as MPC developers use such a common framework to test their implementations, true comparison and assessment can be done relative to other approaches, and development of high-performing, cost-effective MPC approaches can be accelerated.

While development of such a simulation framework presents its own set of challenges, the task is being undertaken in Blum et al. (2019a), presenting a BOPTTEST framework (Building Operation Performance Test) consisting of various building types and software platform for the testing of advanced control strategies. The approach is similar to an existing platform called Alfalfa (National Renewable Energy Laboratory, 2020), which utilizes OpenStudio models for building simulation, implements a Project Haystack (2020) API to connect with potential test controllers and other data analytics platforms, and is designed to be a scalable web-service. The BOPTTEST framework differs in that it utilizes FMI and Modelica for building simulation, will have an API for also providing disturbance forecasts to MPC controllers, utilizes a controller-blocked synchronization scheme rather than a real-time synchronization scheme, and also produces reports on key performance indicators. In the future, the BOPTTEST framework aims to leverage the Alfalfa architecture to provide an industrial-strength tool for controls testing that provides the functionality of BOPTTEST with the scalable architecture of Alfalfa.

11. Conclusions

This paper provides a complete overview and unified framework of MPC for building climate control applications.

MPC theory and problem classification The process of MPC formulation starts with the definition of control loop variables and its interconnections via constraints, objective functions, and a control-oriented building model. The theory behind this process is compactly summarized at the start of this paper. The paper presents three algorithmic principles behind MPC which are essential for real-time implementation. In particular, we talk about receding horizon control (RHC), state estimation, and optimal control solution methods. The details of the particular case, such as building model type (e.g., linear, nonlinear), comfort index (e.g., comfort zone, PMV), and other factors penalized in the objective function, together with imposed constraints are the key building blocks of the MPC formulation. Based on these features, MPC

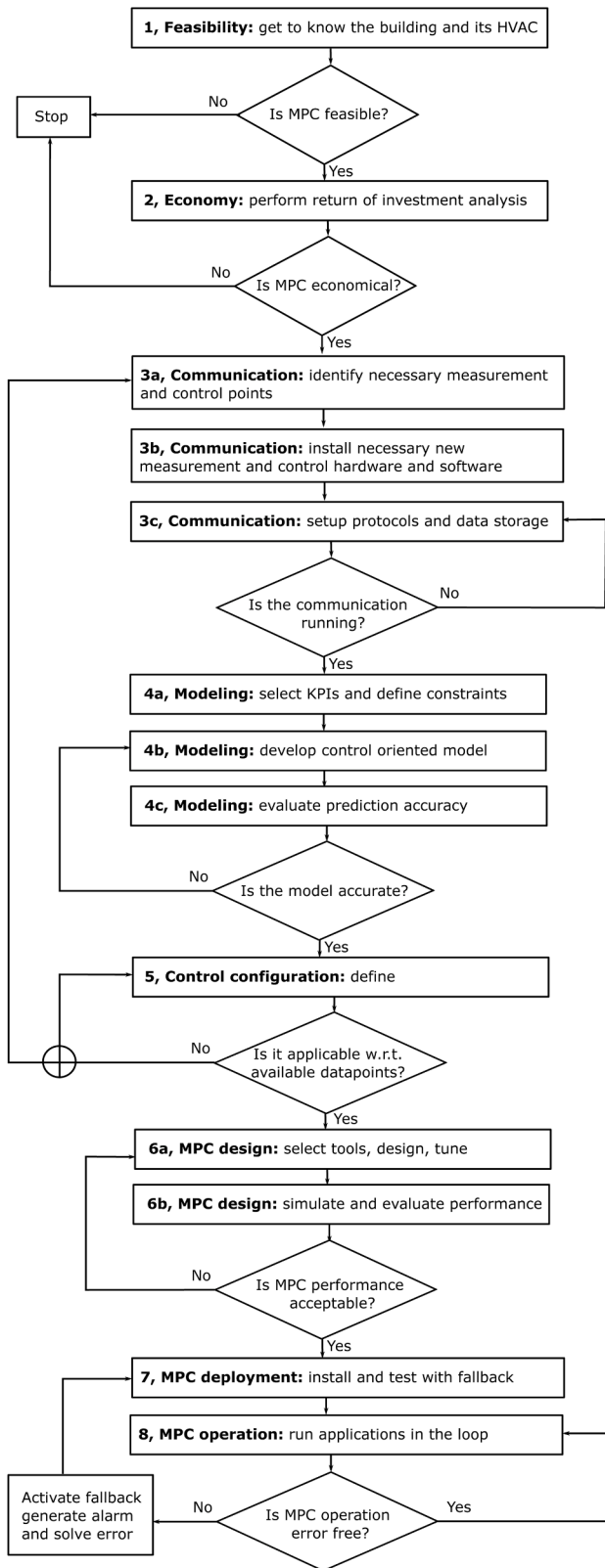


Fig. 13. Flowchart of MPC implementation in real buildings.

problems are classified into three important problem classes (linear, nonlinear, hybrid). Moreover, translation methods for direct optimal control and its use in association with each MPC class are discussed. A linear MPC formulation is computationally least demanding and thus easiest to implement. Many modeling tools support linear MPC with a

wide variety of examples and tutorials. Even though it has certain limitations regarding the formulation flexibility, it is the most commonly used MPC class in the building sector, mainly because the building envelope can be accurately approximated by linear dynamics. Nonlinear MPC provides us with higher flexibility in formulation and possibly increased performance, due to the incorporation of the nonlinear HVAC model. On the other hand, this comes with the cost of more elaborate modeling and increased computational demands for implementation. Hybrid MPC is useful when one needs to deal with integer decision variables or switching dynamics like heat pump modes, etc., a situation very common in building applications. For the cost of increased computational demands, it can provide increased performance compared to the more straightforward linear case.

Algorithmic solutions of MPC Three MPC solution techniques based on direct methods, i.e., implicit, explicit, and approximate MPC, have been discussed with their pros and cons. MPC approaches have been further differentiated based on their problem class, solution approach, and dimensionality of the problem defining the computational complexity of the optimization problem, and thus determining the feasibility, as well as hardware and software requirements for real implementation. Building climate control applications have specific characteristics, such as a large number of state variables and slow dynamics resulting in longer sampling times. For these reasons, and increased availability of computation power, in recent years, MPC is most often being implemented by solving a corresponding optimization problem online in an implicit way. The drawback of this approach is the necessity of available computation power and software dependencies associated with dedicated optimization solvers. Such a method is universally applicable, with the biggest return of investment potential associated with larger tertiary buildings due to the smaller ratio on the investment cost compared to the overall construction or renovation costs. Explicit MPC has been proven to be feasible so far only for small case studies, limiting its applicability in practice in multi-zone building control problems. The potential use of this approach is within low-level control tasks or decentralized single-zone control strategies, e.g. for individual apartments within a block or small residential houses. Approximate explicit MPC solutions appear to be a promising alternative also for large-scale problems providing memory-based control policies with low computational footprints. The main strength of this approach is its numerically robust operation due to lightweight computation requirements with minimal software dependencies and its applicability even on lower-level hardware. The main drawback of such an approach, however, is the requirement of the original MPC and the need for larger training datasets, which can be computationally demanding and hence time-consuming to generate. The theoretical part of the paper is finalized by the formalism of uncertainties in the MPC problem and methods conventionally used for their mitigation. In particular, these methods are offset-free MPC via state augmentation, robust MPC, stochastic MPC, adaptive MPC, and learning-based MPC.

Software tools for building modeling and control For all types of MPC formulations and implementation approaches, a wide variety of modeling and design tools and solvers are available. The wide variety of used modeling tools reflects the lack of understanding of what model formulation and level of detail is best suited for MPC in buildings. The practical part of this paper summarizes an extensive overview and conceptual comparison of dedicated software tools used for building modeling, (co-)simulation, MPC design tools, and available optimization solvers for both desktop as well as embedded platforms. The aim of this overview is to help the reader with a selection of the most appropriate tool from the broad range of options.

Practical deployment of MPC in buildings To facilitate a faster transfer of the technology into practice, a whole section is dedicated to key building blocks and aspects of practical implementation. The underlying implementation framework is defined consisting of the MPC configuration, SCADA architecture, and communication infrastructure. Four conceptual types of MPC configuration are considered, namely

Table 16
Studies comparing two or more MPC formulations.

Ref	MPC comparison	Case	Metric(s)	Result
Oldewurtel et al. (2012)	Stochastic (SMPC) vs. Deterministic (DMPC)	Single room with six variants of HVAC system, European locations, and building construction. Simulation period is one year.	Energy use [kWh/m ² /y] and comfort violations [Kh]	SMPC had comparable energy use (slightly higher) and comfort violations to best case DMPC.
Drgoňa et al. (2013)	Stochastic (SMPC) vs. Deterministic (DMPC)	Single room with simple heating and cooling. Simulation period is nine days.	Energy use [kWh] and comfort violations [% Simulation Samples]	SMPC had comparable energy use (slightly higher) and comfort violations to best case DMPC.
Ma et al. (2015)	Stochastic (SMPC) vs. Deterministic (DMPC)	Multizone VAV HVAC system in Berkeley, CA, USA. Simulation period is 55 days.	Energy savings compared to rule-based control [%], comfort improvement compared to rule-based control [%], thermal efficiency of HVAC system [–]	SMPC had comparable energy savings (slightly less) and comfort improvement over rule-based control to best case DMPC.
Maasoumy et al. (2014)	Robust (RMPC) vs. Deterministic (DMPC)	Single room in Houghton, Michigan, USA with ground-source heat-pump heating system. Simulation period is one day.	Energy use [kWh] and comfort violations [°C-h]	For intermediate levels of model uncertainty, RMPC outperformed DMPC, while DMPC is preferred for low levels of model uncertainty. If model uncertainty is very high, rule-based control is preferred.
Scherer et al. (2014)	Distributed (DisMPC) vs. Centralized (CenMPC)	Multiple zones each served by fan coil units served by common hot and chilled water central plants. Simulation period is one hour.	Integral of squared setpoint error [°C ²]	DisMPC was able to have similar setpoint tracking performance to CenMPC when central plant resources are limited.
Walker, Lombardi, Leseq, and Roshany-Yamchi (2017)	Distributed (DisMPC) vs. Centralized (CenMPC)	Three-zone open office in Cork, Ireland where each zone has radiator and window operation. Simulation period is nine hours.	Energy use [kWh], temperature and CO ₂ setpoint tracking (visually in plots), and normalized computational time [–].	DisMPC had comparable energy use to CenMPC (slightly higher) and similar temperature and CO ₂ tracking with less computational burden on each local controller.
Pcolka, Zacekova, Robinett, Celikovskyy, and Sebek (2014)	Nonlinear (NLMPC) vs. Linear (LMPC)	One zone building with radiant ceiling HVAC system in Prague, Czech Republic. Simulation period is three months.	Energy cost [Euro], maximal comfort violation [°C], and hours of comfort violation larger than 0.2 °C [h].	NLMPC outperforms LMPC by using less energy, having less maximum comfort violation, and having less total hours of discomfort.
Putta, Zhu, Kim, Hu, and Braun (2013)	Affine Quadratic Regulator (AQR) vs. Sequential Quadratic Programming (SQP)	Single room in Indiana, USA with VAV AHU and cooling plant.	Energy cost [\$/day], discomfort cost [\$/occupant/day, and computational time [s/decision]	AQR saved significantly on discomfort costs compared to SQP due to SQP sensitivity to initial guesses and local minima.
Drgoňa and Kvasnica (2013)	Setpoint Tracking (ST) vs. Comfort Tracking (CT) vs. Number Comfort Violation Min (CM)	Single room with simple heating and cooling.	Energy use [kWh], energy savings compared to rule-based control [%], and comfort violations [% Simulation Samples]	ST used most energy with good comfort control. CT used less energy with worse comfort control. CM used least energy with good comfort control.

centralized, decentralized, distributed, and hierarchical configurations, and their usability is discussed. Centralized MPC controls an entire system and is currently the most commonly used configuration in building applications. Decentralized configurations with multiple local MPCs are less favorable for buildings due to the loss of dynamic coupling between controlled subsystems. Distributed MPC represents a more favorable configuration and is based on solving a decoupled problem by communicating the local solutions to other sub-controllers to improve the entire system performance. Meanwhile, hierarchical configurations improve the overall performance when controlling the system over different time scales and including the subsystems with notable differences in their time constants. Examples of such systems are demand response control or long-term behavior of a ground source heat exchanger coupled to the short-term behavior of the building. The SCADA architecture defines the standards for modern industrial hierarchical control systems with four basic levels, which are widely adopted in modern buildings. In practice, this functionality is provided by the building automation systems (BAS) via commercial vendors like Honeywell, Johnson Controls, Priva, Siemens, Schneider Electric, ABB, or Delta Controls, to name the most prominent ones. A functional, automated, and full-scale communication outside the commercial BAS appears to be currently one of the tedious tasks of real MPC implementation. Although they can be built mostly on open standards, the problem lies in a large number of used communication protocols, closed commercial BAS software solutions, and lack of standardized interfaces which make the integration of hardware components from different vendors a real challenge. With practical cases in mind, clear guidelines and a flowchart for MPC implementation are provided for

researchers and early adopters of the technology. The fundamental steps of any successful application are based on preliminary feasibility and economic studies guiding the decision of whether to implement the MPC for a particular case or not. The design phase consists of setting up the communication, followed by control-oriented building modeling, control configuration selection, and MPC design and tuning. The operation phase consists of testing and deployment of the MPC algorithm with backup solutions.

Performance assessment of MPC in buildings Comparison and performance assessment of MPC in buildings plays an important role during the selection of an appropriate strategy for a particular application. However, due to the large solution space, there remain a number of challenges to be tackled on the roadmap towards generalized performance assessment methodology and tools. First initiatives are being taken to standardize this process in a scalable framework built upon next-generation building energy modeling tools that emulate the response of the building system to the MPC controller, using predefined performance indicators and application programming interfaces, all brought together in the BOPTTEST.

Market potential and future of MPC in buildings The practical aspects of integration of MPC algorithms with contemporary BAS create an opportunity for startup companies to deliver customized MPC solutions backed by universal SCADA platforms with multi-protocol, multi-manufacturer compatibility. Examples of such companies are e.g. DeltaQ, IES, BuildingIQ, Feramat Cybernetics, Energozentrum with their Mervis control as a service platform, or QCoefficient, Inc. which successfully operates cloud-based real-time white box MPC based on EnergyPlus models in a number of large commercial office buildings in

the US.

A big leap forward in MPC market penetration can also be made by implementing MPC applications into integrated software platforms, enabling the communication management and control of diverse systems regardless of manufacturer or protocol. The most notable of such communication platforms are the commercial Niagara Framework®, or the open-source Volttron™. It is very hard to make predictions, especially about the future. However, based on the advanced stage of basic research tackling the current bottlenecks of MPC, several pilot case studies, emerging startups, and awareness of the major companies in the field of building controls, the large-scale market penetration of MPC technology for newly built buildings can be optimistically expected to happen within the next decade.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work emerged from the IBPSA Project 1, an international project conducted under the umbrella of the International Building Performance Simulation Association (IBPSA). Project 1 develops and demonstrates a BIM/GIS and Modelica Framework for building and community energy system design and operation. This research was partially supported by the Assistant Secretary for Energy Efficiency and Renewable Energy, Office of Building Technologies of the U.S. Department of Energy, under Contracts nos. DE-AC05-76RL01830, and DE-AC02-05CH11231. The authors acknowledge the financial support by the European Union through the EU-H2020-GEOT&CH project 'Geothermal Technology for Economic Cooling and Heating' and within the H2020-EE-2016-RIA-IA programme for the project 'Model Predictive Control and Innovative System Integration of GEOTABS;-) in Hybrid Low Grade Thermal Energy Systems - Hybrid MPC GEOTABS' (grant number 723649 - MPC;-) GT), and the H2020 programme under Grant Agreement No. 731231 Flexible Heat and Power. The work of Javier Arroyo is financed by Vlaamse Instelling voor Technologisch Onderzoek (VITO) through a Ph.D. Fellowship under the grant no. 1710754.

References

- Abdel-Aziz, H., & Koutsoukos, X. (2017). Online model learning of buildings using stochastic hybrid systems based on Gaussian Processes. *Journal of Control Science and Engineering*, 2017, 18. <https://doi.org/10.1155/2017/3035892>.
- Abreu, A., Bourdais, R., & Guéguen, H. (2018). Hierarchical model predictive control for building energy management of hybrid systems. *IFAC-PapersOnLine*, 51(16), 235–240.
- Adetola, V., & Guay, M. (2011). Robust adaptive MPC for constrained uncertain nonlinear systems. *International Journal of Adaptive Control and Signal Processing*, 25(2), 155–167. <https://doi.org/10.1002/acs.1193>.
- Advanced Process Solutions, LLC APOPT. <http://apopt.com/index.php>.
- Afram, A., & Janabi-Sharifi, F. (2014a). Review of modeling methods for HVAC systems. *Applied Thermal Engineering*, 67(1), 507–519. <https://doi.org/10.1016/j.applthermaleng.2014.03.055>.
- Afram, A., & Janabi-Sharifi, F. (2014b). Theory and applications of HVAC control systems—A review of model predictive control (MPC). *Building and Environment*, 72, 343–355. <https://doi.org/10.1016/j.buildenv.2013.11.016>.
- Afram, A., & Janabi-Sharifi, F. (2017). Supervisory model predictive controller (MPC) for residential HVAC systems: Implementation and experimentation on archetype sustainable house in Toronto. *Energy and Buildings*, 154, 268–282. <https://doi.org/10.1016/j.enbuild.2017.08.060>.
- Afram, A., Janabi-Sharifi, F., Fung, A. S., & Raahemifar, K. (2017). Artificial neural network (ANN) based model predictive control (MPC) and optimization of HVAC systems: A state of the art review and case study of a residential HVAC system. *Energy and Buildings*, 141, 96–113. <https://doi.org/10.1016/j.enbuild.2017.02.012>.
- Afroz, Z., Shafiullah, G. M., Urmee, T., & Higgins, G. (2018). Modeling techniques used in building HVAC control systems: A review. *Renewable and Sustainable Energy Reviews*. <https://doi.org/10.1016/j.rser.2017.10.044>.
- Aghemo, C., Virgone, J., Fracastoro, G. V., Pellegrino, A., Blaso, L., Savoyat, J., & Johannes, K. (2013). Management and monitoring of public buildings through ICT based systems: Control rules for energy saving with lighting and HVAC services. *Frontiers of Architectural Research*, 2(2), 147–161.
- Ahn, K.-U., Kim, D.-W., Kim, Y.-J., Park, C.-S., & Kim, I.-H. (2015). Gaussian process model for control of an existing building. *Energy Procedia*, 78, 2136–2141. <https://doi.org/10.1016/j.egypro.2015.11.295>. 6th International Building Physics Conference, IBPC 2015, Torino, Italy
- Akyol, B., Haack, J. N., Allwardt, C. H., Katipamula, S., Beech, Z. W., Lutes, R. G., ... Monson, K. E. (2016). VOLTTRON™ 2016. *Technical Report*. Pacific Northwest National Laboratory. Available at <https://volttron.org/>
- Ali, J. M., Hoang, N. H., Hussain, M. A., & Dochain, D. (2015). Review and classification of recent observers applied in chemical process systems. *Computers & Chemical Engineering*, 76, 27–41.
- Allen, R. E., Clark, A. A., Starek, J. A., & Pavone, M. (2014). A machine learning approach for real-time reachability analysis. *2014 IEEE/RSJ international conference on intelligent robots and systems* (pp. 2202–2208). <https://doi.org/10.1109/IROS.2014.6942859>.
- American Society of Heating Refrigerating and Air Conditioning Engineers, (2008). ANSI/ASHRAE Standard 140–2007: Standard method of test for the evaluation of building energy analysis computer programs. ASHRAE.
- American Society of Heating Refrigerating and Air Conditioning Engineers, (2013). ANSI/ASHRAE Standard 55-2013: Thermal Environmental Conditions for Human Occupancy. ASHRAE.
- Amos, B., Rodriguez, I. D. J., Sacks, J., Boots, B., & Kolter, J. Z. (2018). Differentiable MPC for end-to-end planning and control. [CoRR abs/1810.13400](https://arxiv.org/abs/1810.13400)<https://arxiv.org/abs/1810.13400>.
- Amrit, R. (2008). Model Predictive Control (MPC) Tools Package. <https://jbrwww.che.wisc.edu/software/mpctools/index.html>.
- Andersen, E. D., & Andersen, K. D. (2000). *The mosek interior point optimizer for linear programming: An implementation of the homogeneous algorithm* (pp. 197–232). Boston, MA: Springer US.
- Andersen, M. S., & Vandenberghe, L. (2018). CVXOPT. <https://cvxopt.org/index.html>.
- Andersson, J. A. E., Gillis, J., Horn, G., Rawlings, J. B., & Diehl, M. (2018). CasADi – a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*. in press.
- Andriamamonjy, R. (2018). *Automated workflows for building design and operation using open BIM and Modelica*. Ph.D. thesis.
- Antonov, S., & Helsen, L. (2016). Robustness analysis of a hybrid ground coupled heat pump system with model predictive control. *Journal of Process Control*, 47, 191–200. <https://doi.org/10.1016/j.jprocont.2016.08.009>.
- Arendt, K., Clausen, A., Mattera, C. G., Jradi, M., Johansen, A., Veje, C., ... Jørgensen, B. N. (2019). Multi-objective model predictive control framework for buildings. *Proceedings of the 16th IBPSA international conference and exhibition building simulation 2019*. <http://buildingsimulation2019.org>. 16th IBPSA International Conference and Exhibition Building Simulation, Building Simulation 2019 ; Conference date: 02-09-2019 Through 04-09-2019
- Arendt, K., Ionesi, A., Jradi, M., Singh, A. K., Kjærsgaard, M. B., Veje, C., & Jørgensen, B. N. (2016). A building model framework for a genetic algorithm multi-objective model predictive control. In P. K. Heiselberg (Ed.), vol. 8. *Clima 2016*. (2016). Aalborg University. Department of Civil Engineering. <http://www.clima2016.org/>. 12th REHVA World Congress CLIMA 2016, CLIMA 2016; Conference date: 22-05-2016 Through 25-05-2016
- Arendt, K., Jradi, M., Shaker, H. R., & Veje, C. (2018a). Comparative analysis of white-, gray- and black-box models for thermal simulation of indoor environment: Teaching building case study. *Proceedings of the 2018 building performance modeling conference and simbuild co-organized by ASHRAE and IBPSA-USA* (pp. 173–180). ASHRAE.
- Arendt, K., Jradi, M., Wetter, M., & Veje, C. (2018b). ModestPy: An open-source python tool for parameter estimation in functional mock-up units. In M. Tiller, H. Tummescheit, & L. Vanfretti (Eds.), *Proceedings of the 1st American modelica conference* (pp. 121–130). Modelica Association and Linköping University Electronic Press.
- Armstrong, P. R., Leeb, S. B., & Norford, L. K. (2006). Control with building mass-part I: Thermal response model. *ASHRAE Transactions*, 112(1), 449.
- Arroyo, J., van der Heijde, B., Spiessens, F., & Helsen, L. (2018). *A python-based toolbox for model predictive control applied to buildings*. West Lafayette, Indiana: Purdue University.
- Asarin, E., Bournez, O., Dang, T., & Maler, O. (2000). Approximate reachability analysis of piecewise-linear dynamical systems. In N. Lynch, & B. H. Krogh (Eds.), *Hybrid systems: Computation and control* (pp. 20–31). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Ascione, F., Bianco, N., De Stasio, C., Mauro, G. M., & Vanoli, G. P. (2016). Simulation-based model predictive control by the multi-objective optimization of building energy performance and thermal comfort. *Energy and Buildings*, 111, 131–144. <https://doi.org/10.1016/j.enbuild.2015.11.033>.
- Ascione, F., Bianco, N., Mauro, G. M., Napolitano, D. F., & Vanoli, G. P. (2019). Weather-data-based control of space heating operation via multi-objective optimization: Application to italian residential buildings. *Applied Thermal Engineering*, 163, 114384. <https://doi.org/10.1016/j.applthermaleng.2019.114384>.
- Aste, N., Manfren, M., & Marenzi, G. (2017). Building automation and control systems and performance optimization: A framework for analysis. *Renewable and Sustainable Energy Reviews*, 75, 313–330. <https://doi.org/10.1016/j.rser.2016.10.072>.
- Aswani, A., Bouffard, P., Zhang, X., & Tomlin, C. (2014). Practical comparison of optimization algorithms for learning-based MPC with linear models.
- Aswani, A., Gonzalez, H., Sastry, S. S., & Tomlin, C. (2013). Provably safe and robust learning-based model predictive control. *Automatica*, 49(5), 1216–1226. <https://doi.org/10.1016/j.automatica.2013.02.003>.

- Aswani, A., Master, N., Taneja, J., Krioukov, A., Culler, D., & Tomlin, C. (2012). Energy-efficient building HVAC control using hybrid system LB MPC.
- Aswani, A., Shen, Z.-J. M., & Siddiqi, A. (2015). Inverse optimization with noisy data.
- Atam, E., & Helsen, L. (2015). A convex approach to a class of non-convex building HVAC control problems: Illustration by two case studies. *Energy and Buildings*, 93, 269–281.
- Avci, M., Erkok, M., Rahmani, A., & Asfour, S. (2013). Model predictive HVAC load control in buildings using real-time electricity pricing. *Energy and Buildings*, 60(0), 199–209. <https://doi.org/10.1016/j.enbuild.2013.01.008>.
- Bacher, P., & Madsen, H. (2011). Identifying suitable models for the heat dynamics of buildings. *Energy and Buildings*, 43(7), 1511–1522.
- Baetens, R., De Coninck, R., Jorissen, F., Picard, D., Helsen, L., & Saelens, D. (2015). OpenIDEAS - an open framework for integrated district energy simulations. *Proceedings of building simulation 2015, Hyderabad, India*.
- Baetens, R., & Saelens, D. (2016). Modelling uncertainty in district energy simulations by stochastic residential occupant behaviour. *Journal of Building Performance Simulation*, 9(4), 431–447. <https://doi.org/10.1080/19401493.2015.1070203>.
- Baldi, S., Michailidis, I., Ravanis, C., & Kosmatopoulos, E. B. (2015). Model-based and model-free “plug-and-play” building energy efficient control. *Applied Energy*, 154, 829–841.
- Baldi, S., Yuan, S., Endel, P., & Holub, O. (2016). Dual estimation: Constructing building energy models from data sampled at low rate. *Applied Energy*, 169, 81–92.
- Balveddi, B. F. a., Ghisi, E., & Lamberts, R. (2018). A review of occupant behaviour in residential buildings. 10.1016/j.enbuild.2018.06.049.
- Bazararaa, M. S., Sherali, H. D., & Shetty, C. M. (2013). *Nonlinear programming: Theory and algorithms*. John Wiley & Sons.
- Baños, R., Manzano-Agugliaro, F., Montoya, F. G., Gil, C., Alcayde, A., & Gómez, J. (2011). Optimization methods applied to renewable and sustainable energy: A review. *Renewable and Sustainable Energy Reviews*, 15(4), 1753–1766. <https://doi.org/10.1016/j.rser.2010.12.008>.
- Beckman, W., Broman, L., Fiksel, A., Klein, S., Lindberg, E., Schuler, M., & Thornton, J. (1994). TRNSYS: The most complete solar energy system modeling and simulation software. *Renewable Energy*, 5(1), 486–488.
- Bemporad, A. (2004). Hybrid Toolbox - User's Guide. <http://cse.lab.imtlucca.it/~bemporad/hybrid/toolbox>.
- Bemporad, A. (2006). Model predictive control design: New trends and tools. *Proceedings of the 45th IEEE conference on decision and control* (pp. 6678–6683). <https://doi.org/10.1109/CDC.2006.377490>.
- Bemporad, A., M., M., Dua, V., & Pistikopoulos, E. N. (2002). The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1), 3–20. [https://doi.org/10.1016/S0005-1098\(01\)00174-1](https://doi.org/10.1016/S0005-1098(01)00174-1).
- Bemporad, A., & Morari, M. (1999a). Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3), 407–427.
- Bemporad, A., & Morari, M. (1999b). Robust model predictive control: A survey. (pp. 207–226).
- Bengea, S., Adetola, V., Kang, K., Liba, M. J., Vrabie, D., Bitmead, R., & Narayanan, S. (2011). Parameter estimation of a building system model and impact of estimation error on closed-loop performance. *2011 50th IEEE conference on decision and control and european control conference* (pp. 5137–5143). <https://doi.org/10.1109/CDC.2011.6161302>.
- Benndorf, G. A., Wyrstcil, D., & Réhault, N. (2018). Energy performance optimization in buildings: A review on semantic interoperability, fault detection, and predictive control. *Applied Physics Reviews*, 5(4), 041501. <https://doi.org/10.1063/1.5053110>.
- Bernal, W., Behl, M., Nghiem, T., & Mangharam, R. (2012). MLE+: a tool for integrated design and deployment of energy efficient building controls. *Proceedings of the fourth ACM workshop on embedded sensing systems for energy-efficiency in buildings, new york, NY, USA* (pp. 123–130). ACM.
- Bernardini, D., & Bemporad, A. (2009). Scenario-based model predictive control of stochastic constrained linear systems. *Proceedings of the 48th IEEE conference on decision and control (CDC) held jointly with 2009 28th chinese control conference* (pp. 6333–6338). <https://doi.org/10.1109/CDC.2009.5399917>.
- Berthold, T., Farmer, J., Heinz, S., & Perregaard, M. (2018). Parallelization of the FICO xpress-optimizer. *Optimization Methods and Software*, 33(3), 518–529. <https://doi.org/10.1080/10556788.2017.1333612>.
- Berthou, T., Stabat, P., Salvazet, R., & Marchio, D. (2014). Development and validation of a gray box model to predict thermal behavior of occupied office buildings. *Energy and Buildings*, 74, 91–100. <https://doi.org/10.1016/j.enbuild.2014.01.038>.
- Bhattacharya, A., Ma, X., & Vrabie, D. (2020). Model predictive control of discrete-continuous energy systems via generalized disjunctive programming.
- Bianchini, G., Casini, M., Pepe, D., Vicino, A., & Zanvettor, G. G. (2017). An integrated MPC approach for demand-response heating and energy storage operation in smart buildings. *2017 IEEE 56th annual conference on decision and control (CDC), Melbourne, Australia* (pp. 3865–3870). <https://doi.org/10.1109/CDC.2017.8264228>.
- Bianchini, G., Casini, M., Vicino, A., & Zarrilli, D. (2016a). Demand-response in building heating systems: A model predictive control approach. *Applied Energy*, 168, 159–170. <https://doi.org/10.1016/j.apenergy.2016.01.088>.
- Bianchini, G., Casini, M., Vicino, A., & Zarrilli, D. (2016b). Demand-response in building heating systems: A model predictive control approach. *Applied Energy*, 168, 159–170.
- Binder, T., Blank, L., Bock, H. G., Bulirsch, R., Dahmen, W., Diehl, M., ... von Stryk, O. (2001). *Introduction to model based optimization of chemical processes on moving horizons*. In M. Grötschel, S. O. Krumke, & J. Rambau (Eds.) (pp. 295–339). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Blanchini, F. (1999). Set invariance in control. *Automatica*, 35, 1747–1767.
- Blochowitz, T., Otter, M., Arnold, M., Bausch, C., Clauß, C., Elmqvist, H., ... Wolf, S. (2011). The functional mockup interface for tool independent exchange of simulation models. *Proc. of the 8th international modelica conference, Dresden, Germany*. Modelica Association. <https://doi.org/10.3384/ecp11063105>.
- Blum, D., Jorissen, F., Huang, S., Chen, Y., Arroyo, J., Benne, K., ... Sofos, M. (2019a). Prototyping the BOPTTEST framework for simulation-based testing of advanced control strategies in buildings. *Proceedings of the 16th IBPSA conference, Rome*.
- Blum, D., & Wetter, M. (2017). MPCPy: An open-source software platform for model predictive control in buildings. *Proceedings of the 15th IBPSA conference, San Francisco, CA, USA* (pp. 1694–1703).
- Blum, D. H., Arendt, K., Rivalin, L., Piette, M. A., Wetter, M., & Veje, C. T. (2019b). Practical factors of envelope model setup and their effects on the performance of model predictive control for building heating, ventilating, and air conditioning systems. *Applied Energy*, 236, 410–425. <https://doi.org/10.1016/j.apenergy.2018.11.093>.
- Blum, D. H., Xu, N., & Norford, L. K. (2016). A novel multi-market optimization problem for commercial heating, ventilation, and air-conditioning systems providing ancillary services using multi-zone inverse comprehensive room transfer functions. *Science and Technology for the Built Environment*, 22(6), 783–797. <https://doi.org/10.1080/23744731.2016.1197718>.
- Bochkanov, S. A. (2019). ALGLIB: a cross-platform numerical analysis and data processing library. <https://www.alglib.net/>.
- Bohlin, T. (2003). Grey-box model calibrator and validator. *IFAC Proceedings Volumes*, 36(16), 1477–1482. [https://doi.org/10.1016/S1474-6670\(17\)34968-6](https://doi.org/10.1016/S1474-6670(17)34968-6). 13th IFAC Symposium on System Identification (SYSID 2003), Rotterdam, The Netherlands, 27–29 August, 2003.
- Bonami, P., Biegler, L. T., Conn, A. R., Cornuejols, G., Grossmann, I. E., Laird, C. D., Lee, J., Lodi, A., Margot, F., Sawayam, N., & Waechter, A. (2005). An Algorithmic Framework for Convex Mixed Integer Nonlinear Programs.
- Bonvini, M., Sohn, M. D., Granderson, J., Wetter, M., & Piette, M. A. (2014). Robust on-line fault detection diagnosis for HVAC components based on nonlinear state estimation techniques. *Applied Energy*, 124, 156–166.
- Boodi, A., Beddiar, K., Benamour, M., Amirat, Y., & Benbouid, M. (2018). Intelligent systems for building energy and occupant comfort optimization: A state of the art review and recommendations. *Energies*, 11(10). <https://doi.org/10.3390/en11102604>. <http://www.mdpi.com/1996-1073/11/10/2604>
- Borrelli, F. (2003). *Constrained optimal control of linear and hybrid systems* (vol. 290). Springer-Verlag.
- Borrelli, F., Bemporad, A., & Morari, M. (2017). *Predictive control for linear and hybrid systems*. Cambridge University Press.
- Borsche, T., Oldewurtel, F., & Andersson, G. (2014). Scenario-based MPC for Energy Schedule Compliance with Demand Response, vol. 47. *19th IFAC World Congress, Cape Town, South Africa* (pp. 10299–10304). <https://doi.org/10.3182/20140824-6-ZA-1003.01284>. <http://www.sciencedirect.com/science/article/pii/S1474667016432489>
- Bovet, G., Ridì, A., & Hennebert, J. (2014). *Toward web enhanced building automation systems*. In N. Bessis, & C. Dobre (Eds.) (pp. 259–283). Cham: Springer International Publishing.
- Boyd, S., El Ghaoui, L., Feron, E., & Balakrishnan, V. (1994). Linear matrix inequalities in system and control theory. In *Studies in Applied Mathematics* (vol. 15). Philadelphia, PA: SIAM.
- Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1), 1–122.
- Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.
- Bradford, E., Insland, L., Zhang, D., & del Rio Chanona, E. A. (2019). Stochastic data-driven model predictive control using Gaussian Processes.
- Broman, D., Brooks, C., Greenberg, L., Lee, E. A., Masin, M., Tripakis, S., & Wetter, M. (2013). Determinate Composition of FMUs for Co-simulation, . In *EMSOFT '13 Proceedings of the eleventh ACM international conference on embedded software* (pp. 2:1–2:12). Piscataway, NJ, USA: IEEE Press. <http://dl.acm.org/citation.cfm?id=2555754.2555756>
- Bujarbaruah, M., Zhang, X., Rosolia, U., & Borrelli, F. (2018). Adaptive MPC for iterative tasks. *2018 IEEE Conference on Decision and Control (CDC)* (pp. 6322–6327).
- Byrd, R. H., Nocedal, J., & Waltz, R. A. (2006). *Knitro: An integrated package for nonlinear optimization*. In G. Di Pillo, & M. Roma (Eds.). *Large-Scale Nonlinear Optimization* (pp. 35–59). Boston, MA: Springer US. https://doi.org/10.1007/0-387-30065-1_4.
- Bünning, F., Huber, B., Heer, P., Aboudonia, A., & Lygeros, J. (2020). Experimental demonstration of data predictive control for energy optimization and thermal comfort in buildings. *Energy and Buildings*, 211, 109792. <https://doi.org/10.1016/j.enbuild.2020.109792>.
- Büskens, C., & Wassel, D. (2013). The ESA NLP solver WORHP. In G. Fasano, & J. D. Pintér (Eds.), vol. 73. *Modeling and optimization in space engineering* (pp. 85–110). Springer New York. https://doi.org/10.1007/978-1-4614-4469-5_4.
- Cagienard, R., Grieder, P., Kerrigan, E. C., & Morari, M. (2004). Move blocking strategies in receding horizon control, vol. 2. *2004 43rd IEEE conference on decision and control (CDC) (IEEE cat. no.04CH37601), Nassau, Bahamas* (pp. 2023–2028). <https://doi.org/10.1109/CDC.2004.1430345>.
- Cai, J., Braun, J. E., Kim, D., & Hu, J. (2016a). General approaches for determining the savings potential of optimal control for cooling in commercial buildings having both energy and demand charges. *Science and Technology for the Built Environment*, 22(6), 733–750.
- Cai, J., Braun, J. E., Kim, D., & Hu, J. (2016b). A multi-agent control based demand response strategy for multi-zone buildings. *American control conference (ACC), 2016, Boston, MA, USA* (pp. 2365–2372). IEEE.
- Cai, J., Kim, D., Putta, V. K., Braun, J. E., & Hu, J. (2015). Multi-agent control for centralized air conditioning systems serving multi-zone buildings. *American control conference (ACC), 2015, Chicago, IL, USA* (pp. 986–993). IEEE.

- Campo, P. J., & Morari, M. (1987). Robust model predictive control, vol. 2. *Proc. American contr. conf.* (pp. 1021–1026).
- Camponogara, E., Jia, D., Krogh, B. H., & Talukdar, S. (2002). Distributed model predictive control. *IEEE Control Systems*, 22(1), 44–52.
- Cao, Y., Du, J., & Soleymanzadeh, E. (2019). Model predictive control of commercial buildings in demand response programs in the presence of thermal storage. *Journal of Cleaner Production*, 218, 315–327. <https://doi.org/10.1016/j.jclepro.2019.01.266>.
- Capozzoli, A., Piscitelli, M. S., Gorrino, A., Ballarini, I., & Corrado, V. (2017). Data analytics for occupancy pattern learning to reduce the energy consumption of HVAC systems in office buildings. *Sustainable Cities and Society*, 35, 191–208. <https://doi.org/10.1016/j.scs.2017.07.016>.
- Carlucci, S., De Simone, M., Firth, S. K., Kjørgaard, M. B., Markovic, R., Rahaman, M. S., ... van Treeck, C. (2020). Modeling occupant behavior in buildings. *Building and Environment*, 174, 106768. <https://doi.org/10.1016/j.buildenv.2020.106768>.
- Castilla, M., Álvarez, J. D., Normey-Rico, J. E., & Rodriguez, F. (2014). Thermal comfort control using a non-linear MPC strategy: A real case of study in a bioclimatic building. *Journal of Process Control*, 24(6), 703–713. <https://doi.org/10.1016/j.jprocont.2013.08.009>. Energy Efficient Buildings Special Issue
- Castilla, M., Álvarez, J. D., Berenguel, M., Rodríguez, F., Guzmán, J. L., & Pérez, M. (2011). A comparison of thermal comfort predictive control strategies. *Energy and Buildings*, 43(10), 2737–2746. <https://doi.org/10.1016/j.enbuild.2011.06.030>. <http://www.sciencedirect.com/science/article/pii/S0378778811002799>
- Castro, P. M., & Grossmann, I. E. (2012). Generalized disjunctive programming as a systematic modeling framework to derive scheduling formulations. *Industrial & Engineering Chemistry Research*, 51(16), 5781–5792. <https://doi.org/10.1021/ie2030486>.
- Cauchi, N., & Abate, A. (2018). Benchmarks for cyber-physical systems: A modular model library for building automation systems. *IFAC PapersOnLine*, 51(16), 49–54. <https://doi.org/10.1016/j.ifacol.2018.08.009>. 6th IFAC Conference on Analysis and Design of Hybrid Systems ADHS 2018, Oxford, UK
- Chandan, V., & Alleyne, A. G. (2013). Optimal partitioning for the decentralized thermal control of buildings. *IEEE Transactions on Control Systems Technology*, 21(5), 1756–1770. <https://doi.org/10.1109/TCST.2012.2219308>.
- Chandan, V., & Alleyne, A. G. (2014). Decentralized predictive thermal control for buildings. *Journal of Process Control*, 24(6), 820–835.
- Chen, B., Cai, Z., & Bergés, M. (2019a). Gnu-RL: A precocial reinforcement learning solution for building HVAC control using a differentiable MPC policy, . In *BuildSys '19 Proceedings of the 6th ACM international conference on systems for energy-efficient buildings, cities, and transportation* (pp. 316–325). New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/3360322.3360849>.
- Chen, S. W., Wang, T., Atanasov, N., Kumar, V., & Morari, M. (2019b). Large scale model predictive control with neural networks and primal active sets.
- Chen, X., Wang, Q., & Srebric, J. (2015). Model predictive control for indoor thermal comfort and energy optimization using occupant feedback. *Energy and Buildings*, 102, 357–369.
- Chen, Y., Shi, Y., & Zhang, B. (2018). Optimal control via neural networks: A convex approach.
- Cigler, J., Gyalistras, D., Široký, J., Tiet, V., & Ferkl, L. (2013a). Beyond theory: The challenge of implementing model predictive control in buildings. *Proceedings of 11th rehva world congress, Clima, Prague, Czech Republic*.
- Cigler, J., Přívara, S., Vána, Z., Žáčková, E., & Ferkl, L. (2012). Optimization of predicted mean vote index within model predictive control framework: Computationally tractable solution. *Energy and Buildings*, 52, 39–49.
- Cigler, J., Široký, J., Korda, M., & Jones, C. (2013b). On the selection of the most appropriate MPC problem formulation for buildings. *Technical Report*.
- Cigler, J., Tomáško, P., & Široký, J. (2013c). BuildingLAB: A tool to analyze performance of model predictive controllers for buildings. *Energy and Buildings*, 57, 34–41. <https://doi.org/10.1016/j.enbuild.2012.10.042>. <http://www.sciencedirect.com/science/article/pii/S0378778812005592>
- Cinquemani, E., Agarwal, M., Chatterjee, D., & Lygeros, J. (2011). Convexity and convex approximations of discrete-time stochastic control problems with constraints. *Automatica*, 47(9), 2082–2087. <https://doi.org/10.1016/j.automatica.2011.01.023>.
- Clarke, J. A. (2001). *Energy simulation in building design* (2nd ed.). Oxford, UK: Butterworth-Heinemann.
- Coffey, B. (2013). Approximating model predictive control with existing building simulation tools and offline optimization. *Journal of Building Performance Simulation*, 6(3), 220–235. <https://doi.org/10.1080/19401493.2012.737834>.
- Coffey, B., Haghighat, F., Morofsky, E., & Kutrowski, E. (2010). A software framework for model predictive control with GenOpt. *Energy and Buildings*, 42(7), 1084–1092. <https://doi.org/10.1016/j.enbuild.2010.01.022>. <http://www.sciencedirect.com/science/article/pii/S0378778810000290>
- Corbin, C., Henze, G., & May-Ostendorp, P. (2012). A model predictive control optimization environment for real-time commercial building application. *Journal of Building Performance Simulation*, 2012. <https://doi.org/10.1080/19401493.2011.648343>.
- Crawley, D., Lawrie, L., Winkelmann, F., Buhl, W., Huang, Y., Pedersen, C., ... Witte, M. (2001). EnergyPlus: creating a new-generation building energy simulation program. *Energy and Buildings*, 33(4), 319–331.
- Cupeiro, I., Drgoňa, J., Abdollahpour, M., Picard, D., & Helsen, L. (2018). State observers for optimal control using white-box building models. *Purdue conferences - 5th international high performance building conference, West Lafayette, IN, USA*. Purdue University, West Lafayette, IN, USA
- Cupeiro Figueroa, I., Cigler, J., & Helsen, L. (2018). Model predictive control formulation: A review with focus on hybrid GEOTABS buildings. *Proceedings of REHVA annual meeting conference low carbon technologies in HVAC, Brussels, Belgium* (pp. 1–9).
- Cupeiro Figueroa, I., Picard, D., & Helsen, L. (2020). Short-term modeling of hybrid geothermal systems for model predictive control. *Energy and Buildings*, 215, 109884. <https://doi.org/10.1016/j.enbuild.2020.109884>. <http://www.sciencedirect.com/science/article/pii/S0378778819331019>
- Cutsem, O. V., Kayal, M., Blum, D., & Pritoni, M. (2019a). Comparison of MPC formulations for building control under commercial time-of-use tariffs. *Ieee powertech Milan*.
- Cutsem, O. V., Kayal, M., Blum, D., & Pritoni, M. (2019b). Comparison of MPC formulations for building control under commercial time-of-use tariffs. *Ieee powertech Milan 2019*.
- Cybersecurity in smart buildings inaction is not an option anymore. (2015). *Technical Report*. Frost and Sullivan the Growth Consulting Company.
- Darivianakis, G., Georghiou, A., Smith, R., & Lygeros, J. (b). The energy hub component modelling (EHCM) toolbox. <https://control.ee.ethz.ch/software/BRCM-Toolbox1.html>.
- Darivianakis, G., Georghiou, A., Smith, R. S., & Lygeros, J. (2015). A stochastic optimization approach to cooperative building energy management via an energy hub. *2015 54th IEEE conference on decision and control (CDC)* (pp. 7814–7819).
- Darivianakis, G., Georghiou, A., Smith, R. S., & Lygeros, J. (2019). The power of diversity: Data-driven robust predictive control for energy-efficient buildings and districts. *IEEE Transactions on Control Systems Technology*, 27(1), 132–145.
- Davidsson, P., & Boman, M. (2005). Distributed monitoring and control of office buildings by embedded agents. *Information Sciences*, 171(4), 293–307. <https://doi.org/10.1016/j.ins.2004.09.007>. Intelligent Embedded Agents
- De Coninck, R., & Helsen, L. (2016). Practical implementation and evaluation of model predictive control for an office building in Brussels. *Energy and Buildings*, 111, 290–298. <https://doi.org/10.1016/j.enbuild.2015.11.014>. <http://www.sciencedirect.com/science/article/pii/S0378778815303790>
- De Coninck, R., Magnusson, F., Åkesson, J., & Helsen, L. (2016). Toolbox for development and validation of grey-box building models for forecasting and control. *Journal of Building Performance Simulation*, 9(3), 288–303. <https://doi.org/10.1080/19401493.2015.1046933>.
- DeHaan, D., Adetola, V., & Guay, M. (2007). Adaptive robust mpc: An eye towards computational simplicity. *IFAC Proceedings Volumes*, 40(12), 228–233. <https://doi.org/10.3182/20070822-3-ZA-2920.00038>. 7th IFAC Symposium on Nonlinear Control Systems
- Diamond, S., & Boyd, S. (2016). CVXPY: A python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83), 1–5.
- Dodier, R. H., & Henze, G. P. (2004). Statistical analysis of neural networks as applied to building energy prediction. *Journal of Solar Energy Engineering*, 126(1), 592–600. <https://doi.org/10.1115/1.1637640>.
- Domahidi, A., Chu, E., & Boyd, S. (2013). ECOS: An SOCP solver for embedded systems. *European control conference (ECC)* (pp. 3071–3076).
- Domahidi, A., Ullmann, F., Morari, M., & Jones, C. N. (2014). Learning decision rules for energy efficient building control. *Journal of Process Control*, 24(6), 763–772. <https://doi.org/10.1016/j.jprocont.2014.01.006>. Energy Efficient Buildings Special Issue
- Dougherty, D., & Cooper, D. (2003). A practical multiple model adaptive strategy for single-loop MPC. *Control Engineering Practice*, 11(2), 141–159. [https://doi.org/10.1016/S0967-0661\(02\)00106-5](https://doi.org/10.1016/S0967-0661(02)00106-5). Automotive Systems
- Dounis, A. I., & Caraiacos, C. (2009). Advanced control systems engineering for energy and comfort management in a building environment-a review. *Renewable and Sustainable Energy Reviews*, 13(6), 1246–1261. <https://doi.org/10.1016/j.rser.2008.09.015>. <http://www.sciencedirect.com/science/article/pii/S1364032108001457>
- Drgoňa, J. (2019). BeSim Toolbox: Fast Development, and Simulation of Advanced Building Control. <https://github.com/drгона/BeSim>.
- Drgoňa, J., & Kvasnica, M. (2013). Comparison of MPC strategies for building control. *Proceedings of the 19th international conference on process control, Štrbské Pleso, Slovakia*.
- Drgoňa, J., Kvasnica, M., Klaučo, M., & Fikar, M. (2013). Explicit stochastic MPC approach to building temperature control. *Ieee conference on decision and control (pp. 6440–6445)*. http://www.kirp.chft.stuba.sk/assets/publication_info.php?id_p ub=1470. Florence, Italy
- Drgoňa, J., Klaučo, M., & Kvasnica, M. (2015). MPC-based reference governors for thermostatically controlled residential buildings. *2015 54th IEEE conference on decision and control (CDC)* (pp. 1334–1339). <https://doi.org/10.1109/CDC.2015.7402396>.
- Drgoňa, J., Picard, D., & Helsen, L. (2020). Cloud-based implementation of white-box model predictive control for a GEOTABS office building: A field test demonstration. *Journal of Process Control*, 88, 63–77. <https://doi.org/10.1016/j.jprocont.2020.02.007>. <http://www.sciencedirect.com/science/article/pii/S0959152419306857>
- Drgoňa, J., Picard, D., Kvasnica, M., & Helsen, L. (2018). Approximate model predictive building control via machine learning. *Applied Energy*, 218, 199–216. <https://doi.org/10.1016/j.apenergy.2018.02.156>. <http://www.sciencedirect.com/science/article/pii/S0306261918302903>
- Dunning, I., Huchette, J., & Lubin, M. (2017). JuMP: A modeling language for mathematical optimization. *SIAM Review*, 59(2), 295–320. <https://doi.org/10.1137/15M1020575>.
- Embotech FORCES Pro. <https://www.embotech.com/forces-pro>.
- Enescu, D. (2017). A review of thermal comfort models and indicators for indoor environments. *Renewable and Sustainable Energy Reviews*, 79, 1353–1379.
- Esther, B. P., & Kumar, K. S. (2016). A survey on residential demand side management architecture, approaches, optimization models and methods. *Renewable and Sustainable Energy Reviews*, 59, 342–351. <https://doi.org/10.1016/j.rser.2016.05.015>.

- rser.2015.12.282.<http://www.sciencedirect.com/science/article/pii/S1364032115016652>
- EUROSIS. Directory of simulation software and tools. <http://www.eurosis.org/cms/?q=node/1318>.
- EU policy. (2018). EU policy: Revised Energy Performance of Buildings Directive (EPBD), EUR-Lex - 32018L0844 - EN. *Technical Report*. European Parliament.
- Fabietti, L. (2014). Control of HVAC systems via explicit and implicit MPC: an experimental case study. Master's thesis. KTH Royal Institute of Technology, Stockholm. XE-EE-RT 2014:006.
- Fagiano, L., Schildbach, G., Tanaskovic, M., & Morari, M. (2015). Scenario and adaptive model predictive control of uncertain systems. *IFAC-PapersOnLine*, 48(23), 352–359. <https://doi.org/10.1016/j.ifacol.2015.11.305>. 5th IFAC Conference on Nonlinear Model Predictive Control NMPC 2015
- Fanger, P. O. (1973). Assessment of man's thermal comfort in practice. *Occupational and Environmental Medicine*, 30(4), 313–324.
- Farina, M., Giulioni, L., & Scattolini, R. (2016a). Stochastic linear model predictive control with chance constraints - a review. *Journal of Process Control*, 44, 53–67. <https://doi.org/10.1016/j.jprocont.2016.03.005>. <http://www.sciencedirect.com/science/article/pii/S0959152416300130>
- Farina, M., Giulioni, L., & Scattolini, R. (2016b). Stochastic linear model predictive control with chance constraints - a review. *Journal of Process Control*, 44, 53–67. <https://doi.org/10.1016/j.jprocont.2016.03.005>. <http://www.sciencedirect.com/science/article/pii/S0959152416300130>
- Feng, J. D., Chuang, F., Borrelli, F., & Bauman, F. (2015). Model predictive control of radiant slab systems with evaporative cooling sources. *Energy and Buildings*, 87, 199–210. <https://doi.org/10.1016/j.enbuild.2014.11.037>. <http://www.sciencedirect.com/science/article/pii/S0378778814009682>
- Ferhatbegović, T., Zucker, G., & Palensky, P. (2012). An unscented Kalman filter approach for the plant-model mismatch reduction in HVAC system model based control. *Iecon 2012-38th annual conference on IEEE industrial electronics society* (pp. 2180–2185). IEEE.
- Ferkl, L., & Siroký, J. (2010). Ceiling radiant cooling: Comparison of ARMAX and subspace identification modelling methods. *Building and Environment*, 45(1), 205–212. <https://doi.org/10.1016/j.buildenv.2009.06.004>
- Ferrarin, L., Mantovani, G., & Costanzo, G. T. (2014). A distributed model predictive control approach for the integration of flexible loads, storage and renewables. *2014 IEEE 23rd international symposium on industrial electronics (ISIE)* (pp. 1700–1705). <https://doi.org/10.1109/ISIE.2014.6864871>.
- Ferreau, H. J., Kirches, C., Potschka, A., Bock, H. G., & Diehl, M. (2014). qpOASES: A parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation*, 6, 327–363.
- Fielsch, S., Grunert, T., Stursberg, M., & Kummert, A. (2017). Model predictive control for hydronic heating systems in residential buildings. *IFAC-PapersOnLine*, 50(1), 4216–4221.
- Figueiredo, J., & Costa, J. S. d. (2012). A SCADA system for energy management in intelligent buildings. *Energy and Buildings*, 49, 85–98. <https://doi.org/10.1016/j.enbuild.2012.01.041>. <http://www.sciencedirect.com/science/article/pii/S0378778812000722>
- Fourer, R., Gay, D. M., & Kernighan, B. W. (2002). *AMPL: A modeling language for mathematical programming* (2nd ed.). Duxbury Press.
- Freire, R., Oliveira, G., & Mendes, N. (2005). Thermal comfort based predictive controllers for building heating systems. *Proc. of the 16th IFAC world congress, Prague, Czech Republic*.
- Freire, R. Z., Oliveira, G. H. C., & Mendes, N. (2008). Predictive controllers for thermal comfort optimization and energy savings. *Energy and buildings*, 40(7), 1353–1365.
- Frison, G., & Jørgensen, J. B. (2013). A fast condensing method for solution of linear-quadratic control problems. *Proceedings of 52nd IEEE conference on decision and control, Florence, Italy* (pp. 7715–7720).
- Frison, G., Sørensen, H. H. B., Dammann, B., & Jørgensen, J. B. (2014). High-performance small-scale solvers for linear model predictive control. *2014 european control conference (ECC), Strasbourg, France* (pp. 128–133). <https://doi.org/10.1109/ECC.2014.6862490>.
- Fritzson, P., Pop, A., Asghar, A., Bachmann, B., Braun, W., Braun, R., ... Östlund, P. (2018). The OpenModelica integrated modeling, simulation and optimization environment. *1st American modelica conference, Cambridge, MA, USA* (pp. 206–219). <https://doi.org/10.3384/ecp18154206>.
- Fux, S. F., Ashouri, A., Benz, M. J., & Guzzella, L. (2014). EKF based self-adaptive thermal model for a passive house. *Energy and Buildings*, 68, 811–817.
- Gambier, A. (2008). MPC and PID control based on multi-objective optimization. *2008 American control conference* (pp. 4727–4732).
- Gao, H., Koch, C., & Wu, Y. (2019). Building information modelling based building energy modelling: A review. *Applied Energy*, 238, 320–343. <https://doi.org/10.1016/j.apenergy.2019.01.032>. <http://www.sciencedirect.com/science/article/pii/S0306261919300327>
- Garifi, K., Baker, K., Touri, B., & Christensen, D. T. (2018). Stochastic model predictive control for demand response in a home energy management system: Preprint. *Ieee power and energy society general meeting, 5–10 august 2018, portland, oregon*.
- Gayeski, N. T., Armstrong, P. R., & Norford, L. K. (2012). Predictive pre-cooling of thermo-active building systems with low-lift chillers. *HVAC&R Research*, 18(5), 858–873. <https://doi.org/10.1080/10789669.2012.643752>.
- Gertz, E. M., & Wright, S. J. (2003). Object-oriented software for quadratic programming. *ACM Trans. Math. Softw.*, 29, 58–81.
- Gill, P., Murray, W., & Saunders, M. (2005a). SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Review*, 47(1), 99–131. <https://doi.org/10.1137/S0036144504446096>.
- Gill, P. E., Murray, W., & Saunders, M. A. (2005b). SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Review*, 47(1), 99–131. <https://doi.org/10.1137/S0036144504446096>.
- Godina, R., Rodrigues, E. M. G., Poursmaeil, E., & Catalão, J. P. S. (2018). Optimal residential model predictive control energy management performance with PV microgeneration. *Computers & Operations Research*, 96, 143–156. <https://doi.org/10.1016/j.cor.2017.12.003>. <http://www.sciencedirect.com/science/article/pii/S0305054817302952>
- Gorecki, T. T., Qureshi, F. A., & Jones, C. N. (2015). OpenBuild: An integrated simulation environment for building control. *Proceedings of IEEE conference on control applications (CCA), Sydney, Australia* (pp. 1522–1527).
- Grant, M., & Boyd, S. (2014). CVX: Matlab software for disciplined convex programming, version 2.1. <http://cvxr.com/cvx>.
- Grossmann, I. E., & Ruiz, J. P. (2012). Generalized disjunctive programming: A framework for formulation and alternative algorithms for MINLP optimization. In J. Lee, & S. Leyffer (Eds.), *Mixed integer nonlinear programming* (pp. 93–115). New York, NY: Springer New York.
- Gupta, S. K., Kar, K., Mishra, S., & Wen, J. T. (2015). Distributed consensus algorithms for collaborative temperature control in smart buildings. *American control conference (ACC), 2015, Chicago, IL, USA* (pp. 5758–5763). IEEE.
- Gurobi Optimization, I. (2012). Gurobi optimizer reference manual. <http://www.gurobi.com>.
- Gutschker, O. (2008). Parameter identification with the software package LORD. *Building and Environment*, 43, 163–169. <https://doi.org/10.1016/j.buildenv.2006.10.010>.
- Gwerder, M., Gyalistras, D., Sagerschnig, C., Smith, R. S., & Sturzenegger, D. (2013). Final Report: Use of Weather And Occupancy Forecasts For Optimal Building Climate Control. Part II: Demonstration (OptiControl-II). *Technical Report*. Automatic Control Laboratory, ETH Zurich, Switzerland. <http://www.opticontrol.ethz.ch>
- Gyalistras, D., Gwerder, M., Schildbach, F., Jones, C. N., Morari, M., Lehmann, B., ... Stauch, V. (2010). Analysis of energy savings potentials for integrated room automation. *Clima - RHEVA world congress, Antalya, Turkey*.
- Gyorodi, C., Gyorodi, R., & Sotoc, R. (2015). A comparative study of relational and non-relational database models in a web-based application. *(IJACSA) International Journal of Advanced Computer Science and Applications*, 6(11). <http://www.ijacsa.th.esai.org>
- Hagan, M. T., Demuth, H. B., & Beale, M. (1996). *Neural network design*. Boston, MA, USA: PWS Publishing Co.
- Harb, H., Boyanov, N., Hernandez, L., Streblov, R., & Müller, D. (2016). Development and validation of grey-box models for forecasting the thermal response of occupied buildings. *Energy and Buildings*, 117, 199–207. <https://doi.org/10.1016/j.enbuild.2016.02.021>.
- Hart, W. E., Laird, C. D., Watson, J.-P., Woodruff, D. L., Hackedbeil, G. A., Nicholson, B. L., & Sirola, J. D. (2017). *Pyomo-optimization modeling in python* (2nd ed.) (2nd ed., vol. 67). Springer Science & Business Media.
- Haystack, P. (2020). Project haystack. <https://www.project-haystack.org/>.
- Hazyuk, I., Ghiaus, C., & Penhouet, D. (2012). Optimal temperature control of intermittently heated buildings using model predictive control: Part I - building modeling. *Building and Environment*, 51, 379–387.
- Hedegaard, R. E., Pedersen, T. H., Knudsen, M. D., & Petersen, S. (2018). Towards practical model predictive control of residential space heating: Eliminating the need for weather measurements. *Energy and Buildings*, 170, 206–216. <https://doi.org/10.1016/j.enbuild.2018.04.014>.
- Hedengren, J. D., Shishavan, R. A., Powell, K. M., & Edgar, T. F. (2014). Nonlinear modeling, estimation and predictive control in APMonitor. *Computers & Chemical Engineering*, 70, 133–148. <https://doi.org/10.1016/j.compchemeng.2014.04.013>. Manfred Morari Special Issue
- Heirung, T. A. N., Paulson, J. A., ÓLeary, J., & Mesbah, A. (2018). Stochastic model predictive control - how does it work? *Computers & Chemical Engineering*, 114, 158–170. <https://doi.org/10.1016/j.compchemeng.2017.10.026>. FOCAP0/CPC 2017
- Heirung, T. A. N., Ydstie, B. E., & Foss, B. (2017). Dual adaptive model predictive control. *Automatica*, 80(12), 340–348. <https://doi.org/10.1016/j.automatica.2017.01.030>.
- Building performance simulation for design and operation*. (2019). In Hensen, J., & Lamberts, R. (Eds.) (2nd ed.), (2019). London: Routledge. <https://doi.org/10.1201/9780429402296>.
- Henze, G. P. (2013). Model predictive control for buildings: A quantum leap? *Journal of Building Performance Simulation*, 6(3), 157–158. <https://doi.org/10.1080/19401493.2013.778519>.
- Herceg, M., Kvasnica, M., Jones, C., & Morari, M. (2013). Multi-parametric toolbox 3.0. *2013 european control conference, zurich, switzerland* (pp. 502–510).
- Hertneck, M., Köhler, J., Trimpe, S., & Allgöwer, F. (2018). Learning an approximate model predictive controller with guarantees. *CoRR abs/1806.04167*.
- Herzog, S., Atabay, D., Jungwirth, J., & Mikulovic, V. (2013). Self-adapting building models for model predictive control. *13th conference of international building performance simulation association, Chambéry, France*.
- Hewing, L., Wabersich, K. P., Menner, M., & Zeilinger, M. N. (2020). Learning-based model predictive control: Toward safe learning in control. *Annual Review of Control Robotics and Autonomous Systems*, 3(1), null. <https://doi.org/10.1146/annurev-control-090419-075625>.
- Hilliard, T., Kavgić, M., & Swan, L. (2015). Model predictive control for commercial buildings: trends and opportunities. *Advances in Building Energy Research*, 2, 172–190.
- Hilliard, T., Swan, L., Kavgić, M., Qin, Z., & Lingras, P. (2016). Development of a whole building model predictive control strategy for a LEED silver community college. *Energy and Buildings*, 111, 224–232. <https://doi.org/10.1016/j.enbuild.2015.11.051>.

- Hilliard, T., Swan, L., & Qin, Z. (2017). Experimental implementation of whole building MPC with zone based thermal comfort adjustments. *Building and Environment*, 125, 326–338. <https://doi.org/10.1016/j.buildenv.2017.09.003>.
- Hong, T., Sun, H., Chen, Y., Taylor-Lange, S. C., & Yan, D. (2016). An occupant behavior modeling tool for co-simulation. *Energy and Buildings*, 117, 272–281. <https://doi.org/10.1016/j.enbuild.2015.10.033>.
- Hou, X., Xiao, Y., Cai, J., Hu, J., & Braun, J. E. (2017). Distributed model predictive control via proximal jacobian ADMM for building control applications. *American control conference (ACC), 2017, Seattle, WA, USA* (pp. 37–43). IEEE.
- Houska, B., Ferreau, H. J., & Diehl, M. (2011). ACADO Toolkit – an open source framework for automatic control and dynamic optimization. *Optimal Control Applications and Methods*, 32(3), 298–312.
- Huang, G. (2011). Model predictive control of VAV zone thermal systems concerning bilinearity and gain nonlinearity. *Control engineering practice*, 19(7), 700–710. <https://doi.org/10.1016/j.conengprac.2011.03.005>.
- Huang, G., Wang, S., & Xu, X. (2010a). Robust model predictive control of VAV air-handling units concerning uncertainties and constraints. *HVAC&R Research*, 16(1), 15–33. <https://doi.org/10.1080/10789669.2010.10390890>.
- Huang, H., Chen, L., & Hu, E. (2014). Model predictive control for energy-efficient buildings: An airport terminal building study. *IEEE International Conference on Control and Automation ICCA Taichung Taiwan* (pp. 1025–1030). <https://doi.org/10.1109/ICCA.2014.6871061>.
- Huang, H., Chen, L., & Hu, E. (2015). A neural network-based multi-zone modelling approach for predictive control system design in commercial buildings. *Energy and Buildings*, 97, 86–97. <https://doi.org/10.1016/j.enbuild.2015.03.045>.
- Huang, R., Biegler, L. T., & Patwardhan, S. C. (2010b). Fast offset-free nonlinear model predictive control based on moving horizon estimation. *Industrial & Engineering Chemistry Research*, 49(17), 7882–7890. <https://doi.org/10.1021/ie901945y>.
- Humphreys, M. A., & Nicol, J. F. (2002). The validity of ISO-PMV for predicting comfort votes in every-day thermal environments. *Energy and Buildings*, 34(6), 667–684. [https://doi.org/10.1016/S0378-7788\(02\)00018-X](https://doi.org/10.1016/S0378-7788(02)00018-X). Special Issue on Thermal Comfort Standards
- Husom, J. K., Poulsen, N. K., Jørgensen, S. B., & Jørgensen, J. B. (2010). Tuning of methods for offset free MPC based on ARX model representations. *Proceedings of the 2010 American control conference* (pp. 2355–2360). <https://doi.org/10.1109/ACC.2010.5530560>.
- ILOG (2007). 11.0 user's manual. ILOG CPLEX Division. Incline Village, NV.
- International Organization for Standardization. (2005). ISO 7730 2005-11-15 Ergonomics of the thermal environment: Analytical determination and interpretation of thermal comfort using calculation of the PMV and PPD indices and local thermal comfort criteria. In *International standards*. ISO.
- IEA International Energy Agency and International Partnership for Energy Efficiency Cooperation. (2015). Building Energy Performance Metrics - Supporting Energy Efficiency Progress in Major Economies. *Technical Report*. IEA Publications.
- Jain, A., Behl, M., & Mangharam, R. (2017a). Data predictive control for building energy management. *2017 American control conference (ACC), Seattle, WA, USA* (pp. 44–49). <https://doi.org/10.23919/ACC.2017.7962928>.
- Jain, A., Nghiem, T., Morari, M., & Mangharam, R. (2018). Learning and control using Gaussian processes. *2018 ACM/IEEE 9th international conference on cyber-physical systems (ICCPS)* (pp. 140–149). IEEE.
- Jain, A., Smarra, F., & Mangharam, R. (2017b). Data predictive control using regression trees and ensemble learning. *2017 IEEE 56th annual conference on decision and control (CDC)* (pp. 4446–4451). <https://doi.org/10.1109/CDC.2017.8264315>.
- Jamshidi, M. (1996). *Large-scale systems: Modeling, control, and fuzzy logic*. Prentice-Hall, Inc.
- Jiang, C., & Wang, J. (1999). Neural networks and system identification. *Journal of Fudan University Natural Science*, 38.
- Jiménez, M. J., Madsen, H., & Andersen, K. K. (2008). Identification of the main thermal characteristics of building components using MATLAB. *Building and Environment*, 43(2), 170–180. <https://doi.org/10.1016/j.buildenv.2006.10.030>. Outdoor Testing, Analysis and Modelling of Building Components
- Johnson, B. J., Starke, M. R., Abdelaziz, O. A., Jackson, R. K., & Tolbert, L. M. (2014). A method for modeling household occupant behavior to simulate residential energy consumption. *2014 IEEE PES innovative smart grid technologies conference, ISGT 2014* (pp. 1–5). Washington, DC, USA: IEEE. <https://doi.org/10.1109/ISGT.2014.6816483>.
- Jorissen, F. (2018). *Toolchain for Optimal Control and Design of Energy Systems in Buildings*. KU Leuven, Belgium. Ph.D. Thesis.
- Jorissen, F., Boydens, W., & Helsen, L. (2017). Simulation-based occupancy estimation in office buildings using CO2 sensors, vol. 15. *15th international conference of IBPSA, San Francisco, USA*. International Building Performance Simulation Association.
- Jorissen, F., Boydens, W., & Helsen, L. (2018a). TACO, an automated toolchain for model predictive control of building systems: Implementation and verification. *Journal of building performance simulation* (pp. 180–192). <https://doi.org/10.1080/19401493.2018.1498537>.
- Jorissen, F., & Helsen, L. (2019). Integrated modelica model and model predictive control of a terraced house using IDEAS. *13th international modelica conference 2019, location: Regensburg, Germany* (pp. 139–148). <https://doi.org/10.3384/ecp19157139>.
- Jorissen, F., Picard, D., Cupeiro Figueroa, I., Boydens, W., & Helsen, L. (2018b). Towards real MPC implementation in an office building using TACO. *5th international high performance building conference at Purdue university, West Lafayette, IN, USA*.
- Jorissen, F., Reynders, G., Baetens, R., Picard, D., Saelens, D., & Helsen, L. (2018c). Implementation and verification of the IDEAS building energy simulation library. *Journal of Building Performance Simulation*, 11(6), 669–688. <https://doi.org/10.1080/19401493.2018.1428361>.
- Jorissen, F., Wetter, M., & Helsen, L. (2018d). Simplifications for hydronic system models in modelica. *Journal of Building Performance Simulation*, 11(6), 639–654. <https://doi.org/10.1080/19401493.2017.1421263>.
- Jradi, M., Arendt, K., Sangogboye, F. C., Mattera, C. G., Markoska, E., Kjaergaard, M. B., ... Jørgensen, B. N. (2018). ObepME: An online building energy performance monitoring and evaluation tool to reduce energy performance gaps. *Energy and Buildings*, 166, 196–209. <https://doi.org/10.1016/j.enbuild.2018.02.005>.
- Kamalapurkar, R. (2017). Simultaneous state and parameter estimation for second-order nonlinear systems. *2017 IEEE 56th annual conference on decision and control (CDC)* (pp. 2164–2169). <https://doi.org/10.1109/CDC.2017.8263965>.
- Kavgic, M., Hilliard, T., & Swan, L. (2015). Opportunities for implementation of MPC in commercial buildings. *Energy Procedia*, 78, 2148–2153. <https://doi.org/10.1016/j.egypro.2015.11.300>. 6th International Building Physics Conference, IBPC 2015, Torino, Italy
- Kelly, M. (2017). An introduction to trajectory optimization: How to do your own direct collocation. *SIAM Review*, 59(4), 849–904. <https://doi.org/10.1137/16M1062569>.
- Kephalopoulos, S., Geiss, O., Barrero-Moreno, J., D'Agostino, D., & Paci, D. (2016). Promoting healthy and energy efficient buildings in the European Union-national implementation of related requirements of the energy performance buildings directive (2010/31/EU). *European commission's science and knowledge service: Brussels, Belgium*.
- Killian, M., & Kozek, M. (2016). Ten questions concerning model predictive control for energy efficient buildings. *Building and Environment*, 105, 403–412. <https://doi.org/10.1016/j.buildenv.2016.05.034>.
- Killian, M., Leitner, A., Goldgruber, R., & Kozek, M. (2017). Adaptive model predictive control for energy-efficient smart homes. *7th international symposium on energy, number 8, Manchester, UK*.
- Kim, D., & Braun, J. E. (2018). Hierarchical model predictive control approach for optimal demand response for small/medium-sized commercial buildings. *2018 annual American control conference (ACC), Milwaukee, WI, USA* (pp. 5393–5398). IEEE.
- Kim, D., Witmer, L., Brownson, J., & Braun, J. (2014). Impact of solar irradiance data on MPC performance of multizone buildings impact of solar estimation on MPC performance of multizone buildings. *International high performance buildings conference, Purdue, USA*.
- Kim, J. (2010). Recent advances in adaptive MPC. *Iccas 2010* (pp. 218–222). <https://doi.org/10.1109/ICCAS.2010.5669892>.
- Kircher, K. J., & Zhang, K. M. (2016). Testing building controls with the BLDG toolbox. *2016 American control conference (ACC), Boston, MA, USA* (pp. 1472–1477). <https://doi.org/10.1109/ACC.2016.7525124>.
- Klauco, M., & Kvasnica, M. (2014). Explicit MPC approach to PMV-based thermal comfort control. *53rd IEEE conference on decision and control, Los Angeles, California, USA* (pp. 4856–4861).
- Klauco, M., Drgoňa, J., Kvasnica, M., & Di Cairano, S. (2014). Building temperature control by simple MPC-like feedback laws learned from closed-loop data. *IFAC Proceedings Volumes*, 47(3), 581–586. <https://doi.org/10.3182/20140824-6-2A-1003.01633>. 19th IFAC World Congress
- Knudsen, M. D., & Petersen, S. (2016). Demand response potential of model predictive control of space heating based on price and carbon dioxide intensity signals. *Energy and Buildings*, 125, 196–204.
- Koller, T., Berkenkamp, F., Turchetta, M., & Krause, A. (2018). Learning-based model predictive control for safe exploration. *2018 IEEE conference on decision and control (CDC)* (pp. 6059–6066). <https://doi.org/10.1109/CDC.2018.8619572>.
- Kothare, M. V., Balakrishnan, V., & Morari, M. (1996). Robust constrained model predictive control using linear matrix inequalities. *Automatica*, 32, 1361–1379.
- Kristensen, N., Madsen, H., & Jørgensen, S. (2004a). Parameter estimation in stochastic grey-box models. *Automatica*, 40, 225–237. <https://doi.org/10.1016/j.automatica.2003.10.001>.
- Kristensen, N., Madsen, H., & Jørgensen, S. (2004b). Parameter estimation in stochastic grey-box models. *Automatica*, 40, 225–237. <https://doi.org/10.1016/j.automatica.2003.10.001>.
- Krupa, P., Danielson, C., Laughman, C., Bortoff, S. A., Burns, D. J., Di Cairano, S., & Limon, D. (2019). Modelica implementation of centralized MPC controller for a multi-zone heat pump. *2019 18th european control conference (ECC)* (pp. 1784–1789). <https://doi.org/10.23919/ECC.2019.8795616>.
- Kuboth, S., Heberle, F., König-Haagen, A., & Brüggemann, D. (2019). Economic model predictive control of combined thermal and electric residential building energy systems. *Applied Energy*, 240, 372–385. <https://doi.org/10.1016/j.apenergy.2019.01.097>.
- Kumar, K., Heirung, T. A. N., Patwardhan, S. C., & Foss, B. (2015). Experimental evaluation of a MIMO adaptive dual MPC. *IFAC-PapersOnLine*, 48(8), 545–550. <https://doi.org/10.1016/j.ifacol.2015.09.024>. 9th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2015
- Kumar, R., Wenzel, M. J., ElBsat, M. N., Risbeck, M. J., Drees, K. H., & Zavala, V. M. (2020). Stochastic model predictive control for central HVAC plants.
- Kusiak, A., & Xu, G. (2012). Modeling and optimization of HVAC systems using a dynamic neural network. *Energy*, 42(1), 241–250. <https://doi.org/10.1016/j.energy.2012.03.063>. 8th World Energy System Conference, WESC 2010
- Kvasnica, M., Takács, B., Holaza, J., & Ingole, D. (2015). Reachability analysis and control synthesis for uncertain linear systems in MPT. *Proceedings of the 8th IFAC symposium on robust control design* (pp. 302–307). Bratislava, Slovak Republic: Elsevier.
- Kvasnica, M., Takács, B., Holaza, J., & Cairano, S. D. (2015b). On region-free explicit model predictive control. *2015 54th IEEE conference on decision and control (CDC)* (pp. 3669–3674). <https://doi.org/10.1109/CDC.2015.7402788>.

- L. Chen, H. H., & Hu, E. (2016). Reducing energy consumption for buildings under system uncertainty through robust MPC with adaptive bound estimator. *4th international high performance buildings conference, West Lafayette, IN, USA* (p. 3684). Berkeley Lab. The simulation research group developed tools. <http://simulationresearch.lbl.gov/>.
- National Renewable Energy Laboratory (2020). Alfalfa. <https://github.com/NREL/alfalfa>.
- Lambrichts, W. (2020). Model Predictive Control of Residential Heating : Accounting for Uncertainties on Weather and Occupancy Behaviour. Master thesis LU Leuven.
- Langson, W., Chrysoschoos, I., Raković, S. V., & Mayne, D. Q. (2004). Robust model predictive control using tubes. *Automatica*, *40*, 125–133.
- Lapusan, C., Balan, R., Hancu, O., & Plesa, A. (2016). Development of a multi-room building thermodynamic model using simscape library. *Energy Procedia*, *85*, 320–328. <https://doi.org/10.1016/j.egypro.2015.12.258>. EENVIRO-YRC 2015 - Bucharest
- Lara, B. G. V., Molina, L. M. C., Yanes, J. P. M., & Borroto, M. A. R. (2016). Offset-free model predictive control for an energy efficient tropical island hotel. *Energy and Buildings*, *119*, 283–292. <https://doi.org/10.1016/j.enbuild.2016.03.040>.
- Lauro, F., Longobardi, L., & Panziri, S. (2014). An adaptive distributed predictive control strategy for temperature regulation in a multizone office building. *2014 IEEE international workshop on intelligent energy systems (IWIES)* (pp. 32–37). <https://doi.org/10.1109/IWIES.2014.6957043>.
- Lazos, D., Sproul, A. B., & Kay, M. (2014). Optimisation of energy management in commercial buildings with weather forecasting inputs: A review. *Renewable and Sustainable Energy Reviews*, *39*, 587–603. <https://doi.org/10.1016/j.rser.2014.07.053>.
- Le, K., Bourdais, R., & Gueguen, H. (2014). Optimal control of shading system using hybrid model predictive control. *European control conference (ECC), 2014, Strasbourg, France* (pp. 134–139). <https://doi.org/10.1109/ECC.2014.6862492>.
- Le, K., Bourdais, R., & Guéguen, H. (2014b). From hybrid model predictive control to logical control for shading system: A support vector machine approach. *Energy and Buildings*, *84*, 352–359. <https://doi.org/10.1016/j.enbuild.2014.07.084>.
- Li, P., Li, D., Vrabie, D., Bengea, S., & Mijanovic, S. (2014). Experimental demonstration of model predictive control in a medium-sized commercial building. *International high performance buildings conference, west lafayette, IN, USA*. Purdue University.
- Li, P., O'Neill, Z., & Braun, J. (2013). Development of control-oriented models for model predictive control in buildings. *Ashrae Trans*, *119*(2).
- Li, P., Vrabie, D., Li, D., Bengea, S. C., Mijanovic, S., & O'Neill, Z. D. (2015). Simulation and experimental demonstration of model predictive control in a building HVAC system. *Science and Technology for the Built Environment*, *21*(6), 721–732.
- Li, X., & Malkawi, A. (2016). Multi-objective optimization for thermal mass model predictive control in small and medium size commercial buildings under summer weather conditions. *Energy*, *112*, 1194–1206. <https://doi.org/10.1016/j.energy.2016.07.021>.
- Liaw, A., & Wiener, M. (2001). Classification and regression by randomforest. *Forest*, *23*.
- van der Linden, A. C., Boerstra, A. C., Raue, A. K., Kurvers, S. R., & de Dear, R. J. (2006). Adaptive temperature limits: A new guideline in the Netherlands: A new approach for the assessment of building performance with respect to thermal indoor climate. *Energy and Buildings*, *38*(1), 8–17. <https://doi.org/10.1016/j.enbuild.2005.02.008>.
- Liu, H., Lee, S. C., Kim, M. J., Shi, H., Kim, J. T., Wasewar, K. L., & Yoo, C. K. (2013). Multi-objective optimization of indoor air quality control and energy consumption minimization in a subway ventilation system. *Energy and Buildings*, *66*, 553–561. <https://doi.org/10.1016/j.enbuild.2013.07.066>.
- Liu, S., & Henze, G. P. (2007). Evaluation of reinforcement learning for optimal control of building active and passive thermal storage inventory. *Journal of Solar Energy Engineering*, *129*(2), 215–225.
- Liu, X., Paritosh, P., Awalgaonkar, N. M., Bilonis, I., & Karava, P. (2018). Model predictive control under forecast uncertainty for optimal operation of buildings with integrated solar systems. *Solar Energy*, *171*, 953–970. <https://doi.org/10.1016/j.solener.2018.06.038>.
- Ljung, L. (1999). System identification: Theory for the user. In *Prentice Hall information and system sciences series*. Prentice Hall PTR.
- Ljung, L. (2006). *System identification toolbox, user's guide, version 6*. The MathWorks, Inc.
- Löfberg, J. (2004). YALMIP : A toolbox for modeling and optimization in MATLAB. *Proc. of the CACSD conference*. Taipei, Taiwan. Available from <http://users.isy.liu.se/johan/lyalmip/>
- Long, Y., Liu, S., Xie, L., & Johansson, K. H. (2014). A scenario-based distributed stochastic MPC for building temperature regulation. *2014 IEEE international conference on automation science and engineering (CASE)* (pp. 1091–1096). <https://doi.org/10.1109/CoASE.2014.6899461>.
- Lorenzen, M., Allgöwer, F., & Cannon, M. (2017). Adaptive model predictive control with robust constraint satisfaction. *Ifac papersonline* (pp. 3313–3318). <https://doi.org/10.1016/j.ifacol.2017.08.512>.
- Lorenzen, M., Allgöwer, F., & Cannon, M. (2017b). Adaptive model predictive control with robust constraint satisfaction. *IFAC-PapersOnLine*, *50*(1), 3313–3318. <https://doi.org/10.1016/j.ifacol.2017.08.512>. 20th IFAC World Congress
- Lorenzen, M., Allgöwer, F., Dabbene, F., & Tempo, R. (2015). Scenario-based stochastic MPC with guaranteed recursive feasibility. *2015 54th IEEE conference on decision and control (CDC)* (pp. 4958–4963). <https://doi.org/10.1109/CDC.2015.7402994>.
- Lorenzen, M., Dabbene, F., Tempo, R., & Allgöwer, F. (2017c). Constraint-tightening and stability in stochastic model predictive control. *IEEE Transactions on Automatic Control*, *62*(7), 3165–3177. <https://doi.org/10.1109/TAC.2016.2625048>.
- Lorenzen, M., Müller, M. A., & Allgöwer, F. (2017d). Stabilizing stochastic MPC without terminal constraints. *2017 American control conference (ACC), seattle, WA, USA* (pp. 5636–5641). <https://doi.org/10.23919/ACC.2017.7963832>.
- Lucia, S., Navarro, D., Karg, B., Sarnago, H., & Lucia, O. (2018). *Deep learning-based model predictive control for resonant power converters*.
- Ma, J., Qin, S. J., & Salsbury, T. (2014). Application of economic MPC to the energy and demand minimization of a commercial building. *Journal of Process Control*, *24*(8), 1282–1291. <https://doi.org/10.1016/j.jprocont.2014.06.011>. Economic nonlinear model predictive control
- Ma, Y. (2012). *Model Predictive Control for Energy Efficient Buildings*. University of California, Berkeley. Ph.D. thesis. Published by ProQuest LLC (2013), UMI:3593911
- Ma, Y., Anderson, G., & Borrelli, F. (2011). A distributed predictive control approach to building temperature regulation. *American control conference (ACC), 2011, San Francisco, California, USA* (pp. 2089–2094). IEEE.
- Ma, Y., Borrelli, F., Hency, B., Coffey, B., Bengea, S., & Haves, P. (2012). Model predictive control for the operation of building cooling systems. *Control Systems Technology IEEE Transactions*, *20*(3), 796–803.
- Ma, Y., Borrelli, F., Hency, B., Coffey, B., Bengea, S., & Haves, P. (2012b). Model predictive control for the operation of building cooling systems. *IEEE Transactions on Control Systems Technology*, *20*(3), 796–803. <https://doi.org/10.1109/TCST.2011.2124461>.
- Ma, Y., Borrelli, F., Hency, B., Packard, A., & Bortoff, S. (2009). Model predictive control of thermal energy storage in building cooling systems. *Proceedings of the 48th IEEE conference on decision and control (CDC) held jointly with 2009 28th chinese control conference* (pp. 392–397). <https://doi.org/10.1109/CDC.2009.5400677>.
- Ma, Y., Matusko, J., & Borrelli, F. (2015). Stochastic model predictive control for building HVAC systems: Complexity and conservatism. *IEEE Transactions on Control Systems Technology*, *23*(1), 101–116. <https://doi.org/10.1109/TCST.2014.2313736>.
- Ma, Y., Vichik, S., & Borrelli, F. (2012c). Fast stochastic MPC with optimal risk allocation applied to building control systems. *Decision and control (CDC), 2012 IEEE 51st annual conference on* (pp. 7559–7564). IEEE.
- Maasoumy, M., Moridian, B., Razmara, M., Shahbakhti, M., & Sangiovanni-Vincentelli, A. (2013). Online simultaneous state estimation and parameter adaptation for building predictive control. *ASME 2013 dynamic systems and control conference*. American Society of Mechanical Engineers. V002T23A006–V002T23A006
- Maasoumy, M., Razmara, M., Shahbakhti, M., & Vincentelli, A. S. (2014). Handling model uncertainty in model predictive control for energy efficient buildings. *Energy and Buildings*, *77*, 377–392.
- Maasoumy, M., & Sangiovanni-Vincentelli, A. (2012). Optimal control of building HVAC systems in the presence of imperfect predictions. *Proceedings of the 5th annual dynamic systems and control conference joint with the 11th motion and vibration conference, Fort Lauderdale, Florida, USA*.
- Maciejowski, J. M. (2002). *Predictive control with constraints*. Prentice Hall.
- Maddalena, E. T., Moraes, C. G. d. S., Waltrich, G., & Jones, C. N. (2019). A neural network architecture to learn explicit MPC controllers from data.
- Madsen, H., & Holst, J. (1995). Estimation of continuous-time models for the heat dynamics of a building. *Energy and Buildings*, *22*(1), 67–79. [https://doi.org/10.1016/0378-7788\(94\)00904-X](https://doi.org/10.1016/0378-7788(94)00904-X).
- Maeder, U., Borrelli, F., & Morari, M. (2009). Linear offset-free model predictive control. *Automatica*, *45*(10), 2214–2222. <https://doi.org/10.1016/j.automatica.2009.06.005>.
- Makhorin, A. (2012). GLPK - GNU Linear Programming Kit.
- Maniar, V. M., Shah, S. L., Fisher, D. G., & Muthas, R. K. (1997). Multivariable constrained adaptive GPC: Theory and experimental evaluation. *International journal of adaptive control and signal processing*, *11*, 343–365.
- del Mar, C. M., Álvarez, J. D., de A., R. F., & Berenguel, M. (2014). *Comfort control in buildings*. Springer-Verlag London.
- van Marken Lichtenbelt, W. D., & Kingma, B. R. (2013). Building and occupant energetics: a physiological hypothesis. *Architectural Science Review*, *56*(1), 48–53.
- Mathworks. Model Predictive Control Toolbox. <https://nl.mathworks.com/products/mpc.html>.
- Mattingley, J., & Boyd, S. (2012). Cvxgen: A code generator for embedded convex optimization. *Optimization and Engineering*, *13*(1), 1–27. <https://doi.org/10.1007/s11081-011-9176-9>.
- May-Ostendorp, P., Henze, G. P., Corbin, C. D., Rajagopalan, B., & Felsmann, C. (2011). Model-predictive control of mixed-mode buildings with rule extraction. *Building and Environment*, *46*(2), 428–437. <https://doi.org/10.1016/j.buildenv.2010.08.004>.
- May-Ostendorp, P., Henze, G. P., Corbin, C. D., Rajagopalan, B., & Felsmann, C. (2011b). Model-predictive control of mixed-mode buildings with rule extraction. *Building and Environment*, *46*(2), 428–437. <https://doi.org/10.1016/j.buildenv.2010.08.004>.
- Mayer, B., Killian, M., & Kozek, M. (2015). Management of hybrid energy supply systems in buildings using mixed-integer model predictive control. *Energy Conversion and Management*, *98*, 470–483. <https://doi.org/10.1016/j.enconman.2015.02.076>.
- Mayne, D. (2016). Robust and stochastic model predictive control: Are we going in the right direction? *Annual Reviews in Control*, *41*, 184–192. <https://doi.org/10.1016/j.arcontrol.2016.04.006>.
- Mayne, D. Q. (2014). Model predictive control: Recent developments and future promise. *Automatica*, *50*(12), 2967–2986. <https://doi.org/10.1016/j.automatica.2014.10.128>.
- Mazar, M. M., & Rezaeizadeh, A. (2020). Adaptive model predictive climate control of multi-unit buildings using weather forecast data. *Journal of Building Engineering*, *32*, 101449. <https://doi.org/10.1016/j.jobbe.2020.101449>.
- Mechri, H. E., Capozzoli, A., & Corrado, V. (2010). Use of the ANOVA approach for sensitive building energy design. *Applied Energy*, *87*(10), 3073–3083. <https://doi.org/10.1016/j.apenergy.2010.04.001>.
- Mesbah, A. (2016). Stochastic model predictive control: An overview and perspectives for future research. *IEEE Control Systems Magazine*, *36*(6), 30–44. <https://doi.org/10.1109/MCS.2016.2602087>.

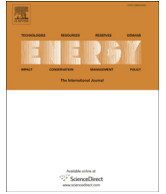
- Mesbah, A. (2018). Stochastic model predictive control with active uncertainty learning: A survey on dual control. *Annual Reviews in Control*, 45, 107–117. <https://doi.org/10.1016/j.arcontrol.2017.11.001>.
- Mirakhorli, A., & Dong, B. (2016). Occupancy behavior based model predictive control for building indoor climate—a critical review. *Energy and Buildings*, 129, 499–513.
- Modelon, A. B. (2017). JModelica.org. <http://www.jmodelica.org>.
- Moore, K. L., Vincent, T. L., Lashhab, F., & Liu, C. (2011). Dynamic consensus networks with application to the analysis of building thermal processes*. *IFAC Proceedings Volumes*, 44(1), 3078–3083. <https://doi.org/10.3182/20110828-6-IT-1002.02549.18th IFAC World Congress>
- Moroşan, P.-D., Bourdais, R., Dumur, D., & Buisson, J. (2010). Building temperature regulation using a distributed model predictive control. *Energy and Buildings*, 42(9), 1445–1452.
- Moroşan, P.-D., Bourdais, R., Dumur, D., & Buisson, J. (2011). A distributed MPC strategy based on benders decomposition applied to multi-source multi-zone temperature regulation. *Journal of Process Control*, 21(5), 729–737.
- Muntwiler, S., Wabersich, K. P., Carron, A., & Zeilinger, M. N. (2019). Distributed model predictive safety certification for learning-based control.
- Muske, K., & Badgwell, T. A. (2002). Disturbance modeling for offset-free linear model predictive control. *Journal of Process Control*, 12(5), 617–632.
- Muske, K. R., & Rawlings, J. B. (1993). Model predictive control with linear models. *AIChE Journal*, 39(2), 262–287.
- Mustafaraj, G., Lowry, G., & Chen, J. (2011). Prediction of room temperature and relative humidity by autoregressive linear and nonlinear neural network models for an open office. *Lancet*, 43, 1452–1460. <https://doi.org/10.1016/j.enbuild.2011.02.007>.
- Müller, D., Lauster, M., Constantin, A., Fuchs, M., & Remmen, P. (2016). AixLib - an open-source modelica library within the IEA-EBC annex 60 framework. *Proceedings of BAUSIM 2016 IBPSA Germany* (pp. 3–9).
- Nageler, P., Schweiger, G., Pichler, M., Brandl, D., Mach, T., Heimrath, R., ... Hoehenauer, C. (2018). Validation of dynamic building energy simulation tools based on a real test-box with thermally activated building systems (TABs). *Energy and Buildings*, 168, 42–55. <https://doi.org/10.1016/j.enbuild.2018.03.025>.
- Naidu, D. S., & Rieger, C. G. (2011). Advanced control strategies for heating, ventilation, air-conditioning, and refrigeration systems an overview: Part I: Hard control. *HVAC&R Research*, 17(1), 2–21. <https://doi.org/10.1080/10789669.2011.540942>.
- Nghiem, T. X. (2011). Green buildings: Optimization and adaptation. <http://www.seas.upenn.edu/~cis800/software.html>.
- Nghiem, T. X., & Jones, C. N. (2017). Data-driven demand response modeling and control of buildings with Gaussian Processes. *2017 American control conference (ACC), Seattle, WA, USA* (pp. 2919–2924). <https://doi.org/10.23919/ACC.2017.7963394>.
- Noh, H. Y., & Rajagopal, R. (2013). Data-driven forecasting algorithms for building energy consumption. *Proc. of SPIE*. <https://doi.org/10.1117/12.2009894>.
- Comite'Europe' en de Normalisation, C. (2007). Indoor environmental input parameters for design and assessment of energy performance of buildings addressing indoor air quality, thermal environment, lighting and acoustics. EN 15251.
- Nytsch-Geusen, C., Banhardt, C., Inderfurth, A., Mucha, K., Möckel, J., Rödler, J., ... Tugores, C. (2016). Buildingsystems - Eine modular hierarchische Modell-Bibliothek zur energetischen Gebäude und Anlagensimulation. *Proceedings of BAUSIM 2016 IBPSA germany* (pp. 473–480).
- Nyvt, O. (2009–2011). *Buses, Protocols and Systems for Home and Building Automation*. Czech Technical University in Prague. Faculty of Electrical Engineering. Department of Control Engineering. Ph.D. thesis.
- O'Dwyer, E., De Tommasi, L., Kouramas, K., Cychowski, M., & Lightbody, G. (2017). Prioritised objectives for model predictive control of building heating systems. *Control Engineering Practice*, 63, 57–68. <https://doi.org/10.1016/j.conengprac.2017.03.018>.
- Oldewurtel, F., Jones, C. N., & Morari, M. (2008). A tractable approximation of chance constrained stochastic MPC based on affine disturbance feedback. *7th IEEE conference on decision and control (CDC)* (pp. 4731–4736). <https://doi.org/10.1109/CDC.2008.4738806>.
- Oldewurtel, F., Jones, C. N., Parisio, A., & Morari, M. (2014). Stochastic model predictive control for building climate control. *IEEE Transactions on Control Systems Technology*, 22(3), 1198–1205. <https://doi.org/10.1109/TCST.2013.2272178>.
- Oldewurtel, F., Parisio, A., Jones, C. N., Gyalistras, D., Gwerder, M., Stauch, V., ... Morari, M. (2012). Use of model predictive control and weather forecasts for energy efficient building climate control. *Energy and Buildings*, 45, 15–27.
- Oldewurtel, F., Parisio, A., Jones, C. N., Morari, M., Gyalistras, D., Gwerder, M., ... Wirth, K. (2010). Energy efficient building climate control using stochastic model predictive control and weather predictions. *American control conference (ACC), 2010, Baltimore, Maryland, USA* (pp. 5100–5105). IEEE.
- Oldewurtel, F., Sturzenegger, D., & Morari, M. (2013). Importance of occupancy information for building climate control. *Applied energy*, 101, 521–532.
- Oldewurtel, F., Ulbig, A., Parisio, A., Andersson, G., & Morari, M. (2010b). Reducing peak electricity demand in building climate control using real-time pricing and model predictive control. *49th IEEE conference on decision and control (CDC)* (pp. 1927–1932).
- Oldewurtel, F., Ulbig, A., Parisio, A., Andersson, G., & Morari, M. (2010c). Reducing peak electricity demand in building climate control using real-time pricing and model predictive control. *Decision and control (CDC), 2010 49th IEEE conference on* (pp. 1927–1932). IEEE.
- Olesen, B. W. (2005). Indoor environment-health-comfort and productivity. *Proceedings of Clima*. <https://orbit.dtu.dk/en/publications/indoor-environment-health-comfo-rt-and-productivity>.
- O'Neill, Z., Narayanan, S., & Brahme, R. (2010). Model-based thermal load estimation in buildings. *Proceedings of SimBuild*, 4(1), 474–481.
- Oravec, J., Pakšiová, D., Bakošová, M., & Fikar, M. (2017). Soft-constrained alternative robust MPC: Experimental study, 20. *Preprints of the 20th IFAC world congress, Toulouse, France* (pp. 11877–11882). <https://doi.org/10.1016/j.ifacol.2017.08.2043>.
- Pang, X., Nouidui, T. S., Wetter, M., Fuller, D., Liao, A., & Haves, P. (2016). Building energy simulation in real time through an open standard interface. *Energy and Buildings*, 117, 282–289. <https://doi.org/10.1016/j.enbuild.2015.10.025>.
- Pannocchia, G., Gabbicini, M., & Artoni, A. (2015). Offset-free MPC explained: Novelty, subtleties, and applications. *IFAC-PapersOnLine*, 48(23), 342–351. <https://doi.org/10.1016/j.ifacol.2015.11.304>. 5th IFAC Conference on Nonlinear Model Predictive Control NMPC 2015
- Pannocchia, G., & Rawlings, J. B. (2003). Disturbance models for offset-free model-predictive control. *AIChE Journal*, 49(2), 426–437.
- Parisio, A., Fabbietti, L., Molinari, M., Varagnolo, D., & Johansson, K. H. (2014). Control of HVAC systems via scenario-based explicit MPC. *Decision and control (CDC), 2014 IEEE 53rd annual conference on* (pp. 5201–5207). <https://doi.org/10.1109/CDC.2014.7040202>.
- Patteeuw, D., Henze, G. P., & Helsen, L. (2016). Comparison of load shifting incentives for low-energy buildings with heat pumps to attain grid flexibility benefits. *Applied Energy*, 167, 80–92. <https://doi.org/10.1016/j.apenergy.2016.01.036>.
- Paulson, J. A., Buehler, E. A., Braatz, R. D., & Mesbah, A. (2017). Stochastic model predictive control with joint chance constraints. *International Journal of Control*, 93(1), 126–139. <https://doi.org/10.1080/00207179.2017.1323351>.
- Pcolka, M., Zacekova, E., Robinett, R., Celikovskiy, S., & Sebek, M. (2014). Economical nonlinear model predictive control for building climate control. *2014 American control conference, Portland, OR, USA* (pp. 418–423). IEEE. <https://doi.org/10.1109/ACC.2014.6858928>.
- Peng, Y., Rysanek, A., Nagy, Z., & Schlüter, A. (2018). Using machine learning techniques for occupancy-prediction-based cooling control in office buildings. *Applied Energy*, 211, 1343–1358. <https://doi.org/10.1016/j.apenergy.2017.12.002>.
- Picard, D., Dragoña, J., Kvasnica, M., & Helsen, L. (2017). Impact of the controller model complexity on model predictive control performance for buildings. *Energy and Buildings*, 152, 739–751. <https://doi.org/10.1016/j.enbuild.2017.07.027>.
- Picard, D., & Helsen, L. (2018). MPC performance for hybrid GEOTABS buildings. *Purdue conferences - 5th international high performance building conference, West Lafayette, IN, USA*. Purdue University, West Lafayette, IN, USA
- Picard, D., Jorissen, F., & Helsen, L. (2015). Methodology for obtaining linear state space building energy simulation models. *Proceedings of the 11th international modelica conference, Paris, France* (pp. 51–58).
- Picard, D., Jorissen, F., & Helsen, L. (2015b). Methodology for obtaining linear state space building energy simulation models. *11th international modelica conference* (pp. 51–58). Paris
- Picard, D., Sourbron, M., Jorissen, F., Cigler, J., Ferkl, L., & Helsen, L. (2016). Comparison of model predictive control performance using grey-box and white-box controller models. *Proceedings of the 4th international high performance buildings conference, West Lafayette, IN, USA* (pp. 1–10). West-Lafayette, Indiana, USA
- Polak, E., & Wetter, M. (2006). Precision control for generalized pattern search algorithms with adaptive precision function evaluations. *SIAM Journal on Optimization*, 16(3), 650–669. <https://doi.org/10.1137/040605527>.
- Pop, P. (2008). Comparing web applications with desktop applications: An empirical study. *Technical report*. Linköping University, Sweden.
- Prívra, S., Široký, J., Ferkl, L., & Cigler, J. (2011). Model predictive control of a building heating system: The first experience. *Energy and Buildings*, 43(2), 564–572.
- Prívra, S., Cigler, J., Vána, Z., Oldewurtel, F., Sagerschnig, C., & Žáčková, E. (2013). Building modeling as a crucial part for building predictive control. *Energy and Buildings*, 56(0), 8–22. <https://doi.org/10.1016/j.enbuild.2012.10.024>.
- Prívra, S., Cigler, J., Vána, Z., Oldewurtel, F., & Žáčková, E. (2013b). Use of partial least squares within the control relevant identification for buildings. *Control Engineering Practice*, 21(1), 113–121. <https://doi.org/10.1016/j.conengprac.2012.09.017>.
- Prívra, S., Vána, Z., Gyalistras, D., Cigler, J., Sagerschnig, C., Morari, M., & Ferkl, L. (2011). Modeling and identification of a large multi-zone office building. *2011 IEEE international conference on control applications (CCA), Denver, CO, USA* (pp. 55–60). <https://doi.org/10.1109/CCA.2011.6044402>.
- Putta, V., Zhu, G., Kim, D., Hu, J., & Braun, J. (2013). Comparative evaluation of model predictive control strategies for a building HVAC system. *2013 American control conference, Washington, DC, USA* (pp. 3455–3460). IEEE. <https://doi.org/10.1109/ACC.2013.6580365>.
- Putta, V., Zhu, G., Kim, D., Hu, J., & Braun, J. E. (2012). A distributed approach to efficient model predictive control of building HVAC systems. *International high performance buildings conference, West Lafayette, IN, USA*.
- Putta, V. K., Kim, D., Cai, J., Hu, J., & Braun, J. E. (2014). Distributed model predictive control for building HVAC systems: a case study. *International high performance buildings conference, West Lafayette, IN, USA*.
- Qin, S. J., & Badgwell, T. A. (2003). A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7), 733–764. [https://doi.org/10.1016/S0967-0661\(02\)00186-7](https://doi.org/10.1016/S0967-0661(02)00186-7).
- Qureshi, F. A., & Jones, C. N. (2018). Hierarchical control of building HVAC system for ancillary services provision. *Energy and Buildings*, 169, 216–227.
- Radecki, P., & Hencye, B. (2012). Online building thermal parameter estimation via unscented kalman filtering. *American control conference (ACC), 2012, Montréal, Canada* (pp. 3056–3062). IEEE.
- Rakovic, S. V., Kerrigan, E. C., Mayne, D. Q., & Lygeros, J. (2006). Reachability analysis of discrete-time systems with disturbances. *IEEE Transactions on Automatic Control*, 51(4), 546–561. <https://doi.org/10.1109/TAC.2006.872835>.

- Rangegowda, P. H., Valluru, J., Patwardhan, S. C., & Mukhopadhyay, S. (2018). Simultaneous state and parameter estimation using receding-horizon nonlinear kalman filter. *IFAC-PapersOnLine*, 51(18), 411–416. <https://doi.org/10.1016/j.ifacol.2018.09.335>. 10th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2018
- Rao, A. V. (2019). A survey of numerical methods for optimal control. Preprint AAS 09–334.
- Åström, K. J., & Wittenmark, B. (2008). *Adaptive control*. Courier Corporation.
- Rawlings, J. B., & Mayne, D. Q. (2009). *Model predictive control: Theory and design*. Nob Hill Pub. Madison, Wisconsin.
- Rawlings, J. B., Patel, N. R., Risbeck, M. J., Maravelias, C. T., Wenzel, M. J., & Turney, R. D. (2018). Economic MPC and real-time decision making with application to large-scale HVAC energy systems. *Computers & Chemical Engineering*, 114, 89–98. <https://doi.org/10.1016/j.compchemeng.2017.10.038>.FOCAPO/CPC 2017
- Rawlings, J. B., Amrit, R. (2008). Nonlinear model predictive control tools package.
- Rehrl, J., & Horn, M. (2011). Temperature control for HVAC systems based on exact linearization and model predictive control. *2011 IEEE international conference on control applications (CCA)*, Denver, CO, USA (pp. 1119–1124). <https://doi.org/10.1109/CCA.2011.6044437>.
- Reynders, G., Diriken, J., & Saelens, D. (2014). Quality of grey-box models and identified parameters as function of the accuracy of input and observation signals. *Energy and Buildings*, 82, 263–274.
- Rincón, F. D., Santoro, B. F., & Mendoza, D. F. (2016). Optimal control of a climatization system using energy and comfort objectives. *IFAC-PapersOnLine*, 49(32), 30–35. <https://doi.org/10.1016/j.ifacol.2016.12.185>. Cyber-Physical & Human-Systems CPHS 2016
- Rockett, P., & Hathway, E. A. (2017). Model-predictive control for non-domestic buildings: a critical review and prospects. *Building Research & Information*, 45(5), 556–571. <https://doi.org/10.1080/09613218.2016.1139885>.
- Rosenthal, R. E. (1988). GAMS: A User's Guide.
- Roth, K. W., Westphalen, D., Dieckmann, J., Hamilton, S. D., & Goetzler, W. (2002). Energy Consumption Characteristics of Commercial Building HVAC Systems - Volume III: Energy Savings Potential. *Technical Report*.
- Rouchier, S., Jiménez, M. J., & Castaño, S. (2019). Sequential Monte Carlo for on-line parameter estimation of a lumped building energy model. *Energy and Buildings*, 187, 86–94. <https://doi.org/10.1016/j.enbuild.2019.01.045>.
- Royer, S., Thil, S., Talbert, T., & Polit, M. (2014). A procedure for modeling buildings and their thermal zones using co-simulation and system identification. *Energy and Buildings*, 78, 231–237. <https://doi.org/10.1016/j.enbuild.2014.04.013>.
- Ruano, A. E., Crispim, E. M., Conceicao, E., & Lucio, M. (2006). Prediction of building's temperature using neural networks models. *Energy and Buildings*, 38(6), 682–694.
- Sagnol, G., & Stahlberg, M. (2018). PICOS—A Python Interface to Conic Optimization Solvers. <https://picos-api.github.io/picos/>.
- Sahinidis, N. V. (2017). BARON 17.8.9: Global Optimization of Mixed-Integer Nonlinear Programs, *User's Manual*.
- Sangoboye, F. C., Arendt, K., Singh, A., Veje, C. T., Kjaergaard, M. B., & Jørgensen, B. N. (2017). Performance comparison of occupancy count estimation and prediction with common versus dedicated sensors for building model predictive control. *Building Simulation*, 10(6), 829–843. <https://doi.org/10.1007/s12273-017-0397-5>.
- Santos, R. M., Zong, Y., Sousa, J. M. C., Mendonca, L., & Thavlov, A. (2016). Nonlinear economic model predictive control strategy for active smart buildings. *2016 IEEE PES innovative smart grid technologies conference europe (ISGT-europe)* (pp. 1–6). <https://doi.org/10.1109/ISGTEurope.2016.7856245>.
- Scattolini, R. (2009). Architectures for distributed and hierarchical model predictive control—a review. *Journal of Process Control*, 19(5), 723–731.
- Scattolini, R., & Colaneri, P. (2007). Hierarchical model predictive control. *46th IEEE conference on decision and control, 2007, New Orleans, LA, USA* (pp. 4803–4808). IEEE.
- Schellen, L., van Marken Lichtenbelt, W. D., Loomans, M., Toftum, J., & De Wit, M. H. (2010). Differences between young adults and elderly in thermal comfort, productivity, and thermal physiology in response to a moderate temperature drift and a steady-state condition. *Indoor Air*, 20(4), 273–283.
- Scherer, H. F., Pasamontes, M., Guzmán, J. L., Álvarez, J. D., Camponogara, E., & Normey-Rico, J. E. (2014). Efficient building energy management using distributed model predictive control. *Journal of Process Control*, 24(6), 740–749. <https://doi.org/10.1016/j.jprocont.2013.09.024>.
- Schildbach, G., Fagiano, L., Frei, C., & Morari, M. (2014). The scenario approach for stochastic model predictive control with bounds on closed-loop constraint violations. *Automatica*, 50(12), 3009–3018. <https://doi.org/10.1016/j.automatica.2014.10.035>.
- Schmelas, M., Feldmann, T., & Bollin, E. (2017). Savings through the use of adaptive predictive control of thermo-active building systems (TABS): A case study. *Applied Energy*, 199, 294–309. <https://doi.org/10.1016/j.apenergy.2017.05.032>.
- Scianca, N., Rosolia, U., & Borrelli, F. (2019). Learning model predictive control for periodic repetitive tasks.
- Serale, G., Fiorentini, M., Capozzoli, A., Bernardini, D., & Bemporad, A. (2018). Model predictive control (MPC) for enhancing building and HVAC system energy efficiency: Problem formulation, applications and opportunities. *Energies*, 11(3). <https://doi.org/10.3390/en11030631>.
- Shaikh, P. H., Nor, N. B. M., Nallagownden, P., Elamvazuthi, I., & Ibrahim, T. (2014). A review on optimized control systems for building energy and comfort management of smart sustainable buildings. *Renewable and Sustainable Energy Reviews*, 34, 409–429. <https://doi.org/10.1016/j.rser.2014.03.027>.
- Shan, K., Fan, C., & Wang, J. (2019). Model predictive control for thermal energy storage assisted large central cooling systems. *Energy*, 179, 916–927. <https://doi.org/10.1016/j.energy.2019.04.178>.
- Shang, C., & You, F. (2019). A data-driven robust optimization approach to scenario-based stochastic model predictive control. *Journal of Process Control*, 75, 24–39. <https://doi.org/10.1016/j.jprocont.2018.12.013>.
- Shi, Z., & O'Brien, W. (2019). Sequential state prediction and parameter estimation with constrained dual extended kalman filter for building zone thermal responses. *Energy and Buildings*, 183, 538–546. <https://doi.org/10.1016/j.enbuild.2018.11.024>.
- Siegelmann, H. T., & Sontag, E. D. (1995). On the computational power of neural nets. *Journal of Computer and System Sciences*, 50(1), 132–150.
- Skeledzija, N., Cesic, J., Koco, E., Bachler, V., Nikola, H. V., & Dzapo, H. (2014). Smart home automation system for energy efficient housing. *2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO)*. University of Zagreb, Croatia. Faculty of Electrical Engineering and Computing.
- Skogestad, S., & Postlethwaite, I. (2007). *Multivariable feedback control: Analysis and design* (vol. 2). Wiley New York.
- Smarra, F., Jain, A., de Rubeis, T., Ambrosini, D., D'Innocenzo, A., & Mangharam, R. (2018). Data-driven model predictive control using random forests for building energy optimization and climate control. *Applied Energy*, 226, 1252–1272. <https://doi.org/10.1016/j.apenergy.2018.02.126>.
- Soloperto, R., Müller, M. A., Trimpe, S., & Allgöwer, F. (2018). Learning-based robust model predictive control with state-dependent uncertainty. *IFAC-PapersOnLine*, 51(20), 442–447. <https://doi.org/10.1016/j.ifacol.2018.11.052>. 6th IFAC Conference on Nonlinear Model Predictive Control NMPC 2018
- Sourbron, M., & Helsén, L. (2011). Evaluation of adaptive thermal comfort models in moderate climates and their impact on energy use in office buildings. *Energy and Buildings*, 43(2–3), 423–432.
- Sourbron, M., Verhelst, C., & Helsén, L. (2013b). Building models for model predictive control of office buildings with concrete core activation. *Journal of Building Performance Simulation*, 6(3), 175–198.
- Sousa, J. (2012). Energy simulation software for buildings: Review and comparison, technical report.
- Sra, S., Nowozin, S., & Wright, S. J. (2011). *Optimization for machine learning*. The MIT Press.
- Stewart, B. T., Venkat, A. N., Rawlings, J. B., Wright, S. J., & Pannocchia, G. (2010). Cooperative distributed model predictive control. *Systems & Control Letters*, 59(8), 460–469.
- Sturm, J. F. (2003). SeDuMi. <http://sedumi.ie.lehigh.edu/>.
- Sturzenegger, D., Gyalistras, D., Gwerder, M., Sagerschnig, C., Morari, M., & Smith, R. S. (2013). Model predictive control of a Swiss office building. *Clima-Rheva world congress* (pp. 3227–3236).
- Sturzenegger, D., Gyalistras, D., Morari, M., & Smith, R. S. (2016). Model predictive climate control of a Swiss office building: Implementation, results, and cost-benefit analysis. *IEEE Transactions on Control Systems Technology*, 24(1), 1–12. <https://doi.org/10.1109/TCST.2015.2415411>.
- Sturzenegger, D., Gyalistras, D., Semeraro, V., Morari, M., & Smith, R. S. (2014). BRCM Matlab Toolbox: Model generation for model predictive building control. *2014 American control conference, Portland, OR, USA* (pp. 1063–1069). <https://doi.org/10.1109/ACC.2014.6858967>.
- Tanaskovic, M., F. L., Smith, R., & Morari, M. (2014). Adaptive receding horizon control for constrained MIMO systems. *Automatica*, 50(12), 3019–3029. <https://doi.org/10.1016/j.automatica.2014.10.036>.
- Tanaskovic, M., Sturzenegger, D., Smith, R., & Morari, M. (2017). Robust adaptive model predictive building climate control. *IFAC-PapersOnLine*, 50(1), 1871–1876. <https://doi.org/10.1016/j.ifacol.2017.08.257>. 20th IFAC World Congress
- Tang, W. S., & Wang, J. (2001). A recurrent neural network for minimum infinity-Norm kinematic control of redundant manipulators with an improved problem formulation and reduced architecture complexity. *IEEE Transactions on Systems Man and Cybernetics Part B*, 31(1), 98–105.
- Tanner, R. A., & Henze, G. P. (2014). Stochastic control optimization for a mixed mode building considering occupant window opening behaviour. *Journal of Building Performance Simulation*, 7(6), 427–444. <https://doi.org/10.1080/19401493.2013.863384>.
- Tarragona, J., Fernández, C., & de Gracia, A. (2020). Model predictive control applied to a heating system with PV panels and thermal energy storage. *Energy*, 197, 117229. <https://doi.org/10.1016/j.energy.2020.117229>.
- Tatjewski, P. (2011). Disturbance modeling and state estimation for predictive control with different state-space process models. *IFAC Proceedings Volumes*, 44(1), 5326–5331. <https://doi.org/10.3182/20110828-6-IT-1002.00440>. 18th IFAC World Congress
- Tesfay, M., Alsaleem, F., Arunasalam, P., & Rao, A. (2018). Adaptive-model predictive control of electronic expansion valves with adjustable setpoint for evaporator superheat minimization. *Building and Environment*, 133, 151–160. <https://doi.org/10.1016/j.buildenv.2018.02.015>.
- The MathWorks, I. (2000). MATLAB® – the language of technical computing. The MathWorks, Inc. Natick, MA.
- Tian, W., Heo, Y., de Wilde, P., L., Z., Yane, D., Park, C. S., ... Augenbro, G. (2018). A review of uncertainty analysis in building energy assessment. *Renewable and Sustainable Energy Reviews*, 93, 285–301. <https://doi.org/10.1016/j.rser.2018.05.029>.
- Toh, K. C., Todd, M. J., & Tütüncü, R. H. (1999). SDPT3 – A matlab software package for semidefinite programming. *Optimization Methods and Software*, 11/12, 545–581.
- Torrisi, G., Frick, D., Robbani, T., Grammatico, S., Smith, R. S., & Morari, M. (2017). FalOpt: First order Algorithm via Linearization of Constraints for OPTimization. <https://github.com/torrisig/FalOpt>.
- Touretzky, C. R., & Baldea, M. (2014). Nonlinear model reduction and model predictive control of residential buildings with energy recovery. *Journal of Process Control*, 24

- (6), 723–739. <https://doi.org/10.1016/j.jprocont.2013.09.022>. Energy Efficient Buildings Special Issue
- Trcka, M., & Hensen, J. L. M. (2010). Overview of HVAC system simulation. *Automation in Construction*, 19(2), 93–99.
- Tridium, I. (2019). Niagara®. Available at <https://www.tridium.com/>.
- Trcka, M., Hensen, J. L. M., & Wetter, M. (2009). Co-simulation of innovative integrated HVAC systems in buildings. *Journal of Building Performance Simulation*, 2(3), 209–230. <https://doi.org/10.1080/19401490903051959>.
- Ullmann, F. (2011). FiOrdOs: A Matlab Toolbox for C-code Generation for First Order Methods. <http://fiordos.ethz.ch/dokuwiki/doku.php>.
- US Department of Energy. Building energy software tools (BEST) directory. <https://www.buildingenergysoftwaretools.com/>.
- Van Overschee, P., & De Moor, B. (1996). *Subspace identification for linear systems* (pp. 57–93). Springer. https://doi.org/10.1007/978-1-4613-0465-4_3.
- Vande Cavey, M., De Coninck, R., & Helsen, L. (2014). Setting up a framework for model predictive control with moving horizon state estimation using JModelica. *10th international modelica conference, March 10–12, 2014, Lund, Sweden*. <https://liirias.kuleuven.be/retrieve/331504DPoster> [freely available]
- Vandenbergh, L., & Boyd, S. (1996). Semidefinite programming. *SIAM Rev.*, 38(1), 49–95. <https://doi.org/10.1137/1038003>.
- Vandermeulen, A., Vandeplas, L., Patteeuw, D., Sourbron, M., & Helsen, L. (2017). Flexibility offered by residential floor heating in a smart grid context: the role of heat pumps and renewable energy sources in optimization towards different objectives. Venkat, A. N., Rawlings, J. B., & Wright, S. J. (2005). Stability and optimality of distributed model predictive control. *Decision and control, 2005 and 2005 european control conference. CDC-ECC'05. 44th IEEE conference on* (pp. 6680–6685). IEEE.
- Verhelst, C. (2012). Model predictive control of ground coupled heat pump systems in office buildings (modelgebaseerde regeling van grondgekoppelde warmtepompsystemen in kantoorgebouwen).
- Verschueren, R., Frison, G., Kouzoupis, D., van Duijkeren, N., Zanelli, A., Novoselnik, B., Frey, J., Albin, T., Quiryren, R., & Diehl, M. (2019). acados: a modular open-source framework for fast embedded optimal control.
- Vogler-Finck, P., Wisniewski, R., & Popovski, P. (2018). Reducing the carbon footprint of house heating through model predictive control—a simulation study in danish conditions. *Sustainable Cities and Society*.
- Vogler-Finck, P. J. C., Pedersen, P. D., Popovski, P., & Wisniewski, R. (2017). Comparison of strategies for model predictive control for home heating in future energy systems. *Powertech, 2017 IEEE manchester* (pp. 1–6). IEEE.
- Vrettos, E., Lai, K., Oldewurtel, F., & Andersson, G. (2013). Predictive control of buildings for demand response with dynamic day-ahead and real-time prices. *European Control Conference (ECC), Zurich, Switzerland* (pp. 2527–2534). IEEE.
- Široký, J., Oldewurtel, F., Cigler, J., & Průvara, S. (2011). Experimental analysis of model predictive control for an energy efficient building heating system. *Applied Energy*, 88(9), 3079–3087.
- Wächter, A., & Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1), 25–57. <https://doi.org/10.1007/s10107-004-0559-y>.
- Walker, S. S. W., Lombardi, W., Lesecq, S., & Roshany-Yamchi, S. (2017). Application of distributed model predictive approaches to temperature and CO2 concentration control in buildings. *IFAC-PapersOnLine*, 50(1), 2589–2594. <https://doi.org/10.1016/j.ifacol.2017.08.107>.
- Wallace, M., Mhaskar, P., House, J., & Salsbury, T. (2014). Offset-free model predictive controller of a heat pump. *2014 American control conference* (pp. 2247–2252). <https://doi.org/10.1109/ACC.2014.6859114>.
- Wang, J.-J., Jing, Y.-Y., Zhang, C.-F., & Zhao, J.-H. (2009). Review on multi-criteria decision analysis aid in sustainable energy decision-making. *Renewable and Sustainable Energy Reviews*, 13(9), 2263–2278. <https://doi.org/10.1016/j.rser.2009.06.021>.
- Wang, S., & Ma, Z. (2008). Supervisory and optimal control of building HVAC systems: A review. *HVAC&R Research*, 14(1), 3–32. <https://doi.org/10.1080/10789669.2008.10390991>.
- Wang, Y., & Boyd, S. (2010). Fast model predictive control using online optimization. *IEEE Transactions on Control Systems Technology*, 18(2), 267–278. <https://doi.org/10.1109/TCST.2009.2017934>.
- Wetter, M. (2001). GenOpt - a generic optimization program. In R. Lamberts, C. O. R. Negrão, & J. Hensen (Eds.), *vol. I. Proc. of the 7th IBPSA conference, Rio de Janeiro* (pp. 601–608).
- Wetter, M. (2004). *Simulation-Based Building Energy Optimization*. University of California at Berkeley. Ph.D. thesis.
- Wetter, M. (2011). *A view on future building system modeling and simulation. book chapter in building performance simulation for design and operation*. Routledge, UK.
- Wetter, M. (2013). Fan and pump model that has a unique solution for any pressure boundary condition and control signal. In J. J. Roux, & M. Woloszyn (Eds.), *Proc. of the 13-th IBPSA conference* (pp. 3505–3512).
- Wetter, M., Bonvini, M., & Nouidui, T. S. (2016). Equation-based languages - A new paradigm for building energy modeling, simulation and optimization. *Energy and Buildings*, 117, 290–300. <https://doi.org/10.1016/j.enbuild.2015.10.017>.
- Wetter, M., & Haugstetter, C. (2006). Modelica versus TRNSYS - A comparison between an equation-based and a procedural modeling language for building energy simulation. *Proceedings of SimBuild, Second National IBPSA-USA Conference, Cambridge, MA*, 2(1).
- Wetter, M., & Haves, P. (2008). A modular building controls virtual test bed for the integration of heterogeneous systems. *Third national conference of IBPSA-USA, Berkeley/California*.
- Wetter, M., & Polak, E. (2004). Building design optimization using a convergent pattern search algorithm with adaptive precision simulations. *Energy and Buildings*, 37(6), 603–612. <https://doi.org/10.1016/j.enbuild.2004.09.005>.
- Wetter, M., & Wright, J. (2004). A comparison of deterministic and probabilistic optimization algorithms for nonsmooth simulation-based optimization. *Building and Environment*, 39(8), 989–999. <https://doi.org/10.1016/j.buildenv.2004.01.022>.
- Wetter, M., Zuo, W., Nouidui, T., & Pang, X. (2014). Modelica buildings library. *Journal of Building Performance Simulation*, 7(4), 253–270.
- Williams, H. P. (1993). *Model building in mathematical programming* (3rd ed.). John Wiley & Sons.
- Wollsen, M. G., & Jørgensen, B. N. (2015). Improved local weather forecasts using artificial neural networks. In S. Omatu, Q. M. Malluhi, S. R. Gonzalez, G. Bociewicz, E. Bucciarelli, G. Giulioni, & F. Iqba (Eds.), *Distributed computing and artificial intelligence, 12th international conference* (pp. 75–86). Cham: Springer International Publishing.
- Xiao, Y., Hou, X., Cai, J., & Hu, J. (2018). A differentially private distributed solution approach to the model predictive control of building clusters. *2018 IEEE conference on decision and control (CDC)* (pp. 7289–7295). IEEE.
- Xie, L., Li, P., & Wozny, G. (2007). *Chance constrained nonlinear model predictive control*. In R. Findeisen, F. Allgöwer, & L. T. Biegler (Eds.) (pp. 295–304). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Xu, X., Wang, S., & Huang, G. (2010). Robust MPC for temperature control of air-conditioning systems concerning on constraints and multitype uncertainties. *Building Services Engineering Research and Technology*, 31(1), 39–55. <https://doi.org/10.1177/0143624409352420>.
- Yadbantung, R., & Bumroongsri, P. (2018). Tube-based robust output feedback MPC for constrained LTV systems with applications in chemical processes. *European Journal of Control*.
- Yahiaoui, A., Hensen, J. L. M., & Soethout, L. (2003). Integration of control and building performance simulation software by run-time coupling. *Proceedings of 8th international IBPSA conference, international building performance simulation association, Eindhoven, The Netherlands* (pp. 1435–1442).
- Yan, D., O'Brien, W., Hong, T., Feng, X., Burak Gunay, H., Tahmasebi, F., & Mahdavi, A. (2015). Occupant behavior modeling for building performance simulation: Current state and future challenges. *Energy and Buildings*, 107, 264–278. <https://doi.org/10.1016/j.enbuild.2015.08.032>.
- Yan, S., Goulart, P., & Cannon, M. (2018). Stochastic model predictive control with discounted probabilistic constraints. *CoRR abs/1807.07465*.
- Yang, S., Wan, M. P., Chen, W., Ng, B. F., & Zhai, D. (2019). An adaptive robust model predictive control for indoor climate optimization and uncertainties handling in buildings. *Building and Environment*, 163, 106326. <https://doi.org/10.1016/j.buildenv.2019.106326>.
- Yang, S., Wan, M. P., Ng, B. F., Zhang, T., Babu, S., Zhang, Z., ... Dubey, S. (2018). A state-space thermal model incorporating humidity and thermal comfort for model predictive control in buildings. *Energy and Buildings*, 170, 25–39. <https://doi.org/10.1016/j.enbuild.2018.03.082>.
- Zakula, T., Armstrong, P. R., & Norford, L. (2014). Modeling environment for model predictive control of buildings. *Energy and Buildings*, 85, 549–559. <https://doi.org/10.1016/j.enbuild.2014.09.039>.
- Zeilinger, M. N., Raimondo, D. M., Domahidi, A., Morari, M., & Jones, C. N. (2014). On real-time robust model predictive control. *Automatica*, 50(3), 683–694. <https://doi.org/10.1016/j.automatica.2013.11.019>.
- Zhang, L., Wang, J., & Wang, B. (2014). A multi-step robust model predictive control scheme for polytopic uncertain multi-input systems. *Preprints of the 19th IFAC world congress, Cape Town, South Africa*.
- Zhang, X., Bujarbaruah, M., & Borrelli, F. (2019). Near-optimal rapid MPC using neural networks: A primal-dual policy learning framework.
- Zhang, X., Grammatico, S., Schildbach, G., Goulart, P., & Lygeros, J. (2014). On the sample size of randomized MPC for chance-constrained systems with application to building climate control. *2014 european control conference (ECC)* (pp. 478–483). <https://doi.org/10.1109/ECC.2014.6862498>.
- Zhang, X., Schildbach, G., Sturzenegger, D., & Morari, M. (2013). Scenario-based MPC for energy-efficient building climate control under weather and occupancy uncertainty. *2013 european control conference (ECC)* (pp. 1029–1034). <https://doi.org/10.23919/ECC.2013.6669664>.
- Zhang, Y., O'Neill, Z., Dong, B., & Augenbroe, G. (2015). Comparisons of inverse modeling approaches for predicting building energy performance. *Building and Environment*, 86, 177–190. <https://doi.org/10.1016/j.buildenv.2014.12.023>.
- Zhao, H., Shen, J., Li, Y., & Bentsman, J. (2017). Preference adjustable multi-objective NMPC: An unreachable prioritized point tracking method. *ISA Transactions*, 66, 134–142. <https://doi.org/10.1016/j.isatra.2016.09.020>.
- Zhao, Y., Lu, Y., Yan, C., & Wang, S. (2015). Mpc-based optimal scheduling of grid-connected low energy buildings with thermal energy storages. *Energy and Buildings*, 86, 415–426. <https://doi.org/10.1016/j.enbuild.2014.10.019>.
- Zhou, X., Hong, T., & Yan, D. (2013). Comparison of building energy modeling programs: HVAC systems. *Building Simulation*.
- Zong, Y., Böning, G. M., Santos, R. M., You, S., Hu, J., & Han, X. (2017). Challenges of implementing economic model predictive control strategy for buildings interacting with smart energy systems. *Applied Thermal Engineering*, 114, 1476–1486.
- Zong, Y., Su, W., Wang, J., Rodek, J. K., Jiang, C., Christensen, M. H., ... Mu, S. (2019). Model predictive control for smart buildings to provide the demand side flexibility in the multi-carrier energy context: Current status, pros and cons, feasibility and

barriers. *Energy Procedia*, 158, 3026–3031. <https://doi.org/10.1016/j.egypro.2019.01.981>. Innovative Solutions for Energy Transitions Zurich, E., Building Automation and Control Tool (BACTool). <http://www.bactool.ethz.ch/>.

Žáčeková, E., Pčolka, M., Tabač, J., Těžký, J., Robinett, R., Čelikovský, S., & Šebek, M. (2015). Identification and energy efficient control for a building: Getting inspired by MPC. *2015 American control conference (ACC), Chicago, IL, USA* (pp. 1671–1676). <https://doi.org/10.1109/ACC.2015.7170973>.



Energy efficient convex-lifting-based robust control of a heat exchanger

Juraj Oravec^{*}, Michaela Horváthová, Monika Bakošová

Slovak University of Technology in Bratislava, Faculty of Chemical and Food Technology, Institute of Information Engineering, Automation and Mathematics, Radlinského 9, SK-812 37, Bratislava, Slovakia

ARTICLE INFO

Article history:

Received 15 January 2020

Received in revised form

6 March 2020

Accepted 6 April 2020

Available online 17 April 2020

Keywords:

Heat exchanger

Energy savings

Robust control

Convex lifting

ABSTRACT

This paper analyses control performance and energy savings reached by application of four robust control approaches implemented on the laboratory plate heat exchanger. The output temperature of the cold medium is the controlled variable and the flow rate of the hot medium is the manipulated variable. Closed-loop control of the heat exchanger aims to ensure offset-free setpoint temperature tracking. First, the computationally demanding robust model predictive control (RMPC) strategy is investigated. Improvements ensured by implementing three modifications of the convex-lifting-based robust control methods are analysed. These modifications are based on the (i) non-tunable, (ii) tunable, and (iii) multiple tunable robust positive invariant (RPI) sets. Designed robust control approaches significantly improved control performance. For considered control setup, the settling time was reduced up to 70%. Energy savings were increased by 82%, when compared to RMPC. Moreover, the considered methods are proper for real-time implementation as they significantly reduce computational demands. The designed robust convex-lifting-based methods are suitable for industrial hardware with limited computational requirements.

© 2020 Elsevier Ltd. All rights reserved.

1. Introduction

Efficiency of energy supply plays an important role in achieving objectives as the health of economies, sustainable industry, and others. Minimizing energy waste is the cornerstone of these objectives. In these days, over 80 % of energy utilization comprises some form of the heat transfer process [1]. Therefore, importance of enhanced heat transfer, thermal process modelling, and integration are significant. Recent developments in applied thermal engineering are listed in Ref. [2].

One of the areas with high energy saving potential is the control system design for heat exchangers. These devices are a major energy-intensive part of the industry. Control performance of these systems is a relevant indicator of process economic efficiency [3]. It is necessary to ensure simultaneously controllability and economy of heat exchanger operation, e.g., see Ref. [4]. Improvement of control performance is often associated with the application of more advanced control approaches.

One of advanced control strategies is the *model predictive control* (MPC). It is a model-based control strategy, that computes optimal control action subject to various control requirements and constraints, see Ref. [5]. MPC and its numerous variations have a great potential to optimize the control performance of heat exchangers. Infinite-horizon-based MPC design taking into account the economic objectives was verified on a heat-exchanger network in Ref. [6]. In Ref. [7], the economic nonlinear MPC was designed to improve the waste heat recovery of Organic Rankine Cycle. MPC designed for the thermal energy storage applied for large central cooling systems ensures energy savings as it is shown in work [8]. The approach based on the neural network and predictive control was designed for the heat exchangers in Ref. [9]. The energy efficiency of the designed approach was compared to the PID controllers. *Balance-Based Adaptive Control* methodology was used for controller synthesis in Ref. [10]. The practical implementation was addressed and the proposed controller was implemented using widely-used industrial Programmable Logic Controllers (PLCs).

The precise mathematical model of a heat exchanger is complex and it has nonlinear and asymmetric behaviour. Moreover, the industrial operation is influenced by various uncertain parameters, e.g., measurement noise, etc. Fouling also significantly affects the

^{*} Corresponding author.

E-mail addresses: juraj.oravec@stuba.sk (J. Oravec), michaela.horvathova@stuba.sk (M. Horváthová), monika.bakosova@stuba.sk (M. Bakošová).

mathematical modeling [11], its reliability [12], and energy-efficient control of heat exchangers [13]. To overcome these obstacles the *robust* model predictive control (RMPC) was introduced, see Ref. [14]. Advanced RMPC method was designed for the shell-and-tube heat exchangers in Ref. [15]. This approach was compared with conventional PID control. RMPC outperformed the conventional PID controllers almost by factor 2 and ensured significant energy savings. The robust control for a plate heat exchanger was designed in Ref. [16] and validated by the experimental case study. The energy savings ensured by RMPC designed for a heat exchanger were analysed in Ref. [17]. The implementation of the soft-constraints improved overall control performance and increased energy savings. However, RMPC is quite computationally demanding. The controller design requires efficient hardware to compute complex optimization problems using the *linear matrix inequalities*, see Ref. [18]. This fact limits RMPC from being implemented using PLCs or other embedded hardware platforms, that are frequently used in the industry.

Recently, in the control theory, the convex lifting is employed in three major areas: (i) design of simplified explicit MPC policies, that still ensure optimality and stability [19,20]; (ii) reduction of the tail of the predicted control sequences by solving an inverse parametric program [21,22]; (iii) design of robust control strategies [23–25]. In Ref. [23], the output feedback control for discrete-time linear systems affected by bounded additive state and output disturbances was designed using the convex lifting. The linear model predictive control deployed on an 8-bit micro-controller unit via convex lifting was designed in Ref. [19]. The 8-bit micro-controller represents a low-cost embedded platform, which is built into mass-produced devices. Therefore, convex-lifting-based approaches could become widely implementable and still provide optimal and stable control performance.

The novel convex-lifting-based robust control approach was introduced in Ref. [24]. Compared to the conventional RMPC, the convex-lifting-based robust control design minimizes computational demand of real-time control. This approach preserves the main benefits of RMPC, i.e., optimizes the control performance w.r.t. the constraints on the manipulated variables and the controlled variables. Compared to conventional RMPC, convex-lifting-based robust control minimizes computational demand. Nevertheless, it can still respect constraints and compute optimal control action similar to RMPC. The control performance is determined by the properties of the so-called robust positive invariant (RPI) set. This approach was modified in Ref. [25] to explicitly take into account the MPC-like quality criteria. It leads to the possibility of *tuning* the properties of the RPI set. Moreover, the multiple RPI sets were designed to improve the control performance.

The main contribution of this paper is the experimental investigation of several robust control strategies implemented on the laboratory plate heat exchanger. This paper directly extends the results presented in Ref. [26], where the convex-lifting-based robust control was designed for the plate heat exchanger. The numerical simulation of the closed-loop control indicated the high potential to ensure significant energy savings. Compared to Refs. [26], this paper presents the experimental results. Three modifications of robust control methods were implemented and the control performance was analysed. The obtained results were compared to the RMPC method. Particularly, this paper analyses the control performance and energy savings of the following convex-lifting-based robust control method: (i) non-tunable approach introduced in Refs. [24], (ii) tunable approach with a single RPI set, and (iii) multiple tunable approach with two RPI sets [25]. The obtained results were compared to the conventional RMPC applied for the same heat exchanger in Ref. [17].

The paper is structured as follows: the considered laboratory

plate heat exchanger is described in Section 2. Section 3 introduces the proposed robust controller design methods. Section 4 investigates the measured results and is followed by the main conclusions summarized in Section 5.

2. Plate heat exchanger

This section describes in the detail the plant and its mathematical model considered in the experimental case study. The uncertain mathematical model is introduced. This model of the plate heat exchanger is augmented for applying integral control action to remove the steady-state control error.

2.1. Description of the heat exchanger

Plate heat exchangers are nonlinear and asymmetric plants widely used in all fields of industry. Effective control of these devices saves energy, resources, and production costs. The plant behaviour and energy efficiency of the heat exchangers differ according to the size of the setpoint change or its initial conditions. However, the main differences occur between negative and positive changes of the setpoint. Applying robust control strategy to a complex plant ensures satisfying control performance for each change of the setpoint, regardless of the initial conditions or size of the step.

A laboratory plate heat exchanger was considered to investigate the energy efficiency of the proposed robust control strategy. The used heat exchanger is shown in Fig. 1. The liquid-liquid plate heat exchanger was manufactured by Armfield, specifically labelled as Armfield Process Control Trainer PCT23, see Ref. [27]. The three-stage plate heat exchanger (Fig. 1, device (I)) provides both, i.e., cooling or heating of the leaking liquid. In this experimental case study, the heating part was considered. The heated medium was cold water stored in two tanks (Fig. 1, device (II)). Hot medium, the hot water, was heated to a constant temperature $T_{\text{hot}} = 70^\circ \text{C}$ in a retention tank with a heating coil (Fig. 1, device (III)). The constant temperature of the hot medium was ensured by an auxiliary closed-loop with a simple PID controller. Both, hot and cold media were fed to the plate heat exchanger by peristaltic pumps (Fig. 1, device (IV), device (V)).

The operational task was to heat the cold medium to a setpoint temperature T_{ref} . From the control viewpoint, the aim of control was to ensure offset-free setpoint tracking. The output temperature of the heated medium T was the controlled variable (CV). As the

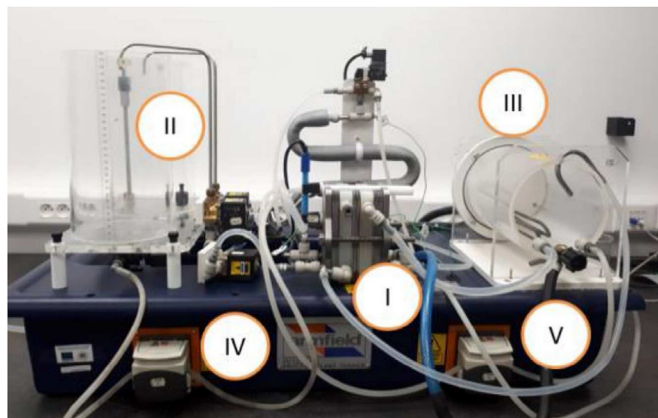


Fig. 1. The controlled plant – plate heat exchanger (I), tanks for cold medium (II), tank for hot medium with heating system (III), peristaltic pump used for cold medium (IV), peristaltic pump used for hot medium (V).

temperature of the heating medium was constant $T_{\text{hot}} = 70^\circ \text{C}$, the temperature of the heated medium T depended on the flow of the heating medium. Therefore, the manipulated variable (MV) was the volumetric flow rate of hot medium q . The actuator was the peristaltic pump for feeding the hot medium (Fig. 1, device (V)) and the sensor was the K -type thermocouple, see Ref. [27].

2.2. Mathematical model of the plate heat exchanger

To implement the robust control strategy to the heat exchanger, we need a sufficiently precise mathematical model of the controlled plant in a proper form. The convex-lifting-based robust control requires a discrete-time state-space model that describes also the effect of interval and additive uncertainties. In order to obtain the minimum and maximum values of the interval uncertainties, the set of multiple-step responses was measured in various working conditions. The step-response-based method of identification described in Ref. [28] was applied to identify the parameters of the transfer function in the following form:

$$G(s) = \frac{Z}{\tau s + 1} e^{-Ds}, \quad (1)$$

where s represents a complex number frequency parameter of Laplace transform, Z is the gain, τ stands for the time constant, and D stands for the time delay. Uncertain parameters were expressed using intervals bounded by minimum and maximum values of each system parameter, i.e., the gains Z_{\min}, Z_{\max} , the time constants τ_{\min}, τ_{\max} . The time delays were not considered as their values were not significant. Using the minimum and maximum values of gains and time constants 4 vertex systems were created. These vertex systems were transformed into state-space models in discrete-time domain.

$$x(k+1) = A(\lambda)x(k) + B(\lambda)u(k) + w(k), \quad (2a)$$

$$y(k) = Cx(k), \quad (2b)$$

$$x(0) = x_0, \quad (2c)$$

$$[A(\lambda), B(\lambda)] \in \mathbb{A}, \quad (2d)$$

$$\mathbb{A} = \text{convhull}((A^{(v)}, B^{(v)}), \forall v), \quad (2e)$$

$$y_{\min} \leq y(k) \leq y_{\max}, \quad (2f)$$

$$u_{\min} \leq u(k) \leq u_{\max}, \quad (2g)$$

where k stands for the discrete time instant, $x(k) \in \mathbb{R}^{n_x}$ are the system states with initial conditions x_0 , $u(k) \in \mathbb{R}^{n_u}$ is the manipulated variable, and $y(k) \in \mathbb{R}^{n_y}$ is the controlled variable. The measurement noise $w(k) \in \mathbb{W} \subset \mathbb{R}^{n_x}$ was considered as an additive disturbance determined as the maximum amplitude of the thermocouple measurement noise. The parameters y_{\min}, y_{\max} are lower and upper bounds of the output variable $y(k)$, and u_{\min}, u_{\max} are lower and upper bounds of the manipulated variable $u(k)$, respectively.

The state, input, and output matrices of the system in (2) are respectively represented by: $A(\lambda) \in \mathbb{R}^{n_x \times n_x}$, $B(\lambda) \in \mathbb{R}^{n_x \times n_u}$ having form

$$A(\lambda) = \sum_{v=1}^{n_v} \lambda_v A^{(v)}, \quad B(\lambda) = \sum_{v=1}^{n_v} \lambda_v B^{(v)}, \quad (3)$$

where the parameter v represents the v -th vertex of the polytopic uncertain system, and parameter $0 \leq \lambda_v \leq 1$ represents the parameter in the convex combination of the uncertain system vertices, see Refs. [25]. Parameter $v = 1, 2, \dots, n_v$, where n_v is the total number of the uncertain system vertices.

2.3. Augmented mathematical model of the plate heat exchanger

The controllers designed for industrial implementation are often expected to ensure offset-free setpoint tracking. To remove the steady-state control error, the mathematical representation of the plant was augmented by the term needed for adding the integral action to the controller. Considered augmented vector of the system states \hat{x} is as follows:

$$\hat{x}(k) = \begin{bmatrix} x(k) \\ \sum_{j=0}^k e(j) \end{bmatrix}, \quad (4)$$

where $e(k)$ is the control error given by

$$e(k) = r(k) - y(k) \quad (5)$$

and $r(k) \in \mathbb{R}^{n_y}$ stands for the setpoint value at the given time instance k .

The augmented matrices of the state space model in (2) are defined as follows:

$$\hat{A}(v) = \begin{bmatrix} A^{(v)} & 0 \\ -t_s C & I \end{bmatrix}, \hat{B}(v) = \begin{bmatrix} B^{(v)} \\ 0 \end{bmatrix}, \hat{C}(v) = [C^{(v)} \quad 0], \quad (6)$$

where t_s is the sampling time and the state, output and input matrices respectively are:

$$\hat{A}(v) = \begin{bmatrix} 0.6649 \pm 0.1470 & 0 \\ -2.6475 \pm 2.4183 & 1 \end{bmatrix}, \quad (7a)$$

$$\hat{B}^{(v)} = \begin{bmatrix} 4.0886 \pm 0.4249 \\ 0 \end{bmatrix}, \quad (7b)$$

$$\hat{C}(v) = [1 \quad 0] \quad (7c)$$

Further technical details regarding the identification of the plant were discussed in Ref. [29].

3. Convex-lifting-based robust control

The convex-lifting-based robust control was designed in two phases: (i) *offline* phase focused on the construction of the controller; and (ii) *online* phase that represents the real-time control.

In the offline phase, the controller is constructed based on the solution of the complex *parametric* optimization problem, see Ref. [30]. The advantage of solving the complex offline phase was to pre-compute the robust control law. The solution of this optimization problem lead to the construction of two *types* of sets: (i) *RPI sets* \mathbb{X}_{rpi} ; and (ii) *convex lifting sets* \mathbb{X}_{lift} .

The construction of the RPI set \mathbb{X}_{rpi} is the crucial part of the proposed approach. We considered three different approaches of RPI set construction that are introduced in detail in Sections

3.1–3.3.

The convex lifting set \mathbb{X}_{lift} was constructed by solving the following optimization problem of multi-parametric linear programming (mpLP):

$$\ell(x) = \min_z \quad (8a)$$

$$\text{s.t.} : x \in \mathbb{X}_{\text{lift}}, \quad (8b)$$

$$\begin{bmatrix} x \\ z \end{bmatrix} \in \Pi, \quad (8c)$$

where the decision variable is $z \in \mathbb{R}^1$ and the parameters are the system states $x \in \mathbb{R}^{n_x}$. The polytopic set $\Pi \in \mathbb{R}^{(n_x+1)}$ is the convex hull of RPI set vertices and vertices of convex lifting set \mathbb{X}_{lift} . Further technical details are in Ref. [24].

In the online phase, the optimal value of manipulated variable u is evaluated based on the pre-computed sets, i.e., either the convex lifting set \mathbb{X}_{lift} or RPI set \mathbb{X}_{rpi} . If the current measurement of the controlled variable y transformed into the system states x lies in the convex lifting set \mathbb{X}_{lift} , then the optimal value of manipulated variable u is computed by solving the following linear programming (LP):

$$\begin{bmatrix} \gamma^\star \\ u(k)^\star \end{bmatrix} = \arg \min_{\gamma, u(k)} \quad (9a)$$

$$\text{s.t.} : a_i^\top (A_j x(k) + B_j u(k) + w) + b_i \leq \gamma \ell(x(k)), \quad (9b)$$

$$\gamma \geq 0, \quad u(k) \in \mathbb{U}, \quad \forall w \in \mathbb{W}, \quad \forall i = 1, 2, \dots, n_p, \quad (9c)$$

$$\forall [A_j, B_j] \in \mathcal{V}(\mathbb{X}_{\text{rpi}}), \quad (9d)$$

where $\mathcal{V}(\mathbb{X}_{\text{rpi}})$ represents set of vertices of RPI set \mathbb{X}_{rpi} , ℓ is the lifting value, a_i , b_i determine polytopes defined above the convex lifting set \mathbb{X}_{lift} , and n_p is the total number of these polytopes. Technical details are described in Ref. [24].

On the other hand, if the system states x lie in RPI set \mathbb{X}_{rpi} , then the optimal value of manipulated variable u is computed by linear control law in the form:

$$u(k) = K x(k), \quad (10)$$

where $K \in \mathbb{R}^{n_u \times n_x}$ is the gain matrix of the controller.

Three convex-lifting-based robust control strategies were investigated in the paper: (i) *non-tunable* approach; (ii) *tunable* approach; (iii) *multiple tunable* approach.

The considered control strategies differ in the construction of RPI sets \mathbb{X}_{rpi} . *Non-tunable* approach was originally proposed in Ref. [24]. This control strategy does not take into account explicit requirements on control performance. *Tunable* approach was presented in Ref. [25]. The RPI set is constructed subject to the MPC-like weighting matrices. Therefore, the requirements on the control performance are implemented into the design of the RPI set. Finally, the *multiple tunable* approach was also introduced in Ref. [25]. This control method considered the construction of multiple RPI sets. Usually, two RPI sets are considered to optimize the control performance. The *non-aggressive* RPI set maximizes its volume to minimize the necessity to solve LP problems. Simultaneously, *aggressive* RPI set optimizes the convergence of the controlled variable into the setpoint. Each RPI set is designed subject to the associated pair of MPC-like weighting matrices.

3.1. Non-tunable approach

The construction of RPI set $\mathbb{X}_{\text{rpi}} \in \mathbb{R}^{n_x \times n_x}$ is crucial to determine control performance. The RPI set $\mathbb{X}_{\text{rpi}} = \mathbb{X}_0$ is constructed by solving the optimization problem of *semidefinite programming* (SDP), see Ref. [31]. SDPs have convex objective function, usually linear, and the constraints have the form of the *linear matrix inequalities* (LMIs), see Ref. [32].

The original *non-tunable* approach does not take into account explicit control performance criteria [24]. The control law in (10) is based on the solution of the following SDP:

$$\min_{X_0, Y} - \log \det(X_0) \quad (11a)$$

$$\text{s.t.} : \begin{bmatrix} X_0 & \star \\ A_j X_0 + B_j Y & X_0 \end{bmatrix} > 0, \quad (11b)$$

where X_0 is the inverted Lyapunov matrix, $Y \in \mathbb{R}^{n_u \times n_x}$ is auxiliary matrix of well-known parametrization of control law in (10):

$$X_0 K_0 = Y. \quad (12)$$

The symbol \star denotes the symmetric structure of LMI in (11b). The gain matrix of the control law in (10) is designed using (12) by

$$K_0 = X_0^{-1} Y. \quad (13)$$

Finally, the procedure of convex-lifting-based robust control considering non-tunable approach is summarized in [Algorithm 1](#), see Ref. [24].

Algorithm 1

Convex-lifting-based robust control using non-tunable RPI set.

-
- 1: **Input:** uncertain model in (2)
 - 2: **Output:** manipulated variable $u(k)$
 - 3: I. Off-line phase
 - 4: construct invariant set \mathbb{X}_0 and associated K_0 using (13)
 - 5: construct convex lifting above set \mathbb{X}_{lift}
 - 6: II. On-line phase
 - 7: **If** $x(k) \in \mathbb{X}_{\text{lift}}$ **then**
 - 8: solve LP in (9) to compute $u(k)$
 - 9: **Else**
 - 10: $u(k) = K_0 x(k)$
 - 11: **end if**
-

3.2. Tunable approach

Considering the tunable RPI set $\mathbb{X}_{\text{rpi}} = \mathbb{X}_1$ enables explicit implementation of control performance criteria, see Ref. [25].

The tunable RPI set \mathbb{X}_1 is constructed considering the MPC-like weighting matrices (Q_1, R_1) , where $Q_1 > 0, \in \mathbb{R}^{n_x \times n_x}, R_1 > 0, \in \mathbb{R}^{n_u \times n_u}$. These weighting matrices are tuned to satisfy

$$V(x(0)) \leq - \sum_{k=0}^{\infty} k = 0(x(k)^\top Q_1 x(k) + u(k)^\top R_1 u(k)), \quad (14)$$

where V is the Lyapunov function.

The RPI set \mathbb{X}_1 is given by the solution of

$$\min_{\gamma_j, X_1, Y_1} \gamma_j \quad (15a)$$

$$\text{s.t. : } \begin{bmatrix} X_j & \star & \star & \star \\ A^\nu X_1 + B^\nu Y_1 & X_1 & \star & \star \\ Q_1^{1/2} X_1 & 0 & \gamma_1 I & \star \\ R_1^{1/2} Y_1 & 0 & 0 & \gamma_1 I \end{bmatrix}, \quad (15b)$$

where $\nu = 1, \dots, n_\nu$, $\gamma_j > 0 \in \mathbb{R}$ is auxiliary optimized parameter, $Y_1 \in \mathbb{R}^{n_u \times n_x}$ is the auxiliary matrix of the control law parametrization, and $X_1 = X_1^\top > 0 \in \mathbb{R}^{n_x \times n_x}$ is the inverted weighted Lyapunov matrix.

The gain matrix of control law in (10) is designed using

$$K_1 = X_1^{-1} Y_1. \quad (16)$$

Finally, the procedure of convex-lifting-based robust control considering tunable approach is proposed in Algorithm 2, see Ref. [25].

Algorithm 2

Convex-lifting-based robust control using tunable RPI set.

-
- 1: **Input:** uncertain model in (2), MPC-like weighting matrices (Q_1, R_1)
 - 2: **Output:** manipulated variable $u(k)$
 - 3: I. Off-line phase:
 - 4: construct invariant set \mathbb{X}_1 and associated K_1 using (16)
 - 5: construct convex lifting above set \mathbb{X}_{lift}
 - 6: II. On-line phase:
 - 7: **if** $x(k) \in \mathbb{X}_{\text{lift}}$ **then**
 - 8: solve LP in (9) to compute $u(k)$
 - 9: **else**
 - 10: $u(k) = K_1 x(k)$
 - 11: **end if**
-

3.3. Multiple tunable approach

The multiple tunable RPI sets $\mathbb{X}_{\text{RPI}} \in \{\mathbb{X}_1, \mathbb{X}_2, \dots, \mathbb{X}_{n_s}\}$ improve the control performance by considering the set of control laws in (10), see Ref. [25]. These multiple tunable RPI sets are constructed considering the set of the MPC-like weighting matrices (Q_j, R_j) , where $Q_j > 0, \in \mathbb{R}^{n_x \times n_x}$, $R_j > 0, \in \mathbb{R}^{n_u \times n_u}$. These weighting matrices are tuned to satisfy

$$V\left(x\left(0\right)\right) \leq -\sum_{k=0}^{\infty}\left(x(k)^{\top} Q_j x(k)+u(k)^{\top} R_j u(k)\right), j=1,2, \dots, n_s, \quad (17)$$

where V is the Lyapunov function and n_s is total number of constructed RPI sets.

The set of n_s weighting matrices (Q_j, R_j) is considered in the following SDPs, see Ref. [18,25]:

$$\min_{\gamma_j, X_j, Y_j} \gamma_j \quad (18a)$$

$$\text{s.t. : } \begin{bmatrix} X_j & \star & \star & \star \\ A^{(\nu)} X_j + B^{(\nu)} Y_j & X_j & \star & \star \\ Q_j^{1/2} X_j & 0 & \gamma_j I & \star \\ R_j^{1/2} Y_j & 0 & 0 & \gamma_j I \end{bmatrix}, \quad (18b)$$

where $\nu = 1, \dots, n_\nu$, $j = 1, 2, \dots, n_s$. Usually, 2 pairs of weighting matrices are considered, i.e., $n_s = 2$.

The set of gain matrices of control law in (10) is designed using

$$K_j = X_j^{-1} Y_j, \quad j = 1, 2, \dots, n_s. \quad (19)$$

Finally, the procedure of convex-lifting-based robust control considering tunable approach is proposed in Algorithm 3, see Ref. [25].

Algorithm 3

Convex-lifting-based robust control using multiple tunable RPI sets.

-
- 1: **Input:** uncertain model in (2), 2 pairs of MPC-like weighting matrices $(Q_1, R_1), (Q_2, R_2)$
 - 2: **Output:** manipulated variable $u(k)$
 - 3: I. Offline phase:
 - 4: construct invariant sets \mathbb{X}_1 and \mathbb{X}_2 and associated K_1 and K_2 using (19)
 - 5: construct convex lifting above set \mathbb{X}_{lift}
 - 6: II. Online phase:
 - 7: **if** $x(k) \in \mathbb{X}_{\text{lift}}$ **then**
 - 8: solve LP in (9) to compute $u(k)$
 - 9: **else if** $x(k) \in \mathbb{X}_2$ **then**
 - 10: $u(k) = K_2 x(k)$
 - 11: **else if** $x(k) \in \mathbb{X}_1$ **then**
 - 12: $u(k) = K_1 x(k)$
 - 13: **end if**
-

4. Results and discussion

RMPC represents a standard and well-known robust control strategy, that requires quite computationally demanding mathematical operations in each sampling time. On the other hand, the convex-lifting-based robust controllers are designed to reduce the computational demand during the online phase, but still having a favourable control performance and energy efficiency. In this section, we analyse the results of the laboratory implementation of 3 proposed theoretical strategies proposed in Section 3 on the plate heat exchanger described in Section 2.

4.1. Experimental setup

Robust control design aimed to ensure the offset-free setpoint tracking of the plate heat exchanger. The setpoint temperature was $T_{\text{ref}} = 40^\circ \text{C}$, while the inlet temperature of the heated medium was approximately $T_0 = 20^\circ \text{C}$.

First, RMPC was considered as the reference control approach, see Ref. [17]. RMPC was tuned w.r.t. the tuning parameters Q, R , i.e., the weighting matrices, to optimize the control performance and energy efficiency. The sampling time of RMPC was 5 s to ensure sufficient time to solve the complex optimization online. Further implementation details are in Refs. [17].

Next, three convex-lifting-based robust control approaches were designed to ensure the offset-free setpoint tracking problem. Sampling time was set to $t_s = 2 \text{ s}$. This value was reduced compared to RMPC design, as the online computational burden was significantly reduced by the convex-lifting-based robust control methods.

To make the measured results fully comparable with the reference RMPC strategy, the control trajectories were analysed considering only initial 60 s of control, i.e., $t_c = 60 \text{ s}$. The real-time control was realized by MATLAB/Simulink R2018b, using CPU i5 2.7 GHz and 8 GB RAM. The optimization problems were parsed using YALMIP toolbox [33]. The semidefinite optimization problems were solved by the solver MOSEK v8 [34], and LPs were handled by solver Gurobi [35]. The parametric optimization and geometry computations were delegated to Multi-Parametric Toolbox [36]. Communication with the plant was ensured by Wifi-based eLab Manager toolbox [37].

Convex-lifting-based robust control was designed and

implemented w.r.t. three different scenarios. At first, we designed (i) original nontunable approach. By modifying the original approach we designed (ii) tunable approach with single pair of tuning MPC-like matrices Q, R , that were systematically tuned as follows:

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad R = [1]. \quad (20)$$

Finally, we implemented multiple tunable approach with 2 pairs of tuning matrices. In this case, we selected two pairs of tuning matrices Q_1, R_1 and Q_2, R_2 as follows:

$$Q_1 = \begin{bmatrix} 5 & 0 \\ 0 & 0.03 \end{bmatrix}, \quad R_1 = [1], \quad (21a)$$

$$Q_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad R_2 = [1]. \quad (21b)$$

For the controller design, input, state, and output variables of system in (2) were defined in the form of the following deviation variables:

$$u(k) = q(k) - q_s, \quad (22a)$$

$$x_1(k) = T(k) - T_s, \quad (22b)$$

$$y(k) = T(k) - T_s, \quad (22c)$$

where T is the controlled temperature of the heated fluid, T_s is the associated working point temperature, q is the volumetric flow rate of the hot medium and q_s is the working point value of the flow rate. The state x_2 is an incremental value of the state x_1 , therefore, it does not have any working point value.

The manipulated variable, the volumetric flow rate of the hot medium q , was restricted within an operating range $[0 - 11.5]$ mL s^{-1} . This range was normalized so that the working point corresponded to $q_s = 5.75$ mL s^{-1} . This value was also the mean value of the operating range. Therefore, the considered constraints of the manipulated variable q were:

$$-5.75 \text{ mL s}^{-1} \leq u(k) \leq 5.75 \text{ mL s}^{-1}. \quad (23)$$

The negative value of the manipulated variable $u(k)$ follows from its definition in (22a). The minimum value of $u(k) = -5.75$ mL s^{-1} corresponds to the flow rate $q(k) = 0$ mL s^{-1} .

The controlled variable, i.e., the temperature of the heated fluid T , was operated within $[15 - 55]$ °C. The limit values of this range correspond to the minimal and maximal values of the manipulated variable bounded as shown in (23). The controlled variable was normalized so that the mean temperature $T_s = 35$ °C corresponds to the operating point. The normalized constraints of the controlled variable were:

$$-20 \text{ °C} \leq y(k) \leq 20 \text{ °C}. \quad (24)$$

The negative value of the controlled variable $y(k)$ follows from its definition in (22c). The minimum value $y(k) = -20$ °C corresponds to the temperature $T(k) = 15$ °C.

Finally, the elements of the augmented vector in (4) were constrained by

$$\begin{bmatrix} -20 \\ -200 \end{bmatrix} \leq \hat{x}(k) \leq \begin{bmatrix} 20 \\ 200 \end{bmatrix}. \quad (25)$$

4.2. Offline phase of convex-lifting-based robust control

This part briefly summarizes the results of the offline phase of the convex-lifting-based robust controller design. The main part of the offline phase of the proposed approaches was to construct the convex lifting set \mathbb{X}_{lift} and a single or the multiple RPI sets \mathbb{X}_{rpi} . The technical details of the construction are described in Section 3. The controllers were designed w.r.t. three different strategies:

- i non-tunable convex-lifting-based robust control: convex lifting and RPI set are depicted in Fig. 2,
- ii tunable convex-lifting-based robust control with single pair of weighting matrices: convex lifting and RPI set are plotted in Fig. 3,
- iii multiple tunable convex-lifting-based robust control with multiple pairs of weighting matrices: convex lifting and two RPI sets are shown in Fig. 4.

4.3. Online phase of convex-lifting-based robust control

The aim of the online phase, i.e., the real-time control, was to

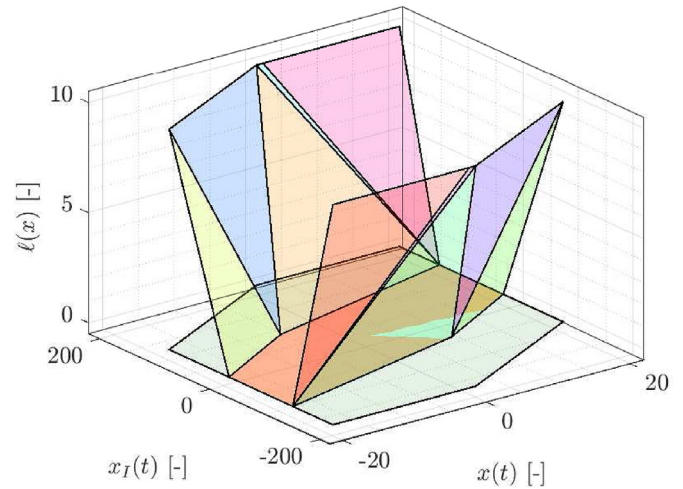


Fig. 2. Parametric solution of convex-lifting-based robust control method for the model of heat exchanger: (i) original non-tunable convex-lifting-based control.

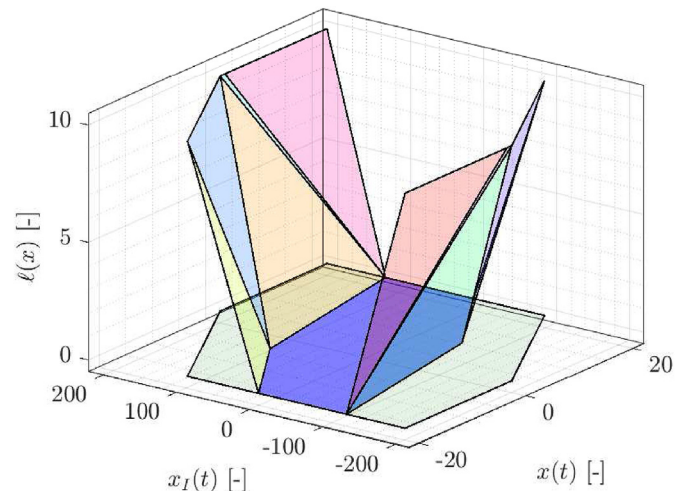


Fig. 3. Parametric solution of convex-lifting-based robust control method for the model of heat exchanger: (ii) tunable convex-lifting-based control with single RPI set.

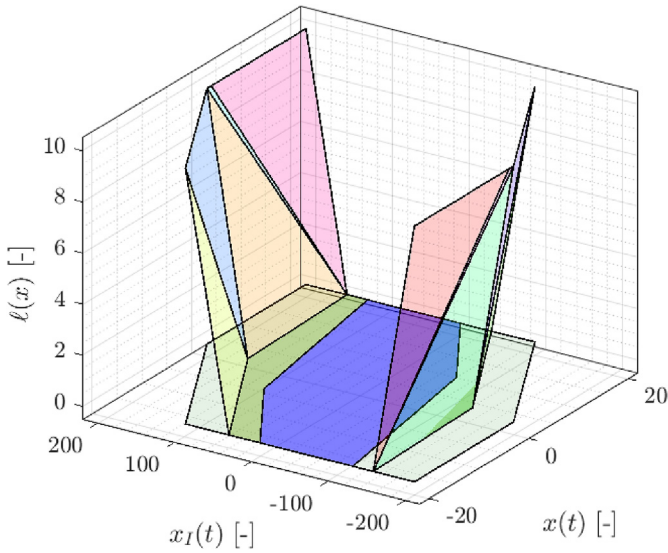


Fig. 4. Parametric solution of convex-lifting-based robust control method for the model of heat exchanger: (iii) multiple tunable convex-lifting-based robust control with 2 pairs of RPI sets.

compute the optimal value of the manipulated variable $u(k)$ at each instance k . The non-tunable robust control strategy could not explicitly take into account the control performance criteria and the energy efficiency during the controller design. On the other hand, the tunable and multiple tunable approaches were experimentally tuned to obtain a satisfactory control performance w.r.t. the minimization of their energy consumption. These control methods were tuned by a single or two pairs of weighting matrices in (20), (21), respectively.

The overall comparison of the control performance of all considered robust control strategies is depicted in Fig. 5. Particularly, the trajectories of the controlled variable, i.e., the temperature of the heated medium, are depicted in Fig. 5 (a), while the associated trajectories of the manipulated variables, i.e., volumetric flow-rate of the hot medium, are shown in Fig. 5 (b).

It can be seen in Fig. 5 (a) that all control approaches ensured offset-free setpoint tracking within 2 min. Simultaneously, the values of the manipulated variables evaluated by all considered control strategies respected the given constraints (Fig. 5 (b), dashed black). As can be seen, the designed convex-lifting-based robust controllers (Fig. 5 (a), blue, yellow, purple) significantly outperformed the RMPC approach (Fig. 5 (b), green). Differences between control trajectories in the analysed approaches were caused by multiple factors. First, RMPC method was focused on the optimization of the performance of the nominal, i.e., average model. The uncertain system vertices were considered just to ensure robust stability guarantee. On the other hand, the convex-lifting-based approaches considered all vertex models in control performance optimization. Next, the structures of the optimization problems themselves were different. Particularly, the RMPC is based on the repetitive solution of the semidefinite programming, while the convex-lifting-based robust control methods solve the LPs or parametric LPs. However, all the approaches are still comparable as they share the same control conditions determined by the uncertain system and the constraints on the manipulated variable and controlled variable. Details of RMPC design applied on the heat exchanger are summarized in Ref. [17].

Fig. 6 provides the detail insight into the initial 60 s to compare the control performance of particular convex-lifting-based robust control approaches. Control trajectory of presented approaches is

depicted in Fig. 6 (a) Corresponding trajectories of the manipulated variables are depicted in Fig. 6 (b). As can be seen, the measured control trajectories differ in the values of settling time and overshoot.

4.4. Analysis of the control performance

The quality of control was judged w.r.t. the control performance of the heat exchanger outlet temperature T , i.e., the system output y . Simultaneously, the quality was analysed subject to the overall energy consumption, i.e., the system input u . The evaluated control performance criteria are summarized in Table 1, where three main control performance quality criteria were analysed, i.e., the settling time t_{set} , overshoot σ_{max} , and the energy consumption E .

The settling time t_{set} was determined as the time, when the controlled variable settled within 2 %-neighborhood of the setpoint value T_{ref} .

The overshoot σ_{max} was computed by:

$$\sigma_{max} = \frac{T_{max} - T_{ref}}{T_{ref} - T_{init}}, \quad (26)$$

where T_{max} is the maximum value of the controlled variable, $T_{ref} = 40^\circ\text{C}$ is the setpoint, $T_{init} = 30^\circ\text{C}$ is the initial value of the controlled variable.

Energy efficiency E was evaluated as the total energy consumption necessary to heat the heating medium and was calculated as follows:

$$E = V_{total} \rho c_p \Delta T = \int_0^{120} q(t) \rho c_p \Delta T dt \approx \sum_{k=0}^{60} q(k) \rho c_p \Delta T t_s, \quad (27)$$

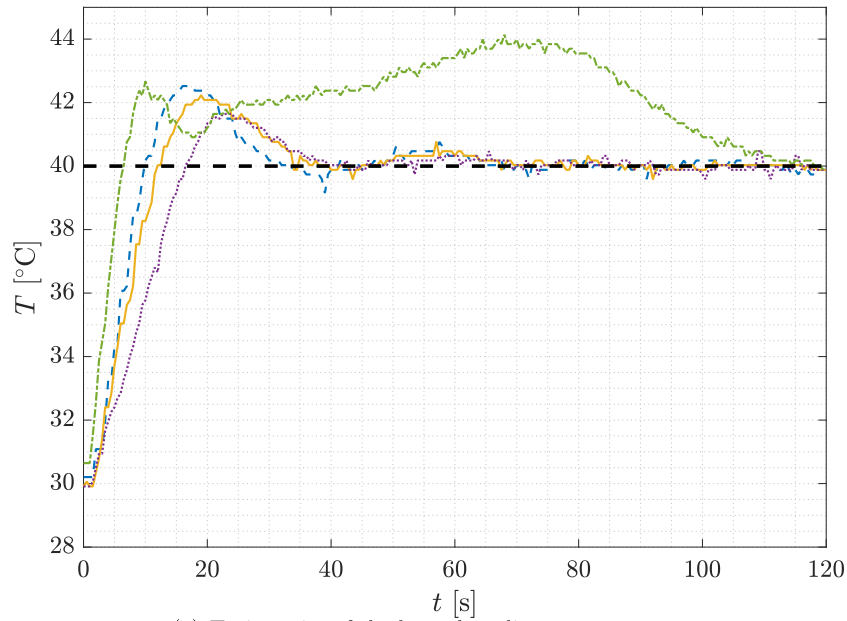
where V_{total} was the total consumption of heating medium, $\rho (35^\circ\text{C}) = 993.94 \text{ kg m}^{-3}$ was the density, $c_p (35^\circ\text{C}) = 4.18 \text{ kJ kg}^{-1} \text{ K}^{-1}$ was the specific heat capacity of hot medium, i.e., water. The temperature difference was $\Delta T = 50^\circ\text{C}$ as the hot medium was heated up from the initial laboratory temperature $T_{lab} = 20^\circ\text{C}$ to the temperature of hot medium $T_{hot} = 70^\circ\text{C}$.

Control trajectory of RMPC had the largest overshoot σ_{max} and settling time t_{set} . Moreover, the energy consumption E had the largest value. Implementation of the convex-lifting-based robust controllers significantly improved all of the evaluated quality criteria, see Table 1. By implementing original non-tunable strategy, overshoot σ_{max} decreased by 40 %, settling time t_{set} decreased by 62 %, and the energy consumption E was reduced by 82 %. Implementing of tunable and multiple tunable strategies, the overshoot and settling time decreased, when compared to the non-tunable approach. Particularly, in comparison to the non-tunable strategy, the multiple tunable approach reduced the overshoot by 36 %.

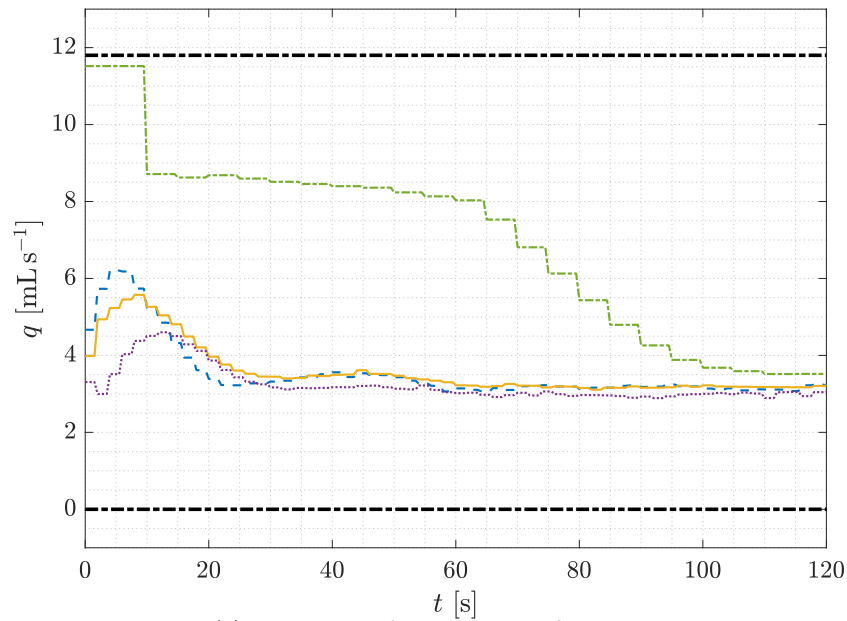
Settling time in multiple tunable strategy was the same as the settling time ensured by the tunable strategy. When compared to the non-tunable strategy, the settling time was reduced by 21 %. Compared to RMPC, the tunable and the multiple tunable strategies reduced the settling time by 70 %.

Comparing non-tunable and tunable approaches, the energy consumption E increased, but the other evaluated criteria were improved. On the other hand, comparing non-tunable and multiple tunable approaches, the energy consumption E decreased by 13 %. Therefore, the multiple tunable strategy was evaluated as the most suitable w.r.t. the judged control performance criteria, see Table 1.

The implementation point of view is important when



(a) Trajectories of the heated medium temperature.



(b) Trajectories of the volumetric flow rate.

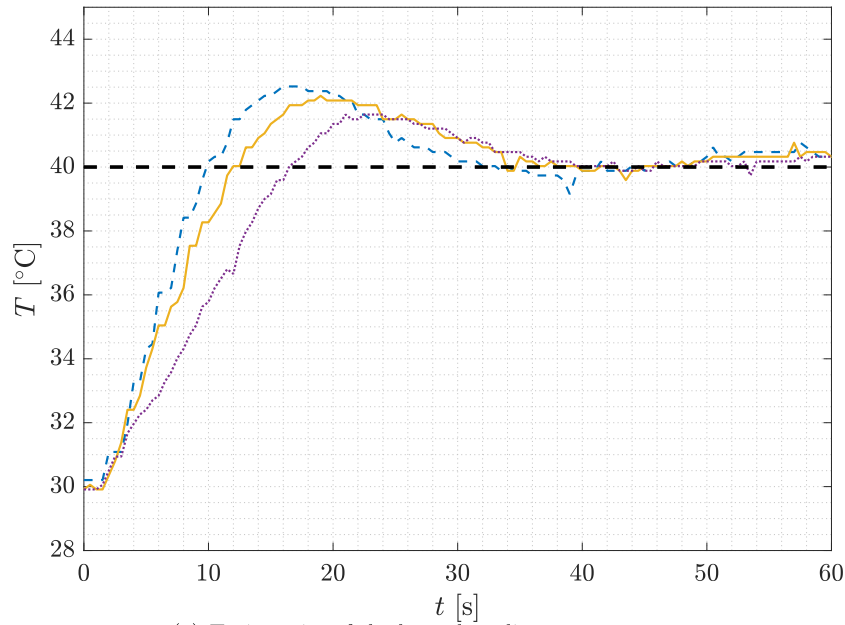
Fig. 5. Control performance: RMPC (dash-dotted green), (i) nontunable approach (dashed blue), (ii) tunable approach with single pair of weighting matrices (solid yellow), (iii) tunable approach with multiple pairs of weighting matrices (dotted purple), setpoint (dashed black), and manipulated variable constraints (dash-dotted black).

considering the industrial implementation of the designed robust control strategies. This paper analysed also the computational demands of each control procedure.

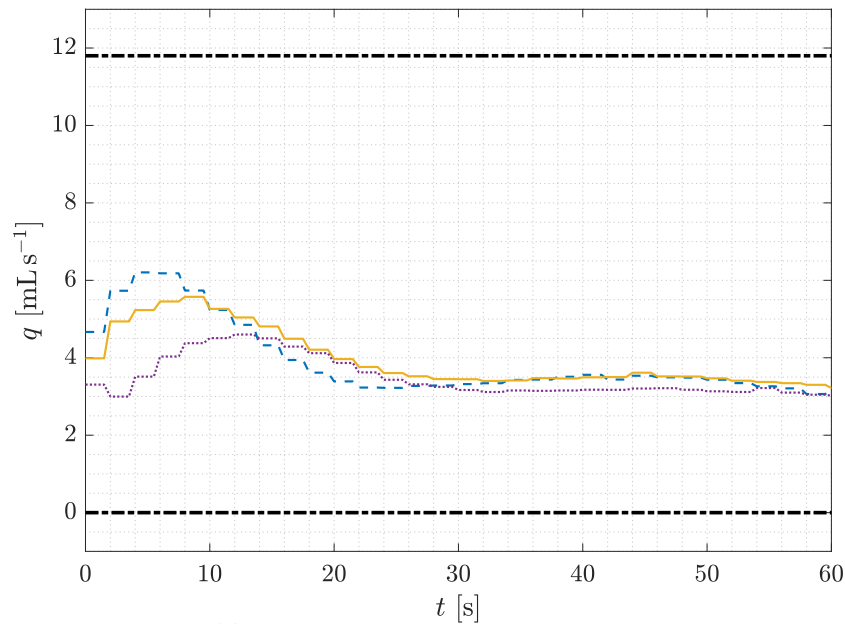
The main limitation of the real-time implementation of RMPC is the computational burden it puts upon the hardware to compute the optimal value of the manipulated variable.

In contrast to RMPC, the convex-lifting-based methods have a preparation offline phase to construct a robust controller. The majority of the optimization problems regarding the controller design were solved during the offline phase. The overall computation time t_{sol} of the offline phase was approximately 3 s. As a consequence, the offline phase significantly decreased the computational demand of computing the optimal value of the manipulated variable during the online phase, i.e., the real-time control.

Table 2 compares the implementation properties of the considered robust control strategies, where t_{online} is the time necessary to compute the optimal value of the manipulated variable during the online phase, i.e., real-time control. As can be seen, convex-lifting-based robust control strategies significantly outperformed the RMPC as the computational time was reduced by 99.4 %. The reason is, that RMPC solved in each control step complex SDP optimization problem. On the other hand, convex-lifting-based robust control strategies solved either LP in (9) or optimization-free state feedback linear control law in (10). The linear control law is evaluated when system states lie inside the RPI set. Therefore, the volume of RPI set V_{rpi} was maximized to increase the probability to evaluate the optimization-free linear control law, see Table 2.



(a) Trajectories of the heated medium temperature.



(b) Trajectories of the volumetric flow rate.

Fig. 6. Control performance: (i) non-tunable approach (dash blue), (ii) tunable approach with single pair of weighting matrices (solid yellow), (iii) tunable approach with multiple pairs of weighting matrices (dotted purple), setpoint (dashed black), and manipulated variable constrains (dash-dotted black).

Table 1
Comparison of the control performance and energy efficiency of the strategies.

Strategy	t_{set} [s]	σ_{max} [%]	E [kJ]
RMPC	102	10.5	780.0
non-tunable	39	6.3	140.5
tunable	31	5.5	141.5
multiple tunable	31	4.0	123.0

Table 2
Comparison of the designed controller properties.

Criteria	t_{online} [ms]	V_{rpi} [$\times 10^3$]
RMPC	160	†
non-tunable	1	5.7
tunable	1	9.5
multiple tunable	1	9.5

Implementation of the tunable strategy decreased the computational demand during the online phase when compared to the non-tunable strategy. The reason is increased volume of the RPI set V_{rpi} , see [Table 2](#).

5. Conclusions

In this paper, we analysed control performance and energy efficiency of four robust control strategies implemented on the

laboratory plate heat exchanger. First, we considered the conventional RMPC as the reference approach. Next, three convex-lifting-based robust control methods were designed and implemented. Compared to RMPC, the non-tunable approach improved control performance in terms of the minimization of the selected quality criteria. Moreover, it increased energy efficiency in 82% w.r.t. RMPC method. The tunable approach with a single RPI set ensured further significant improvement of the control performance. On the other hand, the energy efficacy decreased compared to the non-tunable approach. Multiple tunable approach with 2 RPI sets was designed to preserve the control performance of non-tunable approach and, simultaneously, to ensure further energy savings by 13% w.r.t. the non-tunable approach. Based on the analysis of the measured data, the multiple tunable convex-lifting-based robust controller with two RPI sets was evaluated as the most energy-efficient control strategy for the considered plate heat exchanger. Further research is focused on the investigation of the energy efficacy considering approximated control law.

Acknowledgement

The authors gratefully acknowledge the contribution of the Scientific Grant Agency of the Slovak Republic under the grants 1/0545/20, the Slovak Research and Development Agency under the project APVV-15-0007, and the Research & Development Operational Programme for the project University Scientific Park STU in Bratislava, ITMS 26240220084, supported by the Research 7 Development Operational Programme funded by the ERDF.

References

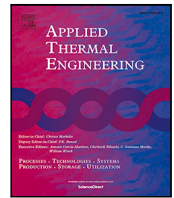
- Chen Q, Wang M, Panc N, Guo Z. Optimization principles for convective heat transfer. *Energy* 2009;34:1199–206. <https://doi.org/10.1016/j.energy.2009.04.034>.
- Zhi-Yong L, Varbanov PS, Klemes JJ, Yong JY. Recent developments in applied thermal engineering: process integration, heat exchangers, enhanced heat transfer, solar thermal energy, combustion and high temperature processes and thermal process modelling. *Appl Therm Eng* 2016;105:755–62. <https://doi.org/10.1016/j.applthermaleng.2016.06.183>.
- Bauer M, Craig I. Economic assessment of advanced process control – a survey and framework. *J Process Contr* 2008;18:2–18. <https://doi.org/10.1016/j.jprocont.2007.05.007>.
- Li C, Liu L, Zhang L, Gu S, Du J. Simultaneous synthesis of controllable heat exchanger networks. *Chemical Engineering Transactions* 2019;79:739–44. <https://doi.org/10.3303/CET1976124>.
- Mayne D. Model predictive control: recent developments and future promise. *Automatica* 2014;12:2967–86. <https://doi.org/10.1016/j.automatica.2014.10.128>.
- Gonzalez AH, Odloak D, Marchetti JL, Sotomayor OAZ. Infinite horizon mpc of a heat-exchanger network. *Chem Eng Res Des* 2006;84:1041–50. <https://doi.org/10.1016/j.energy.2009.04.034>.
- Wu X, Chen J, Xie L. Fast economic nonlinear model predictive control strategy of organic rankine cycle for waste heat recovery: simulation-based studies. *Energy* 2019;180:520–34. <https://doi.org/10.1016/j.energy.2019.05.023>.
- Shan K, Fan C, Wang J. Model predictive control for thermal energy storage assisted large central cooling systems. *Energy* 2019;179:916–27. <https://doi.org/10.1016/j.energy.2019.04.178>.
- Vasičkaninová A, Bakošová M, Mészáros A, Klemes J. Neural network predictive control of a heat exchanger. *Chemical Engineering Transactions* 2010;21:73–8.
- Fratczak M, Czeczot J, Nowak P, Metzger M. Practical validation of the effective control of liquid–liquid heat exchangers by distributed parameter balance-based adaptive controller. *Appl Therm Eng* 2018;129:549–56. <https://doi.org/10.1016/j.applthermaleng.2017.10.056>.
- Markowski M, Trzcinski P. On-line control of the heat exchanger network under fouling constraints. *Energy* 2019;185:521–6. <https://doi.org/10.1016/j.energy.2019.07.022>.
- Wu X, Ze H, Xiaobin J, Xiangcun L, Xuemei W, Gaohong H. Heat exchanger reliability analysis based on fouling growth model by fault tree analysis. *Chemical Engineering Transactions* 2019;79:265–70. <https://doi.org/10.3303/CET1976045>.
- Trafczynski M, Markowski M, Urbaniec K. Energy saving potential of a simple control strategy for heat exchanger network operation under fouling conditions. *Renew Sustain Energy Rev* 2019;111:355–64. <https://doi.org/10.1016/j.rser.2019.05.046>.
- Bemporad A, Morari M. Robust model predictive control: a survey. In: *Robustness in identification and control*. Springer London; 1999. p. 207–26.
- Oravec J, Bakošová M, Trafczynski M, Vasičkaninová A, Mészáros A, Markowski M. Robust model predictive control and PID control of shell-and-tube heat exchangers. *Energy* 2018;159:1–10. <https://doi.org/10.1016/j.energy.2018.06.106>.
- Agner R, Lucas EJ, Olsen DG, Gruber P, Wellig B. Robust control of heat exchangers in stratified storage. *Chemical Engineering Transactions* 2019;79:781–6. <https://doi.org/10.3303/CET1976131>.
- Oravec J, Bakošová M, Galčíková L, Slávik M, Horváthová M, Mészáros A. Soft-constrained robust model predictive control of a plate heat exchanger: experimental analysis. *Energy* 2019;180:303–14. <https://doi.org/10.1016/j.energy.2019.05.093>.
- Kothare MV, Balakrishnan V, Morari M. Robust constrained model predictive control using linear matrix inequalities. *Automatica* 1996;32:1361–79.
- Gulan M, Takács G, Nguyen NA, Olaru S, Rodríguez-Ayerbe P, Rohal-Ilkiv B. Embedded linear model predictive control for 8-bit microcontrollers via convex lifting. *IFAC-PapersOnLine* 2017;50:10697–704. <https://doi.org/10.1016/j.ifacol.2017.08.2220>.
- Nguyen NA, Gulan M, Olaru S, Rodríguez-Ayerbe P. Convex lifting: theory and control applications. *IEEE Trans Automat Contr* 2018;63:1243–58. <https://doi.org/10.1109/TAC.2017.2737234>.
- Nguyen NA, Olaru S, Rodríguez-Ayerbe P. Inverse parametric linear/quadratic programming problem for continuous pwa functions defined on polyhedral partitions of polyhedra. In: 54th IEEE conference on decision and control (CDC). Osaka, Japan: IEEE; 2015. <https://doi.org/10.1109/CDC.2015.7403150>.
- Nguyen NA, Olaru S, Rodríguez-Ayerbe P, Hovd M, Necoara I. Inverse parametric convex programming problems via convex liftings. *IEEE Trans Automat Contr* 2014;47:2489–94. <https://doi.org/10.3182/20140824-6-ZA-1003.02364>.
- Nguyen NA. Stochastic output feedback control: convex lifting approach. *Automatica* 2017;89:212–20. <https://doi.org/10.1016/j.automatica.2017.12.017>.
- Nguyen NA, Olaru S, Rodríguez-Ayerbe P, Kvasnica M. Convex liftings-based robust control design. *Automatica* 2017;77:206–13. <https://doi.org/10.1016/j.automatica.2016.11.031>.
- Oravec J, Holaza J, Horváthová M, Nguyen NA, Kvasnica M, Bakošová M. Convex-lifting-based robust control design using the tunable robust invariant sets. *Eur J Contr* 2019;49:44–52. <https://doi.org/10.1016/j.ejcon.2019.01.002>.
- Oravec J, Bakošová M, Horváthová M, Galčíková L, Slávik M, Vasičkaninová A, Mészáros A. Convex-lifting-based robust control of a laboratory plate heat exchanger. *Chemical Engineering Transactions* 2019;76:733–8. <https://doi.org/10.3303/CET1976123>.
- Armfield. Extracts from instruction manual PCT23.
- Mikláš J, Fikar M. *Process modelling, identification, and control*. Berlin Heidelberg: Springer Verlag; 2007.
- Oravec J, Bakošová M, Pakšiová D, Mikušová N, Batárová K. Advanced robust MPC design of a heat exchanger: modeling and experiments. In: 27th European symposium on computer aided process engineering. Barcelona, Spain: Elsevier; 2017. p. 1585–90. <https://doi.org/10.1016/B978-0-444-63965-3.50266-X>.
- Borrelli F, Bemporad A, Morari M. Predictive control for linear and hybrid systems. Cambridge University Press; 2017. <https://doi.org/10.1017/9781139061759>.
- Vandenberghe L, Boyd S. Semidefinite programming. *SIAM Rev* 1996;38:49–95.
- Boyd S, Ghaoui LE, Feron E, Balakrishnan V. *Linear matrix inequalities in system and control theory*. Philadelphia, USA: Society for Industrial and Applied Mathematics; 1994.
- Löfberg J. Yalmip : a toolbox for modeling and optimization in MATLAB. In: *Proceedings of the CACSD conference*, Taipei, Taiwan; 2004.
- Mosek ApS. Mosek. Mosek ApS; 2019. <https://mosek.com/>.
- Gurobi Optimization. Gurobi optimizer reference manual. Gurobi Optimization; 2019. <http://www.gurobi.com>.
- Herceg M, Kvasnica M, Jones C, Morari M. Multi-parametric toolbox 3.0. In: *Proc. Of the European control conference*, Zürich, Switzerland; 2013. p. 502–10. <http://control.ee.ethz.ch/~mpt>.
- Kalúz M, Círka L, Valo R, Fikar M. Lab of things: a network-based i/o services for laboratory experimentation. In: *Preprints of the 20th IFAC world congress*, Toulouse, France, vol. 20; 2017. p. 14028–33.

Nomenclature

Symbols

- a_i : slope of the i -th polytope
 A : system matrix of the state space system
 $A^{(v)}$: vertex system matrix of the state space system
 $\hat{A}^{(v)}$: vertex system matrix of the augmented state space system
 \mathbb{A} : set of uncertain state space system
 b_i : intercept of the i -th polytope

- B : input matrix of the state space system
 $B^{(v)}$: vertex input matrix of the state space system
 $\hat{B}^{(v)}$: vertex input matrix of the augmented state space system
 c_p : specific heat capacity, $\text{kJ kg}^{-1} \text{K}^{-1}$
 C : output matrix of the state space system
 $C^{(v)}$: vertex output matrix of the state space system
 $\hat{C}^{(v)}$: vertex output matrix of the augmented state space system
 e : control error, $^{\circ}\text{C}$
 E : energy consumption, kJ
 I : identity matrix
 k : sample of discrete time domain, s
 K : gain matrix of the controller
 K_0 : gain matrix associated to non-tunable RPI set
 K_j : gain matrix associated to j -th tunable RPI set
 l : lifting value
 n_p : total number of polytopes
 n_s : total number of RPI sets
 n_{iu} : total number of system inputs
 n_v : total number of the uncertain system vertices
 n_x : total number of system states
 n_y : total number of system outputs
 q : volumetric flowrate of hot medium, i.e., manipulated variable, mL s^{-1}
 q_s : steady-state working point of manipulated variable, mL s^{-1}
 Q : weighting matrix of system states
 Q_j : weighting matrix of system states associated to j -th tunable RPI set
 R : weighting matrix of system inputs
 R_j : weighting matrix of system inputs associated to j -th tunable RPI set
 r : setpoint value, $^{\circ}\text{C}$
 \mathbb{R} : Euclidean space of the real numbers
 t : time, s
 t_c : control time, s
 t_{online} : time necessary to compute optimal value of manipulated variable during online phase, s
 t_s : sampling time, s
 t_{set} : settling time, s
 T : output temperature of the heated medium, i.e., controlled variable, $^{\circ}\text{C}$
 T_0 : inlet temperature of the heated medium, $^{\circ}\text{C}$
 T_{hot} : temperature of the heating medium, $^{\circ}\text{C}$
 T_{init} : initial value of controlled variable, $^{\circ}\text{C}$
 T_{max} : maximal value of controlled variable, $^{\circ}\text{C}$
 T_{lab} : laboratory temperature, $^{\circ}\text{C}$
 T_{ref} : setpoint temperature, $^{\circ}\text{C}$
 T_s : steady-state working point of controlled variable, $^{\circ}\text{C}$
 ΔT : temperature difference, $^{\circ}\text{C}$
 u : control input, i.e., manipulated variable, mL s^{-1}
 u_{max} : maximal value of the manipulated variable, mL s^{-1}
 u_{min} : minimal value of the manipulated variable, mL s^{-1}
 \mathbb{U} : set of constrains on the manipulated variable
 v : vertex of the polytopic uncertain system
 V : Lyapunov function
 V_{rpi} : volume of robust positive invariant set
 V_{total} : total consumption of heating medium, mL
 $\mathcal{V}(\mathbb{X}_{rpi})$: set of vertices of RPI set
 w : measurement noise, i.e., additive disturbance
 \mathbb{W} : set of constrains on the measurement noise
 x : system states
 x_0 : initial conditions of system states
 \tilde{x} : augmented vector of the system states
 X_0 : weighted inverted Lyapunov matrix associated to non-tunable RPI set
 X_j : weighted inverted Lyapunov matrix associated to j -th tunable RPI set
 \mathbb{X}_0 : robust positive invariant set associated to non-tunable RPI set
 \mathbb{X}_j : robust positive invariant set associated to j -th tunable RPI set
 \mathbb{X}_{lift} : convex lifting set
 \mathbb{X}_{rpi} : robust positive invariant set
 y : system output, $^{\circ}\text{C}$
 y_{max} : maximal value of the system output, $^{\circ}\text{C}$
 y_{min} : minimal value of the system output, $^{\circ}\text{C}$
 Y : auxiliary matrix of control law parametrization
 Y_j : auxiliary matrix of control law parametrization associated to j -th tunable RPI set
 z : decision variable of parametric linear programming
 Z : gain
 Z_{max} : maximum gain
 Z_{min} : minimum gain
- Greek letters**
- γ : auxiliary optimized parameter
 γ^* : optimal value of auxiliary optimized parameter
 γ_j : auxiliary optimized parameter associated to j -th tunable RPI set
 λ : parameter of the convex combination of uncertain system vertices
 Π : polytopic set
 ρ : density, kg m^{-3}
 σ_{max} : overshoot, %
 τ : time constant, s
 τ_{max} : maximum time constant, s
 τ_{min} : minimum time constant, s
- Abbreviations**
- CPU: central processing unit
 CV: controlled variable
 LMI: linear matrix inequality
 LP: linear programming
 MPC: model predictive control
 MV: manipulated variable
 PID: proportional–integral–derivative
 PLC: programmable logic controller
 RAM: random access memory
 RMPC: robust model predictive control
 RPI: robust positive invariant



Research Paper

Heat exchanger control using model predictive control with constraint removal[☆]Raphael Dyrška^{a,1}, Michaela Horváthová^{b,*}, Peter Bakaráč^b, Martin Mönnigmann^a, Juraj Oravec^b^a Automatic Control and Systems Theory, Ruhr-Universität Bochum, Germany^b Institute of Information Engineering, Automation, and Mathematics, Faculty of Chemical and Food Technology, Slovak University of Technology in Bratislava, Slovakia

ARTICLE INFO

Keywords:

Heat exchanger
Model predictive control
Constraint removal
Energy consumption
Microcontroller

ABSTRACT

Climate change enforces the implementation of sustainable industrial production with a special focus on pollution reduction, resource management, and energy savings. These goals are addressed by designing advanced control methods using the solution of an adequately formulated optimization problem. Heat exchangers represent particularly energy-demanding plants that are challenging from the advanced controller design point of view. Model predictive control (MPC) is a suitable control strategy to address the relevant control tasks. The complexity of the real-time implementation of MPC directly depends on the number of inequality constraints in the corresponding optimization problem. Therefore, the real-time computational effort can be reduced by removing inactive constraints. Since removing inactive constraints does not change the optimal solution, it is desirable to detect inactive constraints corresponding to the current system state measurement and remove them from the formulation of the MPC problem before running the optimization solver. However, external disturbances, parametric uncertainties, and setpoint changes often impact real plants, limiting the application range of the conventional constraint removal MPC approach. In this paper, we propose a modification of the conventional constraint removal approach to address this issue. The modified constraint removal approach achieves the robustness required for a practical application to a laboratory-scaled heat exchanger. The control performance of the heat exchanger is analyzed from the industrial perspective considering the computational time and energy consumption by implementing the control approach on a 32-bit microcontroller.

1. Introduction

Efficient energy supply plays a crucial role in achieving important goals, such as the health of economies in the presence of sustainable industry. Nowadays, approximately 80% of energy utilization involves some form of heat transfer [1]. Heat exchangers are present in most industrial operations and are thus involved in the energy-intensive part of the operation. As the consequence, the operation of heat exchangers has a strong impact on the economic efficiency of their operation [2]. Therefore, the importance of heat transfer technologies, modeling, and

integration are significant. An overview of current advancements in applied thermal engineering is presented, e.g., in [3].

One of the areas with a promising potential for improvement is the design and application of optimal control algorithms for heat exchangers. Thus, the improved operation of heat exchangers can be directly associated with the implementation of advanced control strategies.

In recent years, advanced optimization-based control methods like model predictive control (MPC) gained popularity [4]. MPC is a control method based on periodically solving a constrained optimal control problem (OCP) in every time step to evaluate the optimal input signal.

[☆] Support by the Alexander von Humboldt Foundation research group linkage cooperation program is gratefully acknowledged. This research is funded by the European Commission under the grant no. 101079342 (Fostering Opportunities Towards Slovak Excellence in Advanced Control for Smart Industries). MH, PB, JO gratefully acknowledge the contribution of the Slovak Research and Development Agency under the project APVV-20-0261, and the Scientific Grant Agency of the Slovak Republic under the grants 1/0297/22, 1/0545/20. RD, MM gratefully acknowledge support by the German Federal Ministry for Economic Affairs and Energy under grant 0324125C and by the Deutsche Forschungsgemeinschaft (DFG), Germany under grant MO 1086/15-1.

* Corresponding author.

E-mail address: michaela.horvathova@stuba.sk (M. Horváthová).

¹ These authors contributed equally to this work.

From the control perspective, the implementation of MPC outperforms any proportional–integral–derivative (PID) controller by its nature, as MPC evaluates the control actions by solving the optimization problem taking into account a wide class of technological constraints and quality criteria. In [5], several case studies comparing MPC with PID control point out the superiority of the optimization-based MPC. A possibility of how to avoid online optimizations while preserving competing results such as an optimal operation was investigated in [6], underlining the wish to profit from the benefits coming with optimization-based control approaches. In, e.g., [7], the aforementioned advantages of MPC, together with the possibility to naturally control systems with multiple inputs and outputs, have been shown to enhance both, the control performance and its robustness compared to the existing PID controller of a heat exchanger processed in a South-East Asian facility. Besides this, further work has dealt with the control of heat exchangers using MPC. In [8], MPC was proposed for the intermittent operation of a solar-assisted ground source heat pump system. Predictive control methods were also considered to control a network of heat exchangers in [9]. In [10], a so-called neural network predictive controller combined with an auxiliary fuzzy controller was successfully applied to a heat exchanger to reduce energy consumption. Compared to conventional APC controller design, the proposed method introduces two main benefits: first, we preserve the optimal evaluation of the control actions by a simultaneous reduction of the computational burden. Secondly, as a consequence, the decreased computational effort leads to reduced energy requirements on the controller-side. Admittedly, MPC comes at a higher implementation cost than APC, but it is evident that more sophisticated control methods, e.g. MPC, may achieve better control performance. Therefore the MPC based methods also affect the energy aspect of the controlled device itself not only the implementation cost.

In MPC, the complexity of the OCP depends on the complexity of the controlled plant, the associated prediction model, and the number of considered physical constraints. The solution of the underlying optimization problem can be computationally intensive. Therefore, extensive research has been devoted to reducing the complexity of the corresponding OCP and to speeding up the solution process [11].

Some well-known methods, e.g., move blocking, decrease the computational effort by reducing the degrees of freedom, see, e.g., [12] for a review of move blocking methods. However, the resulting control action is no longer optimal. From an energy-saving point of view, MPC design methods that preserve the optimality and, simultaneously, minimize the computational burden by exploiting the structure of the OCP (see, e.g., [13,14]) are more relevant. Explicit MPC (see [15,16]) avoids real-time optimization by precomputing the explicit solution map considering the whole set of admissible initial conditions. However, even for linear MPC, the multi-parametric optimization problem of the explicit MPC may be intractably complex, or the constructed explicit solution exceeds the memory limits of industrial hardware for immediate online use. Several methods have been proposed for complexity reduction, e.g., exploiting the geometry of explicit MPC solutions [17], using bilevel optimization [18], or introducing the reachability analysis [19].

Yet another class of approaches detects and removes inactive constraints before delegating the optimization problem to the solver, see [20–23]. These methods aim at preserving the optimal solution while reducing the computational complexity.

The MPC design method proposed in this paper originates from [23], and its nonlinear MPC variant is presented in [24]. The main idea is to evaluate a characteristic number associated with a cost function value that is assigned to each constraint of the OCP. The characteristic values (the σ -bounds introduced in Section 3) are calculated offline, i.e., before the runtime of the controller. Online, i.e., during runtime, the assigned values serve as an indicator to determine if the corresponding constraint is inactive in the current and all subsequent control steps.

The offline calculation of the characteristic values results in additional optimization problems but does not affect the real-time feasibility. Technically, the characteristic value for every constraint represents a bound corresponding to the minimum value that the cost function attains if the corresponding constraint is active. From a theoretical point of view, the cost function is guaranteed to be non-increasing along the control steps, which originates in the asymptotic stability condition of the designed MPC (see, e.g., [25]). As a consequence, once the cost function value decreases below the precomputed characteristic bound, the corresponding inactive constraint is removed for all subsequent time steps. The method does not only remove redundant constraints but progressively reduces the number of constraints during the runtime of the controller. However, the described technique needs to be adopted if an increase in the optimal cost function value can no longer be avoided. This may be the case, for example, due to a disturbance.

The main contribution of this paper is the application of MPC with constraint removal for the setpoint tracking problem of a laboratory plate heat exchanger. To the best of the authors' knowledge, this work presents the first experimental implementation of MPC with constraint removal on a real plant. When controlling a real device, disturbances and plant-model mismatch are inevitable. To eliminate steady-state errors, we will therefore introduce an integrator part to our model. This can lead, however, to an increase of the optimal cost function value over time. Thus we modify the characteristic bounds assigned to every constraint of the OCP to avoid removing constraints that could become active for a later time step due to the step change of the setpoint.

Specifically, we determine conservative values of the original σ -bounds depending on the maximum impact of the setpoint changes on the closed-loop optimal cost function value. Based on the conservative values of the original σ -bounds, inactive constraints in the OCP are detected and removed. Due to the fact that the removed constraints are not active, the control performance is not affected. To confirm this, the control performance of MPC was evaluated by comparing MPC with and without applying the constraint removal approach. To demonstrate the constraint removal approach's practical benefits, the OCP was also implemented and solved on a 32-bit microcontroller. The computational time and energy efficiency of the microcontroller were again evaluated for MPC with and without constraint removal.

In this paper, we focus on the energy efficiency of the microcontroller, therefore the energy aspect of the controlled device itself is beyond its scope. The energy consumption on the side of the controller, however, is often overlooked, but highly relevant to be considered from the authors perspective.

The paper is organized as follows. We introduce the control problem in Section 2. The constraint removal approach is described in detail in Section 3. The considered laboratory plate heat exchanger is introduced in Section 4, and Section 4.3 treats the necessary modifications on the considered constraint removal method. The extensive experimental case study of the heat exchanger control is analyzed in Section 5.1, followed by an evaluation of energy reduction considering a microcontroller in Section 5.2. Section 6 concludes the paper and gives an outlook on future work.

2. Problem statement and notation

Throughout this paper, we consider linear discrete-time systems of the form

$$x(k+1) = Ax(k) + Bu(k), \quad (1a)$$

$$y(k) = Cx(k), \quad (1b)$$

$k \geq 0$, with states $x(k) \in \mathbb{R}^n$, inputs $u(k) \in \mathbb{R}^m$, and outputs $y(k) \in \mathbb{R}^p$, and matrices $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, and $C \in \mathbb{R}^{p \times n}$. We assume (A, B) to

be stabilizable. States and inputs are subject to lower and upper bounds

$$x_{i,\min} \leq x_i(k) \leq x_{i,\max}, \quad (2a)$$

$$u_{j,\min} \leq u_j(k) \leq u_{j,\max}, \quad (2b)$$

for all k and with $i = 1, \dots, n$, $j = 1, \dots, m$. The objective of the MPC design is to regulate the system state (1a) to the origin by periodically solving the optimization problem

$$\min_{x,U} x(N)^T P x(N) + \sum_{k=0}^{N-1} x(k)^T Q x(k) + u(k)^T R u(k) \quad (3a)$$

$$\text{s.t. } x(0) = x_0, \quad (3b)$$

$$x(k+1) = Ax(k) + Bu(k), \quad k = 0, \dots, N-1, \quad (3c)$$

$$x(k) \in \mathcal{X}, \quad k = 0, \dots, N-1, \quad (3d)$$

$$u(k) \in \mathcal{U}, \quad k = 0, \dots, N-1, \quad (3e)$$

$$x(N) \in \mathcal{T}, \quad (3f)$$

on a receding prediction horizon N for the current state $x(0)$. The state and input constraints (2) are stated as compact and convex sets \mathcal{X} and \mathcal{U} , respectively. The decision variables are summarized by $X = (x(1)^T, \dots, x(N)^T)^T$ and $U = (u(0)^T, \dots, u(N-1)^T)^T$, and the weighting matrices P, Q, R have the obvious dimensions. We assume Q to be positive semi-definite and P and R to be positive definite. The terminal set $\mathcal{T} \subseteq \mathcal{X}$, which appears as a constraint on the last state along the prediction horizon N , is assumed to contain the origin in its interior.

By substituting the dynamics of system (1a) into (3), the optimal control problem (3) is rewritten as a quadratic program (QP) of the form

$$\min_U V(x(0), U) \quad (4)$$

$$\text{s.t. } GU \leq w + Ex(0),$$

where $V(x(0), U) = \frac{1}{2}x(0)^T Y x(0) + x(0)^T F U + \frac{1}{2}U^T H U$, $Y \in \mathbb{R}^{n \times n}$, $F \in \mathbb{R}^{n \times mN}$, $H \in \mathbb{R}^{mN \times mN}$, $H > 0$, and $G \in \mathbb{R}^{q \times mN}$, $w \in \mathbb{R}^q$, $E \in \mathbb{R}^{q \times n}$, with q denoting the number of constraints (see, e.g., [23] and the references therein). Since $H > 0$, the solution to (4) is unique, if it exists. After solving (4) for an optimal input trajectory U^* , we apply the first element, i.e., $u^*(0)$ to the system. In the next time step, we solve problem (4) again for the receding horizon N and the updated system state, such that a closed-loop control scheme results (see, e.g., [4,25] for a further introduction). Note that, since the system described by Eq. (1)a-b serves as the prediction model used for the MPC design and the optimization problem described by Eq. (3)a-f does not implement a robust variant of MPC, uncertainties arising in the real plant are not considered in Eq. (1)a-b explicitly.

Notation

Let \mathcal{F} refer to the set of states $x \in \mathcal{X}$ for which (4) has a solution, and let $U^*(x(0))$, for any $x(0) \in \mathcal{F}$, refer to the sequence that optimizes (4). We often write U^* , $U^*(x)$, $V(x, U^*)$, $V(x, U^*(x))$, etc., as short for $U^*(x(0))$, $V(x(0), U^*(x(0)))$, etc. Let $\mathcal{Q} = \{1, \dots, q\}$ collect all constraint indices. The constraint with index $i \in \mathcal{Q}$ is called active for $x(0) \in \mathcal{F}$ if $G_i U^* - w_i - E_i x(0) = 0$, and inactive if $G_i U^* - w_i - E_i x(0) < 0$. Let $\mathcal{A}(x(0))$ and $\mathcal{I}(x(0))$ be defined as the set of indices of all active and inactive constraints for $x(0)$, respectively. For a matrix G and an ordered set $\Sigma \subset \mathcal{Q}$, let G_Σ refer to the submatrix of G with the rows indicated by indices Σ .

3. MPC with constraint removal

In this Section, we introduce the constraint removal method presented in [23] as needed in the present paper. Later, in Section 4.3,

we show the modifications necessary to adopt this method for an application to the laboratory heat exchanger.

Inactive constraints have no influence on the optimal solution. Thus the optimal input sequence $U^*(x)$ resulting from solving (4) does not change if some or all inactive constraints are removed from the original OCP. This observation is stated concisely in the following proposition.

Proposition 1 ([23]). Let $x_0 \in \mathcal{F}$ be arbitrary and let $\tilde{\mathcal{I}} \subset \mathcal{I}$ be an arbitrary subset of the inactive constraints. Consider the reduced optimization problem

$$\min_{\hat{U}} V(x_0, \hat{U}) \quad (5)$$

$$\text{s.t. } G_{\mathcal{Q} \setminus \tilde{\mathcal{I}}} \hat{U} \leq w_{\mathcal{Q} \setminus \tilde{\mathcal{I}}} + E_{\mathcal{Q} \setminus \tilde{\mathcal{I}}} \tilde{x}_0.$$

Then (5) has a unique solution, which we denote by \hat{U}^* . This solution is equal to the solution obtained from (4), i.e., $\hat{U}^* = U^*$ and $V(x_0, \hat{U}^*) = V(x_0, U^*)$.

We use precalculated characteristic bounds on the optimal cost function to detect inactive constraints as proposed in [23]. Hereafter, we denote these bounds σ_i , $i = 1, \dots, q$, with index i corresponding to a constraint of problem (4) and thus a certain line of G, w , and E . Such a bound σ_i is equal to the minimum value the cost function attains such that constraint $i \in \mathcal{A}$, and will be used as a lower bound on the optimal cost function in (4).

Definition 1. Let $i \in \mathcal{Q}$ be arbitrary. If there exists an $x \in \mathbb{R}^n$ such that

$$\min_{x,U} V(x, U) \quad (6)$$

$$\text{s.t. } G_i U - w_i - E_i x = 0,$$

$$G_{\mathcal{Q} \setminus i} U - w_{\mathcal{Q} \setminus i} - E_{\mathcal{Q} \setminus i} x \leq 0$$

has a solution, set σ_i to the minimum that results for (6), i.e., $\sigma_i := V(x^*, U^*)$. Otherwise, let $\sigma_i = \infty$.

Note that $\sigma_i = \infty$ implies constraint i can never be active. Since we need to find the configuration of initial state and input trajectory resulting in the smallest cost function value V^* , the initial state x is an additional degree of freedom in (6), while it is a fixed parameter in (5). The properties of the bounds σ_i can be summarized as follows.

Lemma 1 ([23]). Let i be arbitrary and consider the QP (6). The following statements hold:

- (i) If QP (6) is feasible, it has a unique solution.
- (ii) If (6) is feasible, then $0 < \sigma_i < \infty$.
- (iii) If (6) is infeasible, then constraint i is always inactive in (4), or equivalently, $i \in \mathcal{I}(x)$ for all $x \in \mathcal{X}$.

Proposition 2 is based on Proposition 6 and Corollary 7 in [23] and summarizes how to use the bounds σ_i to detect and remove inactive constraints with respect to problem (4).

Proposition 2 ([23]). Let $i \in \mathcal{Q}$ and $x \in \mathcal{F}$ be arbitrary, and assume σ_i as defined in (6) has been determined. If $V(x, U^*(x)) < \sigma_i$, then constraint i is inactive at the optimal solution of (4) for $x(0) = x$. Furthermore, $V(x(k_0), U^*(x(k_0))) < \sigma_i$ implies $i \in \mathcal{I}(x(k))$ for all $k > k_0$.

The second statement of Proposition 2 holds if the closed-loop optimal cost function is a Lyapunov function and thus nonincreasing. While constraint removal was designed under these conditions in [23], we will see from the results in Section 5 that further modifications are necessary. These are summarized in Section 4.3.

Algorithm 1 summarizes MPC with constraint removal according to Proposition 2. We use the cost trajectory of a hypothetical example depicted in Fig. 1 to explain the steps of Algorithm 1. Let the optimal cost function value $V(x(k), U^*(x(k)))$ be denoted by V^* for short, and let

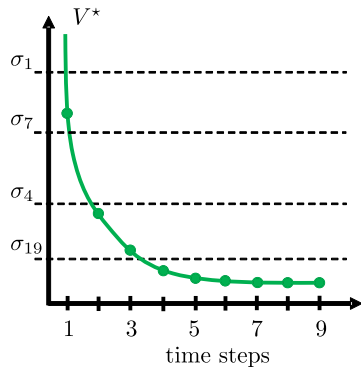


Fig. 1. Optimal cost function trajectory V^* over discrete time steps for a hypothetical example.

\mathcal{R} denote the set of indices corresponding to the constraints that have not been removed and thus can be active or inactive. Assume that, after initially solving the optimization problem in time step $k = 1$, constraint $i = 1$ of this hypothetical example can be detected to be inactive for all future time steps (steps 3 and 4 in Algorithm 1). Therefore, this constraint is removed from the QP (4) (step 7 in Algorithm 1 and topmost dashed line in Fig. 1). The same happens to constraints $i = 7$ and $i = 4$ in time step $k = 2$, and later for $k = 4$ and constraint $i = 19$. Note that the enumeration of the constraints and corresponding bounds $\sigma_i, i \in \mathcal{Q} = \{1, \dots, q\}$ results from the order of the constraints in (4). This order can be chosen arbitrarily but has to be fixed.

Algorithm 1 MPC with constraint removal (see [23]).

- 1: **Input:** $V^*, x^+, \mathcal{R} = \mathcal{Q} \setminus \tilde{\mathcal{I}}, \sigma_i, i \in \mathcal{R}$.
 - 2: **for all** $i \in \mathcal{R}$ **do**
 - 3: **if** $V^* < \sigma_i$ **then**
 - 4: i th constraint will remain inactive: $\tilde{\mathcal{I}} \leftarrow \tilde{\mathcal{I}} \cup \{i\}$.
 - 5: **end if**
 - 6: **end for**
 - 7: Remove inactive constraints: $\mathcal{R} \leftarrow \mathcal{R} \setminus \tilde{\mathcal{I}}$
 - 8: Solve reduced optimization problem in (4) for $x(0) = x^+$ and reduced set of constraints \mathcal{R} .
 - 9: **Output:** Updated $U^*, V^*, x^+, \mathcal{R}$.
-

4. Control of the heat exchanger plant

The proposed constraint removal approach is used to simplify the control of a laboratory liquid–liquid plate heat exchanger. The considered plate heat exchanger, manufactured by Armfield, is shown in Fig. 2.

The three-stage counter-current liquid–liquid plate heat exchanger (Fig. 2, device (I)) can serve to cool or heat the liquids. These three stages are separated, but interconnected at the same time. MPC was applied to the heating part here. The heat exchanger’s dimensions are as follows: outer width, length, and height are 90 mm, 103 mm, and 160 mm.

The cold medium (cold water) is stored in the retention tanks (Fig. 2, device (II)). As the device works in laboratory conditions, after the cold medium exits the heat exchanger it is not used for any other operation. The hot medium (hot water) is pre-heated to the desired temperature $T_{\text{hot}} = 70$ °C using a heating coil inside a retention tank (Fig. 2, device (III)). After the heating medium exits the heat exchanger it enters the heating tank again. The temperature of the hot medium T_{hot} is controlled by an auxiliary PID controller. Both media are fed to the device by two peristaltic pumps (Fig. 2, devices (IV) and (V)).

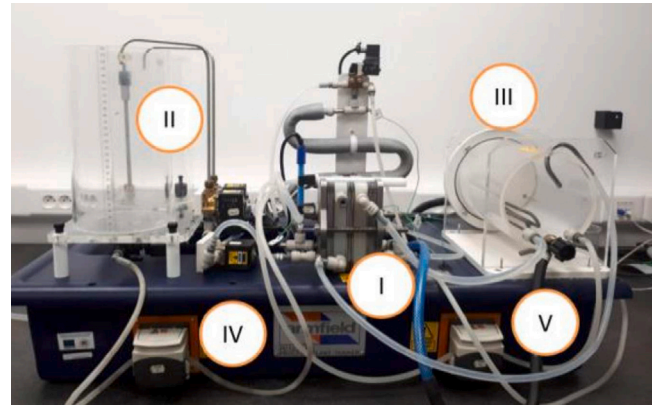


Fig. 2. The controlled plant — plate heat exchanger (I), tanks for cold medium (II), tank for hot medium with heating system (III), peristaltic pump used for cold medium (IV), peristaltic pump used for hot medium (V).

The main objective of the controller design is to heat the cold medium, and, simultaneously, to ensure the offset-free setpoint tracking of the reference temperature T_r . The control output is the outlet temperature of the cold medium T . The value of the temperature mainly depends on the volumetric flow rate of the hot medium \dot{V} , as the temperature of the hot medium is constant with $T_{\text{hot}} = 70$ °C. Hence, the control input is the flow rate of the hot medium. The actuator is the peristaltic pump that feeds the hot medium into the plate heat exchanger (Fig. 2, device (V)). The peristaltic pumps have flexible silicon rubber tubing with wall thickness of 1.6 mm and inner diameter of 3.2 mm.

The inlet temperature of the cold medium is also constant at the temperature $T_0 = 20$ °C. The control output is measured with a K -type thermocouple with operation range 0–150 °C. The detailed scheme of the controlled heat exchanger is presented in Fig. 3, where (I) is the plate heat exchanger, (II) is the peristaltic pump for cold liquid, (III) is the peristaltic pump for hot liquid, (IV) is the retention tank for cold liquid, (V) is the tank with heating coil, (T1–T5) are the temperature sensors, (L1, L2) are the level sensors, (S1–S5) are the solenoid valves, and (W) is the heating coil (see [26] for further information on the plant).

4.1. Mathematical model of the plate heat exchanger

The prediction model for the MPC design purposes was obtained as the nominal model of the intervals bounded by minimum and maximum values for each system parameter, i.e., the gains, and the time constants. These boundary values were evaluated by a set of experimentally collected data generated by a set of laboratory experiments. Then, the step-response-based method of identification was used, see, e.g., [28]. Fig. 4 shows the set of measured and normalized step responses serving for the identification of the parameters of the mathematical model. These were collected by investigating the extensive set of laboratory experiments. The step response of the identified nominal mathematical model is depicted by the black dashed line. The most important property of the mathematical model for controller design purposes, i.e., its ability to track the initial dynamics of the controlled system, is fulfilled by the identified model.

A time delay is not considered as its value is insignificant. Using the minimum and maximum values of the system parameters, a nominal system is created and transformed into a state-space model in the discrete-time domain as defined in (1). Further technical details about the identification of the heat exchanger can be found in [29].

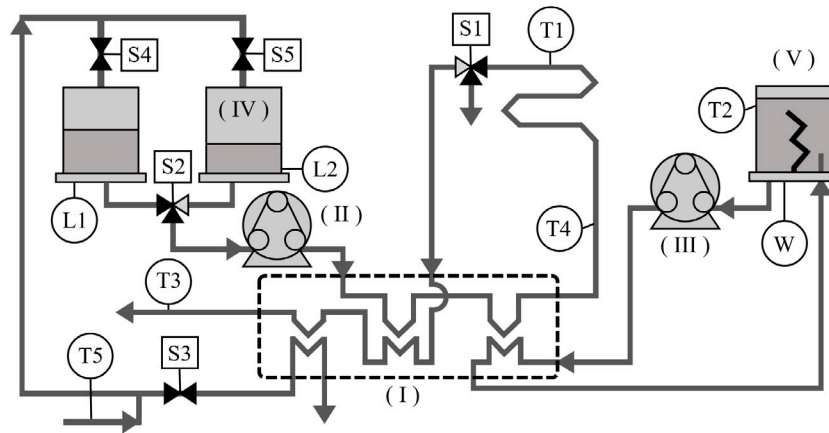


Fig. 3. The detailed scheme of the controlled heat exchanger [27].

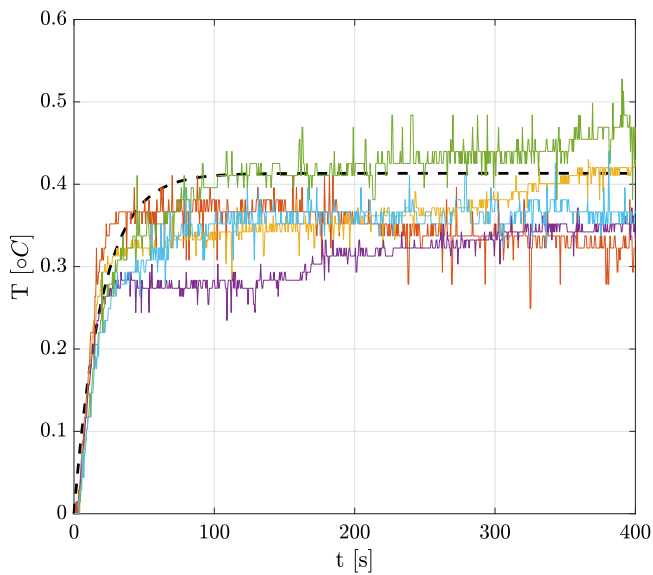


Fig. 4. The set of measured and normalized step responses serves for the identification of the parameters of the mathematical model. The step response of the nominal mathematical model is depicted in the black dashed line.

For the MPC design, the input, state, and output variables of the system in (1) were defined in the form of deviation variables

$$u(k) = \dot{V}(k) - \dot{V}_s, \quad (7a)$$

$$x(k) = T(k) - T_s, \quad (7b)$$

$$y(k) = T(k) - T_s, \quad (7c)$$

where T_s is the operating point of controlled temperature and \dot{V}_s is the operating point of the flow rate.

In industrial applications, controllers are often expected to ensure offset-free setpoint tracking. To remove the steady-state error, the state-space model from (1) is augmented by a term adding the integral action to the controller design. The considered augmented vector of the system states \hat{x} is thus defined in the form

$$\hat{x}(k) = \begin{bmatrix} x(k) \\ \sum_{j=0}^k e(j) \end{bmatrix}, \quad (8)$$

with $e(j) = T_s - T(j) = -x(j)$ the control error, and the negative sign resulting from the laboratory setup.

The matrices of the state-space model considering the augmented vector of states from (8) are defined as

$$\hat{A} = \begin{bmatrix} A & 0 \\ -t_s C & I \end{bmatrix}, \hat{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}, \hat{C} = [C \ 0], \quad (9)$$

where t_s is the sampling time resulting from discretization. Thus the augmented state-space model of the plant has again a structure as in (1)

$$\hat{x}(k+1) = \hat{A}\hat{x}(k) + \hat{B}u(k), \quad (10a)$$

$$y(k) = \hat{C}\hat{x}(k). \quad (10b)$$

State, input, and output matrices read

$$\hat{A} = \begin{bmatrix} 0.8736 & 0 \\ -2 & 1 \end{bmatrix}, \hat{B} = \begin{bmatrix} 0.5562 \\ 0 \end{bmatrix}, \hat{C} = [1 \ 0], \quad (11)$$

respectively. The augmentation of the states only serves to remove the steady-state control error, its influence on the applied constraint removal method will be discussed in Section 4.3. Further technical details regarding the experimental identification of the plant are discussed in [29].

4.2. Control setup

In this experimental case study, the offset-free setpoint tracking of the plant was ensured. The prediction horizon was $N = 7$, and the penalty matrices P , Q , and R were set to

$$P = \begin{bmatrix} 0.1479 & -0.0217 \\ -0.0217 & 0.0053 \end{bmatrix}, Q = \begin{bmatrix} 0.001 & 0 \\ 0 & 0.001 \end{bmatrix}, R = 0.1. \quad (12)$$

Both, the terminal penalty matrix P and the terminal set \mathcal{T} in (3) were determined using the Multi-Parametric Toolbox [30] by solving the Riccati equation. The sampling time considered for the laboratory experiments was $t_s = 2$ s. The weighting matrices Q and R in (3) were systematically tuned. First, we observed the control performance by fixing one of the penalty factors, and, simultaneously, by increasing/decreasing the values of the elements of the remaining weighting matrix. Then, we analyzed the impact of the further increase/decrease of the penalty factor. Finally, we investigated the impact of the other weighting matrix, until a satisfactory control performance was ensured.

Real-time control was implemented with MATLAB/Simulink R2019b on a PC with an i5 CPU (2.7 GHz) and 8 GB RAM. The optimization problems were solved using the MATLAB programming environment [31], and the communication with the plate heat exchanger was implemented with the Wifi-based eLab Manager toolbox [32].

The control input of the plant, i.e., the volumetric flow rate of the hot medium, was constrained to the interval $[0, 11.5] \text{ ml s}^{-1}$ representing the physical constraints. This range was normalized so that the operating point corresponded to \tilde{V}_s . Therefore, the constraints considered on the control input in the deviation form were chosen as

$$-5.75 \text{ ml s}^{-1} \leq u(k) \leq 5.75 \text{ ml s}^{-1}. \quad (13)$$

The outlet temperature T of the cold medium was constrained to the interval $[25, 65] \text{ }^\circ\text{C}$ (i.e. $[298.15, 338.15] \text{ K}$). The state variable x was normalized so that the temperature T_s corresponds to the operating point or the reference temperature. The normalized constraints of the elements of the augmented vector in (8) were defined as

$$\begin{bmatrix} -20 \\ -100 \end{bmatrix} \leq \hat{x}(k) \leq \begin{bmatrix} 20 \\ 100 \end{bmatrix}. \quad (14)$$

The augmented state defined in (8) is an incremental value of the state x and it does not have any operating point.

4.3. Necessary modification of the bounds

The constraint removal method was originally designed from a theoretical point of view assuming a non-increasing optimal cost function value along time steps k . However, to overcome challenges such as measurement noise and plant-model mismatch, the integrator state as described in Section 4.1 was introduced. While this ensures the offset-free tracking of the setpoint value, the integrator state leads to a different development of the closed-loop optimal cost function value. As can be seen in Section 5, a change in the setpoint value T_s in (7b) does not only lead to an increase of V^* due to the new value for the initial state $x(0) = T(0) - T_s$ (in this case a simple *restart* with all constraints added back to the original optimization problem would be possible). However, the integration of the error (see $\hat{x}_2(k)$ in (8)) will result in a further increase before approaching the new setpoint and thus depends on the new setpoint itself.

To make sure constraints that may become active again after the setpoint change are not removed, we present a heuristics to modify the bounds σ_i straightforwardly based on existing experimental data. For ease of presentation, we use the same hypothetical example as in Section 3, augmented by an increase of V^* , in Fig. 5 (red curve).

Every bound σ_i , as the minimum value of the optimal cost function such that constraint i is active, was originally determined assuming a closed-loop cost function trajectory as shown in Fig. 5, green curve. Since the green curve is non-increasing, we can ensure that the corresponding constraint will remain inactive once it was detected to be inactive. However, the setpoint change will lead to a trajectory as shown in Fig. 5, red curve, i.e., the integrator state will lead to a temporary increase of the optimal cost function value V^* for some time steps k . For bound σ_4 corresponding to a hypothetical constraint, e.g., the value of V^* crosses the bound again from below, such that we cannot ensure that constraint $i = 4$ will never become active again. To compensate for this increase, such that the removed constraints will still be inactive despite the increase of V^* , we choose the bounds to be more conservative and thus compensate for the expected increase of the cost function value.

From the set of experimentally collected data generated for the step changes of the setpoint value, we evaluate the maximum increase in V^* caused by the setpoint changes (\tilde{V}^* in Fig. 5(a)). Then we determine sufficiently conservative bounds by reducing their value by the maximum increase in the optimal cost function value V^* , i.e., the value \tilde{V}^* . This is illustrated in Fig. 5(b), where the reduced bounds $\tilde{\sigma}_i$ are shown in comparison to the original σ -bounds. While constraint $i = 4$ was removed in time step $k = 2$ for the original values of the bounds, this constraint is now removed in time step $k = 4$. The conservative bound $\tilde{\sigma}_4$ takes into account the level of the maximum increase in the original cost function V^* , and thus constraint $i = 4$ is not removed too early. This prevents removing a constraint that could become active

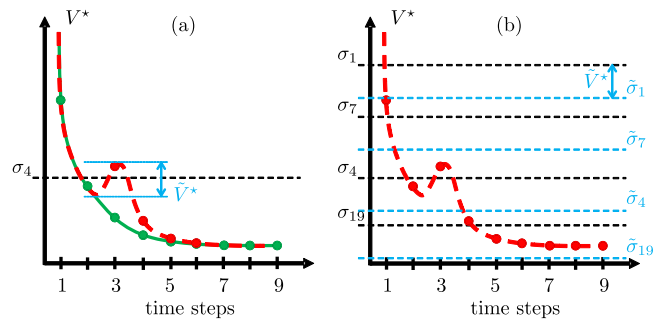


Fig. 5. Modification of the applied constraint removal approach for the hypothetical example from Section 3. (a) MPC with increasing cost (red) and maximum increase \tilde{V}^* . If V^* increases due to the integrator state after a setpoint change, constraints previously detected to be inactive can become active again. (b) Modification of the hypothetical bounds to more conservative values (blue) based on the maximum increase in V^* .

again later during closed-loop control. Obviously, if some σ -bounds are reduced to negative values, they cannot be removed. Therefore, such constraints can be omitted from the evaluation of Algorithm 1, see, e.g., $\tilde{\sigma}_{19}$ in Fig. 5(b). However, constraint removal is efficient also for a subset $\Sigma \subseteq Q$ of all constraints.

For the laboratory case study, we performed an extensive experimental investigation with a conventional MPC to determine the maximum influence of the expected setpoint step changes. Throughout the experiments, we logged the sequences of the closed-loop optimal cost function value. The result is presented in Section 5.1, where two representative control setups were selected.

We stress that this procedure does not give a guarantee that no constraints are removed too early if the conservativeness of the bound on V^* was not appropriately determined during the experiments, similar to the ill-defined level of uncertainty for a robust MPC design. Also, if the setpoints are changing more often or in a different way than expected during the modification process of the σ -bounds, further data evaluation and modification may be necessary. However, we will demonstrate in Section 5.1 that no constraints have been violated when applying MPC combined with the modified constraint removal approach to the laboratory plate heat exchanger. Possible benefits of this approach in terms of energy savings are presented in Section 5.2.

To summarize the steps described so far, the overall process on how to operate MPC with constraint removal on a real plant is sketched in Fig. 6 and can be divided into an offline and an online phase. The offline phase contains the necessary preparations that apply once and before the actual control of the system. It starts with the design of the MPC problem (3), such as deriving a prediction model or tuning the weighting matrices. Once the MPC problem is defined, the sigma bounds need to be computed by solving problem (6) once for each constraint. The last offline step is the modification of the bounds as described in Section 4.3, i.e., by using experiments with expected reference values or existing data of formerly performed experiments.

During the online phase, i.e., for the actual control of the plant, the constraints that may become active have to be reset for a new initial state first, i.e., the constraints are made part of problem (3) again. Then, Algorithm 1 applies to control the system state to its reference. Both steps of the online phase can at least be repeated as long as the initial states and reference values fit the data used during the preparatory offline phase.

5. Experimental results

This section provides two extensive experimental case studies. First, we investigate the control performance of MPC with constraint removal designed for the laboratory plate heat exchanger. Then, we analyze the energy savings achieved by the proposed control method considering an embedded platform suitable for industrial control.

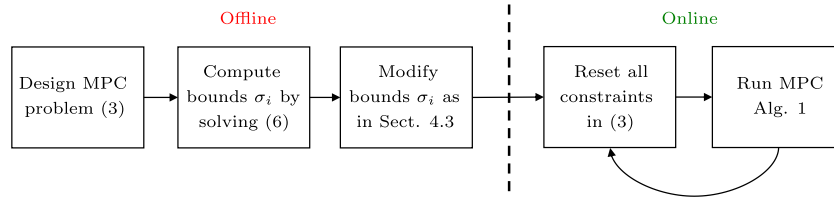


Fig. 6. Workflow of operating MPC with constraint removal on a real plant.

5.1. Experimental validation using the plate heat exchanger

Multiple setpoint step changes were performed and analyzed to demonstrate the efficiency of the proposed acceleration by the constraint removal approach. The closed-loop control trajectories, cost function value V^* , and number of considered constraints c for MPC with constraint removal are depicted in Fig. 7(a) and Fig. 8(a) for two representative setpoint changes, respectively.

The constraint removal strategy (and thus Algorithm 1) started at the same time instance as the setpoint step change, i.e., at $t = 50$ seconds. In all scenarios, the number of considered constraints immediately decreased.

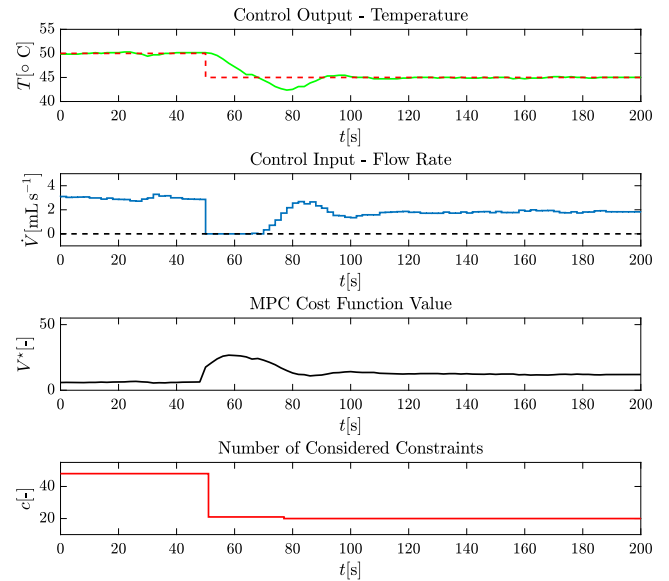
The associated control trajectories and cost function value for the same setpoint changes generated by MPC without constraint removal are depicted in Figs. 7(b)–8(b). These experimentally collected results demonstrate that implementing the proposed constraint-removal-based acceleration technique does not decrease the control performance. Obviously, the control performance in Figs. 7(a)–8(a) can slightly differ from the performance depicted in Figs. 7(b)–8(b) as the presented results are experimentally collected and it is not possible to fully replicate the real plant behavior due to the random impact of disturbances and measurement noise.

For the modification of the bounds as described in Section 4.3, the values of the maximum increase \tilde{V}^* were evaluated based on the results shown in Figs. 7(b)–8(b). By analyzing these sequences, we determined the maximum increase \tilde{V}^* to the value $\tilde{V}^* = 19.06$. The corresponding value of each modified bound σ_i , $i \in \{1, \dots, q\}$ is depicted in Fig. 9. In total, 12 out of all 48 constraints were detected to be redundant, i.e., they will never be active as there exist different, more restrictive constraints. For these 12 redundant constraints, according to Definition 1, $\sigma_i = \infty$ holds (see the constraints depicted on the far right in Fig. 9). They can be removed completely before running the controller. All constraints with bounds below a value of zero will not be removed, as the MPC cost function value is non-negative by definition.

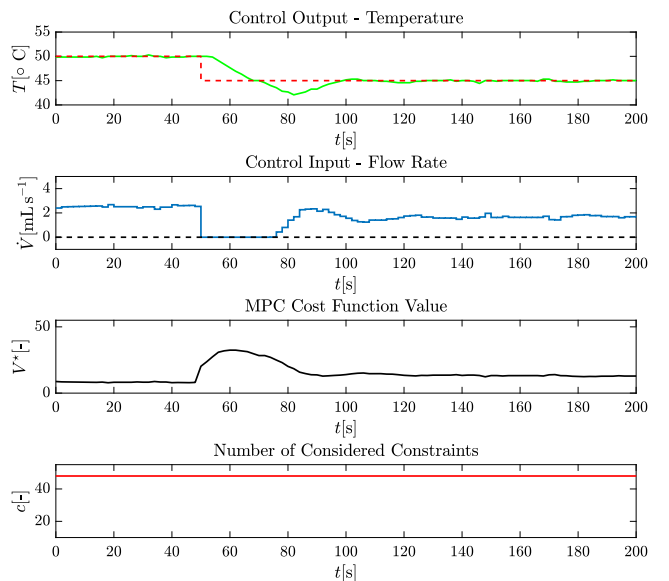
The experimental results in Figs. 7(a)–8(a) show that the designed MPC controller with constraint removal ensured setpoint tracking within 2 minutes. Simultaneously, the computed values of the control input respected the given physical constraints in (13)–(14) even though some of the constraints were removed from the optimization problem by the applied constraint removal method.

At the beginning of the experiment, 48 constraints were considered, and at the end of the time span, only 20 and 18 constraints remained considered in Fig. 7(a) and Fig. 8(a), respectively. Thus, in the experimentally evaluated control scenarios, the MPC cost function value decreased enough to remove at least 28 constraints. Although the first state variable reached the origin, the MPC cost function value depicted in Figs 7–8 is not decreasing to zero, but stays slightly above. Measurement noise and especially the value of the integrator state \hat{x}_2 kept the cost from decreasing further and inhibited removing more than 28 constraints. Also, due to the modification of the bounds, some of the $\tilde{\sigma}_i$ attain negative values as described in Section 4.3 and will thus never be removed.

In conclusion, despite using the conservative bounds $\tilde{\sigma}_i$, between around 58% and 62.5% of all constraints were removed from the original OCP without affecting the control performance negatively.

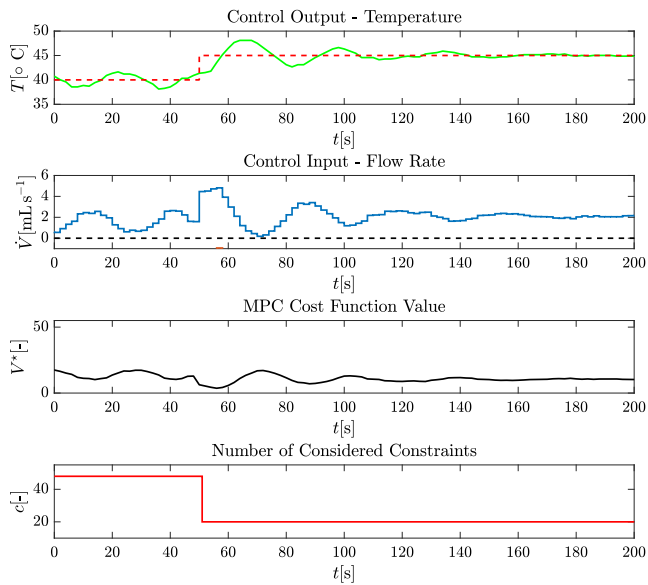


(a) Results generated by MPC with constraint removal.

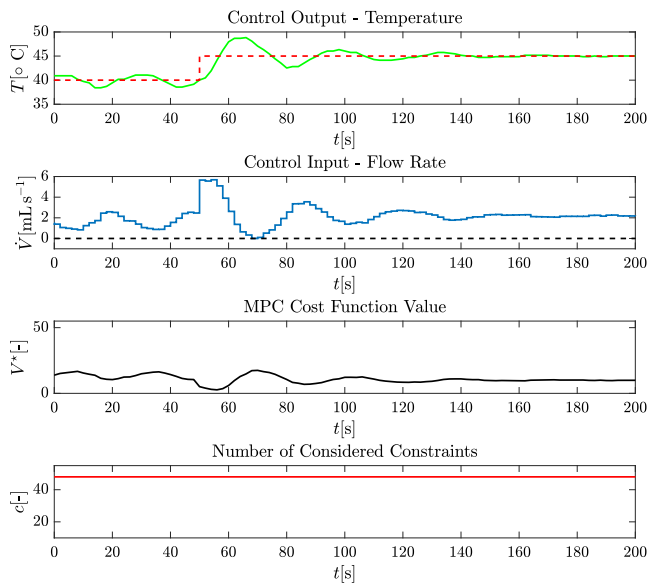


(b) Results generated by MPC without constraint removal.

Fig. 7. Experimental results for MPC with (see (a)) and without (see (b)) constraint removal and setpoint step change 50 °C → 45 °C (323.15 K → 318.15 K): measured control output (solid green), setpoint (dashed red), control input (solid blue), constraint (dashed black), cost function value (solid black), number of considered constraints (solid red).



(a) Results generated by MPC with constraint removal.



(b) Results generated by MPC without constraint removal.

Fig. 8. Experimental results for MPC with (see (a)) and without (see (b)) constraint removal and setpoint step change $40\text{ }^{\circ}\text{C} \rightarrow 45\text{ }^{\circ}\text{C}$ ($313.15\text{ K} \rightarrow 318.15\text{ K}$): measured control output (solid green), setpoint (dashed red), control input (solid blue), constraint (dashed black), cost function value (solid black), number of considered constraints (solid red).

5.2. Experimental evaluation of energy savings

We demonstrate non-negligible energy savings result with the proposed controller. Here, the practical benefits of the applied method were analyzed by evaluating the energy amount consumed by the control unit during its operation.

In practice, control units equipped with a 32-bit microprocessor are gaining more prominence, as they provide sufficient computing power for a wide range of industrial applications. We show that the energy consumption of such a control unit can be reduced considering the presented acceleration method.

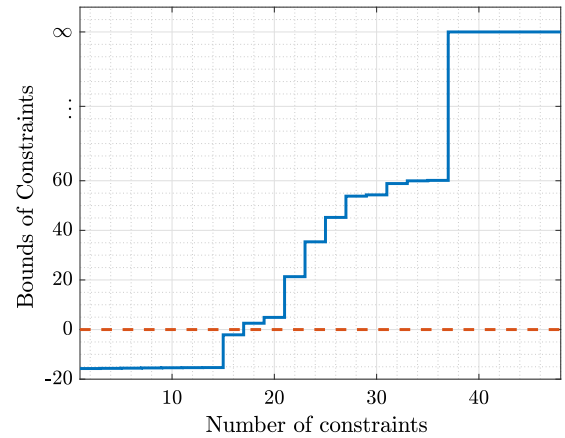


Fig. 9. Values of the modified bounds evaluated for the 48 constraints of the MPC problem, sorted for increasing values. Note that the enumeration here does not reflect the order of the constraints.

In this second case study, the following steps were performed using the microcontroller:

- implementation of a QP solver on a 32-bit microcontroller
- simulation of the control steps
- evaluation of computation time and the corresponding energy consumption by the control unit

The ESP32 DevKit V4 microcontroller platform was used as a control unit. This platform is equipped with a 32-bit microprocessor with 4 MB of Flash memory, which is sufficient to handle the library necessary for solving OCPs having the form of a QP. This library was generated using the CVXGEN tool [33], which created a tailored solver dedicated to solving QP-representable convex optimization problems. The generated solver was exported in C-code, which is compatible with the control unit.

In this case study, the data measured during the experiments was used to simulate the control of the heat exchanger plant using the embedded control unit. Specifically, two sets of data were used: (i) control using MPC without constraint removal, see Fig. 8(b), and (ii) control using MPC with constraint removal, depicted in Fig. 8(a). Both data sets refer to the step change from $40\text{ }^{\circ}\text{C}$ (313.15 K) to $45\text{ }^{\circ}\text{C}$ (318.15 K). Fig. 8(a) shows that considering the constraint removal approach, the number of constraints dropped only once, i.e., from 48 to 20 constraints. Therefore, using this observation, the computational time and the corresponding energy consumption necessary to solve the QP on a microcontroller were compared considering 48 and 20 constraints, respectively.

We used the system states measured during the experiments on the heat exchanger in Section 5.1 as initial conditions for the solution of the online QP implemented on the microcontroller at each sampling instant. Then the associated control input was evaluated, simulating the real-time control of the plant. Within the simulated closed-loop control, it is possible to determine the time required to evaluate the optimal control action by solving the associated QP. The corresponding experimentally generated results are depicted in Fig. 10.

We could assume that solving an optimization problem that considers a smaller number of constraints will consume a less portion of time. However, the procedure of the constraint removal approach also introduces additional operations necessary to evaluate the comparison of the bounds σ_i with the current cost function value V^* to detect the constraints to be removed. To make the results more comparable, the closed-loop control considering 20 constraints also included the operations necessary to identify inactive constraints. Therefore, at each

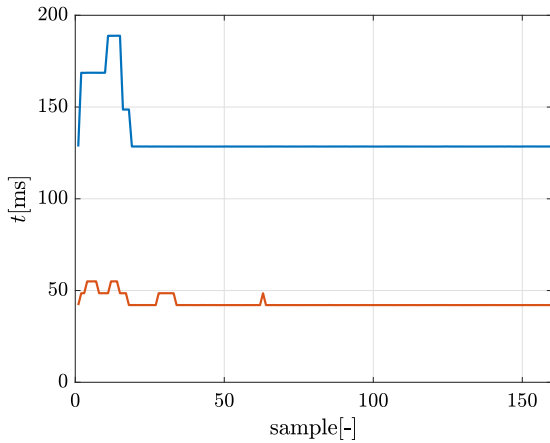


Fig. 10. Time necessary for solving a QP at each sampling instance using the modified constraint removal method (orange) and conventional MPC (blue).

Table 1

Comparison of the average and total solver time for conventional MPC and MPC with constraint removal.

MPC method	t_{avg} [s]	t_{sol} [s]
Conventional MPC	0.133	21.295
MPC with constraint removal	0.0433	6.921

Table 2

Comparison of the energy consumption for conventional MPC and MPC with constraint removal within one sampling period.

MPC method	E_{t_s} [$\times 10^{-3}$ J]
Conventional MPC	38
MPC with constraint removal	12

sampling time, the microcontroller compared the bounds σ_i for each considered constraint with the current cost function value V^* .

Simulation of the closed-loop control showed that the total time required to resolve all QPs throughout the whole time of the operation was significantly lower using the proposed constraint removal approach. Table 1 summarizes the evaluated computational time, where the criterion t_{avg} represents an average computational time evaluated for each control step. This criterion was computed considering the total number of 160 control steps. The criterion t_{sol} represents the total time necessary to solve the QP in each sampling time during the simulation period. As can be seen, the proposed method reduced the total solver time t_{sol} by around 68%.

The energy consumption of the control unit depends on whether calculations are in progress or not. Therefore, the reduced computation time t_{sol} corresponds to the energy savings, i.e., the saved electric power. Within the operation and control of the laboratory plant, we can divide the activities of the control unit into three main groups:

1. The first group is *routine operations* such as the application of a control action to a controlled process and the acquisition of measured values from sensors.
2. The second group includes the *evaluation of optimal control action*, e.g., operations associated with the constraint removal approach and the solution of the QP.
3. The last group is the *sleep mode*, which fills the time until the end of a given sampling period.

Analogously to the case study presented in Section 5.1, we also considered a sampling period of 2 s here. Fig. 11 illustrates the individual stages of the operation within one sampling instance.

The measured results shown in Fig. 11 demonstrate that, depending on the applied control strategy, a sampling instance is split differently into the three groups mentioned above. The electric current drawn by the microcontroller amounts to $I_{\text{comp}} = 61.3$ mA when a QP is solved. On the other hand, during the sleep mode, the value is $I_{\text{sleep}} = 3.92$ mA. Therefore, the decisive factor is the time needed for the solution. The time difference multiplied by the measured current and the set voltage represents the consumed electric power and is calculated by

$$E_{t_s} = (I_{\text{comp}} - I_{\text{sleep}}) U_s t_{\text{avg}}, \quad (15)$$

where E_{t_s} is the energy consumed within one sampling period, and U_s is the supply voltage of the control unit set to $U_s = 5$ V. The energy was computed for the MPC with and without constraint removal. The results are summarized in Table 2. As can be seen, when considering MPC with constraint removal, also the energy consumption is reduced by around 68% compared to the conventional MPC.

The energy consumption saved per hour of operation is then computed as

$$E_h = \frac{3600}{t_s} \Delta E_{t_s} = 41.222 \text{ J}, \quad (16)$$

where ΔE_{t_s} is the difference between energy consumption E_{t_s} computed for conventional MPC and for MPC with constraint removal, see Table 2. Subsequently, it is quite straightforward to calculate the saved annual energy consumption of one control unit per year, $E_y = 361.1$ kJ. Such an amount of energy is equivalent to 20 Ah. The presented results can reflect, e.g., the significantly increased battery life supplying the controller platform.

6. Conclusion

We applied MPC with a constraint removal approach to a laboratory plate heat exchanger. By detecting and removing inactive constraints before solving the underlying optimization problem, this variant of MPC reduces the computational effort associated with solving the optimization problem. The bounds indicating the inactive constraints were modified to overcome the challenges arising in the control setup. In this paper, two experimental case studies were investigated to analyze the properties of the proposed control method - control of the laboratory heat exchanger plant, and the implementation on a microcontroller.

In the real-time experiments on the laboratory plant, the constraint removal approach was able to reduce the number of constraints to be considered in the optimization problem by up to 60% compared to conventional MPC. The results further confirmed that the approach does not affect the control performance in terms of performance losses, resulting in comparable trajectories of the control inputs.

Based on the experimental data, we further implemented and solved the optimization problems corresponding to MPC with constraint removal and to conventional MPC on a 32-bit microcontroller. Both, the computation time and the associated energy consumption decreased by approximately 68% for MPC with constraint removal in contrast to the conventional variant of MPC.

Future research will be focused on the application of nonlinear MPC with constraint removal for the control of the heat exchanger plant and modifications towards robust MPC.

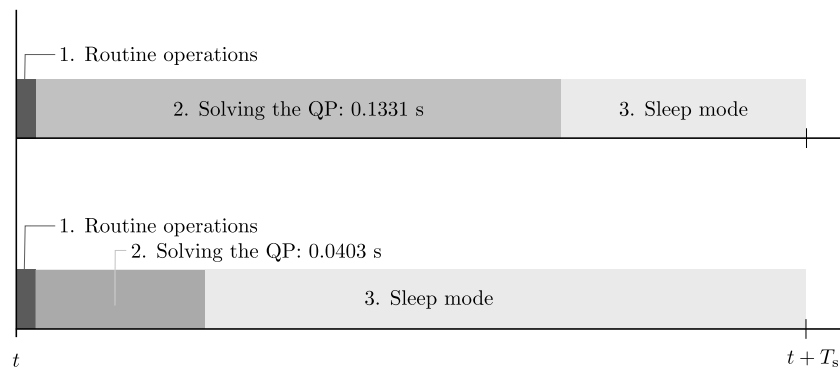


Fig. 11. The timeline of the operations within one sampling instance.

Nomenclature

Symbols

A, B, C	State-space system
$\hat{A}, \hat{B}, \hat{C}$	Augmented state-space system
A, I, \tilde{I}	Active set, inactive set, subset
e	Control error
E_h, E_y	Energy consumption per hour, year
$E_s, \Delta E_s$	Energy consumed within one sampling period, difference
$F, \mathcal{J}, \mathcal{R}$	Feasible set, terminal set, set of remaining constraints
G, w, E	Constraint matrices of the quadratic program
$I_{\text{comp}}, I_{\text{sleep}}$	Electric current consumed during solving, sleep mode
k	Time step
m, n, p, q	Number of inputs, states, outputs, constraints
N	Horizon length
P, Q, R	Weighting matrix for terminal state, states, inputs
Q, Σ	Constraint set, subset
T, T_{hot}, T_s	Temperature of the cold medium, hot medium, setpoint
$t_{\text{sol}}, t_{\text{avg}}$	Total, average solver time
u, y	System input, output
$u_{\text{min}}, u_{\text{max}}$	Lower, upper bound on inputs
U, X	Input, state prediction
\mathcal{U}, \mathcal{X}	Constraint set for inputs, states
U_s	Supply voltage of the control unit
V, \tilde{V}^*	Cost function, maximum increase in V^*
\dot{V}, \dot{V}_s	Volumetric flow rate of the hot medium, operating point
x, x^+, x_0, \hat{x}	System state, subsequent state, initial state, augmented system state
$x_{\text{min}}, x_{\text{max}}$	Lower, upper bound on states
Y, F, H	Matrices of the cost function of the quadratic program
Greek letters	
σ_i	Bound corresponding to constraint i
Abbreviations	
MPC	Model predictive control
OCF	Optimal control problem
PID	Proportional–integral–derivative controller
QP	Quadratic program

References

- [1] Q. Chen, M. Wang, N. Pan, Z.-Y. Guo, Optimization principles for convective heat transfer, *Energy* 34 (2009) 1199–1206.
- [2] F. Friedler, Process integration, modelling and optimisation for energy saving and pollution reduction, *Appl. Therm. Eng.* 30 (16) (2010) 2270–2280.
- [3] L. Zhi-Yong, P.S. Varbanov, J.J. Klemeš, J.Y. Yong, Recent developments in applied thermal engineering: Process integration, heat exchangers, enhanced heat transfer, solar thermal energy, combustion and high temperature processes and thermal process modelling, *Appl. Therm. Eng.* 105 (2016) 755–762.
- [4] C.E. García, D.M. Prett, M. Morari, Model predictive control: Theory and practice – a survey, *Automatica* 25 (3) (1989) 335–348.
- [5] F. Salem, M.I. Mosaad, A comparison between MPC and optimal PID controllers: Case studies, in: Michael Faraday IET International Summit 2015, 2015, pp. 59–65.
- [6] D. Krishnamoorthy, S. Skogestad, Online process optimization with active constraint set changes using simple control structures, *Ind. Eng. Chem. Res.* 58 (30) (2019) 13555–13567.
- [7] A. Kumar, L. Samavedham, I.A. Karimi, R. Srinivasan, Critical assessment of control strategies for industrial systems with input–output constraints, *Ind. Eng. Chem. Res.* 61 (30) (2022) 11056–11070.
- [8] H. Weeratunge, G. Narsilio, J. de Hoog, S. Dunstall, S. Halgamuge, Model predictive control for a solar assisted ground source heat pump system, *Energy* 152 (2018) 974–984.
- [9] A. González, D. Odloak, J.L. Marchetti, Predictive control applied to heat-exchanger networks, *Chem. Eng. Proc.: Proc. Intensif.* 45 (2006) 661–671.
- [10] A. Vasičkaninová, M. Bakošová, Control of a heat exchanger using neural network predictive controller combined with auxiliary fuzzy controller, *Appl. Therm. Eng.* 89 (2015) 1046–1053.
- [11] V.D. Blondel, J.N. Tsitsiklis, A survey of computational complexity results in systems and control, *Automatica* 36 (9) (2000) 1249–1274.
- [12] R. Cagienard, P. Grieder, E. Kerrigan, M. Morari, Move blocking strategies in receding horizon control, *J. Process Control* 17 (6) (2007) 563–570.
- [13] P. Patrinos, A. Bemporad, An accelerated dual gradient-projection algorithm for embedded linear model predictive control, *IEEE Trans. Automat. Control* 59 (1) (2014) 18–33.
- [14] Y. Wang, S. Boyd, Fast model predictive control using online optimization, *IEEE Trans. Control Syst. Technol.* 18 (2) (2010) 267–278.
- [15] A. Bemporad, M. Morari, V. Dua, E.N. Pistikopoulos, The explicit linear quadratic regulator for constrained systems, *Automatica* 38 (1) (2002) 3–20.
- [16] M.M. Seron, G.C. Goodwin, J.A. De Doná, Characterisation of receding horizon control for constrained linear systems, *Asian J. Control* 5 (2) (2003) 271–286.
- [17] M. Kvasnica, M. Fikar, Clipping-based complexity reduction in explicit MPC, *IEEE Trans. Automat. Control* 57 (7) (2012) 1878–1883.
- [18] C.N. Jones, M. Morari, Polytopic approximation of explicit model predictive controllers, *IEEE Trans. Automat. Control* 55 (11) (2010) 2542–2553.
- [19] M. Kvasnica, P. Bakaráč, M. Klaučo, Complexity reduction in explicit MPC: A reachability approach, *Systems Control Lett.* 124 (2019) 19–26.
- [20] M. Jost, M. Mönnigmann, Accelerating model predictive control by online constraint removal, in: Proc. of the 52nd IEEE Conference on Decision and Control, 2013, pp. 5764–5769.
- [21] M. Jost, M. Mönnigmann, Accelerating online MPC with partial explicit information and linear storage complexity in the number of constraints, in: Proc. of the 2013 European Control Conference, 2013, pp. 35–40.
- [22] M. Jost, G. Pannocchia, M. Mönnigmann, Online constraint removal: Accelerating MPC with a Lyapunov function, *Automatica* 57 (2014) 164–169.

- [23] M. Jost, G. Pannocchia, M. Mönnigmann, Accelerating linear model predictive control by constraint removal, *Eur. J. Control* 35 (2017) 42–49.
- [24] R. Dyrška, M. Mönnigmann, Accelerating nonlinear model predictive control by constraint removal, *IFAC-PapersOnLine* 54 (6) (2021) 278–283.
- [25] D.Q. Mayne, J.B. Rawlings, C. Rao, P.O.M. Scokaert, Constrained model predictive control: Stability and optimality, *Automatica* 36 (6) (2000) 789–814.
- [26] Armfield, Instruction Manual Process Plant Trainer PCT23-MKII, Armfield Limited, 2007.
- [27] J. Oravec, M. Bakošová, L. Galčíková, M. Slávik, M. Horváthová, A. Mészáros, Soft-constrained robust model predictive control of a plate heat exchanger: Experimental analysis, *Energy* 180 (2019) 303–314, <http://dx.doi.org/10.1016/j.energy.2019.05.093>, URL https://www.uiam.sk/assets/publication_info.php?id_pub=2003.
- [28] J. Mikleš, M. Fikar, *Process Modelling, Identification, and Control*, Springer Verlag, Berlin Heidelberg, 2007.
- [29] J. Oravec, M. Bakošová, D. Pakšiová, N. Mikušová, K. Batárová, Advanced robust MPC design of a heat exchanger: Modeling and experiments, in: 27th European Symposium on Computer Aided Process Engineering, 2017, pp. 1585–1590.
- [30] M. Herceg, M. Kvasnica, C. Jones, M. Morari, Multi-parametric toolbox 3.0, in: Proc. of the 2013 European Control Conference, 2013, pp. 502–510.
- [31] MATLAB Optimization Toolbox, The MathWorks, Natick, MA, USA, 2019.
- [32] M. Kalúz, L. Čírka, R. Valo, M. Fikar, Lab of things: A network-based I/O services for laboratory experimentation, *IFAC-PapersOnLine* 50 (1) (2017) 13486–13491.
- [33] J. Mattingley, S. Boyd, CVXGEN: A code generator for embedded convex optimization, *Opt. Eng.* 13 (1) (2012) 1–27.



Self-tunable approximated explicit MPC: Heat exchanger implementation and analysis

Lenka Galčíková^{*}, Juraj Oravec

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

ARTICLE INFO

Keywords:

Tunable explicit MPC
Self-tunable technique
Tuning parameter
Heat exchanger
Control performance

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 novel self-tunable control policy that does not require any interventions of the control engineer during operation in order to retune the controller subject to the changed working conditions. 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. This extension 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. 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 procedure. The designed self-tunable controller improved control performance by decreasing sum-of-squared control error, maximal overshoots/undershoots, and settling time compared to the conventional control strategy based on a single (non-tunable) controller.

1. Introduction

The current crisis of energy resources emphasizes the long-term goal of achieving sustainable industrial production and optimal energy utilization. Moreover, minimizing the energy utilization directly reduces the corresponding CO₂ emissions. Therefore, sustainable industrial production is focused on the wide implementation of advanced control methods [1]. A recent survey on applied thermal engineering focused on energy saving and pollution reduction from the industrial perspective is provided in [2], and references therein.

The heat exchangers in their numerous variants are integrated into many industrial plants as the heat transfer represents the crucial phenomena for all thermal energy applications [3]. Simultaneously, the utility generation for heating or cooling is energy-demanding. From the control viewpoint, the controller design for the heat exchangers is a challenging task due to the necessity to take into account the nonlinear and asymmetric behavior of the device, *i.e.*, different plant behavior when the temperature is increasing, in contrast to the behavior when the temperature is decreasing, see [4].

A very common challenge in terms of the time-varying behavior of heat exchangers is fouling. The authors in [5] focus on modeling

the thermal efficiency in a cross-flow heat exchanger using an artificial neural network, which leads to a highly accurate model. In [6], the authors address the effect of fouling by adjusting the parameters of the proportional–integral–derivative (PID) controller.

Although the conventional and widely-used PID controllers are robust and easy to tune, their control performance may not be sufficient. Various extensions built above the well-tuned PID controller were developed to compensate for the nonlinear and asymmetric behavior, often affected by the additional impact of the uncertain parameters. Such widely-used control strategies include, *e.g.*, the robust control [7], the gain-scheduling, and adaptive control. In a recent study [8], the authors suggest to adjust the controller online, based on a minimization of an objective function designed to achieve the desired control performance. For the rigorous mathematical modeling and controller design methods in general, see [9], and for the controller design tailored for the process control engineers see [10].

One of the promising control strategies addressing all these issues in an optimal way came with the formation of the model predictive control (MPC), *e.g.*, see [11]. MPC provides optimal control input

^{*} Corresponding author.

E-mail address: lenka.galcikova@stuba.sk (L. Galčíková).

based on the minimization of a specified cost function while considering a model of the system. Compared to linear quadratic controllers (LQR) [12], model predictive control also includes constraints on the control input or process variables [13], and additional saturation is not necessary. Moreover, as the optimization problem is solved in every control step, MPC represents a receding horizon control policy [14], having a significant benefit mainly in the terms of disturbance rejection. The model predictive control was intensively investigated in connection with heat exchangers. In [15], the authors developed a model predictive control for a shell and tube heat exchanger. Four robust control strategies were presented and compared in [16]. A two-level control structure was applied on a heat exchanger network in [17], where the low level of control was ensured by MPC and the high level by a supervisory online optimizer. The fast nonlinear MPC was designed to optimize the waste heat recovery [18]. The multi-layer control designed in [19] designed the MPC in the leader loop to optimize the thermal response to improved control performance. The applicability of model predictive control expanded with the parametric solution of the MPC optimization problem, known as explicit MPC [20]. As the MPC optimization problem is pre-solved offline, it does not need to be solved in the online phase, *i.e.*, in real-time control. Instead, a piece-wise affine (PWA) control law is evaluated to apply the optimal control action in each control step. The complexity of construction of the explicit MPC controller grows exponentially with the number of considered constraints. If the MPC design problem can be pre-solved explicitly offline, the consequent reduced online computational complexity makes the explicit MPC more suitable for practical industrial implementation. Nevertheless, the explicit MPC is not tunable in default as the conventional approach in [20] considers the penalty matrices with fixed structure and values. The inability to tune the explicit controller online can be a disadvantage due to varying operating conditions when the different setups of the controllers are beneficial.

The possibility to tune the explicit MPC online came with the publishing of [21]. The tuning parameter penalizing the control inputs became a parameter, for which the optimal controller was pre-computed. Nevertheless, the application was limited only to linear cost functions of the optimization problem. To satisfy the demands for often-used quadratic cost functions, the approximated tunable explicit MPC was presented in [22]. The technique is based on two explicit model predictive controllers which differ in the setup of one penalty matrix. The two explicit MPCs provide upper and lower boundary optimal controllers. Based on the evaluation of the two boundary control inputs, the tuned control input is calculated by linear interpolation. The follow-up work [23] provided stability and recursive feasibility guarantees by proper choice of the terminal penalty matrix and terminal set constraint [24]. Moreover, the strategy in [23] extends the tuning ability based on any penalty matrix and not just the input penalty.

The idea of approximated tunable MPC with neural networks is presented in [25]. To ensure the tuning property, the penalty matrices were included in the training process. As a result, it was possible to tune the neural network-based controller online, while mimicking the nearly optimal MPC. In [26], the neural network-based tunable controller MPC was extended with a corrector which steered the controller such that the constraints on the manipulated and process variables were satisfied.

The paper [27] pushes the idea of tunable explicit MPC further and deals with the issues of practical industrial-oriented implementation. In numerous practical applications, the reference value of the controlled variable is changed and acquires values from a wide range of operating conditions. The use of different controller setups can help handle the plant's nonlinear behavior. The paper [27] presents a procedure of the self-tunable controller technique. The controller's aggressivity is tuned based on the difference between the reference value and the steady state corresponding to the model linearization point. In the context of MPC, the aggressiveness is associated with the setup of the penalty matrices, as it determines the aggressiveness of the final control input. In general, higher penalization of the controlled states or control error

in the cost function leads to more aggressive control actions. This process is analogous to increasing the proportional gain in the PID controller. On the contrary, higher penalization of the input variable leads to more sluggish control, *e.g.*, see [13]. In [27], the MPC tuning based on the distance from the steady-state operating point represented a way how to compensate for the system's nonlinear behavior.

This work directly extends our results presented in [27], where the basic principles of the self-tunable approximated explicit MPC were introduced. In this paper, a novel method of self-tuning parameter setup is introduced. Compared to [27], the self-tuning method is based on the size of the reference step change. Moreover, the idea of further scaling of the tuning parameter is elaborated. The interval of the values of the self-tuning parameter is split at some certain value and each part of the interval corresponds to the specific operating conditions defined by the control engineer. In such a way, *e.g.*, the system's asymmetric behavior is compensated. Finally, to investigate the benefits of the proposed approach, the proposed self-tuning control policy was implemented to control a laboratory-scaled counter-current plate heat exchanger. This work provides the control performance evaluation and analysis using the self-tunable controller compared to the boundary explicit MPCs.

The paper is organized as follows. First, the theoretical backgrounds are presented in Section 2, where the explicit MPC, the approximated tunable explicit MPC, and existing self-tunable methods are briefly elaborated. Then, the novel proposed method of self-tunable procedure is explained in detail in Section 3. Finally, the experimental results of the self-tuning controller implementation on a heat exchanger are discussed in Section 4, followed by the main conclusions in Section 5.

2. Theoretical backgrounds

In this section, the theoretical backgrounds necessary for the proposed method are 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 a self-tunable technique of the approximated explicit MPC are presented.

2.1. Explicit model predictive control

Explicit model predictive control [20] utilizes a parametric solution of the model predictive control introducing its application range towards the systems with fast dynamics. Moreover, the explicit solution enables providing 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 industrial-oriented implementation, let us consider the optimization problem in the following form:

$$\min_{u_0, u_1, \dots, u_{N-1}} \sum_{k=0}^{N-1} \left((y_k - y_{\text{ref}})^T Q_y (y_k - y_{\text{ref}}) + u_k^T R u_k + x_{1,k}^T Q_1 x_{1,k} \right) \quad (1a)$$

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

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

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

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

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

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

where k denotes the step of the prediction horizon N . To obtain the offset-free control results, the built-in integrator was included in the state-space model, *e.g.*, see [28]. The prediction model in Eq. (1b)–(1c) has the form of augmented linear time-invariant (LTI) system for a given augmented state matrix $\tilde{A} \in \mathbb{R}^{n_x \times n_x}$, augmented input matrix $\tilde{B} \in \mathbb{R}^{n_x \times n_u}$ and augmented output matrix $\tilde{C} \in \mathbb{R}^{n_y \times n_x}$. Variables $\tilde{x} \in \mathbb{R}^{n_x}$,

$u \in \mathbb{R}^{n_u}$, $y \in \mathbb{R}^{n_y}$ are vectors of corresponding augmented system states, control inputs, and system outputs, respectively. The sets $\mathcal{U} \subseteq \mathbb{R}^{n_u}$, $\mathcal{Y} \subseteq \mathbb{R}^{n_y}$ are convex polytopic sets of physical constraints on inputs and outputs, respectively. These sets include the origin in their strict interiors. The penalty matrix $Q_y \in \mathbb{R}^{n_y \times n_y}$, $Q_y \geq 0$ penalizes the squared control error, *i.e.*, the deviation between the controlled output and output reference value y_{ref} . The matrix $R \in \mathbb{R}^{n_u \times n_u}$, $R > 0$ penalizes the squared value of control inputs. The value of integrator is also penalized in the cost function with the penalty matrix $Q_I \in \mathbb{R}^{n_y \times n_y}$, $Q_I \geq 0$. All the penalty matrices are considered to be diagonal due to the applicability of the self-tunable explicit MPC approach. 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 Eq. (1b)–(1c) is rewritten as follows:

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

$$y_k = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x_k \\ x_{1,k} \end{bmatrix}, \quad (2b)$$

where $x_1 \in \mathbb{R}^{n_y}$ is the integral of the control error, T_s denotes the sampling time, and matrices A , B , C are the well-known state-space matrices that form the augmented LTI model. As a consequence of this extension and penalization in the cost function in Eq. (1a), not only the control error is penalized, but also the integrated value, which leads to analogous offset-free reference tracking results as incorporating an integral part in the PID controller.

The parametric solution of the optimization problem of the quadratic programming (QP) in Eq. (1) leads to the explicit solution in the form of piecewise affine PWA control law defined above the domain 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 $F_i \in \mathbb{R}^{n_u \times n_x}$ and $g_i \in \mathbb{R}^{n_u}$ 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 polytopic critical region \mathcal{R}_i , where the parameter θ belongs, the optimal control input is calculated based on the associated control law in Eq. (3).

Note, many other formulations of the optimization problems for the explicit MPC design were formulated mainly in terms of the definition of the cost functions in Eq. (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 disturbance modeling and estimation. For such an overview see, *e.g.*, [29]

2.2. Tunable explicit model predictive control

The aggressivity of the controller and the whole nature of the control is influenced by appropriate fine-tuning of the penalty matrices in the optimization problem in Eq. (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 afterward without trading off a significant increase in the controller complexity or the performance loss. As the operating conditions and requirements on controller setup may differ throughout the control, the ability to adjust the controller's aggressivity can be very beneficial.

The idea of approximated tunable explicit MPC comes from the work [22], 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 are constructed by solving the optimization problem having the same structure and setup, except for one of the penalty matrices — the tuned one. Based on the specific control application, any penalty matrix can be chosen as the tuned parameter, *i.e.*, this approach is applicable for any penalty matrix. The boundary penalty matrices follow the assumptions on the penalty matrices from Section 2.1 and are diagonal matrices such that $\lambda_{i,L} \leq \lambda_{i,U}$, $\forall i = 1, \dots, s$, 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.

Let us consider the penalty matrices in the cost function in Eq. (1a). The penalty matrices are scaled in the following way:

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

$$Q_I(k) = (1 - \rho(k)) Q_{I,L} + \rho(k) Q_{I,U}, \quad (4b)$$

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

where ρ represents the tuning parameter such that $0 \leq \rho \leq 1$ holds. Based on the rules in Eq. (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 Eq. (4) should be tuned, it is suggested to judge the control performance by systematic tuning of all the penalty matrices. Systematic tuning involves selecting a specific penalty matrix and observing the control results by gradually increasing or decreasing the diagonal elements of the matrix. This process is then repeated for the remaining penalty matrices in a similar manner.

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:

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

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 comes with the cost of storing and evaluating two explicit controllers. Nevertheless, the ability to tune the controller may be more important in many practical applications.

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

Remark 2.1. If the asymptotic stability and recursive feasibility guarantees are required, the reader is referred to follow the instructions from [23]. In order to satisfy these requirements, the study introduces a procedure for computing the common terminal penalty and terminal set for the two boundary controllers.

Remark 2.2. Not only Eq. (5) needs to be chosen for interpolation of the control input. Another way of tuning of the control input can be using some nonlinear relation for the interpolation.

2.3. Self-tunable explicit model predictive control

The advantage of a 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 [27]. 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 [27] focuses on adjusting the penalty matrix when the reference value is changed. The further the reference value is from the steady state, the more aggressively the controller is tuned. The idea behind the suggested scaling lies in compensation for the nonlinear behavior of the system.

Consider a single-input and single-output (SISO) system or a system with 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 the 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 set the maximal deviation d_{\max} based on the constraints on system outputs:

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

where the symbol $|\cdot|$, hereafter, denotes the element-wise absolute value, y_{\min} and y_{\max} are respectively lower and upper bound on the output variable in the deviation form, *i.e.*, zero (origin) 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:

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

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

Note that the reference value must be reachable from the operating range to ensure that $0 \leq \rho \leq 1$ holds. Otherwise, the interpolated control action would be the “extrapolation” leading to the loss of guarantees on the input or state constraints satisfaction, etc.

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 become sluggish.

Remark 2.3. In general, the parameter d_{\max} is a vector, as it depends on the size of the 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 as follows:

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

Note that the relations in Eqs. (7) and (8) operate with the absolute value of the reference. It is not taken into account whether the reference value changed upwards or downwards with respect to the system steady-state value placed in the origin, *i.e.*, whether the inequality $\Delta_{\text{ref}}(k) = y_{\text{ref}}(k) - y_{\text{ref}}(k-1) > 0$ holds or $\Delta_{\text{ref}}(k) < 0$. As many plants have nonlinear behavior with an asymmetric nature (different behavior when the process variable is rising or decreasing), the positivity or negativity of the reference change could be considered in the controller self-tuning procedure to improve the control performance.

3. Methodology

This section extends the ideas of self-tunable explicit MPC in order to improve 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 [27] suggested tuning based on the current reference value distance from the steady state. The aim is to compensate for the nonlinear behavior of a 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 include the size of the reference step change in 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 the control operation:

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

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 Δ_{\max} as the maximal possible reference step change. Analogously to the original approach, the maximal reference step can be set based on the general knowledge of the expected future reference values, *i.e.*, $\Delta_{\max} = \|\Delta_{\text{ref}}(k)\|_{\infty}, \forall k \geq 0$. Another option is to exploit the information about the system constraints and set the parameter Δ_{\max} according to Eq. (6).

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

In Eq. (9), it is suggested to increase the value of tuning parameter ρ with increasing value of reference step change. Note, in this work, the larger value of the tuning parameter leads to adding more weight on the penalty matrices associated with the upper boundary controller, see Eq. (4). If the opposite logic of controller tuning is requested, it is possible to adapt the tuning such that

$$R(k) = \rho(k) R_L + (1 - \rho(k)) R_U, \quad (10)$$

$$Q_I(k) = \rho(k) Q_{I,L} + (1 - \rho(k)) Q_{I,U}, \quad (11)$$

$$Q_y(k) = \rho(k) Q_{y,L} + (1 - \rho(k)) Q_{y,U}, \quad (12)$$

hold. This change leads to adding more weight to the lower boundary controller with the increasing value of the tuning parameter ρ .

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

This paper provides a further extension of the self-tuning method proposed in [27]. 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 $\tilde{\rho}$ is then utilized to scale the control input according to Eq. (5).

Definition 3.1 (Decision Function). For a given interval of tuning parameter ρ , $0 \leq \rho \leq 1$, let ρ_s , $0 < \rho_s < 1$ be a boundary value splitting the interval into two parts. Let $\gamma : \mathbb{R} \rightarrow \mathbb{R}$ be an arbitrary function such that $0 \leq \gamma \leq 1$ holds. Then the decision function γ is constructed to assign its value either $\gamma \leq \rho_s$ or $\gamma \geq \rho_s$.

Various decision functions γ can be considered. In this work, the decision functions according to Eqs. (8) and (9) are suggested, while Eq. (9) was implemented in the experimental case study.

Definition 3.2 (Scaling of the Tuning Parameter). Given the value of tuning parameter ρ , $0 \leq \rho \leq 1$, the splitting value of the tuning parameter interval ρ_s , $0 < \rho_s < 1$, and the value of the decision function γ , $0 \leq \gamma \leq 1$. Then the scaling of the tuning parameter $\tilde{\rho}$ is given by:

$$\tilde{\rho} = \begin{cases} \rho \rho_s & \text{if } \gamma \in \langle 0, \rho_s \rangle, \\ \rho (1 - \rho_s) + \rho_s & \text{else if } \gamma \in \langle \rho_s, 1 \rangle. \end{cases} \quad (13)$$

Remark 3.2. The introduction of splitting the tuning parameter $\tilde{\rho}$ into the tuning intervals in (13) is not limited only to two intervals. If the nature of the controlled plant would benefit from splitting the operating range into more intervals, e.g., when the plant operates in the multiple steady-states values, then these intervals are simply determined by the corresponding values of $\rho_{s,i}$ for each part of the interval. Next, the tuning rules in (13) are adopted in an analogous way.

The following outcomes result from Eq. (13).

Lemma 3.2.1. Given control law in (3), its approximation given by the convex combination in (5), and given scaled tuning parameter $\tilde{\rho}$ according to Definition 3.2. Then the control action approximated into the form:

$$u(k) = (1 - \tilde{\rho}(k)) u_L(k) + \tilde{\rho}(k) u_U(k), \quad (14)$$

preserves the closed-loop system stability and recursive feasibility of the original control law in (3).

Proof. It has been proven [23] that for the asymptotic stable and recursive feasible pair of control inputs (u_L, u_U) , the approximated control law in (3) preserves these properties for any ρ satisfying $0 \leq \rho \leq 1$, see Theorem 3.6 in [23]. It remains to prove that for any value of the scaled tuning parameter $\tilde{\rho}$ according to Definition 3.2 the same results hold. The rest of the proof of Lemma 3.2.1 consists of two parts corresponding to each particular rule in (13).

First, it is proved that Lemma 3.2.1 holds for any $\gamma \leq \rho_s$. Substituting a lower bound $\rho = 0$ into (13) leads to $\tilde{\rho} = 0$. For the upper bound value of $\rho = 1$, from (13) holds $\tilde{\rho} = \rho_s < 1$. Next, for any value $0 < \rho < 1$ evaluation of the linear rule in (13) leads to the convex combination, i.e., $0 < \tilde{\rho} < \rho_s$ holds. Therefore, any value of $\tilde{\rho}$ satisfies $0 \leq \tilde{\rho} \leq \rho_s < 1$. As a consequence, according to the Theorem 3.6 in [23], the asymptotic stability and recursive feasibility of the control law in (14) are preserved.

Secondly, it is proved that Lemma 3.2.1 holds also for any $\gamma \geq \rho_s$. Substituting a lower bound $\rho = 0$ into (13) leads to $\tilde{\rho} = \rho_s$. For the upper bound value of $\rho = 1$, from (13) holds $\tilde{\rho} = 1$. Next, for any value $0 < \rho < 1$ evaluation of the linear rule in (13) leads to the convex combination, i.e., $\rho_s < \tilde{\rho} < 1$ holds. Therefore, any value of $\tilde{\rho}$ satisfies $\rho_s \leq \tilde{\rho} \leq 1$. As a consequence, according to the Theorem 3.6 in [23], the asymptotic stability and recursive feasibility of the control law in (14) are preserved. \square

Remark 3.3. The Lemma 3.2.1 can be extended subject to the multiple intervals in an analogous way following Remark 3.2.

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.

Remark 3.4. 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.

For a detailed insight into the proposed control technique, the procedure of self-tuning evaluation is depicted in Fig. 1.

From the point of computational complexity, the proposed tuning procedure does not lead to any significantly demanding mathematical operations. Simple algebraic operations in Eqs. (9) and (13) are evaluated. Note, the overall control strategy still comes with the cost of storing and evaluating two explicit controllers.

4. Results and discussion

In this section, the results of the proposed self-tuning method are analyzed by an experimental implementation. The self-tuning strategy utilizes tuning parameter calculation 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 [30], see Fig. 2. The schematic of the plant is depicted in Fig. 3. The heat exchanger is 90 mm wide, 103 mm long, and 160 mm high. The heat exchange is performed between the cold medium (water) and the hot medium (water). The cold medium as well as the heating medium are transported to the heat exchanger by two peristaltic pumps with flexible tubing from silicon rubber. The flow rate of the cold medium is constant, while the aim of control is to track the reference value of the outlet cold medium temperature. Therefore, the controlled variable is the cold medium temperature T at the outlet of the heat exchanger. The inlet cold medium temperature was constant during the whole control, i.e., $T_C = 19$, °C. The temperature of the heated cold medium in the outlet stream was measured by the type K thermocouple. The associated manipulated variable is the voltage U corresponding to the power of the pump feeding the heat exchanger by the hot medium. The voltage is within the range of [0–5] V normalized into the relative values in percentage. The maximal voltage 5 V or 100% corresponds to volumetric flow rate 11.5 ml s⁻¹. For further technical specifications of the laboratory heat exchanger, the reader is referred to [30]. As heat exchange is a nonlinear and asymmetric process [10], this heat exchanger represents a suitable candidate for the presented controller tuning strategy. The corresponding illustrative scheme of the implemented closed-loop control setup is in Fig. 6, where the ‘‘Self-tuning’’ block substitutes the more detailed scheme of the tuning procedure in Fig. 1.

The system was identified by experimental identification. The aim was to work with linear nominal model in MPC optimization problem to decrease the numerical complexity. To avoid plant-model mismatch in order to ensure offset-free tracking, either disturbance observer or built-in integrator (Eq. (1)) can be employed. Due to the ease of implementation, in this work, the built-in integrator was considered. The system was identified based on several measured step responses. The step changes were performed in the whole range of admissible values of manipulated variable and every step response was identified by transfer function. It was possible to identify every step response as a first-order system, while the nominal gain and time constant are respectively $K = 0.24$ °C and $\tau = 5.7$ s. Finally, the nominal transfer function was converted to the state-space model. The matrices of the discrete-time state-space model of the plant are

$$A = [0.839], \quad B = [0.039], \quad C = [1], \quad (15a)$$

considering the sampling time $T_s = 1$ s. To respect the physical limitations of the operating conditions, the constraints are considered in the terms of control inputs

$$-15\% \leq u \leq 65\%, \quad (16)$$

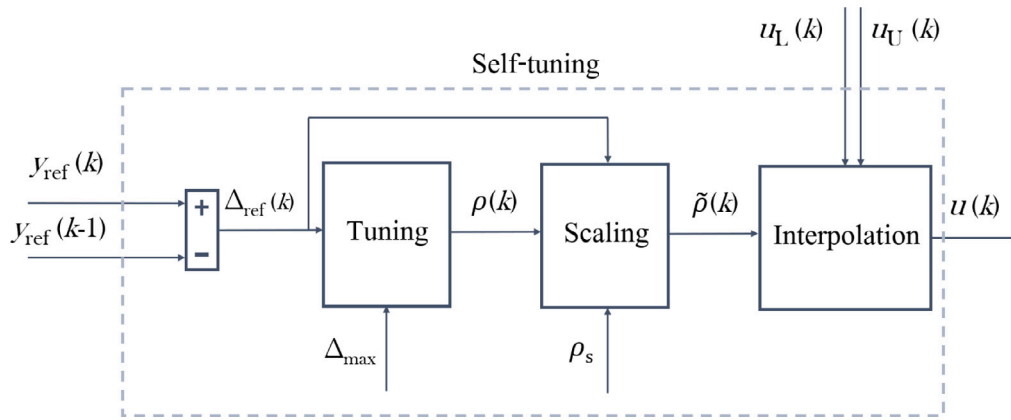


Fig. 1. Scheme of the self-tuning control evaluation.

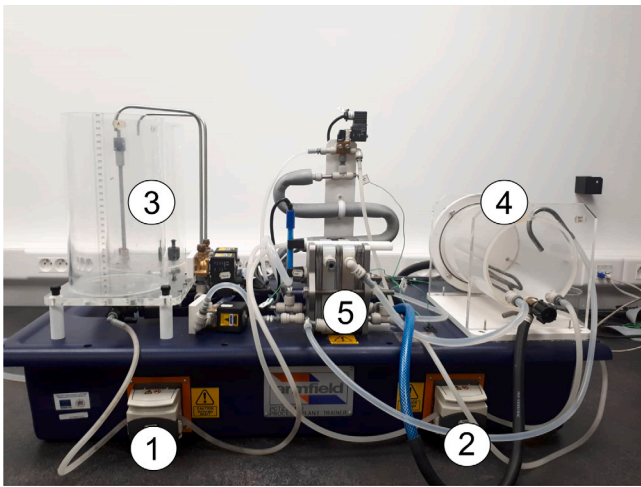


Fig. 2. Laboratory heat exchanger Armfield Process Plant Trainer PCT23: cold medium pump (1), heating medium pump (2), cold medium tanks (3), heater for heating medium (4), heat exchanger (5).

where the variable u represents the control inputs in the deviation form. The values of the heated cold medium temperature and voltage of the heating medium pump corresponding to zero steady states are respectively $T^s = 35, ^\circ\text{C}$ and $U^s = 35\%$. Therefore, the physical constraints on the manipulated variable are actually

$$20\% \leq U \leq 100\%. \quad (17)$$

As the controlled system is naturally stable even if the maximal or minimal value of the manipulated variable is constantly applied, the constraints on the controlled variable in Eq. (1e) could be omitted. On the other hand, unbounded states/outputs lead to higher memory consumption, because covering the whole possible range of parameters requires more critical regions. Therefore, the “redundant” constraints on the system outputs were included in order to reduce the number of critical regions and the overall memory footprint of the explicit controllers. The output constraints were set as:

$$-15\text{ }^\circ\text{C} \leq y \leq 20\text{ }^\circ\text{C}. \quad (18)$$

The constraints in Eq. (18) are equal to physical temperature as follows:

$$20\text{ }^\circ\text{C} \leq T \leq 55\text{ }^\circ\text{C}, \quad (19)$$

which corresponds to the range of temperature values which are achievable in the considered laboratory conditions and setup.

The penalty matrices of the problem in Eq. (1) were systematically tuned, and the corresponding control setup was implemented on the laboratory heat exchanger for each setup of the considered explicit MPC controllers. First, the tuning procedure aimed to determine which penalty matrix is the most suitable for real-time tuning. The most relevant was the penalty matrix Q_y as the tuning is directly associated with a reference value, which takes place in the calculation of the tuning factor ρ . Moreover, the tuning of Q_y preserved a satisfactory control performance. Next, the boundary values of the tunable matrix Q_y were tuned until the following limit values were determined based on the measured closed-loop control data: $Q_{y,L} = 100$ and $Q_{y,U} = 1000$. The built-in integrator was penalized with the fixed penalty matrix $Q_I = 1$ and the control input with the fixed penalty matrix $R = 10$. The prediction horizon N was set to 20 control steps. The explicit model predictive controllers were constructed in MATLAB R2020b using the Multi-Parametric Toolbox 3 [31].

The explicit MPC corresponding to the penalty matrix $Q_{y,U}$ contains 1 680 critical regions, and the explicit MPC with the penalty matrix $Q_{y,L}$ contains 409 critical regions. The corresponding polytopic partitions can be seen in Fig. 4 for the upper boundary controller and Fig. 5 for the lower boundary controller.

The designed explicit model predictive controllers were implemented to track a time-varying PWC reference. For the initial 200 s, the reference temperature was the 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. Specifically, the reference temperature values were $T_{\text{ref}} = 35, ^\circ\text{C}$, $45, ^\circ\text{C}$, $50, ^\circ\text{C}$, $45, ^\circ\text{C}$, $35, ^\circ\text{C}$.

Besides the control design of two boundary explicit MPCs, it was necessary to keep the temperature of the heating medium constant. The heating medium was transported back to the heater after leaving the heat exchanger, i.e., the volume of the heating medium was recycled during the whole operation. The temperature of the heating medium was maintained on the value $70, ^\circ\text{C}$ with a simple proportional controller with proportional gain $P = 20$. The control input from the proportional controller was the electric power which could acquire the values in the range $[0-2]$ kW and was also normalized to percentage.

The control profiles generated for both considered boundary control setups are compared in Fig. 9 for the controlled variable, and in Fig. 10 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 be truncated afterward.

An interesting phenomenon can be observed while tracking the third reference value, i.e., $T_{\text{ref}} = 50, ^\circ\text{C}$. Although the steady-state values of temperature have the same value, the values of the manipulated variable are different. To check the correctness of the results, the

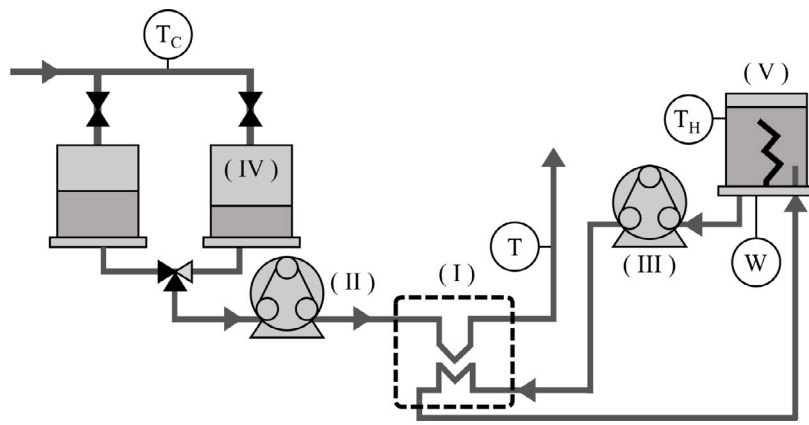


Fig. 3. Scheme of Armfield PCT23. Heat exchanger (I), peristaltic pump for cold medium (II), peristaltic pump for heating medium (III), tank for cold medium (IV), heater for heating medium (V), temperature sensors (T – controlled temperature, T_c – cold outlet cold medium temperature, T_H – heating medium temperature), and electric power for maintaining the temperature of the heating medium (W).

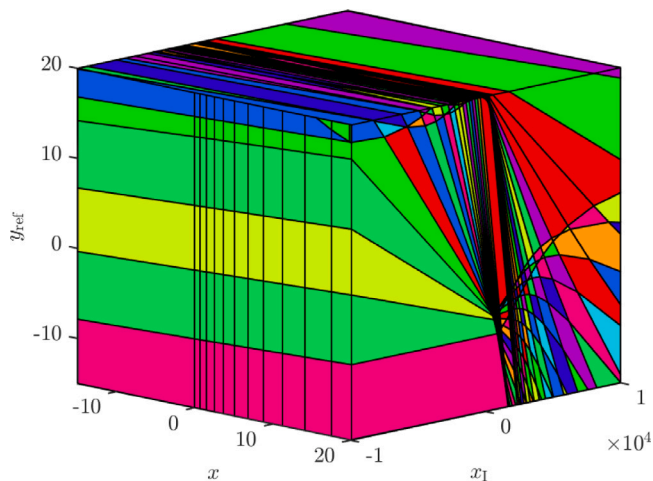


Fig. 4. Polytopic partition of the upper boundary explicit MPC.

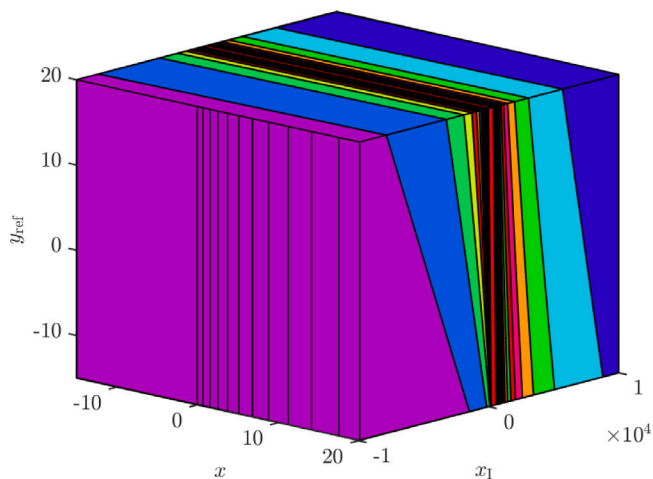


Fig. 5. Polytopic partition of the lower boundary explicit MPC.

measurements were performed multiple times and led to the same behavior. Also, the inlet temperatures of the cold and heating medium were checked to exclude the effect of a disturbance. Regarding the temperature of the cold medium, due to the limited hardware interface, it was not possible to measure the data continuously, store them, and plot the trajectory in a Figure. Nevertheless, the temperature of the cold medium was manually checked multiple times during the experiment and was constant.

Regarding the temperature of the heating medium, the corresponding trajectories of the temperature can be seen in Fig. 11, and the electric power, *i.e.*, the corresponding manipulated variable, can be seen in Fig. 12. Note that the legends correspond to the specific setup of MPC, but the temperature of the heating medium was controlled with a simple P controller with the same proportional gain in every control scenario.

It can be seen that the temperature of the heating medium remains relatively constant during the whole control, except for the undershoots in the scenario with upper boundary MPC, *i.e.*, blue trajectory. The undershoots can be easily associated with the trajectory of the voltage on the pump dosing the heating medium (and ultimately the heating medium flow rate). As the upper boundary MPC calculated “aggressive” control inputs, the increased flow rate of the heating medium led to a slight decline in the heating medium temperature. After approximately 100 s, the heating medium warmed up to the reference value, *i.e.*, $T_{H,ref} = 70, ^\circ\text{C}$ and remained constant within the accuracy of the temperature sensor. It can be seen that although the temperature of the heating medium is constant and identical for all control scenarios (MPC setups), the value of the voltage on the pump dosing the heating medium is not the same when tracking the temperature $T_{ref} = 50, ^\circ\text{C}$. Therefore, the same conditions were fulfilled for all control scenarios.

The reason for this behavior could be explained by the peak of the manipulated variable associated with the upper boundary controller at time 800 s, see Fig. 10, blue. After approximately 100 s, the value of the manipulated variable dropped and settled at a value lower than the value associated with the lower boundary controller, see Fig. 10, red. This is a consequence of the heat accumulated inside the heat exchanger plates, and therefore, less heating medium was necessary to heat the cold medium. This phenomenon does not happen when tracking the reference value $T_{ref} = 45, ^\circ\text{C}$, which originates in the nonlinear nature of the heat transfer process. When working in a higher temperature range, the gain of the heat transfer process decreases, and the sensitivity to changes in the heating medium flow is lower. Therefore, even different flow rates of the heating medium lead to the same temperature at the outlet.

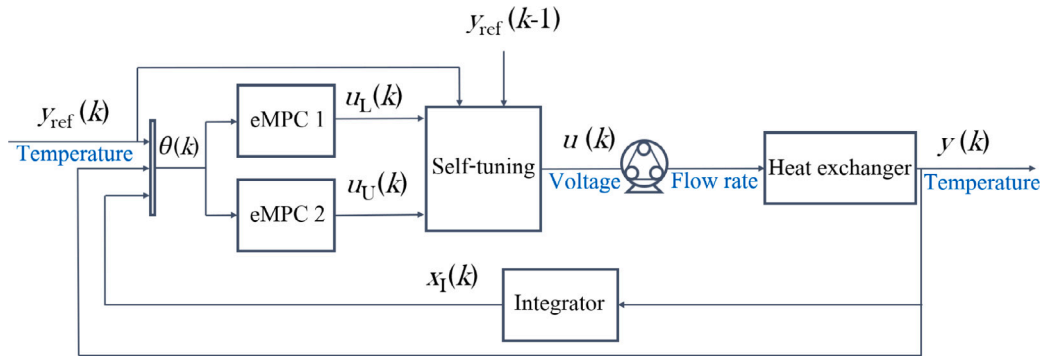


Fig. 6. Scheme of the implemented closed-loop control setup, where “eMPC” denotes explicit MPC.

The trajectories in Fig. 9 show the asymmetric nature of controlling the plant of plate heat exchanger mainly when observing the overshoots and undershoots. When applying the control inputs associated with the lower boundary penalty matrix $Q_{y,L}$ in Eq. (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 $Q_{y,U}$ in Eq. (1a), the undershoots are negligible, but significant overshoots can be seen when tracking the reference upwards, see Fig. 9, blue.

These main experimental observations established the base for the strategy of controller self-tuning. The strategy follows the ideas summarized in Section 3. Utilizing the nature of the boundary controller with the penalty matrix $Q_{y,L}$ is preferred when the reference changes upwards. Therefore, in these operating conditions, the tuning factor is scaled in the first part of the whole interval, *i.e.*, closer to the lower bound. On the contrary, tuning the controller closer to the upper boundary controller associated with $Q_{y,U}$ is preferred for negative reference step changes. Therefore, in these operating conditions, the tuning factor is scaled above the splitting value ρ_s , *i.e.*, closer to the upper bound. The splitting value of the tuning parameter was chosen simply in the middle of the interval, *i.e.*, $\rho_s = 0.5$. The remaining parameter that needed to be set was the maximal admissible size of the reference step change Δ_{max} , which was determined to 15°C as the investigated range of controlled temperature was $[35\text{--}50]^\circ\text{C}$. Based on the aforementioned parameters and real-time information about the current reference change, the tuning factor was updated during control. The evolution of the scaled tuning factor $\tilde{\rho}$ can be seen in Fig. 7. When the positive reference changes are tracked, the tuning factor is scaled below the splitting value ρ_s . On the contrary, when the reference changes are negative, the tuning factor is scaled above the splitting value ρ_s .

The setup of the tuning factor can be associated with tuning of the penalty matrix Q_y according to Eq. (4c). The evolution of the penalty matrix Q_y during control is depicted in Fig. 8. Note, the penalty matrix evolution in Fig. 8 does not correspond to tuning of the optimal MPC, but serves for a deeper insight into the association of the interpolated control inputs with the optimal explicit MPC setup.

The control input is applied to the system each second, so there is a possible concern regarding the speed at which two explicit MPCs are evaluated. By analyzing the computational speed, it was concluded that the approximate control input can be generated in an average time of 0.01 s, which is 100 times faster than the sampling time.

The control results of the self-tunable technique compared to the boundary controllers can be seen in Fig. 9 for the controlled variable, and in Fig. 10 for the manipulated variable. 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

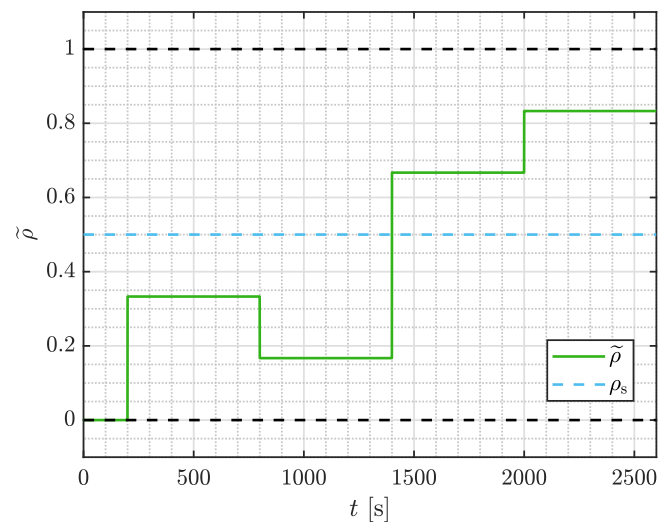


Fig. 7. Evolution of the scaled tuning factor $\tilde{\rho}$ during real-time control. When tracking positive reference changes, the tuning factor is scaled below the splitting value ρ_s (200–1400 s). On the contrary, when the reference changes are negative, the tuning factor is scaled above the splitting value ρ_s (1400–2600 s).

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 $Q_{y,U}$. The similarity with the boundary controllers can be seen also on the manipulated variable 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. Table 1 summarizes the evaluated control performance criteria computed for the two boundary controllers and the self-tuned controller. The control performance is evaluated for each reference step change separately. The considered quality criteria are: sum-of-squared control error SSE, maximal overshoot/undershoot σ_{max} and the settling time t_e for 5%-neighborhood of the reference temperature T_{ref} . To provide 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, the real-time self-tuning of the explicit MPC controller helped to improve two to three criteria when tracking each reference value. The relative improvement in the percentage, denoted by δ , of using the self-tunable controller is summarized in Table 2

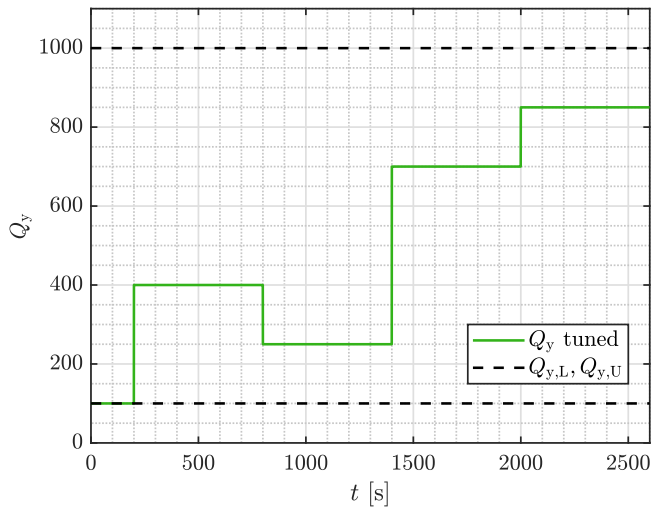


Fig. 8. Evolution of the penalty matrix Q_y during real-time control. When tracking positive reference changes, the controller is tuned to operate closer to the lower boundary matrix $Q_{y,L}$ (200–1400 s). On the contrary, when the reference changes are negative, the controller is tuned to operate closer to the lower boundary matrix $Q_{y,U}$ (1400–2600 s).

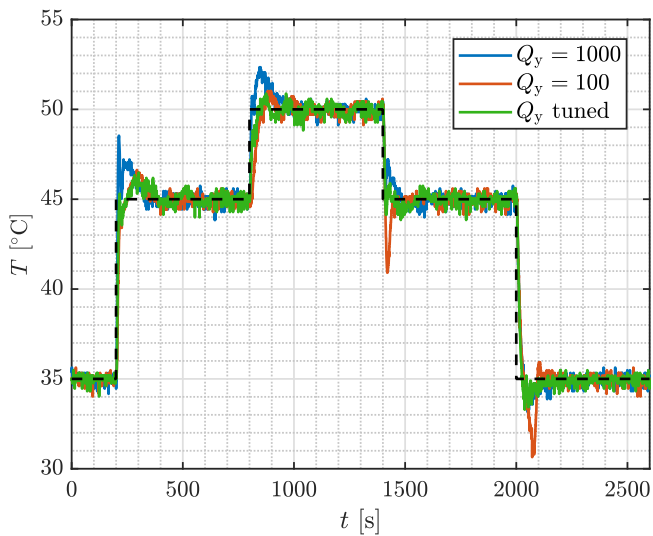


Fig. 9. Explicit MPC: Controlled variable trajectory for two boundary controllers and the tuned one. The solid lines represent the controlled temperature T and the dashed line represents the reference value.

for each reference step change separately. The values were computed as the difference between two criteria values corresponding to the optimal and self-tunable MPC, referred to the self-tunable MPC. The negative numbers represent deterioration of the specific performance criterion in the corresponding reference tracking.

Compared to the considered non-self-tunable controllers, the control trajectories and the evaluated quality criteria confirmed the improved control performance for the reference tracking control problem of the heat exchanger with the non-linear and asymmetric behavior. Implementing a self-tunable explicit MPC controller leads to improved control performance in the most analyzed quality criteria, see Table 2. In average, the control performance criteria improved compared to the

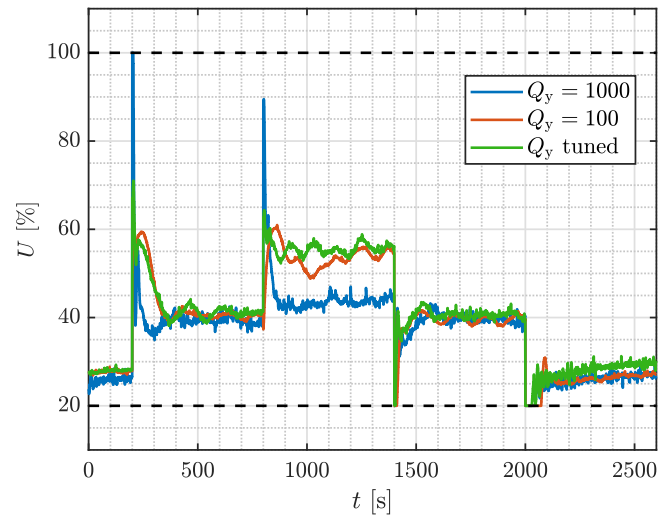


Fig. 10. Explicit MPC: Manipulated variable trajectory for two boundary controllers and the tuned one. The solid lines represent the voltage U and the dashed lines represent the constraints.

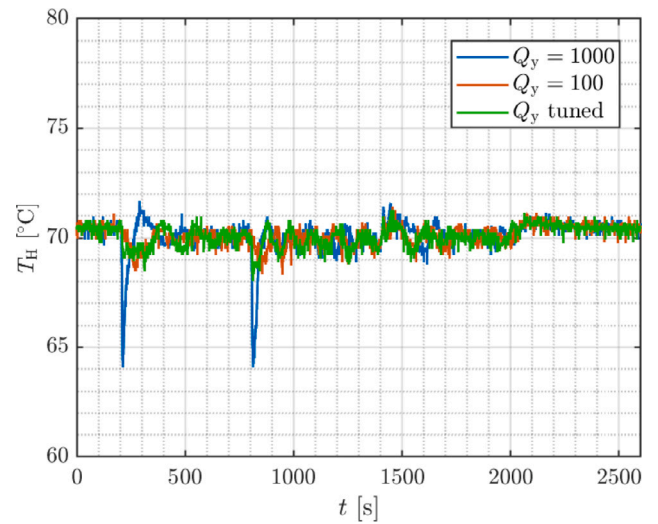


Fig. 11. Auxiliary P controller: The trajectory of heating medium temperature control.

Table 1

Control performance criteria.

Reference step change	Q_y	SSE [$^{\circ}\text{C}^2 \text{s}$]	σ_{\max} [%]	t_{ϵ} [s]
35 $^{\circ}\text{C}$ \rightarrow 45 $^{\circ}\text{C}$	1000	714	33.5	16.5
	100	867	16.7	12.5
	self-tuned	678	15.2	9.5
45 $^{\circ}\text{C}$ \rightarrow 50 $^{\circ}\text{C}$	1000	365	47.2	5
	100	606	23.3	26.5
	self-tuned	248	19.1	9.5
50 $^{\circ}\text{C}$ \rightarrow 45 $^{\circ}\text{C}$	1000	245	18.9	6.5
	100	398	79.6	31
	self-tuned	186	24.6	6.5
45 $^{\circ}\text{C}$ \rightarrow 35 $^{\circ}\text{C}$	1000	1024	18.4	22.5
	100	1402	41.9	90
	self-tuned	967	16.5	18.5

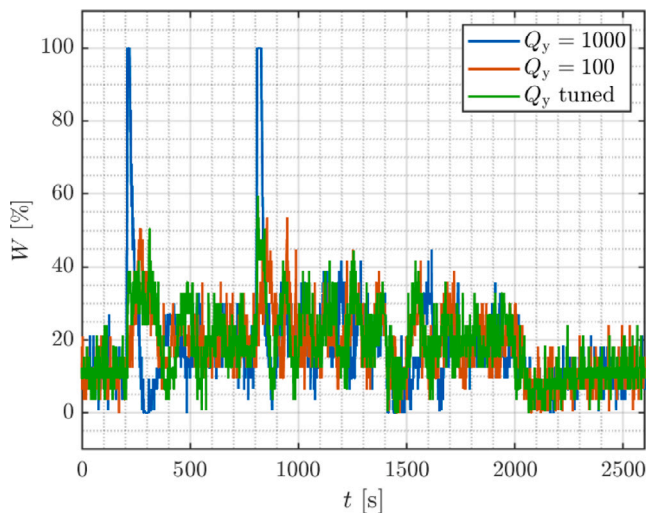


Fig. 12. Auxiliary P controller: The trajectory of electric power controlling the heating medium temperature.

Table 2

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

	Comparison with Q_y setup	δ SSE [%]	$\delta\sigma_{\max}$ [%]	δt_e [%]
35 °C → 45 °C	1000	5	121	74
	100	28	10	32
45 °C → 50 °C	1000	47	147	-47
	100	144	22	179
50 °C → 45 °C	1000	32	-23	0
	100	114	224	377
45 °C → 35 °C	1000	6	12	22
	100	45	154	386
Average	1000	23	64	12
	100	83	102	244

upper and lower boundary MPC respectively as follows: the squared-error-based criterion (SSE) reduced by 23% and 83%, the maximal overshoot/undershoot σ_{\max} reduced by 64% and 102%, and the settling time t_e reduced by 12% and 244%.

In general, utilizing the proposed controller with a scalable aggressiveness according to the operating conditions leads to higher accuracy (lower SSE), lower value of the overshoots (reduced σ_{\max}), and faster achieving the reference value (decreased t_e).

Obviously, if there exists a well-tuned “universal” controller that satisfies the requirements on the control performance in the whole range of the considered operating conditions, then the implementation of the self-tuning procedure is out of scope for such control application. Nevertheless, in numerous practical situations, using only one controller with a constant setup leads to poor or just “satisfactory” control results, *i.e.*, the reference value is achieved, but with worse control performance, *e.g.*, leading to high overshoots or settling times. When working on our laboratory case study, a set of different setups of penalty matrices was investigated. In every control scenario, the setup was beneficial only in some working conditions (tracking the reference upwards or downwards). Therefore, the closed-loop control performance is improved by introducing the benefits of the self-tuning method based on the two boundary MPC controllers.

Note that this strategy relies on a proper design of the two boundary controllers. In case a non-negligible disturbance occurs, both boundary controllers should be able to solve a disturbance rejection problem as the final value of the manipulated variables is interpolated between

them. To address the impact of the disturbances directly in constructing the MPC controller design, a robust MPC strategy should be considered, *e.g.*, see [32]. Any robust MPC design method leads to conservative control actions as some portion of the performance is sacrificed to compensate for the impact of the disturbances. Nevertheless, if it is possible to obtain the explicit (multi-parametric) solution of the robust explicit MPC offline, then the same self-tuning procedure is applicable to interpolate between the control actions from the robust controllers.

5. Conclusions

This paper deals with the experimental implementation and analysis of the novel self-tunable approximated explicit model predictive control method and provides a strategy for an effective self-tuning controller design. Based on the current value of the piece-wise constant reference, the tuning parameter is scaled using linear interpolation. The previously published work related to the self-tunable explicit MPC suggested tuning based on the distance of the reference value from the system steady-state value. This paper presents a novel perspective idea of self-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 whenever the reference changes. The tuning value is calculated as the ratio between the size of the reference change and the maximal admissible size of the reference change, which is specified before operation. Another novel contribution addresses the challenging control problem of asymmetric system behavior by splitting the interval of the self-tuning parameter into two ranges, while both intervals are assigned to different operating conditions. The proposed method is implemented on a laboratory-scaled heat exchanger with nonlinear and asymmetric behavior. The asymmetry makes the plant a suitable candidate to analyze the benefits of 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 selected 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 properly investigate the control results, the control performance was also judged quantitatively using a set of quality criteria. The self-tunable control approach outperformed the conventional control strategy handling just a single controller, *i.e.*, non-tunable controller. In average, the control performance criteria improved compared to the upper and lower boundary MPC respectively as follows: the squared-error-based criterion (SSE) reduced by 23% and 83%, the maximal overshoot/undershoot σ_{\max} reduced by 64% and 102%, and the settling time t_e reduced by 12% and 244%.

The approach of the self-tunable technique was successfully implemented on a SISO system but can also be extended to multivariable systems by utilizing only a single value of the tuning parameter ρ to interpolate the values of every control input. It is suggested in Eq. (8) that the tuning parameter ρ can be calculated as the maximal value of all the tuning parameters computed for every output reference. However, this is straightforward to implement only for decoupled systems. If there are strong interactions between the system states, the self-tunable technique is challenging to design. In such a case, it is necessary to include expert knowledge about the system state interactions, and the resulting value of the tuning parameter ρ could be computed, *e.g.*, as a weighted average of the individual tuning parameters.

Nomenclature

Symbols

A	system state matrix
\tilde{A}	augmented system state matrix
B	system input matrix
\tilde{B}	augmented system input matrix
C	system output matrix
\tilde{C}	augmented system output matrix
d_{\max}	maximal deviation from the steady-state value
F	slope of the affine control law
g	section of the affine control law
I	identity matrix
k	step of the prediction horizon
K	system gain, °C
N	prediction horizon
n_u	size of system inputs
n_y	size of system outputs
$n_{\tilde{x}}$	size of augmented system states
P	proportional gain of proportional controller
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_ϵ	settling time, s
T	temperature, °C
T_C	cold medium temperature, °C
T_H	heating medium temperature, °C
$T_{H,\text{ref}}$	heating medium reference temperature, °C
T_{ref}	reference 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
W	electric power, %
x	system states
\tilde{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
O	zero matrix

Greek letters

δ	relative improvement, %
Δ_{\max}	maximal size of the reference change, °C
Δ_{ref}	size of the reference change, °C
ρ	tuning factor
$\tilde{\rho}$	scaled tuning factor

ρ_s	splitting value of the tuning factor
σ_{\max}	maximal overshoot, %
τ	system time constant, s
θ	parameter of optimization problem
Θ	set of parameter values

Abbreviations

eMPC	explicit model predictive control
LTI	linear time-invariant (system)
LQR	linear-quadratic regulator
MPC	model predictive control
mp-QP	multi-parametric quadratic programming (problem)
PID	proportional–integral–derivative (controller)
PWA	piece-wise affine (function)
PWC	piece-wise constant (function)
QP	quadratic programming (problem)
SISO	single-input and single-output (system)
SSE	sum-of-squared error

CRedit authorship contribution statement

Lenka Galčíková: Writing – review & editing, Writing – original draft, Validation, Methodology, Investigation, Data curation. **Juraj Oravec:** Writing – review & editing, Validation, Supervision, Project administration, Methodology, Investigation, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors are unable or have chosen not to specify which data has been used.

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, and 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). The authors also acknowledge Petronela Belková for help with generating the experimental data.

References

- [1] M.M. Morato, J.E. Normey-Rico, O. Senam, Model predictive control design for linear parameter varying systems: A survey, *Annu. Rev. Control* 49 (2020) 64–80, <http://dx.doi.org/10.1016/j.arcontrol.2020.04.016>.
- [2] L. Zhi-Yong, P.S. Varbanov, J.J. Klemeš, J.Y. Yong, Recent developments in applied thermal engineering: Process integration, heat exchangers, enhanced heat transfer, solar thermal energy, combustion and high temperature processes and thermal process modelling, *Appl. Therm. Eng.* 105 (2016) 755–762, <http://dx.doi.org/10.1016/j.applthermaleng.2016.06.183>.
- [3] J. Klemeš, P. Varbanov, Heat transfer improvement, energy saving, management and pollution reduction, *Energy* 162 (2018) 267–271, <http://dx.doi.org/10.1016/j.energy.2018.08.014>.
- [4] W. Roetzel, X. Luo, D. Chen, Chapter 1 - heat exchangers and their networks: A state-of-the-art survey, in: W. Roetzel, X. Luo, D. Chen (Eds.), *Design and Operation of Heat Exchangers and their Networks*, Academic Press, 2020, pp. 1–12, <http://dx.doi.org/10.1016/B978-0-12-817894-2.00001-7>.
- [5] S. Aguel, Z. Meddeb, M.R. Jeday, Parametric study and modeling of cross-flow heat exchanger fouling in phosphoric acid concentration plant using artificial neural network, *J. Process Control* 84 (2019) 133–145, <http://dx.doi.org/10.1016/j.jprocont.2019.10.001>.

- [6] M. Traczymanski, M. Markowski, S. Alabrudzinski, K. Urbaniec, The influence of fouling on the dynamic behavior of PID-controlled heat exchangers, *Appl. Therm. Eng.* 109 (2016) 727–738, <http://dx.doi.org/10.1016/j.applthermaleng.2016.08.142>.
- [7] Y. Wang, S. You, W. Zheng, H. Zhang, X. Zheng, Q. Miao, State space model and robust control of plate heat exchanger for dynamic performance improvement, *Appl. Therm. Eng.* 128 (2018) 1588–1604, <http://dx.doi.org/10.1016/j.applthermaleng.2017.09.120>.
- [8] J. van Niekerk, J. le Roux, I. Craig, On-line automatic controller tuning of a multivariable grinding mill circuit using Bayesian optimisation, *J. Process Control* 128 (2023) 103008, <http://dx.doi.org/10.1016/j.jprocont.2023.103008>.
- [9] J. Mikleš, M. Fikar, *Process Modelling, Identification, and Control*, Springer, 2007.
- [10] B.G. Liptak, fourth ed., *Instrument Engineers' Handbook*, vol. 2: *Process Control and Optimization*, CRC Press, London, 2005, <http://dx.doi.org/10.1201/9781315219028>.
- [11] M. Morari, J. H. Lee, Model predictive control: Past, present and future, *Comput. Chem. Eng.* 23 (4) (1999) 667–682, [http://dx.doi.org/10.1016/S0098-1354\(98\)00301-9](http://dx.doi.org/10.1016/S0098-1354(98)00301-9).
- [12] C. Hajiyev, H. Soken, S. Vural, Linear quadratic regulator controller design, in: *State Estimation and Control for Low-Cost Unmanned Aerial Vehicles*, Springer, Cham, 2015, http://dx.doi.org/10.1007/978-3-319-16417-5_10.
- [13] J. Maciejowski, *Predictive Control with Constraints*, Prentice Hall, London, 2000.
- [14] J. Mattingley, Y. Wang, S. Boyd, Receding horizon control, *IEEE Control Syst. Mag.* 31 (3) (2011) 52–65, <http://dx.doi.org/10.1109/MCS.2011.940571>.
- [15] K.V. Vinaya, K. Ramkumar, V. Alagesan, Control of heat exchangers using model predictive controller, in: *IEEE-International Conference on Advances in Engineering, Science and Management, ICAESM -2012*, 2012, pp. 242–246.
- [16] J. Oravec, M. Bakošová, A. Mészáros, N. Míková, Experimental investigation of alternative robust model predictive control of a heat exchanger, *Appl. Therm. Eng.* 105 (2016) 774–782, <http://dx.doi.org/10.1016/j.applthermaleng.2016.05.046>.
- [17] A.H. González, D. Odloak, J.L. Marchetti, Predictive control applied to heat-exchanger networks, *Chem. Eng. Process.: Process Intensif.* 45 (8) (2006) 661–671, <http://dx.doi.org/10.1016/j.cep.2006.01.010>.
- [18] X. Wu, J. Chen, L. Xie, Fast economic nonlinear model predictive control strategy of organic rankine cycle for waste heat recovery: Simulation-based studies, *Energy* 180 (2019) 520–534, <http://dx.doi.org/10.1016/j.energy.2019.05.023>.
- [19] Z. Dong, Z. Zhang, Y. Dong, X. Huang, Multi-layer perception based model predictive control for the thermal power of nuclear superheated-steam supply systems, *Energy* 151 (2018) 116–125, <http://dx.doi.org/10.1016/j.energy.2018.03.046>.
- [20] A. Bemporad, M. Morari, V. Dua, E.N. Pistikopoulos, The explicit linear quadratic regulator for constrained systems, *Automatica* 38 (2002) 3–20, [http://dx.doi.org/10.1016/S0005-1098\(01\)00174-1](http://dx.doi.org/10.1016/S0005-1098(01)00174-1).
- [21] M. Baric, M. Baotic, M. Morari, On-line tuning of controllers for systems with constraints, in: *Proceedings of the 44th IEEE Conference on Decision and Control*, 2005, pp. 8288–8293, <http://dx.doi.org/10.1109/CDC.2005.1583504>.
- [22] M. Klaučo, M. Kvasnica, Towards on-line tunable explicit MPC using interpolation, in: *Preprints of the 6th IFAC Conference on Nonlinear Model Predictive Control, IFAC, Madison, Wisconsin, USA*, 2018.
- [23] J. Oravec, M. Klaučo, Real-time tunable approximated explicit MPC, *Automatica* 142 (2022) 110315, <http://dx.doi.org/10.1016/j.automatica.2022.110315>.
- [24] D. Mayne, J. Rawlings, C. Rao, P. Scokaert, Constrained model predictive control: Stability and optimality, *Automatica* 36 (6) (2000) 789–814, [http://dx.doi.org/10.1016/S0005-1098\(99\)00214-9](http://dx.doi.org/10.1016/S0005-1098(99)00214-9).
- [25] K. Kiš, M. Klaučo, A. Mészáros, Neural network controllers in chemical technologies, in: *2020 IEEE 15th International Conference of System of Systems Engineering, SoSE*, 2020, pp. 397–402, <http://dx.doi.org/10.1109/SoSE50414.2020.9130425>.
- [26] K. Kiš, P. Bakaráč, M. Klaučo, Nearly optimal tunable MPC strategies on embedded platforms, in: *18th IFAC Workshop on Control Applications of Optimization, IFAC-PapersOnline*, 2022, pp. 326–331, <http://dx.doi.org/10.1016/j.ifacol.2022.09.045>.
- [27] L. Galčíková, M. Horváthová, J. Oravec, M. Bakošová, Self-tunable approximated explicit model predictive control of a heat exchanger, *Chem. Eng. Trans.* 94 (94) (2022) 1015–1020, <http://dx.doi.org/10.3303/CET2294169>.
- [28] D. Di Ruscio, Model predictive control with integral action: A simple MPC algorithm, *Model. Identif. Control* 34 (3) (2013) 119–129, <http://dx.doi.org/10.4173/mic.2013.3.2>.
- [29] M. Klaučo, M. Kvasnica, *MPC-Based Reference Governors*, Springer, 2019, <http://dx.doi.org/10.1007/978-3-030-17405-7>.
- [30] *Introduction Manual, PCT23-MkII Process Plant Trainer*, Armfield, 2007.
- [31] M. Herceg, M. Kvasnica, C. Jones, M. Morari, Multi-parametric toolbox 3.0, in: *European Control Conference*, 2013, pp. 502–510, <http://dx.doi.org/10.23919/ECC.2013.6669862>.
- [32] G. Pannocchia, J.B. Rawlings, S.J. Wright, Conditions under which suboptimal nonlinear MPC is inherently robust, *Systems Control Lett.* 60 (9) (2011) 747–755, <http://dx.doi.org/10.1016/j.sysconle.2011.05.013>.



Fixed complexity solution of partial explicit MPC

Lenka Galčíková, Juraj Oravec*

Institute of Information Engineering, Automation, and Mathematics, Faculty of Chemical and Food Technology, Slovak University of Technology in Bratislava, Radlinského 9, SK812-37 Bratislava, Slovak Republic

ARTICLE INFO

Article history:

Received 25 May 2021

Revised 21 September 2021

Accepted 18 November 2021

Available online 23 November 2021

Keywords:

Parametric optimization

Fixed complexity solution

Explicit model predictive control

ABSTRACT

Solving large-scale optimization problems with numerous constraints and optimization variables is a challenging task. Partial explicit MPC enables solving the large-scale optimization problem efficiently. This paper pushes the idea of partial explicit MPC to the fixed complexity parametric solution. The idea is to replace the polytopic critical regions that have a variable number of halfspaces with the maximal volume inner approximation based on the Chebyshev balls. As the approximation has a fixed and known structure, the memory footprint of the parametric solution is also fixed and known in advance, without the necessity to solve the large-scale optimization problem. This valuable property enables scaling the solution size a priori to meet the requirements of the hardware, where the MPC controller will be installed. The proposed method also dramatically reduced the memory burden of the partial explicit solution. Moreover, the proposed method improves the accuracy of the initialization of the hot-start procedure.

© 2021 Elsevier Ltd. All rights reserved.

1. Introduction

In the past three decades, model predictive control (MPC) became a widely used control strategy due to its many practical benefits, see, e.g. Morari and Lee (1999), Darby and Nikolaou (2012), Morato et al. (2020), and references therein. As MPC is a receding horizon control strategy, the optimization of control action is performed in each control step. The possibility to include constraints on the input, output, or state variable is nonnegligible. These properties make MPC an attractive way of process control compared to the well-known proportional-integral-derivative (PID) controllers for Single-Input Single-Output (SISO) systems or linear-quadratic (LQ) optimal controllers for Multiple-Input Multiple-Output (MIMO) systems. As a consequence, some form of MPC-based control is present in approximately 90% of industrial implementation of multivariable control, see (Lu, 2015), Qin and Badgwell (2003). However, the necessity to solve an optimization problem in each control step is a very challenging task due to the computational complexity. The challenges of the real-time MPC applications include complex systems with many constraints, e.g., robust and stochastic MPC design (Mayne, 2016), distributed control (Li and Swartz, 2019), problems with a long prediction hori-

zon and a high number of states and control actions (Darby et al., 2011).

One of the ways to implement MPC despite its demands on real-time computational complexity is an explicit solution using multiparametric programming, see, e.g., Bemporad et al. (2002), Burnak et al. (2019). The essence of the explicit model predictive control lies in the division of implementation into two separate phases. First, in the offline phase, the controller is constructed. Particularly, the optimization problem is computed for a predefined set of parameter values and the corresponding control law is determined. For a multiparametric quadratic problem (mpQP), the control law has the form of a piecewise affine (PWA) function over a polytopic partition composed of a set of the convex critical regions. Next, in the online phase, i.e., in the real-time control, the optimal control action is evaluated from the control law after identifying the critical region corresponding to the current measurement by solving the point location problem in some form of a lookup table.

Although the application range of explicit MPC is wide, two interconnected issues arise from its implementation: memory consumption and runtime effort. One of the possibilities of memory burden reduction is a regionless explicit model predictive control presented in Kvasnica et al. (2015). The authors showed that the geometrical construction and storage of the critical regions are not required. Instead, the active sets are considered, which provides significant savings in memory preserving the optimal solution. Other efficient constructions of the ex-

* Corresponding author.

E-mail addresses: lenka.galcikova@stuba.sk (L. Galčíková), juraj.oravec@stuba.sk (J. Oravec).

explicit partition using the dynamic programming were introduced in Mönnigmann (2019), Mitze and Mönnigmann (2020). The memory savings were achieved in Kvasnica and Fikar (2012), using the clipping function eliminating the number of regions of the PWA function over which the control law attains a saturated value. The utilization of the large set of critical regions evaluating the saturated control law is removed using the polynomial separator function in Kvasnica et al. (2013). The polynomial separator was replaced by various convex sets in Oravec et al. (2013).

The complexity reduction techniques do not target only memory footprint, but also accelerating the evaluation of the optimal control action in the online phase. One of the methods speeding up the online phase was suggested in Holaza et al. (2020), where the critical regions are sorted based on the minimal or maximal value of the corresponding value function. Using the proposed *smart* order, real-time control is significantly accelerated on average. Another technique leading to the decreased computational effort was an online removal of inactive constraints introduced in Jost et al. (2017). In Kvasnica et al. (2019), the online runtime is reduced by simplifying the point location problem without sacrificing closed-loop performance. The irrelevant critical regions are removed using the reachability analysis.

Several works bridge the gap between optimization-less real-time implementation of *explicit* MPC and *implicit* (non-explicit) MPC suitable for large-scale systems.

Many later works were inspired by Ferreau et al. (2008), where the online solution of the MPC problem was accelerated using the warm-start strategy based on the knowledge of the optimal active set from the previous control step. An efficient approach of semi-explicit MPC was presented in Goebel and Allgöwer (2017). This method is based on the offline state-dependent parametrization of the optimization variables using the tailored subspace clustering algorithm and the training data consisting of the MPC optimization problem solutions. A semi-explicit approach was introduced into the move-blocking-based MPC design in Son et al. (2020), where the time-varying blocking structure also guarantees the recursive feasibility and closed-loop system stability. Learning approximate semi-explicit MPC for a hybrid system was designed in Masti et al. (2020). In Zeilinger et al. (2011), the real-time sub-optimal MPC was designed combining the approximated explicit solution of the MPC problem used for warm-start of the active set method.

The trade-off between the benefits and limitations of the explicit MPC implementation for the large-scale MPC problems is well-balanced in Katz and Pistikopoulos (2020). This work presents a novel perspective concept of a *partial* multiparametric solution which places it on the road between *explicit* MPC and *implicit* MPC. Without loss of optimality, closed-loop system stability and recursive feasibility of the large-scale MPC problem, the partial multiparametric solution utilized in the framework of the explicit MPC improves initialization of the *hot-start* strategy for the real-time implementation.

First, in the offline phase, a partial solution of an explicit MPC optimization problem is evaluated. The positions of the critical regions are represented by the centers of the Chebyshev balls. In the online phase, when the measurement is obtained, the critical region with the nearest center of the Chebyshev ball is identified. This critical region is used for the hot-start strategy to initialize solving the large-scale optimization problem.

The main contribution of this work is to push the idea towards the *fixed complexity* parametric solution of the partial explicit MPC. Throughout this paper, the term “fixed complexity” denotes that the size of the memory footprint necessary to store the parametric solution is determined in advance, i.e., before solving the multiparametric optimization problem. Inspired by the ideas presented in Katz and Pistikopoulos (2020), the crucial idea of this paper is

not to store the polytopic critical region, but only its maximal volume inner approximation using the Chebyshev ball, i.e., data defining its center and radius. Just these data will be used in the online phase for the hot-start strategy. As the critical polytopic regions do not have the same number of halfspaces, the data size needed to be stored is not known in advance. On the other hand, storing the Chebyshev ball approximation provide us with fixing the memory of each considered critical region. As a consequence, the partial solution of explicit MPC is fixed in advance. It enables scaling the size of the solution a priori without the necessity to solve an optimization problem.

Moreover, compared to Katz and Pistikopoulos (2020), the proposed method significantly reduces the memory consumption of the partial solution, as the storage of the Chebyshev balls, i.e., centers and radii, requires much lower memory, compared to the storage of the polytopes.

Finally, the proposed method improves also the initialization of the hot-start procedure for solving the large-scale optimization problem. In contrast to Katz and Pistikopoulos (2020), we do not consider only the centers of Chebyshev balls but also their radii, which provide much more accurate information to identify the nearest region for the hot-start strategy.

The paper is organized as follows: first, the preliminaries containing the concepts of explicit MPC and partial explicit MPC are introduced. In the following section, the novel method of designing the fixed complexity partial explicit MPC is presented. The next section analyses results together with a discussion focused on the main benefits of the proposed fixed complexity approach. Finally, the main conclusions are summarized.

2. Preliminaries

2.1. Explicit MPC

In this section, the concept of explicit MPC is briefly recalled. Let us consider the model predictive control problem given in the form of multiparametric quadratic programming

$$\min_U \frac{1}{2} U^T H U + \theta^T F U \quad (1a)$$

$$\text{s.t. } G U \leq E \theta + w, \quad (1b)$$

where $\theta \in \mathbb{R}^n$ is the vector of parameters. The vector of the optimization variable $U \in \mathbb{R}^m$ is the vector of the manipulated variable optimized for the whole prediction horizon N , i.e., $U^* = [u_0^*, \dots, u_{N-1}^*]^T$. Matrices $H \in \mathbb{R}^{m \times m} > 0$, $F \in \mathbb{R}^{n \times m}$, $G \in \mathbb{R}^{c \times m}$, $E \in \mathbb{R}^{c \times n}$, and vector $w \in \mathbb{R}^c$ define the problem data describing the system model and its limitations, and c represents the number of optimization problem constraints. Typically, the parameter θ defines the set of initial conditions of system states, for which the problem is solved in the offline phase. Further technical details are, e.g., in Bemporad et al. (2002).

Using the well-known transformation, the optimization problem (1) can be rewritten for

$$z = U + H^{-1} F^T \theta, \quad (2a)$$

$$S = E + G H^{-1} F^T, \quad (2b)$$

into the equivalent following form

$$\min_z \frac{1}{2} z^T H z \quad (3a)$$

$$\text{s.t. } G z \leq S \theta + w. \quad (3b)$$

Finally, the constraints of the optimization problem (3) can be divided as follows:

$$\min_z 1/2 z^T H z \quad (4a)$$

$$\text{s.t. } G_A z = S_A \theta + w_A, \quad (4b)$$

$$G_N z < S_N \theta + w_N, \quad (4c)$$

where \mathcal{A} denotes the rows of matrices G , S , and vector w where the equality holds, i.e., the constraints are active. On the contrary, \mathcal{N} denotes the inactive constraints. The index sets \mathcal{A} and \mathcal{N} are disjoint, i.e., $\mathcal{A} \cap \mathcal{N} = \emptyset$ and $\mathcal{A} \cup \mathcal{N} = \{1, \dots, c\}$. Further technical details are, e.g., in [Borrelli \(2017\)](#).

The result of the optimization problem in (4) is an affine relation between the optimization variable z and parameter θ as follows:

$$z^* = F(\mathcal{A})\theta + f(\mathcal{A}), \quad (5)$$

where the slope $F(\mathcal{A})$ and the section $f(\mathcal{A})$ are functions of the combinations of the active and inactive constraints

$$F(\mathcal{A}) = H^{-1} G_A^T (G_A H^{-1} G_A^T)^{-1} S_A, \quad (6a)$$

$$f(\mathcal{A}) = H^{-1} G_A^T (G_A H^{-1} G_A^T)^{-1} w_A. \quad (6b)$$

The subset of the parametric space, where the affine control law in (5) is optimal, is defined as a critical region \mathcal{R} . The critical region \mathcal{R} is closed and bounded convex polytope

$$\mathcal{R} = \{\theta \in \mathbb{R}^n \mid A\theta \leq b\}, \quad (7)$$

where

$$A = \begin{bmatrix} G_N F(\mathcal{A}) - S_N \\ (G_A H^{-1} G_A^T)^{-1} S_A \end{bmatrix}, \quad (8a)$$

$$b = \begin{bmatrix} w_N - G_N f(\mathcal{A}) \\ -(G_A H^{-1} G_A^T)^{-1} w_A \end{bmatrix}. \quad (8b)$$

We can see in (6) and (8), that every feasible combination of the active and inactive constraints defines a specific critical region \mathcal{R} with its corresponding affine control law in (5). Therefore, when the multiparametric QP in (4) is solved for the whole parametric space, one can obtain the complete piecewise affine control law defined over all regions \mathcal{R}_i , $i = 1, \dots, R_{\text{total}}$, i.e., over the *polytopic partition* given by $\cup_i^{R_{\text{total}}} \mathcal{R}_i$, where R_{total} denotes the total number of the generated critical regions. The more constraints the optimization problem has, the higher is the potential to form more combinations of the active sets, and the corresponding computational time necessary to solve the multiparametric optimization problem exponentially increases.

In the online phase of explicit MPC, the point location problem is solved. According to system state measurement θ , the corresponding critical region is located such that $\theta \in \mathcal{R}_i$ holds. When the corresponding critical region is detected, the associated affine control law is applied to implement the optimal control action u_0^* into the system. However, with increasing problem size, the number of critical regions is also rising. Then the point location problem becomes more complex, and the usage of the explicit solution is hardly tractable.

2.2. Partial explicit MPC

One of the perspective techniques on how to handle large-scale optimization problems with numerous active sets is a partial solution of the explicit MPC, see [Katz and Pistikopoulos \(2020\)](#). The main idea is to solve the problem of explicit MPC only for a particular set of initial points from the feasible domain, i.e., *feasible seeding points*. In [Katz and Pistikopoulos \(2020\)](#), the authors suggest

using the procedure of *random walks* to obtain a random set of initial conditions from the parameter space. Therefore, only a subset of all critical regions is determined and stored, i.e., \mathcal{R}_i , $i = 1, \dots, R$, where R denotes the number of the evaluated and stored critical regions. This method leads to decreased memory burden compared to the whole explicit solution, i.e., $R \ll R_{\text{total}}$. In addition, the computational complexity of the offline phase is significantly reduced, since it is not necessary to construct the full explicit solution of the given large-scale optimization problem.

On the other hand, since we do not construct all critical regions in the offline phase, it can often occur in the online phase that the current state measurement θ does not lay inside any of them, i.e., $\theta \notin \mathcal{R}_i$, $\forall i = 1, \dots, R$. In such cases, we need to solve the optimization problem in (3) to evaluate the optimal control action using (5). To initialize solving the large-scale optimization problem, a near critical region is utilized. This is a useful tool to streamline searching for the critical region where the measurement belongs. This procedure is called *hot-start* strategy ([Ferreau et al., 2008](#)). The hot-start strategy utilizes the previous parameter realization θ_i , its corresponding region \mathcal{R}_i , and the current parameter θ_j , to find the critical region \mathcal{R}_j such that $\theta_j \in \mathcal{R}_j$. Once the critical region or associated active set is identified, then the corresponding control law is determined to find the optimal control action. For a more detailed description of the hot-start procedure see [Katz and Pistikopoulos \(2020\)](#). Implementation of the hot-start strategy requires storing one point which lies inside every stored critical region, a so-called *feasible point*. In [Katz and Pistikopoulos \(2020\)](#), it is suggested to construct the Chebyshev ball inside each stored critical region and determine its center. The Chebyshev balls centers C_i are then stored along with the associated critical regions \mathcal{R}_i , $\forall i = 1, \dots, R$.

Besides the polytope data and Chebyshev ball center C_i , we store a binary vector \mathcal{I}_i defining the indices of active constraints \mathcal{A} of the optimization problem in (3) for each critical region \mathcal{R}_i , $\forall i = 1, \dots, R$. The associated vector \mathcal{I}_i is utilized for initializing the hot-start procedure in the online phase. This vector has a cardinality given by the number of constraints, i.e., $|G|$ in (3). Particularly, the elements of this vector represent the fixed-ordered indices of constraints and their binary values indicate, if the particular constraint is active or inactive, e.g., see [Mönnigmann \(2019\)](#).

Remark 2.1. (Storing the control law) As binary vectors \mathcal{I}_i , $i = 1, \dots, R$, defining the active sets are stored, the matrices $F(\mathcal{A})$ and $f(\mathcal{A})$ are recovered from the problem matrices H , G , w , and S to apply the optimal control action u_0^* . It is also possible to store the matrices $F(\mathcal{A})$ and $f(\mathcal{A})$, along with the polytopic regions \mathcal{R}_i , but it leads to an unnecessary increased memory burden.

Remark 2.2. (Number of critical regions) The random distribution of the feasible seeding points p_i , $i = 1, \dots, R$, that serve to initialize the partial parametric solution of the optimization problem in the offline phase could lead to multiple evaluations of the same critical region, i.e., $p_i \mapsto \mathcal{R}_i$, $p_j \mapsto \mathcal{R}_i$, for $p_i \neq p_j$. In such case, it is sufficient to store such a critical region just once. Then two options are: (i) accept the number of the unique critical regions lower than the number of the feasible seeding points, or (ii) insert a new random feasible seeding point until the required number of the unique critical regions is evaluated.

The procedure of the partial explicit MPC based on ([Katz and Pistikopoulos, 2020](#)) is described in the following algorithms, where [Algorithm 1](#) summarizes the steps of the offline phase, and [Algorithm 2](#) summarizes the steps of the online phase.

For large-scale optimization problems, it is necessary to investigate the memory burden. In the online phase, the controller utilizes the data H , G , w , and S of the optimization problem in (3), as well as the partial solution represented by matrices A_i , b_i defining

Algorithm 1 Offline phase of partial explicit MPC (Katz and Pistikopoulos, 2020).

Inputs: Set of R random feasible seeding points $\{p_1, p_2, \dots, p_R\}$
Outputs: Matrices A_i and b_i of critical regions \mathcal{R}_i in (3), centers of Chebyshev balls C_i , binary vectors \mathcal{I}_i , $i = 1, \dots, R$

```

1: for each  $p_i$  do:
2:   solve the QP in (3) for  $\theta \leftarrow p_i$ 
2:   find and store optimal active set  $\mathcal{I}_i$ 
3:   construct the critical region  $\mathcal{R}_i$  in (8)
4:   store the polytope matrices  $A_i, b_i$ 
5: for each  $\mathcal{R}_i$  do:
6:   construct Chebyshev ball  $\tilde{\mathcal{R}}_i$ 
7:   store Chebyshev ball center  $C_i$  of  $\tilde{\mathcal{R}}_i$ 

```

Algorithm 2 Online phase of partial explicit MPC (Katz and Pistikopoulos, 2020).

Inputs: Parameter value θ , matrices A_i and b_i of critical regions \mathcal{R}_i , centers of Chebyshev balls C_i , binary vectors \mathcal{I}_i , $i = 1, \dots, R$, optimization problem matrices H, G, w, S in (3)

Output: Optimal control action u_0^*

```

1: for each  $\mathcal{R}_i$  do:
2:    $v_i = \|\theta - C_i\|$ 
3: solve  $v^* = \min v_i, \forall v_i$ .
4: find the region  $\mathcal{R}_i$ , associated active set  $\mathcal{I}_i$  corresponding to the minimal  $v^*$ 
5: if  $\theta \in \mathcal{R}_i$ :
6:   apply the optimal control action  $u_0^*$  using  $\mathcal{I}_i$  and (5)
7: else:
8:   find optimal active set  $\mathcal{I}^*$  from hot-start procedure using  $\mathcal{I}_i$ 
9:   apply the optimal control action  $u_0^*$  using  $\mathcal{I}^*$  and (5)

```

the critical regions \mathcal{R}_i in (7), see inputs of Algorithm 2. Moreover, the feasible points of all stored regions, i.e., the Chebyshev centers C_i , $i = 1, \dots, R$, are saved. The problem size is given by the parameter dimension n , dimension of the optimization variable m , and the length of the prediction horizon N . Obviously, if the problem size is large, then the demands on memory storage become hardly tractable.

Lemma 2.3 (Memory footprint of polytopic region). *Given mpQP in (3). The memory footprint of the i th critical region $\mathcal{R}_i \subset \mathbb{R}^n$ having a form of a polytope given by h_i halfspaces defined in (7) requires following number of floating-point numbers necessary to store this polytope*

$$n(\mathcal{R}_i) = h_i \times n + h_i. \quad (9)$$

Proof. Proof of Lemma 2.3 directly follows from the structure of the polytopic i th critical region \mathcal{R}_i , i.e., the data necessary to store h_i halfspaces determined by a matrix $A_i \in \mathbb{R}^{h_i \times n}$ and vector $b_i \in \mathbb{R}^{h_i}$ in (7). \square

The memory footprint necessary to store the data of partial solution is not only large but it is also unpredictable. Although the number of the critical regions is determined by the number of random points R , and the upper bound on the data size defining the critical region is known, it is not possible to predict the exact size of the data that we need to store in advance before solving the optimization problem. It is obvious from Lemma 2.3, that the number of floating-point numbers $n(\mathcal{R}_i)$ in (9) necessary to store the critical region \mathcal{R}_i is not fixed and varies from one region to another. Particularly, the value of $n(\mathcal{R}_i)$ depends on the number of halfspaces h_i that define the i th specific critical region.

Theorem 2.4 (Memory footprint of polytopic partial solution). *Given mpQP in (3). The total memory footprint of partial solution consisting of R critical regions including the optimization problem matrices H, G, S, w , and R binary vectors of active constraints \mathcal{I} , requires following number of floating point numbers*

$$n(\forall \mathcal{R}) = \sum_{i=1}^R (n(\mathcal{R}_i)) + Rn + R \frac{c}{64} + (m \times m + c \times m + c \times n + c), \quad (10)$$

where n is the parameter dimension, m is number of the optimization variables, c represents the number of constraints, and $n(\mathcal{R}_i)$ denotes the memory footprint of i th specific critical region in (9).

Proof. Proof of Theorem 2.4 directly follows from the structure of the polytope in (9), dimensions of the optimization problem matrices in (3), and the number of the critical regions R . According to Lemma 2.3, the memory consumption of the i th polytopic critical regions $n(\mathcal{R}_i)$ is defined by finite number of halfspaces h_i determined by a matrix $A_i \in \mathbb{R}^{h_i \times n}$ and vector $b_i \in \mathbb{R}^{h_i}$ in (7). The remaining data contains: memory footprint of corresponding binary vectors \mathcal{I}_i , $i = 1, \dots, R$ and matrices of the optimization problem H, G, S, w in (3). Note, the memory footprint necessary to store the binary vector \mathcal{I} defining the active sets is divided by 64 to transform the binary format to the double floating point numbers. \square

According to Theorem 2.4, it is necessary to solve the large-scale optimization problem to determine the total memory footprint.

3. Fixed complexity partial explicit MPC

This section is devoted to the main contribution of this paper, i.e., designing the fixed complexity parametric solution of partial explicit MPC. The term “fixed complexity” denotes that the size of the memory footprint necessary to store the parametric solution is determined in advance, i.e., before solving the multiparametric optimization problem in (3). This is a groundbreaking benefit enabling us to scale the solution of the partial explicit MPC to respect the limited memory of the hardware, where the controller will be installed.

We recall that in the MPC framework introduced in (Katz and Pistikopoulos, 2020), the partial solution of explicit MPC is evaluated. The critical regions are stored in the form of polytopes \mathcal{R}_i in (7), and also the Chebyshev balls centers C_i are stored, see Algorithm 1. The centers of Chebyshev balls C_i are then used in the online phase to evaluate the distance to the measurement θ . The Chebyshev ball center nearest to the measurement θ is determined and the corresponding critical region \mathcal{R}_i is used for initialization of the hot-start procedure to solve the large-scale optimization problem in (3) for a given state measurement θ , see Algorithm 2.

The main idea of a fixed memory footprint is to replace the polytopic region \mathcal{R}_i with its maximal volume inner approximation $\tilde{\mathcal{R}}_i$ using the Chebyshev ball. In other words, it is not necessary to store the large data A_i, b_i defining the polytopes \mathcal{R}_i in (7), but only the light-weight data defining the Chebyshev balls $\tilde{\mathcal{R}}_i$ are stored. As a consequence, the memory footprint of each critical region is fixed. Except for fixing the memory footprint, this approach leads to significant memory savings. Compared to storing the polytopic representation of the critical region \mathcal{R}_i , the memory savings are ensured, as just a single point (center of Chebyshev ball C_i) and a scalar (radius of Chebyshev ball r_i) are stored for each critical region, $\tilde{\mathcal{R}}_i, \forall i = 1, \dots, R$.

Lemma 3.1 (Memory footprint of approximated region). *Given mpQP in (3). The memory footprint of the approximation of the i th critical region $\tilde{\mathcal{R}}_i \subset \mathbb{R}^n$ using the Chebyshev ball is fixed and requires*

following number of floating-point numbers

$$n(\tilde{\mathcal{R}}) = 1 + n. \quad (11)$$

Proof. Proof of [Lemma 3.1](#) directly follows from the fixed structure of the Chebyshev ball, i.e., the data necessary to store the Chebyshev balls' radius $r \in \mathbb{R}$ and coordinates of its center $C \in \mathbb{R}^n$. \square

Remark 3.2. The fixed memory footprint of $\tilde{\mathcal{R}}_i$ could be ensured also by other well-known maximal volume inner approximations of the polytopic critical region \mathcal{R}_i in (7), e.g., by hyperboxes or ellipsoids. In this work, we consider the Chebyshev balls as they lead to a reasonable trade-off between the numerical complexity and the volume of the approximation $\tilde{\mathcal{R}}_i$. Obviously, introducing inner approximation may lead to a situation when the critical region is not detected in the online phase, although the critical region is known, i.e., $\theta \in \tilde{\mathcal{R}}_i \Rightarrow \theta \in \mathcal{R}_i$, but $\theta \in \mathcal{R}_i \not\Rightarrow \theta \in \tilde{\mathcal{R}}_i$.

As the inner approximation $\tilde{\mathcal{R}}_i$ using the Chebyshev ball of each critical region \mathcal{R}_i has the same fixed structure $\forall i = 1, \dots, R$, the fixed memory footprint of each region is enforced, see [Lemma 3.1](#). Next, the fixed-size memory footprint necessary to store all data utilized in the online phase is determined.

Theorem 3.3 (Memory footprint of approximated solution). *Given mpQP in (3). The total memory footprint of partial solution consisting of R approximated critical regions, including the optimization problem matrices H, G, S, w , and R binary vectors indicating the set of active constraints \mathcal{I}_i requires following number of floating point numbers*

$$n(\forall \tilde{\mathcal{R}}) = Rn(\tilde{\mathcal{R}}) + R\frac{c}{64} + (m \times m + c \times m + c \times n + c). \quad (12)$$

where $n(\tilde{\mathcal{R}})$ denotes the memory footprint of one critical region approximated using the Chebyshev ball in (11).

Proof. Proof of [Theorem 3.3](#) directly follows from the fixed structure of the Chebyshev ball, dimensions of the optimization problem matrices, and the number of the critical regions R . According to [Lemma 3.1](#), the memory consumption of the approximated critical regions $n(\tilde{\mathcal{R}})$ is defined by Chebyshev balls' radii r_i and centers C_i . The remaining data contains: memory footprint of corresponding binary vectors $\mathcal{I}_i, i = 1, \dots, R$ and matrices of the optimization problem H, G, S, w in (3). We recall, the memory footprint necessary to store the binary vector \mathcal{I} defining the active sets is divided by 64 to transform the binary format to the double floating point numbers. \square

Corollary 3.3.1. *The total memory footprint of a partial solution $n(\forall \tilde{\mathcal{R}})$ in (3.3) is an affine function of the number of the considered approximated critical regions R , and is independent on the solution of the optimization problem in (3).*

Proof. First, we show that (12) is an affine function of R . According to [Theorem 3.3](#), the total memory footprint of partial solution is given by (12), that can be rewritten into an equivalent form of a affine function of R given by:

$$n(\forall \tilde{\mathcal{R}}) = \alpha R + \beta, \quad (13)$$

where the slope is $\alpha = (n(\tilde{\mathcal{R}}) + \frac{c}{64})$ and the section is $\beta = (m \times m + c \times m + c \times n + c)$. Next, we show that (13) is independent on the solution of the optimization problem in (3). Function (13) is defined by α, β , and depends on variable R . The parameters α, β are determined only by the size of the optimization problem in (3), and the number of feasible points R is given in advance. Therefore, α, β, R are evaluated without the necessity to solve the optimization problem in (3). \square

Moreover, another benefit of this approach is that the evaluation of the nearest critical region to the current state measurement θ is more accurate compared to the one introduced in [Katz and](#)

[Pistikopoulos \(2020\)](#). In [Katz and Pistikopoulos \(2020\)](#), the distance v_i between the current system measurement θ and the critical region \mathcal{R}_i is evaluated by the center of the Chebyshev ball C_i , i.e., $v_i = \|\theta - C_i\|$. In contrast, our approach evaluates the distance v_i using the boundary of Chebyshev ball given by its radii r_i , i.e., $v_i = \|\theta - C_i\| - r_i$. Therefore, the distance v_i between the current system measurement θ and the original polytopic set \mathcal{R}_i approximated by the Chebyshev ball $\tilde{\mathcal{R}}_i$ is more accurate in comparison to the original approach based just on a center of inscribed Chebyshev ball C_i .

Remark 3.4. (Nearest critical region) Online evaluation of the optimal nearest critical region is hardly tractable. Analogous to [Katz and Pistikopoulos \(2020\)](#), throughout this paper, we denote the "nearest" critical region the set \mathcal{R}_i that corresponds to the minimum distance v^* evaluated between the current system measurement θ and the approximation of the critical regions. The minimum distance v^* is evaluated either using the center of the Chebyshev ball (Step 2 of [Algorithm 2, Katz and Pistikopoulos, 2020](#)) or using the boundary of the Chebyshev ball, i.e., both approaches evaluating the distance v^* are based on the Chebyshev ball. However, the Chebyshev ball inscribed in the given polytopic critical region \mathcal{R}_i is not unique in general, as in some situations, multiple maximal volume balls can be inscribed inside the polytope, e.g., when the critical region has a shape of a hyperbox, trapezoid, parallelotope, zonotope, etc. [Boyd and Vandenberghe \(2004\)](#). Moreover, approximation of the polytope \mathcal{R}_i need not lead to the optimal evaluation of the nearest critical region in general. Therefore, the identification of the "nearest" critical region could be suboptimal. Nevertheless, the control action u_0 applied for control is optimal thanks to the hot-start strategy.

Finally, [Algorithms 1, 2](#) were revisited to demonstrate the fixed complexity partial explicit MPC procedure.

The offline phase procedure of fixed complexity partial explicit MPC is evaluated by [Algorithm 3](#). Compared to the original pro-

Algorithm 3 Offline phase of fixed complexity partial explicit MPC.

Input: Set of R random feasible points $\{p_1, p_2, \dots, p_R\}$

Outputs: Centers of Chebyshev balls C_i , radii of Chebyshev balls r_i , binary vectors $\mathcal{I}_i, i = 1, \dots, R$

- 1: **for each** p_i **do:**
 - 2: solve the QP in (3) for $\theta \leftarrow p_i$
 - 3: find and store optimal active set binary vector \mathcal{I}_i
 - 4: construct the critical region \mathcal{R}_i in (8)
 - 5: **for each** \mathcal{R}_i **do:**
 - 6: construct Chebyshev ball $\tilde{\mathcal{R}}_i$
 - 7: store center C_i and radius r_i of $\tilde{\mathcal{R}}_i$
-

cedure in [Algorithm 1](#), our approach is extended by storing the Chebyshev ball radii r_i . On the other hand, the polytopic representations of the critical regions A_i, b_i in (7) are not stored. If necessary, A_i, b_i can be recovered in the online phase from the binary vector \mathcal{I}_i defining the corresponding active set \mathcal{A} and the matrices H, G, S, w of the optimization problem in (3).

The online phase of the fixed complexity partial explicit MPC is described in [Algorithm 4](#), where the distances from the boundaries of the Chebyshev balls are identified ([Algorithm 4, Step 2](#)), in contrast to the original approach in [Katz and Pistikopoulos \(2020\)](#), cf. [Algorithm 2, Step 2](#).

In [Algorithm 2](#), if current system measurement θ lies inside some of the approximated critical regions $\tilde{\mathcal{R}}_i$, then the optimal control action is evaluated using the corresponding control law in (5). Otherwise, if θ does not lie inside any of the Chebyshev

Algorithm 4 Online phase of fixed complexity partial explicit MPC.

Inputs: Parameter value θ , centers of Chebyshev balls C_i , radii of Chebyshev balls r_i , binary vectors \mathcal{I}_i , $i = 1, \dots, R$, optimization problem matrices H, G, S, w

Output: Optimal control action u_0^*

```

1: for each  $C_i$  do:
2:    $v_i = \|\theta - C_i\| - r_i$ 
3:   if ( $v_i < r_i$ ):
4:     apply the optimal control action  $u_0^*$  using  $\mathcal{I}_i$ 
5:     break
6: solve  $v^* = \min v_i, \forall v_i$ 
7: identify active set  $\mathcal{I}_i$  of the nearest region for  $v^*$ 
8: construct the polytopic critical region  $\mathcal{R}_i$  using  $\mathcal{I}_i$  and optimization problem matrices  $H, G, S, w$ 
9: if  $\theta \in \mathcal{R}_i$ :
10:  apply the optimal control action  $u_0^*$  using  $\mathcal{I}_i$ 
11: else:
12:  find optimal active set  $\mathcal{I}^*$  from hot-start procedure using  $\mathcal{I}_i$ 
13:  apply the optimal control action  $u_0^*$  using  $\mathcal{I}^*$  and (5)

```

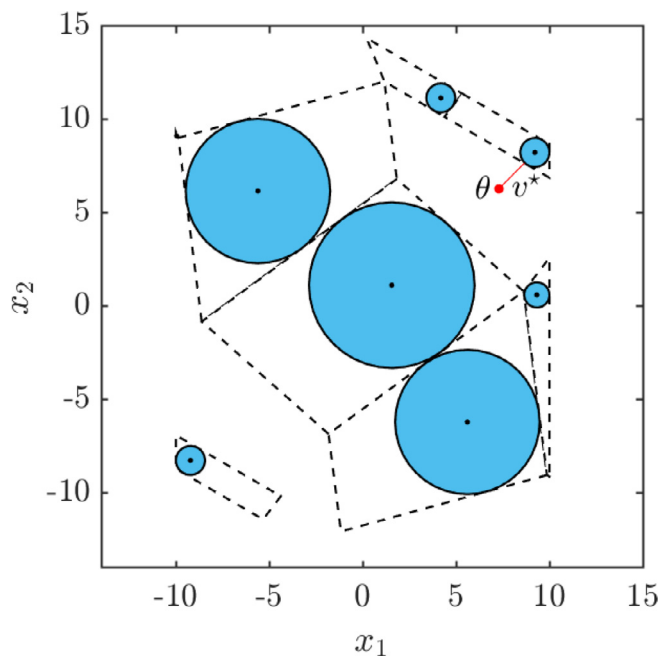


Fig. 1. Fixed complexity partial solution of explicit MPC. The polytopic regions \mathcal{R}_i (black dashed) are not stored. Instead, the Chebyshev ball approximations $\tilde{\mathcal{R}}_i$ are stored (blue). In the online phase, the minimum distance v^* from the set of Chebyshev balls (red line) to the current measurement θ is found. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

balls, i.e., $\theta \notin \tilde{\mathcal{R}}_i, \forall i = 1, \dots, R$, then the nearest Chebyshev ball is used for hot-started solution of the large-scale optimization problem in (3). The rest of the procedure remains the same as proposed in Katz and Pistikopoulos (2020). The illustration of identifying the nearest Chebyshev ball $\tilde{\mathcal{R}}_i$ is shown in Fig. 1.

Note, the proposed approach interferes with the online phase of the original approach adapted from Katz and Pistikopoulos (2020) in one major step – construction of the nearest polytope, see Algorithm 4, Step 8. After the nearest critical region is estimated, the rest of the online procedure remains the same. The remainder of the improvements represents a negligible computational intervention compared to the construction of the polytope,

i.e., checking whether the parameter lies in the Chebyshev ball in Algorithm 4, Step 3, and subtracting the radius in Algorithm 4, Step 2.

Remark 3.5 (Degeneracies in mpQP). In practical applications, degeneracies in mpQP problem solutions may occur. The primal degeneracy occurs if linear inequality constraint qualification (LICQ) does not hold, see Tøndel et al. (2003). As a consequence, it leads to the presence of the lower-dimensional critical regions in the (partial) explicit solution. These lower-dimensional critical regions can be excluded from the solution, as they form the facets between neighboring full-dimensional critical regions. Then, during the hot start strategy in the online phase, it is checked whether LICQ holds on the facet of a polytope intersecting the direction vector. If LICQ does not hold, a QP is solved with improved initialization based on the known critical region. For a more detailed overview on the evaluation of the online control algorithm, see Katz (2020). The dual degeneracy (Bemporad et al., 2002) cannot occur as $H > 0$, see problem formulation in (1) and the corresponding assumptions below.

Remark 3.6 (Accelerating the hot-start procedure). Following the ideas of the concept of *accelerating explicit MPC* introduced in Holaza et al. (2020), an alternative to the *geometrical*-perspective based on the evaluation of the distance v^* , is the *penalty-gap*-driven procedure using the difference between the current value of some Lyapunov-function-like value function $V(\theta)$ and its minimal/maximal value V_i^* corresponding to each critical region $\mathcal{R}_i, \forall i = 1, \dots, R$. Then, the “nearest” critical region is determined based just on using the comparison of two scalars, i.e., V_i^* and $V(\theta)$. This approach overcomes the main issues related to the evaluation of the distance for nonlinear MPC, where non-convex critical regions occur. Moreover, this approach also enables the pruning of the considered regions, see Holaza et al. (2020).

4. Results and discussion

In this section, we demonstrate the main benefit of fixed complexity partial explicit MPC on three sets of randomly generated large-scale optimization problems. The aim is to analyze the memory footprint of the original approach based on the polytopic representation of critical regions \mathcal{R}_i presented in Katz and Pistikopoulos (2020) and the proposed fixed complexity approach based on the maximum volume inner approximation using the Chebyshev balls $\tilde{\mathcal{R}}_i$.

Analogous to the case study presented in Katz and Pistikopoulos (2020), we generated 5 sets of large-scale systems. Each set was represented by a specific problem size and contained 5 different randomly generated systems.

Particularly, the memory footprint was determined for each of the 5×5 large-scale systems using both methods. Therefore, in total, 25 randomly generated large-scale systems were analyzed using 2 methods. For each of 25 investigated large-scale systems, the sets of random feasible seeding points were generated to obtain 300 unique critical regions, i.e., $\mathcal{R}_i, i = 1, \dots, 300$, see Remark 2.2 Then, the average values of the memory footprints were analyzed for each set of 5 large-scale systems using (10), (12).

The generated results for the considered set of 5 types of large-scale systems are summarized in Table 1. The first two columns provide the information about the given problem size in (3), i.e., the number of optimization variables m and the number of optimization problem constraints c are stated. The third column contains the information about the memory footprint of the optimization problem matrices H, G, S, w , which is the same for both approaches. The next two columns compare the memory footprint necessary to store the solution considering the polytopic approach

Table 1

Results generated for the partial solution of 300 critical regions: memory footprint comparison using the polytopic regions and Chebyshev balls for different problem sizes.

Optimization variables	Constraints	Problem matrices [kB]	Memory footprint of solution[kB]		Memory savings [%]	
			Polytopic regions	Chebyshev balls	Solution	Total
765	6 030	44 578	31 813	349	98.9	41.2
900	6 300	55 007	33 056	359	98.9	37.3
990	6 480	62 445	32 781	365	98.9	34.2
1 125	6 750	74 332	39 472	376	99.0	34.6
1 350	7 200	96 088	35 312	392	98.9	26.7

(using Lemma 2.3) and the novel fixed-complexity approach (using Lemma 3.1). Finally, the last two columns focus on memory savings which arise from the fixed complexity approach, while the first column contains the memory savings of the solution (using Lemma 3.1), and the second one contains the total memory savings considering also the problem matrices (using Theorem 3.3).

The results presented in Table 1 were generated using MATLAB R2020b, YALMIP R20200930 (Löfberg, 2004) and solver GUROBI 9.1. The results were performed on a computer running 8 cores and AMD Ryzen 7 PRO 4750U at 4.1 GHz, and 16 GB RAM.

The generated results collected in Table 1 confirmed that considering the proposed fixed memory approach significantly outperformed the original method presented in Katz and Pistikopoulos (2020). Particularly, when considering just the solution of the partial explicit MPC, the memory footprint necessary to store the fixed memory solution composed of the Chebyshev balls centers C_i and radii r_i , and corresponding binary vectors indicating active constraints \mathcal{I}_i is dramatically reduced compared to the memory footprint needed to store the polytopic solution composed of the polytopes \mathcal{R}_i , binary vectors \mathcal{I}_i , and centers of the Chebyshev balls C_i , $\forall i = 1, \dots, 300$.

The memory footprint necessary to store the solution considering the fixed memory approach requires only around 1% of the memory footprint corresponding to the original method, i.e., the memory savings of fixed memory partial solution reach up to 99%, see Table 1.

We recall, that also the matrices H , G , S , w of large-scale optimization problem in (3) have to be stored to evaluate the online phase, see inputs of Algorithms 2,4.

Note, the contribution of the large-scale optimization problem matrices H , G , S , w to the total memory consumption is significant and given, see Theorems 2.4, 3.3. Therefore, it is necessary to evaluate the memory savings also considering the large-scale optimization problem matrices H , G , S , w .

Moreover, with increasing problem size, the contribution of the large-scale optimization problem to the total memory footprint significantly increases, see Table 1, column "Problem matrices". Therefore, the gap between the total memory footprints of the considered two methods is lower, see Table 1, column "Total memory savings". Nevertheless, considering the smallest problem size (765 optimization variables and 6 030 constraints), when the problem data requires the lowest memory, the memory savings are the highest and reach up to 41.2%, see Table 1, column "Total memory savings".

The generated results in Table 1 demonstrate that the proposed fixed complexity partial explicit MPC is not only an effective approach to determine the size of the partial solution in advance, but this method also provides a significant improvement in the memory savings compared to the original method in Katz and Pistikopoulos (2020).

5. Conclusion

The proposed method improved the original partial explicit MPC method presented in Katz and Pistikopoulos (2020) in 3 main

properties: (i) fixed structure of the partial explicit solution, (ii) significant memory savings, and (iii) improved initialization of the hot-start procedure to solve the large-scale optimization problem. Particularly, we propose the novel approach of the partial solution of the explicit MPC while fixing the size of the memory footprint. It is provided by approximation of the critical regions using the fixed-structured maximal volume inner approximation based on the Chebyshev balls. As the approximation has a fixed structure, the data size necessary to store the partial solution is determined in advance. We suggest storing the Chebyshev ball center and radius, together with a binary vector defining the optimal active set of the associated critical region. The proposed fixed complexity method significantly increases the memory savings, which were demonstrated using 5 sets of 5 randomly generated large-scale systems. Third, the initialization of the hot-start procedure to solve the large-scale optimization problem was improved as the nearest critical region was evaluated using more relevant information of the Chebyshev ball boundary, in contrast to considering only its center.

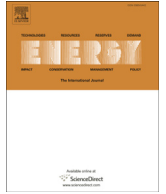
Acknowledgments

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

References

- Bemporad, A., Morari, M., Dua, V., Pistikopoulos, E.N., 2002. The explicit linear quadratic regulator for constrained systems. *Automatica* 38, 3–20.
- Borrelli, F., 2017. Constrained Optimal Control of Linear and Hybrid Systems. Springer Berlin Heidelberg doi:10.1007/3-540-36225-8.
- Boyd, S., Vandenberghe, L., 2004. *Convex Optimization*. Cambridge University Press.
- Burnak, B., Diangelakis, N.A., Katz, J., Pistikopoulos, E.N., 2019. Integrated process design, scheduling, and control using multiparametric programming. *Comput. Chem. Eng.* 125, 164–184. doi:10.1016/j.compchemeng.2019.03.004.
- Darby, M.L., Nikolaou, M., 2012. MPC: current practice and challenges. *Control Eng. Pract.* 20 (4), 328–342. doi:10.1016/j.conengprac.2011.12.004.
- Darby, M.L., Nikolaou, M., Jones, J., Nicholson, D., 2011. RTO: an overview and assessment of current practice. *J. Process Control* 21 (6), 874–884. doi:10.1016/j.jprocont.2011.03.009.
- Ferreau, H.J., Bock, H., Diehl, M., 2008. An online active set strategy to overcome the limitations of explicit MPC. *Int. J. Robust Nonlinear Control* 18, 816–830.
- Goebel, G., Allgöwer, F., 2017. Semi-explicit MPC based on subspace clustering. *Automatica* 83, 309–316. doi:10.1016/j.automatica.2017.06.036.
- Holaza, J., Oravec, J., Kvasnica, M., Dyrská, R., Mönnigmann, M., Fikar, M., 2020. Accelerating explicit model predictive control by constraint sorting. In: *Preprints of the 21st IFAC World Congress (Virtual)*, Berlin, Germany, Vol. 21, pp. 11520–11525.
- Jost, M., Pannocchia, G., Mönnigmann, M., 2017. Accelerating linear model predictive control by constraint removal. *Eur. J. Control* 35, 42–49.
- Katz, J., 2020. *Advancing multiparametric programming for model predictive control*. Texas A&M University Ph.D. thesis.
- Katz, J., Pistikopoulos, E.N., 2020. A partial multiparametric optimization strategy to improve the computational performance of model predictive control. *Comput. Chem. Eng.* 118, 107057.
- Kvasnica, M., Bakaráč, P., Klaučo, M., 2019. Complexity reduction in explicit MPC: reachability approach. *Syst. Control Lett.* 124, 19–26. doi:10.1016/j.sysconle.2018.12.002.
- Kvasnica, M., Fikar, M., 2012. Clipping-based complexity reduction in explicit MPC. *IEEE Trans. Automat. Control* 57 (7), 1878–1883. doi:10.1109/TAC.2011.2179428.

- Kvasnica, M., Hledík, J., Rauová, I., Fikar, M., 2013. Complexity reduction of explicit model predictive control via separation. *Automatica* 49 (6), 1776–1781. doi:10.1016/j.automatica.2013.02.018.
- Kvasnica, M., Takács, B., Holaza, J., Cairano, S.D., 2015. On region-free explicit model predictive control. In: 54th IEEE Conference on Decision and Control, Vol. 54, pp. 3669–3674. Osaka, Japan
- Li, H., Swartz, C.L.E., 2019. Dynamic real-time optimization of distributed MPC systems using rigorous closed-loop prediction. *Comput. Chem. Eng.* 122, 356–371. doi:10.1016/j.compchemeng.2018.08.028. 2017 Edition of the European Symposium on Computer Aided Process Engineering (ESCAPE-27)
- Löfberg, J., 2004. YALMIP : a toolbox for modeling and optimization in MATLAB. In: *Proc. of the CACSD Conf., Taipei, Taiwan*, pp. 284–289.
- Lu, J.Z., 2015. Closing the gap between planning and control: amultiscale MPC cascade approach. *Annu. Rev. Control* 40, 3–13. doi:10.1016/j.arcontrol.2015.09.016.
- Masti, D., Pippia, T., Bemporad, A., Schutter, B.D., 2020. Learning approximate semi-explicit hybrid MPC with an application to microgrids. *IFAC-PapersOnLine* 53 (2), 5207–5212. doi:10.1016/j.ifacol.2020.12.1192. 21th IFAC World Congress
- Mayne, D., 2016. Robust and stochastic model predictive control: are we going in the right direction? *Annu. Rev. Control* 41, 184–192. doi:10.1016/j.arcontrol.2016.04.006.
- Mitze, R., Mönnigmann, M., 2020. A dynamic programming approach to solving constrained linear-quadratic optimal control problems. *Automatica* 120, 109132. doi:10.1016/j.automatica.2020.109132.
- Morari, M., Lee, J.H., 1999. Model predictive control: past, present and future. *Comput. Chem. Eng.* 23 (10), 667–682.
- Morato, M.M., Normey-Rico, J.E., Sename, O., 2020. Model predictive control design for linear parameter varying systems: a survey. *Annu. Rev. Control* 49, 64–80. doi:10.1016/j.arcontrol.2020.04.016.
- Mönnigmann, M., 2019. On the structure of the set of active sets in constrained linear quadratic regulation. *Automatica* 106, 61–69. doi:10.1016/j.automatica.2019.04.017.
- Oravec, J., Blažek, S., Kvasnica, M., Cairano, S.D., 2013. Polygonic representation of explicit model predictive control. In: *IEEE Conference on Decision and Control*, pp. 6422–6427. Florence, Italy
- Qin, S.J., Badgwell, T.A., 2003. A survey of industrial model predictive control technology. *Control Eng. Pract.* 11 (7), 733–764. doi:10.1016/S0967-0661(02)00186-7.
- Son, S.H., Oh, T.H., Kim, J.W., Lee, J.M., 2020. Move blocked model predictive control with improved optimality using semi-explicit approach for applying time-varying blocking structure. *J. Process Control* 92, 50–61. doi:10.1016/j.jprocont.2020.04.002.
- Tøndel, P., Johansen, T.A., Bemporad, A., 2003. An algorithm for multi-parametric quadratic programming and explicit MPC solutions. *Automatica* 39 (3), 489–497.
- Zeilinger, M.N., Jones, C.N., Morari, M., 2011. Real-time suboptimal model predictive control using a combination of explicit MPC and online optimization. *IEEE Trans. Automat. Control* 56 (7), 1524–1534. doi:10.1109/TAC.2011.2108450.



Soft-constrained robust model predictive control of a plate heat exchanger: Experimental analysis

Juraj Oravec^{*}, Monika Bakošová, Lenka Galčíková, Michal Slávik, Michaela Horváthová, Alajos Mészáros

Slovak University of Technology in Bratislava, Faculty of Chemical and Food Technology, Institute of Information Engineering, Automation, and Mathematics, Radlinského 9, SK-812 37, Bratislava, Slovak Republic

ARTICLE INFO

Article history:

Received 18 January 2019

Received in revised form

11 April 2019

Accepted 13 May 2019

Available online 16 May 2019

Keywords:

Plate heat exchanger

Robust control

Model predictive control

Soft constraints

Energy saving

ABSTRACT

Real processes with heat exchange have usually complex behaviour and are energy intensive. In practical applications, the process variables are always bounded, and it is suitable to include these boundaries into the controller design. The soft-constrained robust model predictive controller has been designed to improve the control performance and energy consumption in comparison with the robust model predictive control with only hard constraints. Experimental application of soft-constrained robust model predictive control (SCR MPC) for a laboratory plate heat exchanger is presented in this paper. The plate heat exchanger is a non-linear process with asymmetric dynamics and is modelled as a system with parametric uncertainties. The controlled variable is the temperature of the heated fluid at the outlet of the heat exchanger and the manipulated variable is the volumetric flow rate of the heating fluid. The actuator is a peristaltic pump and the influence of the linear and non-linear actuator characteristics on the control performance is also investigated. The set-point tracking using SCR MPC is studied for the laboratory plate heat exchanger in an extensive case study. The experimental results confirmed the improvement of the control responses and reduction of energy consumption by introducing the soft constraints into MPC design.

© 2019 Elsevier Ltd. All rights reserved.

1. Introduction

Optimal control of heat exchangers is one of the most important aspects of industrial production, as their operation is really energy intensive and the heat transfer represents the crucial phenomena for all thermal energy applications [1]. Model Predictive Control (MPC) represents a state-of-the-art optimal control strategy for multivariable constrained systems [2]. The values of the manipulated variables are optimized in each control step subject to various constraints on system states, controlled variables, and manipulated variables [3]. Current MPC originates in Dynamic Matrix Control (DMC) and Generalized Predictive Control (GPC) [4]. The development and implementation of DMC and GPC began in the 1970s in chemical and petrochemical industries. The tractable formulation of control problem was focused on the plants with relatively slow dynamics enabling to solve the optimization problems within the

sampling time.

The drawback of the original MPC setup was that uncertain process parameters were not taken into account. This drawback was overcome by introducing Robust MPC, e.g., see Ref. [5]. The tractable formulation of the optimization problem was ensured by using the linear matrix inequalities (LMIs) by the pioneer work [6]. This work was further developed, e.g., see Ref. [7], and references therein.

It was demonstrated by plenty of research works and industrial applications that MPC and/or robust MPC improved control performance and optimized energy consumption for a wide class of energy-intensive processes. In Ref. [8], MPC was used to design multivariable temperature control of a continuous fermentation unit. The manipulated variables were the feed flow rate and the flow rate of the cooling fluid. Simulation results confirmed the improved control performance. The on-line control maximizing the heat recovery of the heat exchanger network was investigated in Ref. [9]. The accurate model of the considered plant was derived and the closed-loop control performance was evaluated. The well-tuned PID controller-based heat recovery maximization of the heat

^{*} Corresponding author.

E-mail address: juraj.oravec@stuba.sk (J. Oravec).

exchanger network coupled with a crude distillation unit in the presence of fouling was investigated in Ref. [10]. The authors of [11] investigated the improvement of the closed-loop control performance of an air-to-water heat pump ensured by the predictive control. Global optimization method, particularly the genetic algorithm, was used to optimize the design of a high-temperature heat exchanger for waste heat cascade recovery from exhaust flue gases in Ref. [12]. In Ref. [13], the proposed cascade control introduced the MPC in the master loop to optimize the thermal response of a nuclear superheated-steam system. The simulation results confirmed the improved control performance. The authors of [14] designed MPC for a cycle plant combined with integrated solar collectors. The proposed MPC strategy outperformed the considered PI controller in energy consumption. The energy savings in an air-cooled steam condenser were ensured by introducing optimization into the closed-loop control design in Ref. [15]. For the process of waste heat energy conversion with organic Rankine cycle, the robust H_2 control was designed in Ref. [16] to improve the operating efficiency. Robust MPC design for a heat exchanger network was analysed in Ref. [17]. The fouling was handled via parametric uncertainty and the designed robust MPC ensured the significant energy savings compared to the conventional PID control. In Ref. [18], the robust MPC subject to non-symmetric hard constraints was designed and the closed-loop experimental results were analysed. The implementation of non-symmetric hard constraints improved control performance. The implementation of soft constraints was not considered.

Between heat exchangers, the plate heat exchangers attract the attention of engineers and researchers as they are more efficient than the shell and tube exchangers. The simplified dynamical input-output modelling of plate heat exchangers was developed in Ref. [19]. The experimental results confirmed the advantages of the proposed strategy also for the advanced controller design. Control performance of the adaptive cascade control for liquid-liquid plate heat exchanger was investigated in Ref. [20]. The laboratory implementation confirmed that the designed controller outperformed the conventional PID controller. The implementation and control performance of the alternative approaches of robust MPC design for a laboratory plate heat exchanger were analysed in Ref. [21].

This paper significantly extends the results of the works [22–24]. The concept of LMI-based soft-constrained robust MPC design was introduced in Ref. [22] and the theoretical results and numerical examples were presented. The simulation results confirmed the significantly improved control performance. Although the possibility to use the soft-constrained robust (SCR) MPC was studied intensively by simulation experiments [22], there was still a lack of experimental studies in this field. Moreover, the developed SCR MPC did not have an integral action and was not able to ensure the offset-free reference tracking. The mathematical model of the laboratory plate heat exchanger was identified and the SCR MPC without integral action was implemented for control of this heat exchanger in Ref. [23]. The closed-loop experimental results were analysed and they confirmed improved control performance. Although the control performance was improved, the experimental results showed the necessity to add an integral action to the SCR MPC in the future to ensure the offset-free control response. In the paper [24], SCR MPC with integral action was designed. The soft-constraints on the manipulated variable (MV) and controlled variable (CV) were considered simultaneously. Simulations results were obtained and improved control performance was confirmed. But, the control performance evaluation was based only on simulation results and no experimental validation was performed.

This paper brings new results in comparison with previously

mentioned papers. More control scenarios of SCR MPC with integral action are analysed, in which the soft-constraints are implemented simultaneously for CV and MV, and also separately just for MV, and just for CV. All control scenarios were tested experimentally on the laboratory plate heat exchanger and totally new experimental results were obtained.

The laboratory plate heat exchanger was a non-linear process with asymmetric dynamics and was modelled as a system with parametric uncertainties. The controlled variable (CV) was the temperature of the heated fluid at the outlet of the heat exchanger and the manipulated variable (MV) was the volumetric flow rate of the heating fluid. The actuator was a peristaltic pump and the influence of the linear and non-linear actuator characteristics on the control performance was also investigated. In comparison to the paper [23], the integral action was added to the soft-constrained robust model predictive controller and the control performance of the laboratory plate heat exchanger ensured by the designed SCR MPC with integral action was investigated experimentally. The controller with integral action removed the steady-state error. In comparison to the paper [24], more control scenarios and influence of various soft constraints on control performance were experimentally studied. The soft-constraints were implemented simultaneously for CV and MV, and also separately only for MV, and only for CV. All experimental results were obtained in an extensive case study for the plate laboratory heat exchanger using SCR MPC with integral action in the set-point tracking. Various quality criteria were evaluated to judge the quality of the achieved results. The experimental results confirmed improvements of control responses and reduction of energy consumption by introducing the soft constraints into the robust MPC design.

The paper is organized as follows. Section 2 introduces the main properties of the controlled plate heat exchanger and its mathematical model. The implemented control strategy is described in Section 3. The experimental results are discussed in Section 4 followed by concluding remarks in Section 5.

2. Plate heat exchanger

2.1. Description of a laboratory plate heat exchanger

The controlled process was a laboratory heat exchanger of the Armfield Process Control Trainer PCT23 (Fig. 1), see Ref. [25].

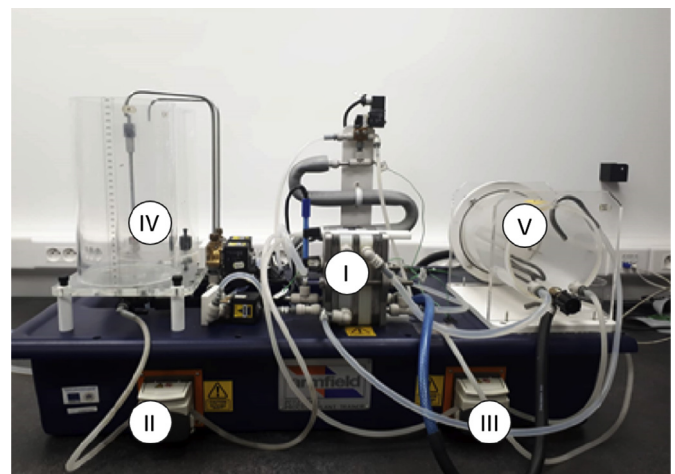


Fig. 1. Plate heat exchanger (I), peristaltic pump for heated fluid (II), peristaltic pump for cold water (III), tank for cold fluid (IV), tank with heating for preparing hot water (V).

Specifically, it was the three-stage indirect liquid-liquid plate heat exchanger (PHE) (Fig. 1 device (I)). These three stages were separated, but interconnected, and were able to provide (i) cooling, (ii) heating, and (iii) feed preheat/regeneration. For the control design, the heating stage was considered. The heat exchanger's outer width was 90 mm, the length was 103 mm and the height was 160 mm.

The heated fluid – cold water was stored in a retention tank (Fig. 1 device (IV)). The heating fluid – hot water was preheated to the fixed temperature $T_{hot} = 70\text{ }^{\circ}\text{C}$ in the heating tank (Fig. 1 device (V)). Two peristaltic pumps pumped heated (Fig. 1 device (II)) and heating (Fig. 1 device (III)) fluids into the plate heat exchanger. The flow rate of cold water was constant, meanwhile, the flow rate of hot water was a manipulated variable. The peristaltic pump pumping the hot water into the heat exchanger was an actuator.

The peristaltic pumps had flexible tubing made from the silicon rubber. The wall thickness was 1.6 mm and inner diameter was 3.2 mm. The temperature of the heated fluid in the outlet stream was measured by the type K thermocouple operating within 0 – 150 °C.

The scheme of the Armfield PCT23 arrangement is presented in Fig. 2, where (I) is the plate heat exchanger, (II) is the peristaltic pump for cold fluid, (III) is the peristaltic pump for hot fluid, (IV) is the tank for heated fluid, (V) is the tank with heating for preparing hot fluid, (L1, L2) are the level sensors, (T1 – T5) are the temperature sensors, (S1 – S5) are the solenoid valves, and (W) is the heat supply for preparing hot fluid, see Refs. [21,25]. Further technical specifications of the plate heat exchanger can be found in Ref. [25].

2.2. Static characteristics of the controlled process and actuator

An advanced controller design requires a thorough knowledge of the behaviour of the controlled plant. Therefore, experimental analysis of the steady-state operating modes of the laboratory plate heat exchanger (PHE) was performed first and the static characteristics of the controlled process and actuator were measured. The static characteristic is the dependence of the output variable on the input variable in steady operating modes.

The results are shown in Fig. 3. Fig. 3(a) presents the static characteristic of the PHE. The output variable of the PHE is the temperature of the heated fluid at the outlet of the PHE, T , and the input variable is the volumetric flow rate of the hot fluid entering the plate heat exchanger, \dot{V}_{hot} . The measured data are represented by blue circles and the green curve is the quadratic approximation

of the measured data. This static characteristic demonstrates the non-linearity of the PHE. Fig. 3(b) shows two static characteristics of the actuator, which is the peristaltic pump (PP) feeding the hot fluid to the PHE. The output variable of the PP is the volumetric flow rate of the hot fluid entering the plate heat exchanger, \dot{V}_{hot} , and the input variable is the voltage, U , in the range 1 – 5 V. The real measured static characteristic of the pump is linear (blue line). To compensate for the non-linearity of the PHE, the non-linear equal percentage characteristic of the PP was also considered (green line). This characteristic was obtained as an approximation of the measured data, see Ref. [26].

The mathematical description of the linear static characteristic of the PP (Fig. 3(b), blue) has the form:

$$\dot{V}_{hot} = 0.1338 U - 1.8633. \tag{1}$$

The mathematical description of the non-linear characteristic of the PP (Fig. 3(b), green) is following:

$$\dot{V}_{hot} = \frac{e^h}{\tau \dot{V}_{max}}, \quad h = U_{norm} \log(\tau), \quad \tau = \frac{\dot{V}_{max}}{\dot{V}_{min}}, \tag{2}$$

where \dot{V}_{min} and \dot{V}_{max} are the minimum and maximum volumetric flow rates of the heating fluid, respectively. U_{norm} is the normalized voltage manipulating the PP, i.e., the voltage recalculated to the range 0 – 1 V.

Another result of this analysis was the determination of the boundary values of the controlled variable (CV) and the manipulated variable (MV). The CV is the temperature of the heated fluid in the outlet stream of the plate heat exchanger, T , and its range is 30 – 54 °C. The manipulated variable (MV) is the volumetric flow rate of the hot fluid entering the plate heat exchanger, \dot{V}_{hot} , and its range is 0 – 11.5 ml s⁻¹. These boundary values represent the hard constraints of the variables.

Based on the results described above, the reference value $T_{ref} = 40\text{ }^{\circ}\text{C}$ was selected for the PHE control.

Static characteristics of the actuator (Fig. 3(b)) also served to find mathematical descriptions of the calibration curves needed to convert the value of the MV, i.e., the volumetric flow rate \dot{V}_{hot} , calculated by the control algorithm to the voltage U controlling the PP. Mathematical descriptions of the calibration curves were obtained as inverse models for the models in (1) and (2).

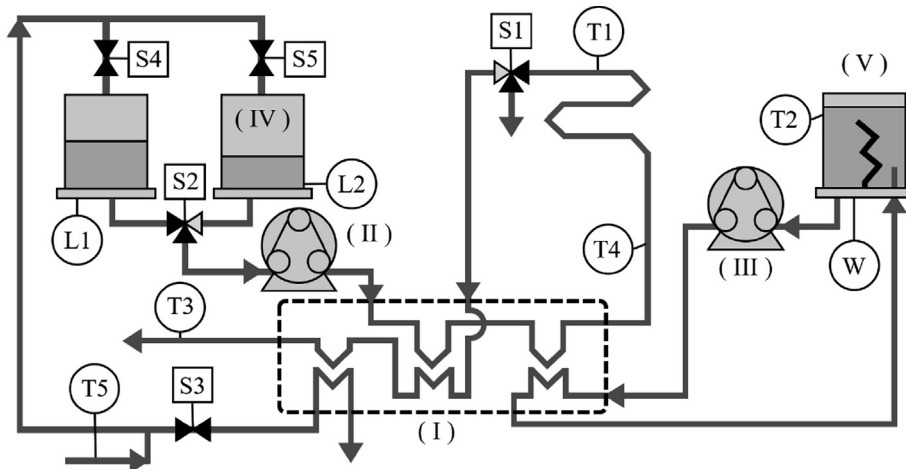
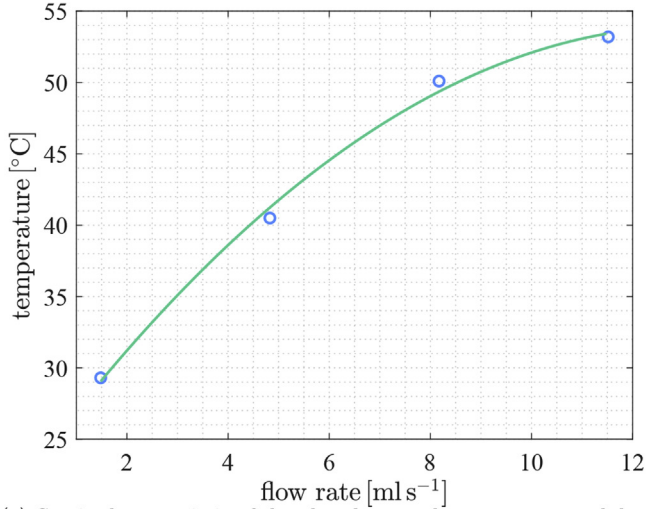
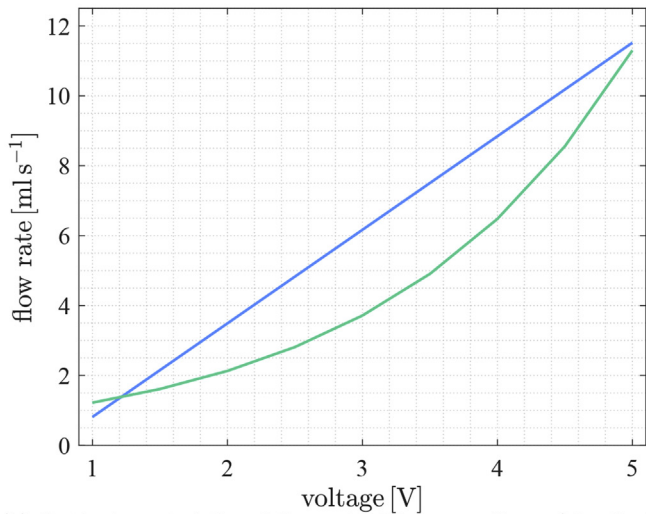


Fig. 2. Scheme of Armfield PCT23: PHE (I), peristaltic pump for cold water (II), peristaltic pump for hot water (III), tank for cold water (IV), tank with heating for preparing hot water (V), liquid level sensors (L1, L2), temperature sensors (T1 – T5), solenoid valves (S1 – S5), and heat supply for preparing hot fluid (W) [21].



(a) Static characteristic of the plate heat exchanger: measured data (blue circles), quadratic approximation (green curve).



(b) Static characteristics of the peristaltic pump: linear (blue line) and non-linear (green curve).

Fig. 3. Static characteristics of the controlled process and actuator.

2.3. Mathematical model of the plate heat exchanger for the soft-constrained robust model predictive control design

Soft-constrained robust model predictive control (SCR MPC) design requires a mathematical model of a controlled process with uncertainties in the form of a state space system in the discrete time domain:

$$x(k+1) = A(\lambda)x(k) + B(\lambda)u_{\text{sat}}(k), x(0) = x_0, \quad (3a)$$

$$y(k) = Cx(k), \quad (3b)$$

$$[A(\lambda), B(\lambda)] \in \mathbb{A}, \mathbb{A} = \text{convhull}\left([A^{(v)}, B^{(v)}]\right), \quad (3c)$$

where $k \geq 0$ is a time instant in the discrete-time domain for the sampling time $t_s = 5$ s, $x(k) \in \mathbb{R}^{n_x}$ are states of the controlled system, $u_{\text{sat}}(k) \in \mathbb{R}^{n_u}$ are saturated control inputs, $y(k) \in \mathbb{R}^{n_y}$ are system outputs, x_0 are initial conditions. Following the linear control

theory principles, the control inputs $u_{\text{sat}}(k)$ and system outputs $y(k)$, respectively, are MVs and CVs in the incremental form, i.e., they are given by:

$$u_{\text{sat}}(k) = \Delta \dot{V}_{\text{hot}}(k) = \dot{V}_{\text{hot}}(k) - \dot{V}_{\text{hot}}^s, \quad (4a)$$

$$y(k) = \Delta T(k) = T(k) - T^s, \quad (4b)$$

where $\dot{V}_{\text{hot}}(k)$ is MV, $T(k)$ is CV, and \dot{V}_{hot}^s , T^s are corresponding steady-state values.

Matrices $A(\lambda) \in \mathbb{R}^{n_x \times n_x}$ and $B(\lambda) \in \mathbb{R}^{n_x \times n_u}$ in (3a) are state and input matrices of the controlled system given by

$$A(\lambda) = \sum_{v=1}^{n_v} \lambda_v A^{(v)}, \quad B(\lambda) = \sum_{v=1}^{n_v} \lambda_v B^{(v)}, \quad \sum_{v=1}^{n_v} \lambda_v = 1, \quad (5)$$

where the parameter $0 \leq \lambda_v \leq 1$, superscript (v) represents the v -th vertex, $v = 1, 2, \dots, n_v$. Parameter n_v is the total number of vertices of the uncertain system \mathbb{A} .

Matrix $C \in \mathbb{R}^{n_y \times n_x}$ in (3b) is the output matrix. The uncertain system \mathbb{A} in (3c) is a polytope represented as a convex hull (convhull) of vertex systems, see e.g. Ref. [27], chap. 2.1.4.

The mathematical model of the laboratory PHE was obtained by step-response identification techniques, e.g., see Ref. [28]. As the PHE is a non-linear process with asymmetric behaviour, several step responses were measured within the whole range of the operating conditions. The identified transfer functions were transformed into the form of a state-space system with parametric uncertainty. Technical details related to the heat exchanger identification can be found in Ref. [23].

3. Soft-constrained robust MPC design with integral action

For the PHE, the soft-constrained robust model-based predictive control (SCR MPC) was designed. For the controller design, the mathematical model representing a nominal system is needed. The nominal system is an idealized system in which the influence of uncertainties can be neglected and has the form

$$x(k+1) = A^{(0)}x(k) + B^{(0)}u_{\text{sat}}(k), x(0) = x_0, \quad (6a)$$

$$y(k) = Cx(k). \quad (6b)$$

The matrix pair $[A^{(0)}, B^{(0)}]$ can be obtained as the analytic centre of a polytope \mathbb{A} (see e.g. Ref. [27], chap. 8.5.3).

3.1. SCR MPC design with integral action

To improve control, an integral action was added to the SCR MPC. The basis for adding the integral action is augmentation of the state vector of the process model in (3) and obtaining an augmented system. The augmented vector of states has the form

$$z(k) = \begin{bmatrix} x(k) \\ \sum_{i=0}^k e(i) \end{bmatrix}, \quad (7)$$

where the control error is calculated as $e(k) = w(k) - y(k)$, and $w \in \mathbb{R}^{n_y}$ is the set-point. For the PHE, the control error is calculated: $e(k) = T_{\text{ref}}(k) - T(k)$.

The model of the augmented system is:

$$z(k+1) = \tilde{A}(\lambda) z(k) + \tilde{B}(\lambda) u(k), \quad z(0) = z_0, \quad (8a)$$

$$y(k) = \tilde{C} z(k), \quad (8b)$$

$$[\tilde{A}(\lambda), \tilde{B}(\lambda)] \in \tilde{\mathbb{A}}, \quad \tilde{\mathbb{A}} = \text{convhull}([\tilde{A}^{(v)}, \tilde{B}^{(v)}]), \quad (8c)$$

where:

$$\tilde{A}^{(v)} = \begin{bmatrix} A^{(v)} & 0 \\ -t_s C & I \end{bmatrix}, \quad \tilde{B}^{(v)} = \begin{bmatrix} B^{(v)} \\ 0 \end{bmatrix}, \quad \tilde{C} = [C \quad 0]. \quad (9)$$

The matrices in (9) for the laboratory PHE had the form:

$$\tilde{A}^{(v)} = \begin{bmatrix} 0.6649 \pm 0.1470 & 0 \\ -0.1833 & 1 \end{bmatrix}, \quad (10a)$$

$$\tilde{B}^{(v)} = \begin{bmatrix} 4.0886 \pm 0.4249 \\ 0 \end{bmatrix}, \quad (10b)$$

$$\tilde{C} = [0.1833 \quad 0]. \quad (10c)$$

More technical details can be find, e.g., in Ref. [18].

3.2. Hard and soft constraints

Hard constraints are real-world boundaries on MVs and CVs that cannot be overcome, e.g., the minimum/maximum flow rate, the minimum/maximum temperature, see Fig. 3(b), and others.

The considered SCR MPC design procedure takes into account the symmetric hard constraints on MVs and the mathematical representation of the constraints on the MVs may have the form:

$$u_{\text{sat}} \in \mathbb{U}_{\text{max}}, \quad \mathbb{U}_{\text{max}} = \{u_{\text{sat}} : \|u_{\text{sat}}\| \leq \|u_{\text{max}}\|\}, \quad (11)$$

where $u_{\text{max}} \in \mathbb{R}^{n_u}$ are the maximal admissible values of control inputs.

So, the control inputs generated by the SCR MPC are in the range between the minimum and maximum boundary value of the input. It is possible to define a saturation function $f_{\text{sat}}(u(k), u_{\text{max}})$ to calculate the saturated control input u_{sat} in (11), e.g., see Ref. [23].

The mathematical representation of the symmetric hard constraints on the system outputs may have the form:

$$y(k) \in \mathbb{Y}, \quad \mathbb{Y} = \{y : \|y\| \leq \|y_{\text{max}}\|\}, \quad (12)$$

and $y_{\text{max}} \in \mathbb{R}^{n_y}$ are the maximal admissible values of the system outputs.

Besides the hard constraints, there are also soft constraints that can be used if necessary. The mathematical representation of soft constraints can be following:

$$u(k) \in \mathbb{U}_{\text{soft}} \subset \mathbb{U}_{\text{max}}, \quad y \in \mathbb{Y}_{\text{soft}} \subset \mathbb{Y}_{\text{max}}, \quad (13)$$

where the boundaries of soft-constraints are determined by the sets:

$$\mathbb{U}_{\text{soft}} = \left\{ u \in \mathbb{R}^{n_u} : \|u\| \leq \|u_{\text{soft}}\| \right\}, \quad (14a)$$

$$\mathbb{Y}_{\text{soft}} = \left\{ y \in \mathbb{R}^{n_y} : \|y\| \leq \|y_{\text{soft}}\| \right\}, \quad (14b)$$

where $u_{\text{soft}} \in \mathbb{R}^{n_{su}}$, $y_{\text{soft}} \in \mathbb{R}^{n_{sy}}$ are the symmetric soft constraints on control inputs and system outputs, respectively. The technical

details were discussed, e.g., in Ref. [22].

3.3. Feedback control law

The developed SCR MPC design procedure is based on the minimization of the LQR-based quadratic cost function

$$J = \sum_{k=0}^{n_k} (z(k)^\top \tilde{Q} z(k) + u(k)^\top R u(k)), \quad (15)$$

where $\tilde{Q} \in \mathbb{R}^{(n_x+n_y) \times (n_x+n_y)} > 0$ is the weighting matrix for the augmented vector of states z in (7) and $R \in \mathbb{R}^{n_u \times n_u}$ is the weighting matrix for the control input u .

The cost function in (15) is minimized using the closed-loop control of the system in (8) calculated as

$$u(k) = [F_p(k) \quad F_I(k)] z(k) = \tilde{F}(k) z(k), \quad (16)$$

where F_p, F_I are the proportional and the integral parts of the state-feedback linear control law, respectively. $\tilde{F} \in \mathbb{R}^{n_u \times 2n_x}$ is the gain matrix in the compact form.

3.4. Optimization problem in SCR MPC design

The aim of SCR MPC design is to determine a state-feedback control law in (16) for the system in (8) so that the control is optimal and takes into account the hard constraints in (11)–(12), and soft constraints in (14). This aim can be mathematically transformed to the optimization problem of the semidefinite programming (SDP):

$$\min_{\gamma, X, Y, Z, \tilde{s}_y, \tilde{s}_u} \gamma + Q_{\text{soft}}^\top \tilde{s}_y + R_{\text{soft}}^\top \tilde{s}_u \quad (17a)$$

$$\text{s.t.} \begin{bmatrix} X_k & \star & \star & \star \\ A^{(0)}X_k + B^{(0)}K_k & X_k & \star & \star \\ \sqrt{Q}X_k & 0 & \gamma_k I & \star \\ \sqrt{R}K_k & 0 & 0 & \gamma_k I \end{bmatrix} > 0, \quad (17b)$$

$$\begin{bmatrix} X_k & \star \\ A^{(v)}X_k + B^{(v)}K_k & X_k \end{bmatrix} > 0, \quad (17c)$$

$$\begin{bmatrix} 1 & \star \\ z_k & X_k \end{bmatrix} > 0, \quad (17d)$$

$$\begin{bmatrix} U_{\text{max}} & \star \\ Z_k^\top & X_k \end{bmatrix} > 0, \quad (17e)$$

$$\begin{bmatrix} X_k & \star \\ C[A^{(v)}X_k + B^{(v)}K_k] & Y_{\text{max}} \end{bmatrix} > 0, \quad (17f)$$

$$\begin{bmatrix} X_k & \star \\ E_u Y_k & U_{\text{soft},k} \end{bmatrix} > 0, \quad (17g)$$

$$\begin{bmatrix} X_k & \star \\ E_y C[A^{(v)}X_k + B^{(v)}K_k] & Y_{\text{soft},k} \end{bmatrix} > 0. \quad (17h)$$

for $\forall v = 1, 2, \dots, n_v$. In 17, the symbol \star is used as an element of a matrix and denotes symmetric structure of a matrix. The subscript k represents the discrete-time dependence. In (17b), I is an identity matrix and 0 is a zero matrix of appropriate dimensions.

Optimization problem in (17) is resolved in each control step to optimize the values of the following variables: $\gamma, X, Y, Z, K, U_{\text{soft}}, Y_{\text{soft}}, \tilde{s}_u, \tilde{s}_y$. The robust closed-loop system stability is guaranteed via the Lyapunov function $V(x) = x^T P x$, where $P = P^T > 0, X = \gamma P^{-1}$. Variables \tilde{s}_u and \tilde{s}_y are square values of overruns of the soft constraints on MV and CV, respectively. Matrices Z, K ensure the closed-loop stability subject to the control input saturation.

The feedback gain matrix in (16) is calculated from the feasible solution of the optimization problem in (17) using the optimized variables X, Y :

$$\tilde{F} = Y X^{-1}. \quad (18)$$

The technical details were discussed in Ref. [23].

4. Results and discussion

This section presents the main results obtained implementing the SCR MPC for the laboratory plate heat exchanger. At first, the considered experiment setup is introduced. Then, various control strategies are implemented to control the PHE, and the closed-loop control trajectories are analysed. Finally, the analytical criteria are compared.

4.1. Experimental setup of robust SCR MPC

The scheme of the closed-loop SCR MPC with integral action implemented for control of the *Plate Heat Exchanger* is shown in Fig. 4, where the closed-loop input was the *Reference temperature* T_{ref} , the closed-loop output was the *Controlled Variable*, i.e., the temperature of the heated fluid at the outlet of PHE and the *Manipulated Variable* was the volumetric flow rate of the hot fluid. As the robust *Controller* and the *Integrator* needed information about the controlled system *States*, see (3), the *Observer* ensuring *State Estimation* was used. The steady-state error was removed using the *Integrator*. SCR MPC in each control step calculated the optimal value of *Manipulated Variable*.

The aim of robust SCR MPC of the PHE was to ensure offset free set-point tracking for the reference temperature $T_{\text{ref}} = 40^\circ\text{C}$. The initial temperature of the heated fluid was 30°C . For comparison of results, the time of control was $t_{\text{reg}} = 400\text{ s}$. The hard constraints on MV were the boundaries of the interval $[0, 11.5]\text{ ml s}^{-1}$, and the hard constraints on CV were the boundaries of the interval $[30, 54]^\circ\text{C}$.

According to the considered soft-constraints, four robust MPC scenarios were investigated:

- 1 Robust MPC designed without soft constraints;
- 2 Robust MPC designed for the soft-constrained controlled variable;
- 3 Robust MPC designed for soft-constrained controlled variable and manipulated variable;
- 4 Robust MPC designed for soft-constrained manipulated variable.

Each control scenario was implemented considering both, the linear and the non-linear characteristics of the peristaltic pump.

The real-time control of the PHE was accomplished using MATLAB/Simulink R2018b environment, CPU i5 1.7 GHz and 6 GB RAM. The soft-constrained robust model-based predictive controllers were designed using MUP toolbox [29]. The semidefinite optimization problems in the robust MPC design were parsed using the YALMIP toolbox [30] and solved by solver MOSEK [31]. The real-time WiFi-based communication with the plant was provided by eLab Manager, as a part of the concept of the *Laboratory of Things*, see Ref. [32].

4.2. Robust MPC without soft constraints

In this control scenario, the robust MPC was designed without soft constraints, i.e., only hard constraints were considered. The control performance ensured by this control scenario served as the reference performance for investigating the impact of the soft constraints. The closed loop control trajectories obtained using robust MPC without soft constraints are depicted in Fig. 5. The possibility to compensate for the non-linearity of the controlled process with a non-linear actuator was also tested. The trajectories of the controlled variable are plotted in Fig. 5(a), where the results are presented for both, the actuator with linear static characteristic (Fig. 5(a), blue) and the actuator with non-linear equal percentage static characteristic (Fig. 5(a), green). As can be seen in Fig. 5(a), the robust MPC with both actuators assured the set-point tracking without steady-state error. But, control using both types of actuators led to significant overshoots in the control responses. The actuator with linear static characteristic allowed greater overshoot than the actuator with the non-linear characteristic. On the other hand, the settling time increased when the actuator with non-linear static characteristic was used. The trajectories of the manipulated variable are shown in Fig. 5(b). The flow rate of hot water pumped by the actuator with the linear characteristics is represented by the blue line and the flow rate pumped by the actuator with non-linear characteristic is represented by the green line.

4.3. SCR MPC with soft-constrained controlled variable

To avoid the overshoots in robust MPC (Fig. 5), the intuitive way is to implement the soft constraints on the controlled output to force the value of the controlled variable into the close neighbourhood of the set-point. Therefore, the following soft constraints on the controlled variable were considered:

$$\mathbb{Y}_{\text{soft}} = \{y \mid -2 \leq y \leq 2\}, \quad (19)$$

The soft constraints were chosen not to allow the controlled temperature to overcome the reference temperature more than 2°C . The penalisation matrix in (17a) was:

$$Q_{\text{soft}} = [0.07 \ 3]^T \times 10^3. \quad (20)$$

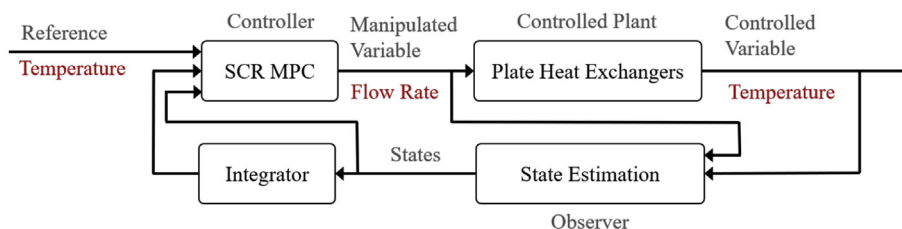
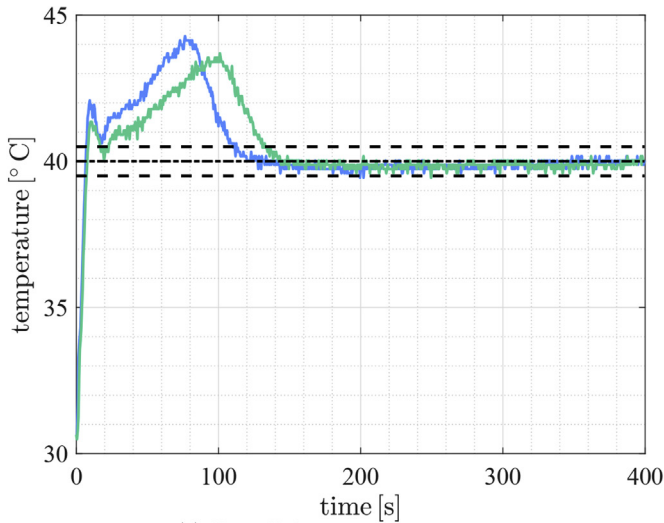
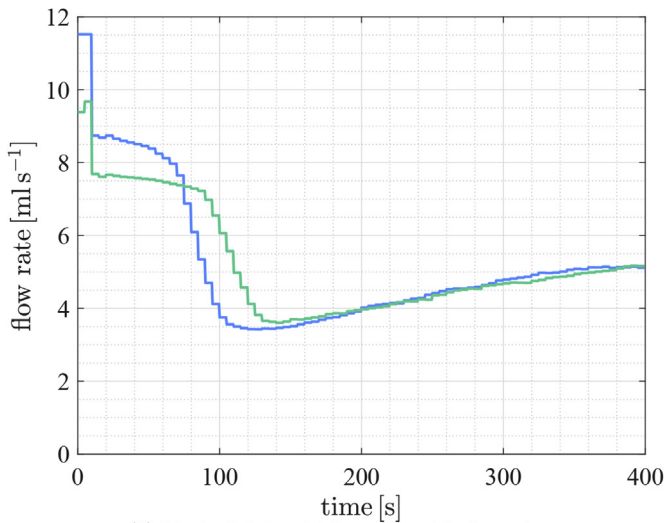


Fig. 4. Scheme of the closed-loop PHE control using SCR MPC with integral action.



(a) Controlled output – temperature.



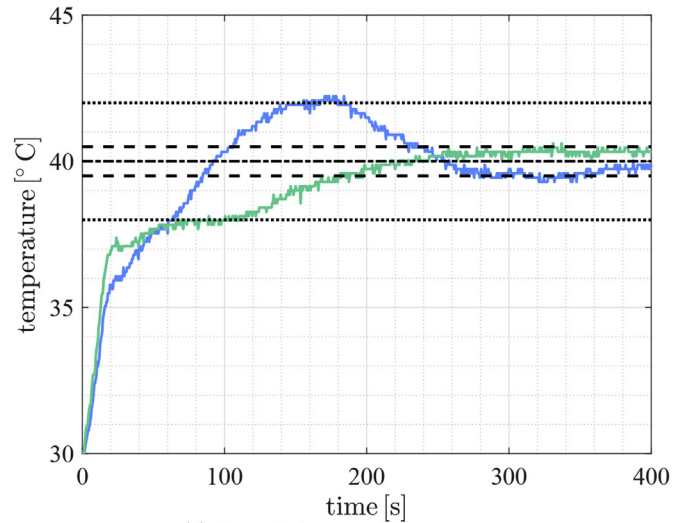
(b) Manipulated variable – volumetric flow rate.

Fig. 5. Control performance with robust MPC without soft-constraints: actuator with linear (blue) and non-linear (green) characteristics, set-point (dash-dotted), and allowed control inaccuracy (dashed). (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

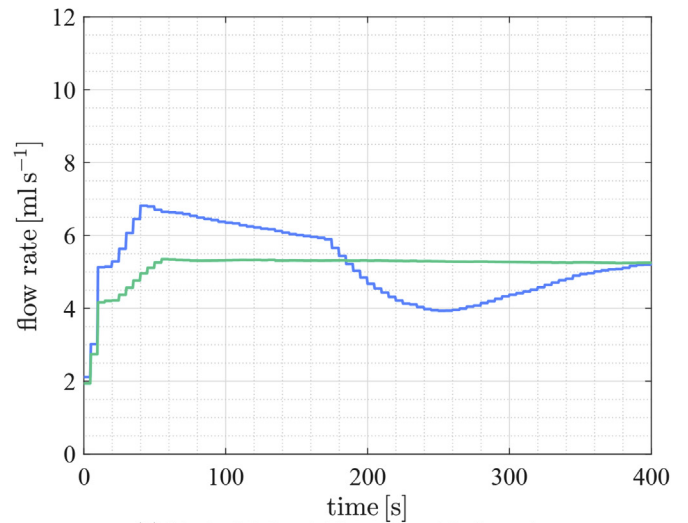
The control trajectories are depicted in Fig. 6, where the results obtained using the actuator with linear (Fig. 6(a), blue) and the non-linear (Fig. 6(a), green) characteristics are presented. SCR MPC with constrained CV ensured the set-point tracking without steady-state error with both actuators. SCR MPC and the actuator with the linear characteristic reduced the overshoot in about 2°C, meanwhile, SCR MPC with the actuator with the non-linear characteristic removed the overshoot. But, reducing of overshoots caused increasing of the settling time. The trajectories of the manipulated variables are shown in Fig. 6(b), where the volumetric flow rate of hot water pumped by the actuator with linear characteristics is drawn by the blue line and the volumetric flow rate of hot water dosed by the actuator with non-linear characteristics is presented by the green line.

4.4. SCR MPC with soft-constrained controlled and manipulated variables

This subsections investigates SCR MPC of the PHE with soft-



(a) Controlled output – temperature.



(b) Manipulated variable – volumetric flow rate.

Fig. 6. Control performance with SCR MPC and with soft-constrained CV: actuator with linear (blue) and non-linear (green) characteristics, set-point (dashed-dotted), allowed control inaccuracy (dashed), soft constraints on CV (dotted). (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

constrained both, the controlled variable and the manipulated variable. The soft constraints on CV had the form (19) and the penalisation matrix Q_{soft} was the same as in (20). The soft constraints on the manipulated variable were

$$\mathcal{U}_{\text{soft}} = \{u \mid -0.7 \leq u \leq 0.7\}, \quad (21)$$

to keep the volumetric flow rate close to its value corresponding to the set-point. The penalisation matrix in (17a) was:

$$R_{\text{soft}} = [30 \times 10^3]. \quad (22)$$

The control trajectories are depicted in Fig. 7. The trajectory of the controlled temperature obtained using SCR MPC and the actuator with the linear characteristic is represented by the blue line and the trajectory obtained by SCR MPC with the actuator with the non-linear characteristic by the green line. The trajectories of the controlled temperature were obtained without steady-state error in both cases, but the control responses are unacceptably

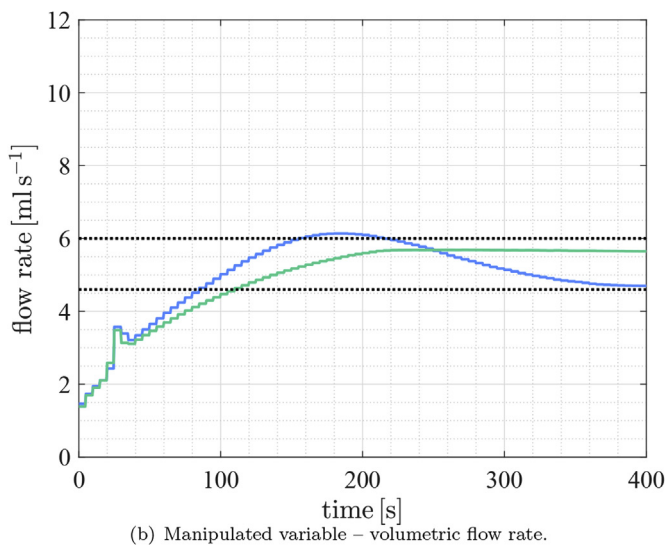
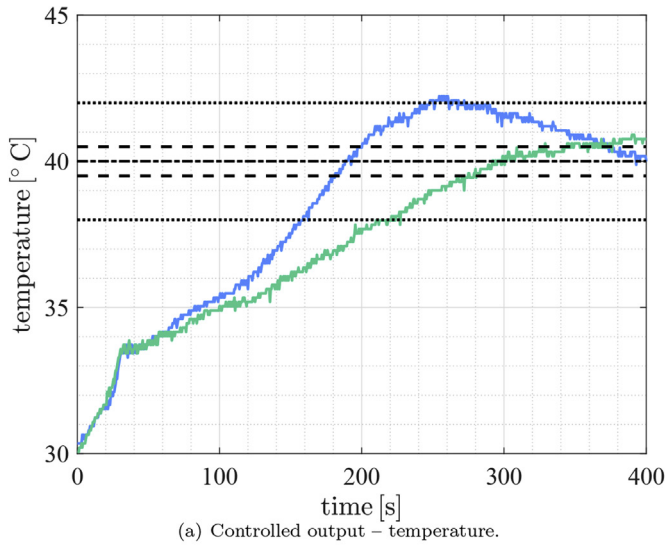


Fig. 7. Control performance of robust MPC with soft-constrained CV and MV: actuator with linear (blue) and non-linear (green) characteristics, set-point (dashed-dotted), allowed control inaccuracy (dashed), and soft constraints (dotted). (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

slow. As far as the overshoots, the results presented in Fig. 7(a) are comparable to the results in Fig. 6(a). The trajectories of the manipulated variable are shown in Fig. 7(b). The trajectory of the volumetric flow rate pumped into the PHE by the actuator with the linear characteristic is represented by the blue line and by the actuator with the non-linear characteristics by the green line.

4.5. SCR MPC with soft-constrained manipulated variable

This subsection investigates the SCR MPC with soft constraints on the manipulated variable. These soft constraints had the same form as in (21) and the penalisation matrix R_{soft} was as in (22). The aim was to minimize the overshoots by the optimized soft-constrained manipulated variable.

The control trajectories obtained by the SCR MPC with the soft-constrained MV are shown in Fig. 8. The controlled output obtained using the actuator with linear static characteristic is represented by the blue line and using the actuator with the non-linear

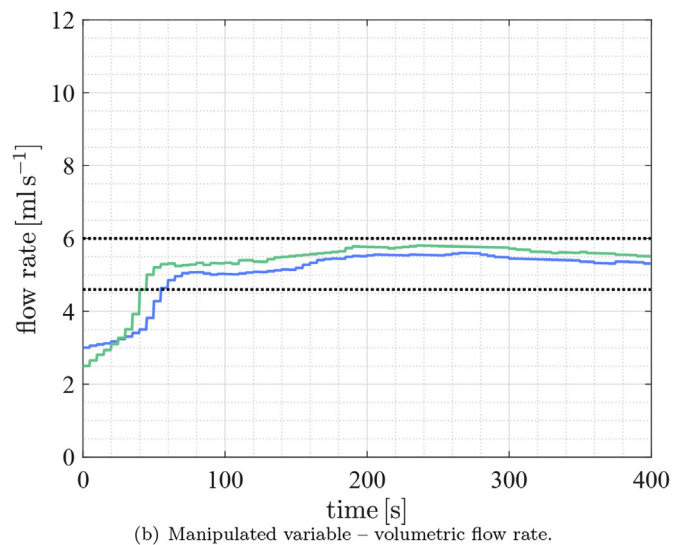
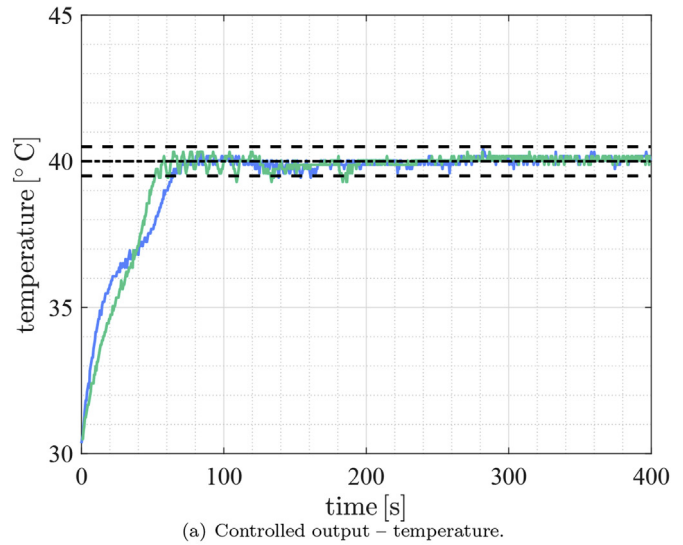


Fig. 8. Control performance of SCR MPC with soft-constrained MV: actuator with linear (blue) and non-linear (green) characteristics, set-point (dashed-dotted), allowed control inaccuracy (dashed), and soft constraints (dotted). (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

characteristics by the green line, see Fig. 8(a). The controlled output reached the setpoint without a steady-state error in both cases. In comparison to the control trajectories in Figs. 6(a) and 7(a), this approach significantly reduced the settling time. The settling time is even shorter than in robust MPC without soft constraints, cf. Fig. 5(a). The trajectories of the manipulated variable are shown in Fig. 8(b). The volumetric flow rate of the hot fluid dosed by the actuator with the linear characteristic is represented by the blue line and with the non-linear characteristic by the green line.

4.6. Analytical criteria of the control performance

Robust MPC of the PHE in all investigated control scenarios was analysed also using various analytical criteria. The performance of CV was evaluated by the maximum overshoot σ_{max} , the settling time t_{set} , and the integral criterion ISE . These criteria were calculated using the following formulas. The maximum overshoot is given as:

$$\sigma_{\max} = \frac{T_{\max} - T_{\text{ref}}}{T_{\text{ref}} - T_{\text{init}}} \times 100\%, \quad (23)$$

where T_{\max} is the maximum value of CV, T_{ref} is the reference value of CV, and T_{init} is the initial value of CV. The settling time t_{set} is the time that is needed to assure control with the required accuracy. In the PHE control, it was the time that CV needed to settle inside the interval $[39.5, 40.5]^\circ\text{C}$, i.e., the allowed control inaccuracy was $\pm 0.5^\circ\text{C}$. The integral criterion ISE is defined as:

$$ISE = \int_0^\infty e(t)^2 dt \approx \sum_{k=0}^{n_k} e(k)^2 t_s, \quad (24)$$

where e is the control error given by the difference between SP and CV, i.e., $e(k) = T_{\text{ref}} - T(k)$, n_k is the total number of the control steps, and t_s is the sampling time.

The performance of MV was evaluated by the integral criterion ISU , the total consumption of hot fluid V_{hot} , and the energy consumption E during control. The values of these criteria were calculated using the following formulas. The integral criterion ISU is given as:

$$ISU = \int_0^\infty (\dot{V}_{\text{hot}}(t) - \dot{V}_{\text{hot}}^s)^2 dt \approx \sum_{k=0}^{n_k} (\dot{V}_{\text{hot}}(k) - \dot{V}_{\text{hot}}^s)^2 t_s, \quad (25)$$

where \dot{V}_{hot} is MV and \dot{V}_{hot}^s is its steady-state value corresponding to the set-point. The total consumption of hot fluid V_{hot} is calculated as:

$$V_{\text{hot}} = \int_0^\infty \dot{V}_{\text{hot}}(t) dt \approx \sum_{k=0}^{n_k} \dot{V}_{\text{hot}}(k) t_s, \quad (26)$$

The total energy consumption E was calculated using following formula:

$$E = m_{\text{hot}} c_p \Delta T = \rho_{\text{hot}} V_{\text{hot}} c_p \Delta T, \quad (27)$$

where m_{hot} is the total mass of hot fluid consumed during control, c_p is the specific heat capacity of the hot fluid, ρ_{hot} is the density of the hot fluid, and ΔT is the temperature difference between the temperature of the hot fluid used for heating $T = 70^\circ\text{C}$ and the initial temperature of the hot fluid $T_{\text{init}} = 20^\circ\text{C}$ entering the heating tank. The total energy consumption E is represented as the heat needed for preparing hot water.

From the control viewpoint, the aim was to reach minimum

values of all introduced quality criteria. The values of all criteria are summarized in Tables 1 and 2, where the data in the first row, i.e., in the robust MPC design without soft constraints, served as the reference values.

As can be seen in Table 1, the overshoot σ_{\max} was reduced by introducing soft constraints into robust MPC design. When soft-constraints on controlled variable were active (scenario 2 and 3), the overshoot was twice reduced. The zero value of overshoot σ_{\max} was ensured when only soft-constraints on manipulated variable were considered (scenario 4). The settling time t_{set} was reduced when only soft-constrained controlled variable (scenario 2) or only soft-constrained manipulated variable (scenario 4) was used. The minimum value of the settling time t_{set} was ensured when only soft-constraints on manipulated variable were considered (scenario 4). Considering both, the constraints of CV and MV gave the worst results. The integral criteria ISE and ISU increased when soft constraints on the controlled variable were active (scenarios 2 and 3). The minimum values of integral criteria were ensured by the soft-constrained manipulated variable (scenario 4). Analogous performance analysis holds also for the total consumption of hot fluid V_{hot} and total energy consumption E . To conclude, the analytical quality criteria confirmed the improved control performance and energy savings ensured by introducing the soft constraints on manipulated variable into robust MPC design.

As can be seen in Table 2, the overshoot σ_{\max} was reduced by introducing soft constraints into robust MPC design. The zero values of the overshoot σ_{\max} were ensured when only soft-constrained controlled variable (scenario 2) or only soft-constrained manipulated variable (scenario 4) was used. The overshoot was four times smaller with soft constraints on CV and MV (scenario 3). The settling time t_{set} was reduced only with the soft constraints on the manipulated variable (scenario 4). The integral criteria ISE and ISU increased when soft constraints were introduced (scenarios 2, 3, 4). The minimum increasing of ISE and ISU was reached with the soft-constrained manipulated variable (scenario 4). Analogous performance analysis holds also for the total consumption of hot fluid V_{hot} and total energy consumption E .

The relative squared errors of the soft-constraints' violation resulting from the implementation of SCR MPC with integral action on PHE were also investigated. The relative squared errors were calculated as follows

$$\varepsilon_T = \int_0^{t_{\text{reg}}} \left(\max\left(0; T - \left(T_{\text{ref}} + T_{\text{soft}}\right)\right) \right)^2 dt, \quad (28a)$$

Table 1
Values of the quality criteria for robust MPC of PHE using the actuator with the linear static characteristic.

scenario	soft constraints	σ_{\max} [%]	t_{set} [s]	ISE [$^\circ\text{C}^2$ s]	ISU [ml^2]	V_{hot} [ml]	E [kJ]
1	not used	44	107	1163	12166	2082	428
2	CV	20	85	2160	24132	3081	634
3	CV & MV	22	181	5961	22659	2980	613
4	MV	0	64	1035	5650	1313	417

Table 2
Values of the quality criteria for robust MPC of PHE using the actuator with the non-linear static characteristic.

scenario	soft constraints	σ_{\max} [%]	t_{set} [s]	ISE [$^\circ\text{C}^2$ s]	ISU [ml^2]	V_{hot} [ml]	E [kJ]
1	not used	43	104	845	11015	2036	418
2	CV	0	182	1793	23482	3058	629
3	CV & MV	12	565	6653	22551	2972	612
4	MV	0	51	1286	11406	2106	438

$$\varepsilon_V = \int_0^{t_{\text{reg}}} \left(\max \left(0; \dot{V}_{\text{hot}} - \left(\dot{V}_{\text{hot}}^S + \dot{V}_{\text{hot,soft}} \right) \right) \right)^2 dt, \quad (28b)$$

$$\varepsilon_{\text{total}} = \varepsilon_T + \varepsilon_V, \quad (28c)$$

where ε_T , ε_V , $\varepsilon_{\text{total}}$, respectively, are the relative squared error of the CV soft-constraints' violation, the relative squared error of the MV soft-constraints' violation, and the total relative squared error of the CV and MV soft-constraints' violation. $T_{\text{soft}} = 2^\circ\text{C}$ and $\dot{V}_{\text{hot,soft}} = 0.7\text{ml s}^{-1}$ are the soft-constraints on CV and MV, respectively. $T_{\text{ref}} = 40^\circ\text{C}$ is the reference temperature and $\dot{V}_{\text{hot}}^S = 5.3\text{ ml s}^{-1}$ is the steady-state value of MV corresponding with the reference temperature. $t_{\text{reg}} = 400\text{ s}$ is the total time of control. The values ε_T , ε_V , $\varepsilon_{\text{total}}$ were calculated also in the cases if soft constraints on CV and/or MV were not active and therefore the errors are denoted as relative. Table 3 summarizes the computed relative squared errors of the soft-constraints' violation for the actuator with the linear static characteristic. The minimum value of $\varepsilon_{\text{total}}$ was ensured considering only soft-constrained MV (scenario 4). The larger value of ε_V than ε_T was obtained only in the scenario 1 when no soft constraints were active. The biggest relative squared errors of the soft-constraints' violation ε_T and $\varepsilon_{\text{total}}$ resulted in the scenario 3 when simultaneously soft constraints on CV and MV were active. Table 4 shows the errors for the actuator with the non-linear static characteristic. The results are analogous to the results presented in Table 3.

The analytical quality criteria confirmed that SCR MPC with soft-constrained manipulated variable together with the actuator with the non-linear characteristic ensured the best control performance subject to the maximum overshoot and the minimum settling time. The other quality criteria had the minimum values when SCR MPC with soft-constrained MV and the actuator with the linear characteristic was used. Using the actuator with the non-linear characteristic ensures the minimum values of all quality criteria only in robust MPC without soft constraints.

5. Conclusions

The main contributions of this paper can be summarized as follows: (i) design of SCR MPC with integral action for the laboratory plate heat exchanger; (ii) experimental closed-loop control of the laboratory PHE using SCR MPC with integral action; (iii) using

Table 3
Relative squared errors of the soft-constraints' violation for robust MPC of PHE using the actuator with the linear static characteristic.

scenario	soft constraints	ε_T	ε_V	$\varepsilon_{\text{total}}$
1	not used	443	1610	2052
2	CV	1397	346	1743
3	CV & MV	4983	746	5729
4	MV	986	182	1169

Table 4
Relative squared errors of the soft-constraints' violation for robust MPC of PHE using the actuator with the non-linear static characteristic.

scenario	soft constraints	ε_T	ε_V	$\varepsilon_{\text{total}}$
1	not used	414	1762	1176
2	CV	1059	189	1248
3	CV & MV	5625	831	6456
4	MV	1380	197	1578

the actuator with the linear and non-linear static characteristics; the second one used for compensation of the non-linear behaviour of the plate heat exchanger; (iv) extensive analysis of the control performance of the PHE using various quality criteria in various SCR MPC scenarios in the set-point tracking.

The overshoot of the controlled variable, i.e., the temperature of the heated fluid at the outlet of the PHE was removed by introducing the soft constraints on the manipulated variable, i.e., the volumetric flow rate of the heating fluid. Simultaneously, all other quality criteria and the energy consumption were minimum when the actuator with the linear static characteristic was used. Using the actuator with the nonlinear characteristic outperformed the actuator with linear static characteristic only in robust MPC without soft constraints. The integral action ensured the offset-free control performance in all investigated control scenarios.

Finally, using soft constraints on manipulated variable improved the control performance and is recommended. Compensation of the non-linearity of the controlled process using the actuator with the non-linear static characteristic did not offer better values of the quality criteria in comparison with the actuator with the linear static characteristic.

Further research will be focused on the design and implementation of the advanced robust MPC strategies for the complex interconnected systems, such as heat exchanger networks. From an implementation viewpoint, the aim is to develop the complexity reduction techniques to make the real-time application of robust MPC more attractive.

Acknowledgement

The authors gratefully acknowledge the contribution of the Scientific Grant Agency of the Slovak Republic under the grants 1/0112/16, 1/0585/19, the Slovak Research and Development Agency under the project APVV-15-0007, and the Research & Development Operational Programme for the project University Scientific Park STU in Bratislava, ITMS 26240220084, supported by the Research 7 Development Operational Programme funded by the ERDF.

References

- [1] Klemeš J, Varbanov P. Heat transfer improvement, energy saving, management and pollution reduction. *Energy* 2018;162:267–71. <https://doi.org/10.1016/j.energy.2018.08.014>.
- [2] Mayne DQ. Model predictive control: recent developments and future promise. *Automatica* 2014;12:2967–86. <https://doi.org/10.1016/j.automatica.2014.10.128>.
- [3] Maciejowski J. *Predictive control with constraints*. London: Prentice Hall; 2000.
- [4] Cutler CR, Ramaker BL. Dynamic matrix control – a computer control algorithm. In: *American institute of chemical engineers (AIChE) national meeting*; 1979.
- [5] Bemporad A, Morari M. Robust model predictive control: a survey. In: *Robustness in identification and control*. Springer London; 1999. p. 207–26.
- [6] Kothare MV, Balakrishnan V, Morari M. Robust constrained model predictive control using linear matrix inequalities. *Automatica* 1996;32:1361–79.
- [7] Zhang L, Wang J, Li K. Min-max MPC for LPV systems subject to actuator saturation by a saturation-dependent Lyapunov function. In: *Chinese control conference*. China: Xi'an; 2013. p. 4087–92.
- [8] Violaro F, Rivera E, Alvarez L. Multivariable model predictive control of a continuous fermentation unit for first-generation ethanol production. *Chem. Eng. Trans.* 2018;(65):67–72. <https://doi.org/10.3303/CET1865012>.
- [9] Markowski M, Trzcinski P. On-line control of the heat exchanger network under industrial constraints. *Chem. Eng. Trans.* 2018;(70):517–22. <https://doi.org/10.3303/CET1870087>.
- [10] Trafczynski M, Markowski M, Urbaniec K, Grabarczyk R. Energy saving potential and the efficacy of using different control strategies for the heat exchanger network operation. *Chem. Eng. Trans.* 2018;(70):823–8. <https://doi.org/10.3303/CET1870138>.
- [11] Pospíšil J, Špiláček M, Kudela L. Potential of predictive control for improvement of seasonal coefficient of performance of air source heat pump in central european climate zone. *Energy* 2018;154:415–23. <https://doi.org/10.1016/j.energy.2018.04.131>.

- [12] Zhang P, Ma T, Li W, Ma G, Wang Q. Design and optimization of a novel high temperature heat exchanger for waste heat cascade recovery from exhaust flue gases. *Energy* 2018;160:3–18. <https://doi.org/10.1016/j.energy.2018.06.216>.
- [13] Dong Z, Zhang Z, Dong Y, Huang X. Multi-layer perception based model predictive control for the thermal power of nuclear superheated-steam supply systems. *Energy* 2018;151:116–25. <https://doi.org/10.1016/j.energy.2018.03.046>.
- [14] Ponce CV, Sáez D, Bordons C, Núñez A. Dynamic simulator and model predictive control of an integrated solar combined cycle plant. *Energy* 2016;109:974–86. <https://doi.org/10.1016/j.energy.2016.04.129>.
- [15] Yang T, Wang W, Zeng D, Liu J, Cui C. Closed-loop optimization control on fan speed of air-cooled steam condenser units for energy saving and rapid load regulation. *Energy* 2017;135:394–404. <https://doi.org/10.1016/j.energy.2017.06.142>.
- [16] Zhang J, Lin M, Chen J, Xu J, Li K. PLS-based multi-loop robust H_2 control for improvement of operating efficiency of waste heat energy. *Energy* 2017;123:460–72. <https://doi.org/10.1016/j.energy.2017.01.131>.
- [17] Oravec J, Bakošová M, Trafczynski M, Vasičkaninová A, Mészáros A, Markowski M. Robust model predictive control and pid control of shell-and-tube heat exchangers. *Energy* 2018;159:1–10. <https://doi.org/10.1016/j.energy.2018.06.106>.
- [18] Oravec J, Bakošová M, Pakšiová D, Mikušová N, Batárová K. Advanced robust MPC design of a heat exchanger: modeling and experiments. In: 27th european symposium on computer aided process engineering, Barcelona, Spain; 2017. p. 1585–90.
- [19] Fraczak M, Nowak P, Czeczot J, Metzger M. Simplified dynamical input–output modeling of plate heat exchangers – case study. *Appl Therm Eng* 2016;98:880–93. <https://doi.org/10.1016/j.applthermaleng.2016.01.004>.
- [20] Fraczak M, Czeczot J, Nowak P, Metzger M. Practical validation of the effective control of liquid–liquid heat exchangers by distributed parameter balance-based adaptive controller. *Appl Therm Eng* 2018;129:549–56. <https://doi.org/10.1016/j.applthermaleng.2017.10.056>.
- [21] Oravec J, Bakošová M, Mészáros A, Míková N. Experimental investigation of alternative robust model predictive control of a heat exchanger. *Appl Therm Eng* 2016;105:774–82. <https://doi.org/10.1016/j.applthermaleng.2016.05.046>.
- [22] Oravec J, Bakošová M. Soft constraints in the robust MPC design via LMIs. In: *American control conference, Boston, Massachusetts, USA*; 2016. p. 3588–93.
- [23] Oravec J, Pakšiová D, Bakošová M, Fikar M. Soft-constrained alternative robust MPC: experimental study. *Preprints of the 20th IFAC world congress, vol. 20*; 2017. p. 11877–82. <https://doi.org/10.1016/j.ifacol.2017.08.2043>. Toulouse, France.
- [24] Oravec J, Bakošová M, Vasičkaninová A, Mészáros A. Robust model predictive control of a plate heat exchanger. *Chem. Eng. Trans.* 2018;70:25–30. <https://doi.org/10.3303/CET1870005>.
- [25] Armfield. *Extracts from instruction manual PCT40*. Armfield; 2007.
- [26] Lipták BG. *Instrument engineers' handbook. In: Process control and optimization, 4th edition, chapter 8.29 heat exchanger control and optimization, vol. 2*. USA: CDC Press; 2006.
- [27] Boyd S, Vandenberghe L. *Convex optimization*. New York, USA: Cambridge University Press; 2004.
- [28] Mikleš J, Fikar M. *Process modelling, identification, and control*. Berlin Heidelberg: Springer Verlag; 2007.
- [29] Oravec J, Bakošová M. *MUP toolbox for MATLAB*. 2014. <https://bitbucket.org/oravec/mup/wiki/Home>.
- [30] Löfberg J. *YALMIP : a toolbox for modeling and optimization in MATLAB*. In: *Proc. Of the CACSD conf., Taipei, Taiwan*; 2004. URL, <http://users.isy.liu.se/johan/yalmp>.
- [31] MOSEK ApS. *MOSEK*. 2019. <https://mosek.com/>.
- [32] Kalúz M, Cirka L, Valo R, Fikar M. *Lab of things: a network-based I/O services for laboratory experimentation*. *Preprints of the 20th IFAC world congress, Toulouse, vol. 20*. France; 2017. p. 14028–33.

Nomenclature

Symbols

A : system matrix of the state space system
 $A^{(0)}$: system matrix of the nominal state space system
 $A^{(v)}$: vertex system matrix of the uncertain state space system
 \tilde{A} : system matrix of the augmented state space system
 $\tilde{A}^{(v)}$: vertex system matrix of the augmented uncertain state space system
 \mathbb{A} : set of uncertain state space systems
 $\tilde{\mathbb{A}}$: set of augmented uncertain state space systems
 B : input matrix of the state space system
 $B^{(0)}$: input matrix of the nominal state space system
 $B^{(v)}$: vertex input matrix of the uncertain state space system
 \tilde{B} : input matrix of the augmented state space system
 $\tilde{B}^{(v)}$: vertex input matrix of the uncertain augmented state space system

C : output matrix of the state space system
 \tilde{C} : output matrix of the augmented state space system
 c_p : specific heat capacity, $\text{kJ kg}^{-1} \text{K}^{-1}$
 e : control error, $^{\circ}\text{C}$
 E : total energy consumption, kJ
 E_u : auxiliary matrix of soft constraints evaluation
 E_y : auxiliary matrix of soft constraints evaluation
 \tilde{F} : gain of the state-feedback control law
 F_I : integral gain of the state-feedback control law
 F_P : proportional gain of the state-feedback control law
 h : auxiliary variable of non-linear static characteristic
 I : identity matrix
 J : quadratic quality criterion
 k : sample of discrete time domain, s
 K : auxiliary matrix of controller design
 m_{hot} : total mass of hot fluid, kg
 n_k : number of control steps
 n_u : total number of system inputs
 n_v : total number of system vertices
 n_x : total number of system states
 n_y : total number of system outputs
 P : Lyapunov matrix
 Q : weighting matrix of system states
 Q_{soft} : weighting on soft-constrained controlled variable
 \tilde{Q} : weighting matrix of the augmented system states
 R : weighting matrix of system inputs
 R_{soft} : weighting on soft-constrained manipulated variable
 \mathbb{R} : Euclidean space of real numbers
 \tilde{s}_u : squared value of soft constraints violation on MV
 \tilde{s}_y : squared value of soft constraints violation on CV
 t : time, s
 t_{reg} : total time of control, s
 t_s : sampling time, s
 t_{set} : settling time, s
 T : controlled temperature, $^{\circ}\text{C}$
 T_{init} : initial value of temperature, $^{\circ}\text{C}$
 T_{hot} : temperature of hot fluid, $^{\circ}\text{C}$
 T_{max} : maximal value of temperature, $^{\circ}\text{C}$
 T_{ref} : reference temperature, $^{\circ}\text{C}$
 T_{soft} : soft constraints on temperature, $^{\circ}\text{C}$
 T^s : steady-state temperature of hot fluid, $^{\circ}\text{C}$
 ΔT : temperature difference, $^{\circ}\text{C}$
 u : inputs of the state space system, ml s^{-1}
 u_{max} : maximal value of the control input, ml s^{-1}
 u_{sat} : saturated control input of the state space system, ml s^{-1}
 U : voltage, V
 U_{norm} : normalized voltage, 1
 U_{max} : auxiliary matrix of controller design
 \mathbb{U}_{max} : set of hard constraints on the controlled variables
 U_{soft} : auxiliary matrix of controller design
 \mathbb{U}_{soft} : set of soft constraints on the manipulated variables
 v : vertex system
 V : Lyapunov function
 V_{hot} : total consumption of hot fluid, ml
 \dot{V}_{hot} : volumetric flow rate, ml s^{-1}
 $\dot{V}_{hot,soft}$: soft constraints on volumetric flow rate, ml s^{-1}
 \dot{V}_{hot}^s : steady-state volumetric flow rate, ml s^{-1}
 \dot{V}_{min} : minimum volumetric flow rate, ml s^{-1}
 \dot{V}_{max} : maximum volumetric flow rate, ml s^{-1}
 $\Delta \dot{V}_{hot}$: volumetric flow rate difference, ml s^{-1}
 w : set-point, $^{\circ}\text{C}$
 x : states of the state space system, $^{\circ}\text{C}$
 x_0 : initial conditions of the state space system, $^{\circ}\text{C}$
 X : weighted inverted Lyapunov matrix
 y : outputs of the state space system, $^{\circ}\text{C}$
 y_{max} : maximal value of output variable, $^{\circ}\text{C}$
 Y : auxiliary matrix of controller design
 \mathbb{Y}_{max} : set of hard constraints on the controlled variables
 Y_{max} : auxiliary matrix of controller design
 Y_{soft} : auxiliary matrix of controller design
 \mathbb{Y}_{soft} : set of soft constraints on the controlled variables
 z : states of the augmented state space system, $^{\circ}\text{C}$
 z_0 : initial conditions of the augmented state space system
 Z : auxiliary matrix of controller design
 0 : zero matrix

Greek letters

ϵ_{total} : relative squared error of total soft constraints violation
 ϵ_T : relative squared error of soft constraints violation on temperature
 ϵ_V : relative squared error of soft constraints violation on volumetric flow rate

γ : weighting parameter of Lyapunov matrix
 λ : auxiliary weighting parameter of convex combination
 ρ : density, kg m^{-3}
 σ_{max} : maximal overshoot, %
 τ : auxiliary parameter of non-linear characteristic

Abbreviations

CV: controlled variable
DMC: dynamic matrix control

GPC: generalized predictive control
ISE: integral square error of controlled variable
ISU: integral square error of manipulated variable
LMI: linear matrix inequality
MPC: model predictive control
MV: manipulated variable
PHE: plate heat exchanger
PP: peristaltic pump
SCR MPC: soft-constrained robust model predictive control
SDP: semidefinite programming



Approximated MPC for embedded hardware: Recursive random shooting approach

Peter Bakaráč, Michaela Horváthová^{*}, Lenka Galčíková, Juraj Oravec, Monika Bakošová

Slovak University of Technology in Bratislava, Faculty of Chemical and Food Technology, Institute of Information Engineering, Automation, and Mathematics, Radlinského 9, Bratislava SK812-37, Slovak Republic

ARTICLE INFO

Keywords:
Model predictive control
Random shooting
Embedded hardware

ABSTRACT

Advanced, optimization-based control methods are implemented at each level of industrial production. Although the model predictive control (MPC) represents a state-of-the-art control strategy maximizing profit, and, simultaneously, minimizing the energy losses, its industrial implementation is limited by the requirements on the software and hardware resources. This paper proposes an approximated MPC for such cases when the implementation of implicit MPC has a prohibitive effort on industrial hardware and its explicit counterpart is limited by its memory footprint requirements. The recursive random shooting-based approach is introduced to eliminate the conventional optimization and minimize the memory requirements, meanwhile guarantying the asymptotic stability subject to the physical constraints on control inputs and system states. The benefits of the proposed method are significantly reduced computational complexity by 93%, and decreased energy consumption by 93%, compared to non-recursive random shooting, while the performance loss is approximately 8%.

1. Introduction

In recent years, model predictive control (MPC) became a standard control approach for plants in the various fields of industry (Qin and Badgwell, 2003). MPC is the most effective method for control of complex multivariable systems, where technological constraints and control performance optimization are requested (Mayne, 2014). The optimal control problem of conventional MPC is solved in a *receding-horizon* manner, meaning that it is solved repeatedly over a finite sequence of control inputs $\{u_0, u_1, \dots, u_{N-1}\}$ for prediction horizon N at each sample, while the current states of the system are updated. After the state measurement, only the first control input out of the sequence is applied to the controlled plant.

When considering the widely-used LQR-like penalty function, then the MPC formulation leads to the problem of the quadratic programming (QP), e.g., see Boyd and Vandenberghe (2004). The problem of large-scale QPs in the MPC design framework was investigated in Bartlett et al. (2002), where an effective method to solve large-scale QPs using the Schur complement algorithm is proposed. In Brand et al. (2011), another method to speed-up solving QPs was presented. The evaluation time was significantly reduced using the parallel computations to solve the QPs. In the framework of the embedded MPC design,

the active set method based on nonnegative least squares to solve strictly convex QPs was presented in Bemporad (2016). The benefits of the proposed method are the reduced evaluation time, ease of implementation, and an effective evaluation of the optimization problem feasibility.

Embedded hardware is becoming increasingly popular due to its low-cost, open architecture, and a wide range of relevant applications. However, the computational power and memory storage of the embedded hardware is still low compared to other devices used in the industry. As the consequence, there is a significant requirement to implement such control strategies that provide inexpensive and low-power consuming controllers. For instance, the well-known proportional-integral-derivative (PID) controllers. In Youness et al. (2014) a detailed case study analyzing the implementation of various PID controllers considering multiprocessor system-on-chip and multicore microcontrollers is conducted. The designed controllers are compared in terms of design time, design effort, and closed-loop control performance. The paper also summarizes the rules for handling the multiprocessor systems for embedded control design.

A robust fuzzy PID controller implemented using a low-cost microcontroller for an uncertain model of an inverted pendulum was designed in El-Nagar and El-Bardini (2014). The laboratory implementation

^{*} Corresponding author.

E-mail address: michaela.horvathova@stuba.sk (M. Horváthová).

demonstrates that even with the limited memory and computational capacity of the embedded hardware, the implemented controller successfully stabilized a strongly uncertain system.

Although the PID controllers are suitable for embedded platforms, they lack the guarantees of constraints satisfaction and the closed-loop system stability. To overcome this obstacle, the MPC design method were implemented on the embedded hardware. Among other methods, a dynamic matrix control (DMC) and a generalized model predictive control (GPC) were designed and investigated. In [Chaber and Lawrynczuk \(2019\)](#), the fast DMC and fast GPC variants of MPC were proposed and analyzed using a microcontroller. Compared to the conventional MPC, considering DMC and GPC methods ensured a significant reduction in evaluation time.

A recent paper ([Ndje et al., 2021](#)) presents a computationally effective method of a nonlinear DMC implemented on a microcontroller. This approach reduces the computational burden by solving an optimal control action just if it is necessary, otherwise, applies a solution of unconstrained optimization.

Although various modifications of MPC controllers were designed for embedded platforms, there are still barriers limiting its widescale implementation, especially, when handling the multivariable systems with fast dynamics. Therefore, there is still a significant requirement to design and implement an MPC-like control strategy ensuring close-to-optimal control performance on embedded hardware that is inexpensive and low-power consuming, simultaneously, guaranteeing the constraints satisfaction, recursive feasibility, and the closed-loop system stability.

A significant number of methods were introduced to decrease the computational effort of a conventional MPC. One way to reduce the computational effort is to solve the optimal control problem *offline*. It is possible to determine the optimal control law for every possible combination of initial conditions utilizing parametric programming. This technique is also known as *explicit* MPC, see [Bemporad et al. \(2002\)](#). The main benefit of explicit MPC is preserving the optimality of the solution, and, simultaneously, removing the real-time optimization.

Practical issues of hardware synthesis of explicit MPC controllers are analyzed in [Johansen et al. \(2007\)](#). In this work, the small-scale explicit MPC was designed with a special focus on the implementation on the application-specific integrated circuit (ASIC) and field-programmable gate array (FPGA). One of the showcases of the direct implementation of MPC in embedded hardware is presented in [Hrustic and Prljaca \(2020\)](#).

The case study presented in [Shoukry et al. \(2010\)](#), performs the implementation of MPC in safety-critical automotive systems. In [Krishnamoorthy and Skogestad \(2022\)](#), the authors provide an overview of various different approaches that also aim to preserve the optimal solution of the MPC controller, without the need to solve optimization problems *online*. The main limitation of these control methods is a hardly tractable memory footprint required to store the explicit solution. Moreover, the corresponding point-location problem can become overly time-demanding.

To overcome these obstacles, various methods introducing an approximation of the optimal feedback control law were developed. In [Chen et al. \(2018\)](#), the neural networks were considered in the design of an explicit linear feedback law. The downside of these types of methods is the loss of optimality and the lack of any stability guarantee. Moreover, one still needs to evaluate the sufficiently large set of optimal control problems to generate the data for the neural network training.

In the past two decades, the stochastic methods as *random shooting*-based (RS) method gained popularity. The RS-based approach in the robust controller design framework was analyzed in [Vidyasagar \(2001a\)](#). Here, the statistical learning theory was considered to introduce the randomized algorithms for a wide class of controller design problems. The detailed tutorial overview on randomized algorithms for robust controller synthesis using statistical learning theory was presented in [Vidyasagar \(2001b\)](#). A generalized approach to solve the

convex optimization problems using the randomized algorithms was introduced in [Dyer et al. \(2014\)](#). Here, under some mild assumptions, the proposed algorithm evaluated a close-to-optimal solution in a number of iterations that is limited by a polynomial function.

Similar to the neural-network-based MPC, RS leads to a suboptimal solution and lacks any type of stability guarantee. In contrast with the neural-network-based MPC, it does not require solving any type of optimal control problem. Therefore, it is fully implementable on embedded platforms widely used in industrial applications and so-called industrial internet of things (IIoT) services.

The control performance ensured by designing RS-based controllers was analyzed in several case studies. The numerical simulation of the closed-loop control of the nonlinear chemical reactor governed by an RS-based controller was investigated in [Bakarác and Kvasnica \(2018\)](#). The RS-based approach significantly reduced the complexity of the prediction of the nonlinear behavior of the chemical reactor. In [Piovesan and Tanner \(2009\)](#), the problem of nonlinear MPC design was approximated using the randomized algorithms. In [Fedorová et al. \(2019\)](#), the agile maneuvers of a pendulum on a cart were controlled by RS-based approximation of nonlinear MPC. Here, the control performance was investigated by the implementation of the designed RS-based controller on the laboratory plant. Based on the results presented in these papers, and the references therein, the RS-based approach can be considered a close-to-optimal controller design method suitable for challenging systems and for the systems with fast dynamics.

The RS method is based on random generation of control sequences. Randomly generated control sequences are inquired for feasibility, i.e., constraint satisfaction, and their performance indices. The feasible random sequence with the best performance index is determined at the end of the runtime and the first element of this sequence is applied to the system. Even though the selected control sequence is suboptimal, it is still feasible, thus guaranteeing a safe operation. Suboptimality of the solution decreases with an increasing number of generated control sequences. Specifically, if the considered optimization problem is convex, runtime and the suboptimality of the solution can be estimated in advance ([Dyer et al., 2014](#)). The versatility of this method does not end there. It allows to solve problems, where the prediction model or cost function are discontinuous, or even the constraints are non-convex.

The main contribution of this work is to propose extremely lightweight implementation of close-to-optimal RS-based control on the embedded/industrial hardware. The aim is to merge the benefits of both methods, i.e., optimal evaluation of control input using MPC and RS-based approximation of optimal solution enabling library-free, and easy-to-implement computation of close-to-optimal control input on embedded platforms. This method results in straightforward and extremely fast implementation of control input.

In this work, we introduce a novel *recursive* RS-based control approach overcoming the main drawbacks of its well-known counterpart limiting its wide industrial application. Namely, the presented method provides guarantees on the physical constraint satisfaction, recursive feasibility, and asymptotic stability.

Technically, we introduce a so-called *support controller* to guarantee these closed-loop system properties, e.g., see [Oravec et al. \(2017\)](#) and references therein. The idea of a support (backup) controller is widely-known, but to the best authors' knowledge, this approach has not been introduced in the random shooting-based approximation of MPC design, yet.

The benefit of this approach, compared to the earlier works is, that we use the precomputed support controller just to initialize the evaluation of an asymptotically stable controller, and its closed-loop performance is continuously improved by recursive learning strategy.

Moreover, the other significant benefit of the proposed method is scaling of the time necessary to evaluate close-to-optimal control input, while, simultaneously, preserving the above-listed benefits.

The paper is organized as follows: first, the preliminaries containing the concepts of MPC and its formulations are introduced. In the

following section, the novel method of recursive random shooting computing the close-to-optimal control input is presented. The next section analyses the case study generated using the embedded hardware, together with a discussion focused on the main benefits of the proposed control strategy. Finally, the main conclusions are summarized.

2. Preliminaries

2.1. Model predictive control

The MPC represents the state-of-the-art optimization-based control method worth implementing in the industrial conditions to increase the yields and the associated profit, and, simultaneously, minimize the energy consumption and reduce the corresponding carbon footprint. Throughout this paper, we consider a following MPC formulation:

$$\min_{u_0, u_1, \dots, u_{N-1}} x_N^T P x_N + \sum_{k=0}^{N-1} (x_k^T Q x_k + u_k^T R u_k) \quad (1a)$$

$$\text{s.t.: } x_{k+1} = A x_k + B u_k, \quad (1b)$$

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

$$x_k \in \mathcal{X}, \quad (1d)$$

$$x_N \in \mathcal{T}, \quad (1e)$$

$$x_0 = x(t), \quad (1f)$$

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

where N is prediction horizon, $P \in \mathbb{R}^{n_x \times n_x}$, $Q \in \mathbb{R}^{n_x \times n_x}$, $R \in \mathbb{R}^{n_u \times n_u}$ are terminal, state, and input penalty matrices, respectively. Prediction model in (1b) has the form of linear time invariant (LTI) system for given state matrix $A \in \mathbb{R}^{n_x \times n_x}$ and input matrix $B \in \mathbb{R}^{n_x \times n_u}$. In LTI system (1b), $x \in \mathbb{R}^{n_x}$, $u \in \mathbb{R}^{n_u}$ are vectors of corresponding system states and control inputs, respectively. $\mathcal{U} \subseteq \mathbb{R}^{n_u}$, $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ are sets of physical constraints on inputs and system states, respectively. $\mathcal{T} \subseteq \mathcal{X}$ is set of terminal constraint.

Assumption 2.1. Let MPC problem (1) be asymptotically stable and recursively feasible. Assume, in (1) hold:

1. sets $\mathcal{U}, \mathcal{X}, \mathcal{T}$ are closed convex polyhedra containing the origin in their strict interiors,
2. matrices $P \succ 0, Q \succ 0, R \succ 0$ are positive definite,
3. terminal set $\mathcal{T} \subseteq \mathcal{X}$ is control invariant set, where no physical constraints in (1c)–(1d) are active.

From the first point of Assumption 2.1, it follows that the domain of the optimization variables represents a convex set. From the second point of Assumption 2.1, it directly follows that the objective function is strictly convex. Therefore, solving the MPC problem in (1) leads to the strictly convex optimization problem of quadratic programming having a unique optimizer, see Boyd and Vandenberghe (2004).

For a sake of simplicity, we introduce a simplified notation for penalty function in (1a) as follows:

$$J = x_N^T P x_N + \sum_{k=0}^{N-1} (x_k^T Q x_k + u_k^T R u_k) = \ell_N(x_N) + \sum_{k=0}^{N-1} \ell(x_k, u_k), \quad (2)$$

where $\ell_N : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^1$ is a quadratic terminal penalty function and $\ell : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^1$ is a quadratic stage cost function.

The MPC is implemented in so-called receding horizon fashion, i.e., in each control step just the first control input is applied to the controlled plant and the optimization problem is resolved in the next control step for the updated system measurement, see Maciejowski (2000). The MPC formulation in (1) considers an idealized system without a significant

impact of uncertain parameters. If the inherent robustness of the receding horizon framework is not sufficient, then we may robustify the MPC controller, e.g., see Bemporad and Morari (1999).

Remark 2.2. (Robust terminal set) In case the MPC formulation in (1) is implemented on the plant, where plant-model mismatch threatens the invariant properties of the maximal control invariant set \mathcal{T} in (1f) (Borrelli, 2017), the robust control invariant set $\mathcal{T}_{\text{robust}}$ can be constructed and substituted into (1e), e.g., see Kvasnica et al. (2015).

Finally, we formulate the controller design problem.

Problem 2.3. (Problem statement) The problem is to approximate the solution of the MPC problem in (1) in a finite number of optimization-free iterations, such that the evaluated control input guarantees satisfaction of physical constraints in (1c)–(1d), recursive feasibility of optimization problem in (1), and the asymptotic stability for admissible measurements of system states, i.e., $\forall x \in \mathcal{X}$.

Remark 2.4. (Application range) Although the MPC formulation in (1) represents a simple regulatory problem, the control method proposed in this paper is not limited just to this MPC formulation. Any stabilizing MPC formulation satisfying Assumption 2.1 can be considered for its recursive-random shooting-based approximation as described in Section 3, among others: the reference tracking problem, soft-constraints, slew-rate control (Maciejowski, 2000), robust MPC (Kvasnica et al., 2015), etc.

2.2. Random shooting

In this section, we describe the approximation of the MPC control problem in (1) considering the stochastic approach of the so-called random shooting-based method. In this paper, the random shooting-based approach is considered in its formulation presented in Bakarác and Kvasnica (2018). This method evaluates a randomized sequence of the feasible control inputs with the best performance index, i.e., the minimum cost function value in (1a).

The enumeration of the feasible control input is performed considering a large number of randomly generated sequences. Obviously, the selected control sequence $\{\tilde{u}_0, \tilde{u}_1, \dots, \tilde{u}_{N-1}\}$ is suboptimal compared to the exact solution of MPC problem in (1). Compared to original approach in Bakarác and Kvasnica (2018), the asymptotic stability of the closed-loop system is guaranteed.

First, the Algorithm 1 explains the basic principles of the random shooting method in its original formulation. In Algorithm 1, $\hat{u}, \hat{x}, \hat{J}_r$ are variables for particular random shoot, and \tilde{u}, \tilde{J} are variables corresponding to the close-to-optimal solution of random shooting based approximation of MPC problem in (1). Scalar i is a counter for random shoots, r is a counter for feasible random shoots, binary δ is a feasibility indicator (flag), and N_F is maximal number of feasible random shoots. The remaining variables in Algorithm 1 came from MPC problem in (1).

The problem of implementing Algorithm 1 is, that this approach does not guarantee to find N_F feasible random shoots during a given sampling time t_s , i.e., the reaching suboptimality level is not guaranteed. Even worse, in general, Algorithm 1 does not guarantee finding any feasible solution. Therefore, it may happen, in the worst-case, that there will be no available feasible control input u_0 . Moreover, when comes to recursive feasibility, Algorithm 1 does not guarantee that there will be found any feasible solution in the next control step.

This paper aims to address the above-mentioned issues preserving the benefits of random shooting-based approximation of the MPC problem solution.

3. Recursive random shooting-based evaluation of MPC

This section is devoted to the main contribution of this paper, i.e., designing the recursive random shooting-based evaluation of MPC. The

INPUT: system state measurement x_0 , number N_F of feasible random shoots, maximal number of random shoots N_{\max} , MPC problem $(A, B, N, \mathcal{U}, \mathcal{X}, \mathcal{T}, \ell(x, u), \ell_N(x_N))$ in (1)

OUTPUT: sequence of control inputs $\{\tilde{u}_0, \tilde{u}_1, \dots, \tilde{u}_{N-1}\}$

```

1:  $\tilde{J} \leftarrow \infty; \tilde{U} \leftarrow \emptyset; r \leftarrow 0, i \leftarrow 0$  // Initialize random shooting
2: while  $r < N_F$  and  $i < N_{\max}$  do
3:    $\hat{x}_0 \leftarrow x_0, \hat{J}_r \leftarrow 0, \delta \leftarrow 1, i \leftarrow i + 1$  // Reset variables
4:   for  $j = 0, 1, \dots, N - 1$  do
5:     if  $\hat{x}_j \in \mathcal{X}$  then
6:        $\hat{u}_j \leftarrow \text{random}\{\hat{u} : \hat{u} \in \mathcal{U}\}$  // Random shoot
7:        $\hat{J}_r \leftarrow \hat{J}_r + \ell(\hat{x}_j, \hat{u}_j)$  // Penalty update
8:        $\hat{x}_{j+1} \leftarrow A \hat{x}_j + B \hat{u}_j$  // System state update
9:     else
10:       $\delta \leftarrow 0$  // Primal infeasible
11:      break
12:    end if
13:  end for
14:  if  $\delta == 1$  and  $x_N \in \mathcal{T}$  then
15:     $\hat{J}_r \leftarrow \hat{J}_r + \ell_N(x_N)$  // Terminal penalty update
16:     $r \leftarrow r + 1$  // Update primal feasibility counter
17:    if  $\hat{J}_r < \tilde{J}$  then
18:       $\tilde{U} \leftarrow \{\hat{u}_0, \hat{u}_1, \dots, \hat{u}_{N-1}\}$  // Update approximated solution
19:       $\tilde{J} \leftarrow \hat{J}_r$  // Update reference penalty
20:    end if
21:  end if
22: end while
23: return  $\{\tilde{u}_0, \tilde{u}_1, \dots, \tilde{u}_{N-1}\} \leftarrow \tilde{U}$ 

```

Algorithm 1. Random shooting-based approximation of MPC problem Bakarčić and Kvasnica (2018).

term “recursive” points out the learning-like self-improvement of the closed-loop control performance.

3.1. Principles of recursive random shooting-based control

Compared to standard random shooting implementation in Algorithm 1, the essential difference of the recursive random shooting-based method is that we do not erase the best solution of the previous control step. In the conventional random shooting approach, in each control step, the best-found solution is “forgotten”. In contrast to Algorithm 1, we reuse the best solution of the previous control step to be reconsidered in the current control step. If no better solution is found then we reapply the so far best solution in receding horizon control fashion, see Maciejowski (2000).

The so-called *dual-mode* control approach is introduced to ensure asymptotic stability, see Darup and Mönningmann (2018). The dual-mode control as described in Algorithm 2, consists of two modes: (i) Mode I: LQ optimal control that is implemented when the system states $x \in \mathcal{T}$ and (ii) Mode II: otherwise, apply the control input evaluated by a random shooting.

Compared to the conventional implementation of MPC, introducing the dual-mode control approach enables drastic reduction of the average real-time effort of the evaluation of the control input, as the enumeration of the control input melts down just into a simple matrix multiplication in (3), once the system states enter the invariant terminal set \mathcal{T} . It improves the resources-aware policy of controller operation, e.g., significantly prolonged battery life plugging the embedded hardware, etc.

The asymptotic stability of Mode I is ensured by design, see Darup and Mönningmann (2018). First, we construct a terminal set \mathcal{T} according to Assumption 2.1. Having a proper terminal set, we construct a stabilizing controller F_{LQ} above this terminal set \mathcal{T} .

Definition 3.1. (Terminal set controller) The terminal set controller $F_{LQ} \in \mathbb{R}^{n_x \times n_x}$ of state-feedback control law:

$$u(k) = F_{LQ} x(k) \tag{3}$$

is given by the solution of discrete matrix algebraic Riccati equation for LTI system in (1b), e.g., see Borrelli (2017).

Lemma 3.1. (Stabilizing terminal controller Borrelli (2017)) Let \mathcal{T} satisfy Assumption 2.1. The terminal set controller F_{LQ} in (3) guarantees the asymptotic stability of LTI system in (1b) for $\forall x \in \mathcal{T}$.

The task is to guarantee the asymptotic stability of Mode II. Therefore, we introduce the support control law.

Definition 3.2. (Support controller) The support controller $\bar{\kappa} : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_u}$ of support control law

$$u(k) = \bar{\kappa}(x(k)), \tag{4}$$

where $\bar{\kappa}(x(k))$ is constructed to guarantee the asymptotic stability of LTI system in (1b) for $\forall x \in \mathcal{X}$.

Assumption 3.2. (Stabilizing controller) Consider that there exists a support controller $\bar{\kappa}(x)$ such that the LTI system in (1) is asymptotically stable according to Mayne et al. (2000) for $\forall x$ subject to the physical

constraints in (1c)–(1d).

Note, the proper design method of the support controller according to Definition 3.2 is beyond the scope of this paper. Nevertheless, there are various methods available in the literature.

In this paper, we propose the approach of (robust) MPC design proposed in Kothare et al. (1996). Then, the following optimization problem of semidefinite programming (SDP) is obtained:

$$\min_{\gamma, X, Y, U} \gamma \tag{5a}$$

$$\text{s.t. : } \begin{bmatrix} 1 & \star \\ x_0 & X \end{bmatrix} \geq 0, \tag{5b}$$

$$\begin{bmatrix} X & \star & \star & \star \\ AX + BY & X & \star & \star \\ Q^{1/2}X & 0 & \gamma I & \star \\ R^{1/2}Y & 0 & 0 & \gamma I \end{bmatrix} \geq 0, \tag{5c}$$

$$\begin{bmatrix} X & \star \\ Y & U \end{bmatrix} \geq 0, \quad U_{i,i} \leq u_{\text{sat},i}, \forall i = 1, 2, \dots, n_u, \tag{5d}$$

$$\begin{bmatrix} X & \star \\ (AX + BY) & x_{\text{sat}}^2 I \end{bmatrix} \geq 0, \tag{5e}$$

where the decision variables are: $X = X^T \in \mathbb{R}^{n_x \times n_x}$ the positive definite weighted inverse-matrix of the Lyapunov matrix, $Y \in \mathbb{R}^{n_u \times n_x}$, $U \in \mathbb{R}^{n_u \times n_u}$, $\gamma \in \mathbb{R}$ the weight parameter of X , and symbol \star denotes a symmetric structure of linear matrix inequalities (LMIs). Note, just the symmetric constraints are considered in (5d), (5e). Therefore, the conservative inner box approximation of the sets of physical constraints in (1c), (1d) is considered for in (5d), (5e):

$$u_{\text{sat}} = \min\{|u_{\min}|, |u_{\max}|\}, \quad x_{\text{sat}} = \min\{|x_{\min}|, |x_{\max}|\} \tag{6}$$

Following the approach in Kothare et al. (1996), the support controller has the form of state feedback controller given by

$$\bar{F} = Y X^{-1}, \tag{7}$$

and the corresponding closed-loop control law in (4) is

$$u(k) = \bar{F} x(k), \tag{8}$$

see Kothare et al. (1996) for the stability proof and further technical details.

As the mathematical model is imperfect, the predicted behavior of the plant differs from the real system behavior. If the controlled plant is affected by the serious impact of uncertain parameters, then the plant-model is non-negligible. In such cases, it is recommended to design the support controller using the means of the robust controller design methods, e.g., following the procedure in Kothare et al. (1996).

In general, the support controller \bar{F} in (8) is designed subject to the conservative subset of physical constraints and leads to the suboptimal control performance. Nevertheless, the support controller serves just to

INPUT: system state measurement x_0 , terminal set \mathcal{T} in (1), LQ optimal controller F_{LQ} , and inputs of Algorithm 3

OUTPUT: control input u_0

- 1: **if** $x_0 \in \mathcal{T}$ **then**
- 2: $u_0 = F_{LQ} x_0$ // LQ optimal control input
- 3: **else**
- 4: $u_0 \leftarrow$ recursive random shooting (Algorithm 3) // Approximate MPC solution
- 5: **end if**

Algorithm 2. Dual mode control.

provide a backup control input in case no better solution is found.

Finally, we introduce the recursive random shooting-based approach in [Algorithm 3](#).

In [Algorithm 3](#), bar-symbol ($\bar{\cdot}$) denotes variables recursively passed into random shooting, hat-symbol ($\hat{\cdot}$) denotes variables of the current random shoot, and tilde-symbol ($\tilde{\cdot}$) denotes variables corresponding to the approximated solution of MPC problem in (1).

Compared to [Algorithm 1](#), [Algorithm 3](#) enters two auxiliary inputs: \bar{U}, \bar{J} . These variables are initialized before running the closed-loop control and introduce the recursive improvement of random shooting-based evaluation.

The sequence of support control inputs $\bar{U} = \{\bar{u}_0, \bar{u}_1, \dots, \bar{u}_{N-1}\}$ is initialized using a state-feedback control law given by:

$$\bar{u}_k = \kappa(\bar{x}_k), \quad \forall k = 0, 1, \dots, N-1, \quad (9)$$

where κ is a support controller according to [Definition 3.2](#).

Support penalty $\bar{J}(\bar{x}, \bar{u})$ is evaluated by

$$\bar{J}(\bar{x}, \bar{u}) = \sum_{k=0}^{N-1} \ell(\bar{x}_k, \bar{u}_k) + \ell_N(\bar{x}_N), \quad (10)$$

where \bar{u}_k, \bar{x}_k for $\forall k = 0, 1, \dots, N-1$, are vectors of support control inputs and the corresponding support system state trajectory, respectively.

We point out that compared to [Algorithm 1](#), [Algorithm 3](#) aims to evaluate $(N_F - 1)$ random shoots, i.e., [Algorithm 3](#) skips evaluation of one random shoot. The difference is given by the availability of a support (backup) sequence of control inputs \bar{U} that ensures evaluation of \bar{U} satisfying the physical constraints, recursive feasibility, and the asymptotic stability.

Lemma 3.3. (Satisfaction of physical constraints) Given MPC design problem in (1) and the support controller defined according to [Definition 3.2](#). Assume that [Assumption 3.2](#) holds. Then the control input $u(k)$ evaluated by [Algorithm 3](#) satisfies the physical constraints in (1c)–(1d) for all control steps $k \geq 0$.

Proof. First, we prove the satisfaction of physical constraints on control inputs in (1c). The control input applied to the controlled plant u_0 is evaluated either by recursive random shooting approach in [Algorithm 3](#), or by a support controller. Any feasible random shoot in [Algorithm 3](#) satisfies the constraints in (1c). If no feasible random shoot is found, then the support controller enumerates u_0 using (8). According to [Definition 3.2](#), the support controller satisfies the constraints in (1c) by design. Next, we need to prove the satisfaction of physical constraints on system states in (1d). As the control input u_0 is either feasible random shoot evaluated subject to (1d), or enumerated using the support controller in (8) constructed subject to (1d), then (1c) holds. \square

Lemma 3.4. (Recursive feasibility) Given MPC design problem in (1) and the support controller defined according to [Definition 3.2](#). Assume [Assumption 3.2](#) holds. Then the control input $u(k)$ evaluated by [Algorithm 3](#) ensures recursive feasibility of MPC problem in (1) for all control steps $k \geq 0$.

Proof. Proof of [Lemma 3.4](#) directly follows from the [Lemma 3.3](#). If for any $x_0 \in \mathcal{X}$ the physical constraints in (1c)–(1d) hold for $\forall k \geq 0$, then the recursive feasibility condition holds. \square

Theorem 3.5. (Asymptotic stability) Given MPC design problem in (1) and the support controller defined according to [Definition 3.2](#). Assume [Assumption 3.2](#) holds. Then the control input $u(k)$ evaluated by [Algorithm 3](#) ensures asymptotic stability of problem in (1) for all control steps $k \geq 0$.

Proof. Proof of [Lemma 3.5](#) directly follows from the [Lemmas 3.3, 3.4](#), and $x_N \in \mathcal{F}$ in (1e), where \mathcal{F} is control invariant set. For any $x_0 \in \mathcal{X}, u_0$ is evaluated either using random shooting in [Algorithm 3](#), or using the support controller $\bar{\kappa}$ in (8). In the both cases, u_0 is such that $x_N \in \mathcal{F}$ hold. According to dual-mode control in [Algorithm 2](#), once $x_0 \in \mathcal{F}$, then u_0 is evaluated using LQ optimal control input in (3). The asymptotic stability

of LQ optimal control is given by design, see [Borrelli \(2017\)](#). \square

Lemma 3.6. (Sample size bound for worst-case performance [Bakarác and Kvasnica \(2018\)](#), [Tempo et al. \(2013\)](#)) If the number of feasible random shoots N_F satisfies

$$N_F \geq \frac{\log \frac{1}{1-\alpha}}{\log \frac{1}{1-\varepsilon}} \quad (11)$$

for any $\varepsilon \in (0, 1)$ and $\alpha \in (0, 1)$, then the control sequence $\{\tilde{u}_0, \tilde{u}_1, \dots, \tilde{u}_{N-1}\}$ generated by [Algorithm 3](#) is by $\varepsilon \cdot 100\%$ suboptimal approximation of the solution to MPC problem in (1) with a confidence of $\alpha \cdot 100\%$.

Proof. The proof of [Lemma 3.6](#) directly follows from the proof of [Theorem 8.1](#) in [Tempo et al. \(2013\)](#), p. 118. \square

[Fig. 1](#) shows the exponential dependence of the number of feasible random shoots N_F on the suboptimality level ε and confidence value α given by (11). As can be seen in [Fig. 1](#), the minimum number of feasible random shoots N_F necessary to guarantee the required level of performance increases exponentially with increasing value of confidence α and decreasing value of the suboptimality level ε .

Remark 3.7. (Scaling of time evaluation) Given MPC design problem in (1) and the support controller according to [Definition 3.2](#). Assume [Assumption 3.2](#) holds. Then the evaluation of the control input $u(k)$ evaluated by [Algorithm 3](#) is scalable within an interval $[t_{\min}, t_{\max}]$ for all control steps $k \geq 0$, where t_{\min}, t_{\max} are the minimum and maximum portion of time available within one control step determined by the sampling time t_s .

Corollary 3.7.1. (Current suboptimality) For any $\alpha \in (0, 1)$ and given number of feasible random shoots generated by [Algorithm 3](#), the suboptimality level $\varepsilon \cdot 100\%$ of the MPC problem in (1) is given by

$$\varepsilon = 1 - (1 - \alpha)^{1/r}. \quad (12)$$

Proof. The proof of [Corollary 3.7.1](#) directly follows from the [Lemma 3.6](#) by substituting r into N_F and evaluating α from (11), see (4.3) in [Vidyasagar \(2001a\)](#), p.5. \square

Remark 3.8. (Suboptimality level) Compared to an optimal control input evaluated by MPC problem in (1) (denoted by u^*), the evaluation of approximated control input \tilde{u} using [Algorithm 3](#) leads to suboptimal solution. The required level of suboptimality could be enforced according to [Lemma 3.6](#) by evaluation of specific number of random shoots N_F , see [Piovesan and Tanner \(2009\)](#), [Dyer et al. \(2014\)](#). We point out, the sampling time t_s needs to be sufficiently long to proceed such number of random shoots. Otherwise, the realized number of random shoots r determines an upper bound on the reached suboptimality level according to [Corollary 3.7.1](#).

Remark 3.9. (Approximation of nonlinear MPC) We point out that the proposed method of recursive random shooting-based approximation of MPC is not limited to linear systems, as it can be directly extended into the framework of nonlinear MPC.

Remark 3.10. (Approximated terminal set) In case the polytopic terminal set \mathcal{F} in (1e) requires significant portion of time to evaluate the point location problem, i.e., the terminal set \mathcal{F} has relatively high number of facets (polytopic halfspaces), then its maximal volume inner approximation $\tilde{\mathcal{F}}$ can be implemented to reduce the real time effort of evaluation $x \in \mathcal{X}$, and, simultaneously, preserving the invariant properties of original set \mathcal{F} . Various approximation methods can be considered, e.g., Chebyshev ball, ellipsoidal approximation, box approximation, etc. [Boyd and Vandenberghe \(2004\)](#).

Remark 3.11. (Approximation of Robust MPC) Following up [Remark 2.2](#), in the case the controlled plant is affected by a serious impact of

INPUT: system state measurement x_0 , number N_F of feasible random shoots, maximal number of random shoots N_{\max} , MPC problem $(A, B, N, \mathcal{U}, \mathcal{X}, \ell(x, u), \ell_N(x_N))$ in (1), LQ optimal controller F_{LQ} , support controller \bar{F} , initialization of support control inputs \bar{U}

OUTPUT: sequence of control inputs $\{\bar{u}_0, \bar{u}_1, \dots, \bar{u}_{N-1}\}$, support control inputs \bar{U} , support penalty \bar{J}

```

1:  $r \leftarrow 0, i \leftarrow 0$  // Initialize random shooting
2:  $\bar{U} \leftarrow \bar{U}, \bar{u}_N \leftarrow \emptyset$  // Initialize recursive random shooting
3: for  $i = 0, 1, \dots, N - 1$  do
4:    $\bar{J} \leftarrow \bar{J} + \ell(\bar{x}_i, \bar{u}_i)$  // Penalty update
5:    $\bar{x}_{i+1} \leftarrow A \bar{x}_i + B \bar{u}_i$  // System state update
6: end for
7:  $\bar{J} \leftarrow \bar{J} + \ell_N(\bar{x}_N)$  // Terminal penalty update
8:  $\bar{J} \leftarrow \bar{J}$ 
9: while  $r < (N_F - 1)$  and  $i < N_{\max}$  do
10:   $\hat{x}_0 \leftarrow x_0, \hat{J}_r \leftarrow 0, \delta \leftarrow 1, i \leftarrow i + 1$  // Reset variables
11:  for  $j = 0, 1, \dots, N - 1$  do
12:    if  $\hat{x}_j \in \mathcal{T}$  then
13:       $\hat{u}_j \leftarrow F_{LQ} \hat{x}_j$  // LQ optimal control input
14:    else
15:       $\hat{u}_j \leftarrow \text{random}\{\hat{u} : \hat{u} \in \mathcal{U}\}$  // Random shoot
16:    end if
17:    if  $\hat{x}_j \in \mathcal{X}$  then
18:       $\hat{J}_r \leftarrow \hat{J}_r + \ell(\hat{x}_j, \hat{u}_j)$  // Penalty update
19:       $\hat{x}_{j+1} \leftarrow A \hat{x}_j + B \hat{u}_j$  // System state update
20:    else
21:       $\delta \leftarrow 0$  // Primal infeasible
22:      break
23:    end if
24:  end for
25:  if  $\delta == 1$  and  $\hat{x}_N \in \mathcal{T}$  then
26:     $\hat{J}_r \leftarrow \hat{J}_r + \ell_N(\hat{x}_N)$  // Terminal penalty update
27:     $r \leftarrow r + 1$  // Update primal feasibility counter
28:    if  $\hat{J}_r < \bar{J}$  then
29:       $\bar{U} \leftarrow \{\hat{u}_0, \hat{u}_1, \dots, \hat{u}_{N-1}\}$  // Update approximated solution
30:       $\bar{J} \leftarrow \hat{J}_r$  // Update reference penalty
31:       $\bar{u}_N \leftarrow F_{LQ} \bar{x}_N$  // Update support control input
32:       $\bar{x}_{N+1} \leftarrow A \bar{x}_N + B \bar{u}_N, \bar{\ell}_N(\bar{x}_{N+1})$  // Update terminal penalty
33:    end if
34:  end if
35: end while
36: if  $\bar{u}_N == \emptyset$  then
37:   $\bar{u}_N \leftarrow F_{LQ} \bar{x}_N$  // Update recursive support control input
38: end if
39: return  $\{\bar{u}_0, \bar{u}_1, \dots, \bar{u}_{N-1}\} \leftarrow \bar{U}, \bar{U} \leftarrow \{\bar{u}_1, \bar{u}_2, \dots, \bar{u}_N\}, \bar{J}$ 

```

Algorithm 3. Recursive-random shooting-based approximation of MPC problem.

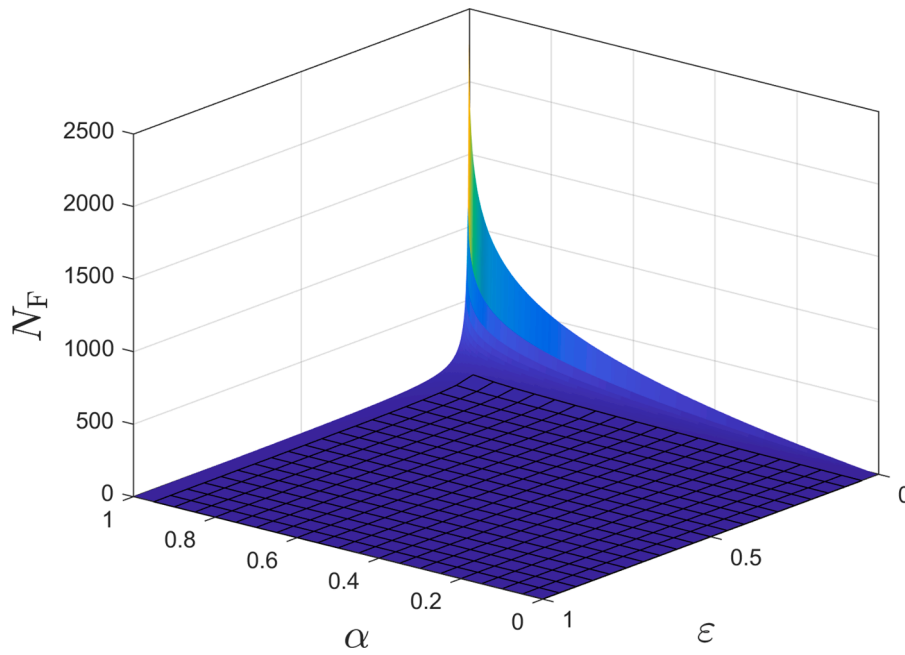


Fig. 1. Number of feasible random shoots N_F as a function of the suboptimality level ε and confidence value α in (11).

uncertain parameters threatening the system stability, and/or the implementation of support sequence of control inputs \bar{U} in open-loop fashion when no better solution is found by random shooting-based strategy, then MPC formulation in (1) should be reformulated to some relevant Robust MPC strategy, e.g., Kvasnica et al. (2015), Mayne et al. (2005), etc.

3.2. Properties of the recursive random shooting-based control

The properties of the main state-of-the-art methods are briefly summarized in Table 1, where we judge the properties of the conventional implicit (non-explicit) MPC (Maciejowski, 2000), conventional explicit MPC (EMPC) (Bemporad et al., 2002), random shooting-based approximation of MPC (RS) (Bakarác and Kvasnica, 2018), and the proposed recursive random shooting-based approximation of MPC (recursive RS). To make the results comparable, the term “conventional” denotes the well-known formulation, i.e., there is no implemented special method focused on improving the selected property. Obviously, introducing such a specific method improving selected property (e.g., complexity reduction techniques) could decline the other judged property (suboptimality, etc.).

Table 1

Properties of considered methods: MPC, explicit MPC (EMPC), standard random shooting (RS), recursive random shooting (recursive RS).

method	MPC (Maciejowski, 2000)	EMPC (Bemporad et al., 2002)	RS (Bakarác and Kvasnica, 2018)	Recursive RS
solution of (1)	optimal	optimal	suboptimal	suboptimal
physical constraints	guaranteed	guaranteed	non-guaranteed	guaranteed
recursive feasibility	guaranteed	guaranteed	non-guaranteed	guaranteed
stability	guaranteed	guaranteed	non-guaranteed	guaranteed
realtime tuning	tunable	non-tunable	tunable	tunable
enumeration memory footprint	intensive low	negligible high	negligible low	negligible low

We point out that MPC and EMPC evaluate the optimal solution of the optimization problem in (1). On the other hand, the evaluation of both RS and recursive RS methods does not guarantee reaching the optimal solution. We recall, that evaluating a given number of random shoots N_F leads to specific confidence of the suboptimality level, see Lemma 3.6.

MPC and EMPC satisfy the physical constraints on control inputs and system states in (1c), (1d) and the associated recursive feasibility by design, as they are directly integrated in the optimization problem in (1). If the original RS approach finds any feasible solution, then this method guarantees satisfaction of the physical constraints. But, it does not imply the recursive feasibility, i.e., that there will find any feasible solution in the next control step. Therefore, the main drawback of the original RS method is that, in general, this approach does not guarantee finding any feasible solution of the problem in (1) and the ability to find it also in the next control step. On the other hand, recursive RS addresses the both limitations of the original RS method: (i) the satisfaction on the physical constraints in (1c), (1d) is guaranteed by the support controller in (9), and (ii) the recursive feasibility is enforced by the support sequence of control inputs \bar{U} evaluated by (9).

If the formulation of MPC problem in (1) ensures stability conditions according to Mayne et al. (2000), then MPC and EMPC guarantee asymptotic stability by design. Due to the absence of the guarantee of finding any feasible solution using the RS method, this method does not guarantee asymptotic stability. On the other hand, recursive RS provides a stability guarantee via the existence of the support controller in (9).

The MPC outperforms EMPC by the ability of the real-time tuning of the penalty factors Q, R in (2). In contrast to MPC, EMPC enables the real-time tuning of the penalty matrices Q, R in (1). Similar to MPC approach, RS and recursive RS methods also enable real-time tuning of penalty matrices Q, R .

The real-time solving/enumeration of optimization problem in (1) limits the implementation of MPC on the embedded hardware for the large-scale systems.

Although the complexity of the optimization problem decreases the ability to find any feasible solution by the random shooting-based approaches, the support controller enforces the availability of the feasible solution. The random shooting-based approach enables scaling of the suboptimality level subject to the available portion of the sampling time, see Remark 3.8. Moreover, implementation of the dual mode control

approach in Algorithm 2 significantly reduces the real-time effort necessary to find an optimal control input by solving a simple matrix equation in (3). Analogous to random shooting-based approaches, EMPC enables the evaluation of (optimal) control input by a negligible enumeration effort. These control methods are easy-to-implement without the need for special libraries. If the controlled problem is of modest size, the implementation of EMPC seems to be an optimal option, until it comes to the limits of the memory footprint. In such a case, the random shooting-based approach outperforms the EMPC method.

4. Experimental case study

In this section, we demonstrate the efficiency of the proposed control approach. The properties were investigated using the inverted pendulum benchmark. The successful closed-loop control of this plant also demonstrates that this approach is suitable for systems with fast dynamics with multiple states. Moreover, the proposed control strategy was implemented using embedded platforms.

4.1. Mathematical model of the inverted pendulum

To investigate the properties of the proposed recursive RS control approach, the laboratory inverted pendulum device was considered, see Bakarac et al. (2017). The inverted pendulum device has fast dynamics and the corresponding sampling time is $t_s = 20 \times 10^{-3}$ s. Therefore, the inverted pendulum device is an appropriate candidate for validation of the properties of the proposed recursive random shooting-based approximated MPC. The inverted pendulum device has four system states and one control input. Technically, the system states are the angle of the pendulum x_1 , angular velocity x_2 , cart position x_3 , and speed of the cart x_4 . The control input u is the acceleration of the cart. The discrete-time state-space matrices of the linearized LTI system in (1b) acquire the following form:

$$A = \begin{bmatrix} 1.01 & 0.02 & 0 & 0 \\ 0.70 & 0.99 & 0 & 0 \\ 0 & 0 & 1.00 & 0.02 \\ 0 & 0 & 0 & 1.0 \end{bmatrix}, \quad B = \begin{bmatrix} 7 \\ 709 \\ 2 \\ 200 \end{bmatrix} \times 10^5. \quad (13)$$

The inverted pendulum device has two natural equilibrium points, both in the vertical position of the pendulum's rod. The upward position of the pendulum represents the unstable equilibrium point, that represents a challenging stabilizing control problem.

4.2. Control setup

In this experimental case study, the offset-free regulatory problem of the inverted pendulum device was investigated considering two control strategies: (i) recursive RS-based evaluation of MPC described in Section 3 and (ii) conventional MPC described in Section 2.1. In order to make the results comparable, both control methods shared the same values of the prediction horizon $N = 3$, number of evaluated control steps $N_{\text{sim}} = 300$, and the same initial conditions of system states $x_0 = [0.2, 0, -0.1, 0]^T$. The penalty matrices P, Q, R of cost function in (1a) were systematically tuned to have the following values:

$$P = \begin{bmatrix} 1.22 & 0.13 & -0.13 & -0.18 \\ -0.13 & 0.02 & -0.02 & -0.4 \\ -0.13 & -0.02 & -0.09 & -0.07 \\ -0.18 & -0.4 & -0.07 & -0.1 \end{bmatrix} \times 10^5, \quad (14a)$$

$$Q = \begin{bmatrix} 10^4 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 10^2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (14b)$$

$$R = 10. \quad (14c)$$

The implemented control methods were designed to respect the physical constraints on system states and control inputs given by:

$$\begin{bmatrix} -15 \\ -20 \\ -0.25 \\ -2 \end{bmatrix} \leq x(k) \leq \begin{bmatrix} 15 \\ 20 \\ 0.25 \\ 2 \end{bmatrix}, \quad (15a)$$

$$-15 \leq u(k) \leq 15. \quad (15b)$$

For the controller design purposes, the control input, state, and control output of the system were defined in the form of deviation variables, which explains the negative values in (15) and in the vector of initial conditions x_0 . In this case study, the regulatory problem was solved by random shooting-based method (Section 2.2) and recursive random shooting-based method (Section 3). Both RS-based controllers were implemented using a 32-bit microcontroller. Specifically, the ESP32 DevKit V4 microcontroller was considered. It operates with 4 MB of Flash memory. A Flash memory of this size enables solving quadratic programs for smaller systems. The proposed recursive RS control approach (Algorithms 2, 3), was exported to the programming language C, which is fully compatible with the considered microcontroller device.

As described in Algorithm 3, the proposed approach also considered the required number of feasible shoots $N_F = 29$, which was determined using (11) to satisfy $\epsilon = 0.1$ suboptimality level with a confidence of $\alpha = 0.95$. The maximal number of random shoots was limited by $N_{\text{max}} = 182$. The gain of the support controller in (8) was designed as follows:

$$\bar{F} = [-52.50, -2.49, 1.163, 1.73]. \quad (16)$$

The LQ optimal controller gain F_{LQ} in (3), was evaluated as follows:

$$F_{LQ} = [-45.89, -5.84, 2.64, 4.42]. \quad (17)$$

The conventional MPC described in Section 2.1 was implemented using MATLAB/Simulink R2021b on a PC with an i5 CPU (2.7 GHz) and 8 GB RAM. The optimization problems were solved using MATLAB programming environment (MATLAB Optimization Toolbox, 2019). The optimal control problems were formulated by Multi-Parametric Toolbox (Hergeg et al., 2013) and solved by Quadprog a built-in QP solver of MATLAB. We point out, that the implementation of the MPC approach served solely to evaluate the optimal control performance. This reference control performance serves to evaluate the suboptimality level of the proposed recursive RS-based approach. The implicit MPC controller was not implemented on the embedded platform, because the constructed QP corresponding to the MPC design problem in (1) had larger computational time than the sampling time of the controlled system. Simultaneously, the explicit MPC is not applicable on this embedded platform, as the explicit solution map has greater size than the available flash memory.

4.3. Results

In this section, we demonstrate the main benefits of the proposed recursive RS control method by using the benchmark model of the inverted pendulum. Four control strategies (Table 1) were investigated in this case study and two of them were analyzed from the closed-loop control performance point of view. First, the proposed recursive RS-

based evaluation of MPC described in Section 3 was implemented. To compare the recursive RS control method in terms of suboptimality level, conventional MPC described in Section 2.1 was implemented.

Fig. 2 depicts the closed-loop control trajectories evaluated by recursive RS-based method and of MPC approach. These data were generated an embedded platform - 32-bit microcontroller. As can be seen in Fig. 2, the offset-free regulatory problem of the inverted pendulum device was ensured in both cases.

According to Algorithm 3, the sequence of control inputs is generated considering by one of the following: (i) random shoot, (ii) support controller in (4), (iii) LQ optimal controller in (3). Figure 2 depicts a control trajectory in dashed blue, which was generated considering Algorithm 3. The initial four steps of the control input trajectory were enumerated using the support controller in (4) and the following five control steps were computed using random shooting. Although the generated control actions may differ from the optimal control actions computed by the conventional MPC method, the evaluated control actions still guarantee the closed-loop stability and satisfaction of physical constraints of system inputs and states. Then, in the tenth step of the closed-loop simulation, the system states entered the terminal set \mathcal{T} . Since then, the control inputs were evaluated by LQ optimal controller in (3), see Algorithm 3. As the consequence, such control actions do not differ from the optimal control actions evaluated by the conventional MPC method.

Table 2 summarizes the numerical results of the investigated methods. The parameter t_{avg} represents an average time required to compute the control input. This parameter is computed for all control steps N_{sim} evaluated using the microcontroller.

The parameter J_{total} represents the total sum of all penalty functions computed in each step of the closed-loop control:

Table 2

Comparison of the selected methods: MPC, explicit MPC (EMPC), standard random shooting (RS), recursive random shooting (Recursive RS).

method	MPC	EMPC	RS	Recursive RS
t_{avg} [ms]	†	†	18.21	1.20
$J_{total} [\times 10^4]$	6.33	6.33	28.97	6.88
E [mJ]	†	†	5.23	0.34

$$J_{total} = \sum_{i=1}^{N_{cl}} J_{cl,i}. \quad (18)$$

In case of conventional MPC (Section 2.1) and EMPC, the criterion $J_{cl,i}$ is defined as J in (1a). In case of the random shooting-based methods (Section 2.2, Section 3), the criterion $J_{cl,i}$ is defined as \tilde{J} in Algorithm 1 and Algorithm 3, respectively.

Moreover, the energy consumption of the controller during the computation of control input was evaluated. The parameter E , represents the electric power consumed within one sampling period to perform the operations necessary to compute the control input. It is computed according to the following formula:

$$E = I_{con} U_{sup} t_{avg}, \quad (19)$$

where $I_{con} = 57.38$ mA is the value of measured electric current consumed by the microcontroller during the computation of control input and $U_{sup} = 5$ V is the supply voltage of the microcontroller.

The symbol † in Table 2 denotes that the corresponding criterion was not applicable. In the case of EMPC, the parameter t_{avg} and E were not evaluated because the flash memory of the considered microcontroller is 4 MB, and the control law of EMPC in form of partitions has a memory

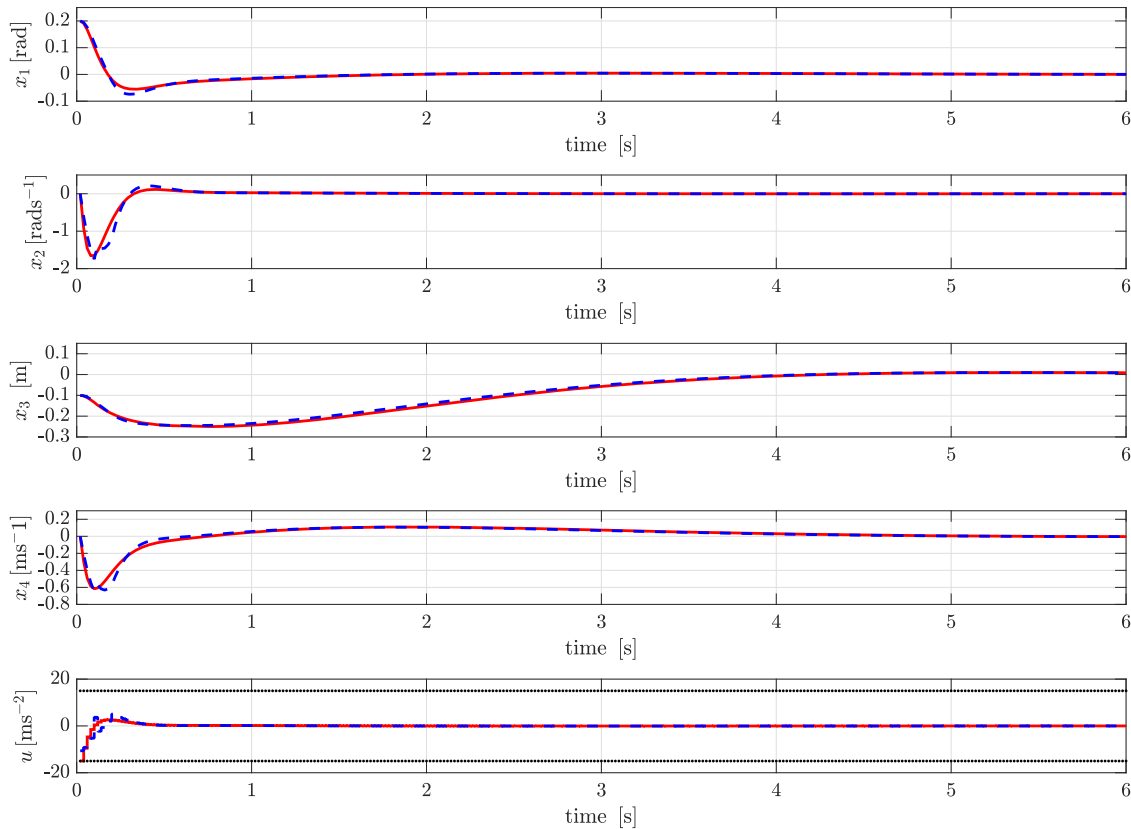


Fig. 2. The control performance of stabilizing the inverted pendulum from non-zero initial conditions into origin (the regulatory problem) using: the proposed recursive RS-based controller (dashed blue), the conventional MPC controller (solid red). The physical constraints on control input are depicted in black-dotted lines. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

footprint of 21 MB. This size of memory footprint prevents the implementation of EMPC for the inverted pendulum benchmark on the considered microcontroller. A different obstacle occurred, when implementing the conventional MPC. Although the corresponding quadratic program has a memory footprint of 665 kB, the corresponding average computational time $w_{\text{avg}} = 113 \times 10^{-3}$ s, which is far greater than the sampling time of the inverted pendulum $t_s = 20 \times 10^{-3}$ s. Thus, conventional MPC is not an option when it comes to control of the inverted pendulum benchmark on the considered microcontroller.

Comparing RS and recursive RS methods, the implementation of recursive RS decreased the t_{avg} . Simultaneously, the corresponding energy consumption E was reduced by 93%. The time and energy savings are ensured by the dual mode control of recursive RS, i.e., a computationally cheap linear control law is evaluated once the system states enter the terminal set \mathcal{T} , see Algorithm 2. This reduction could significantly prolong the battery life of the embedded platforms and reduce the associated carbon footprint.

The suboptimality level of the closed-loop performance, judged by the parameter J_{total} , was approximately 8%. This value corresponds to the suboptimality level ϵ required in (11) leading to $N_F = 29$ which was determined based on the requirement $\epsilon = 0.10$ with confidence level of $\alpha = 0.95$ in (11).

5 Conclusion

This paper proposes the recursive random-shooting-based approximation of the MPC problem suitable for implementation on the embedded hardware. The method is focused on systems, where the implementation of implicit MPC or explicit MPC is limited due to high computational complexity or strict memory requirements on embedded platforms, i.e., microcontrollers or programmable logic controllers. The main contribution of the proposed approach is the guarantee of asymptotic stability with respect to the physical constraints, despite its stochastic nature. To investigate the properties of the proposed approach, a case study using an inverted pendulum benchmark was conducted. This multivariable system is naturally unstable and has fast dynamics. The regulatory problem of the inverted pendulum benchmark was implemented on the 32-bit microcontroller. The results of the implementation were analyzed considering the average computational time and energy consumption of the microcontroller. The criteria were computed considering four controller design methods: the proposed method, the non-recursive random shooting method, the implicit MPC, and the explicit MPC. The proposed method reduced computational time and energy consumption by 93%, when compared to the non-recursive random shooting method. The implementation of an explicit MPC controller or implementation of an implicit MPC controller was restricted due to the strict memory requirements of the microcontroller and the fast dynamics of the inverted pendulum. Considering the results of the optimal control action evaluated by MPC, the suboptimality of the closed-loop performance was approximately 8%. Thus, the generated control trajectories could be considered close to optimal. Future work will include the implementation and validation of the proposed method on the laboratory plant.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

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. The authors P. Bakarác, M. Horváthová, L. Galčíková would like to thank for financial contribution from the STU Grant scheme for Support of Young Researchers.

References

- Bakarác, P., Kvasnica, M., 2018. Fast nonlinear model predictive control of a chemical reactor: a random shooting approach. *Acta Chimica Slovaca* 11 (2), 175–181. <https://doi.org/10.2478/acs-2018-0025>.
- Bakarác, P., Kalúz, M., Čirka, L., 2017. Design and development of a low-cost inverted pendulum for control education. 2017 21st International Conference on Process Control (PC), pp. 398–403. <https://doi.org/10.1109/PC.2017.7976247>.
- Bartlett, R.A., Biegler, L.T., Backstrom, J., Gopal, V., 2002. Quadratic programming algorithms for large-scale model predictive control. *J. Process Control* 12 (7), 775–795. [https://doi.org/10.1016/S0959-1524\(02\)00002-1](https://doi.org/10.1016/S0959-1524(02)00002-1).
- Bemporad, A., 2016. A quadratic programming algorithm based on nonnegative least squares with applications to embedded model predictive control. *IEEE Trans. Automat. Control* 61 (4), 1111–1116. <https://doi.org/10.1109/TAC.2015.2459211>.
- Bemporad, A., Morari, M., 1999. *Robust model predictive control: a survey. Robustness in Identification and Control*. Springer, London, pp. 207–226.
- Bemporad, A., Morari, M., Dua, V., Pistikopoulos, E.N., 2002. The explicit linear quadratic regulator for constrained systems. *Automatica* 38 (1), 3–20. [https://doi.org/10.1016/S0005-1098\(01\)00174-1](https://doi.org/10.1016/S0005-1098(01)00174-1).
- Borrelli, F., 2017. *Constrained Optimal Control of Linear and Hybrid Systems*. Springer Berlin Heidelberg. <https://doi.org/10.1007/3-540-36225-8>.
- Boyd, S., Vandenberghe, L., 2004. *Convex Optimization*, 4th ed. Cambridge University Press.
- Brand, M., Shilpiekandula, V., Yao, C., Bortoff, S.A., Nishiyama, T., Yoshikawa, S., Iwasaki, T., 2011. A parallel quadratic programming algorithm for model predictive control. *IFAC Proc. Vol. 44 (1)*, 1031–1039. <https://doi.org/10.3182/20110828-6-IT-1002.03222>. 18th IFAC World Congress.
- Chaber, P., Lawrynczuk, M., 2019. Fast analytical model predictive controllers and their implementation for STM32 ARM microcontroller. *IEEE Trans. Ind. Inf.* 15 (8), 4580–4590. <https://doi.org/10.1109/TII.2019.2893122>.
- Chen, S., Saulnier, K., Atanasov, N., Lee, D.D., Kumar, V., Pappas, G.J., Morari, M., 2018. Approximating explicit model predictive control using constrained neural networks. 2018 Annual American Control Conference (ACC), pp. 1520–1527. <https://doi.org/10.23919/ACC.2018.8431275>.
- Darup, M.S., Mönnigmann, M., 2018. Optimization-free robust MPC around the terminal region. *Automatica* 95, 229–235. <https://doi.org/10.1016/j.automatica.2018.05.025>.
- Dyer, M., Kannan, R., Stougie, L., 2014. A simple randomised algorithm for convex optimisation. *Math. Program.* 147, 207–229. <https://doi.org/10.1007/s10107-013-0718-0>.
- El-Nagar, A.M., El-Bardini, M., 2014. Practical implementation for the interval type-2 fuzzy PID controller using a low cost microcontroller. *Ain Shams Eng. J.* 5 (2), 475–487. <https://doi.org/10.1016/j.asej.2013.12.005>.
- Fedorová, K., Bakarác, P., Kvasnica, M., 2019. Agile manoeuvres using model predictive control. *Acta Chimica Slovaca* 12 (1), 136–141. <https://doi.org/10.2478/acs-2019-0019>.
- Herceg, M., Kvasnica, M., Jones, C.N., Morari, M., 2013. *Multi-parametric toolbox 3.0*. Proc. of the European Control Conference, pp. 502–510.
- Hrusic, N., Prljaca, N., 2020. An implementation and evaluation of fast embedded model predictive control. 19th International Symposium INFOTEH-JAHORINA, pp. 1–5. <https://doi.org/10.1109/INFOTEH48170.2020.9066290>.
- Johansen, T.A., Jackson, W., Schreiber, R., Tondel, P., 2007. Hardware synthesis of explicit model predictive controllers. *IEEE Trans. Control Syst. Technol.* 15 (1), 191–197. <https://doi.org/10.1109/TCST.2006.883206>.
- Kothare, M.V., Balakrishnan, V., Morari, M., 1996. *Robust constrained model predictive control using linear matrix inequalities*. *Automatica* 32, 1361–1379.
- Krishnamoorthy, D., Skogestad, S., 2022. Real-time optimization as a feedback control problem - a review. *Comput. Chem. Eng.* 107723. <https://doi.org/10.1016/j.compchemeng.2022.107723>.
- Kvasnica, M., Takács, B., Holaza, J., Ingole, D., 2015. Reachability analysis and control synthesis for uncertain linear systems in MPT. *Proceedings of the 8th IFAC Symposium on Robust Control Design*. Elsevier, Bratislava, Slovak Republic, pp. 302–307.
- Maciejowski, J., 2000. *Predictive Control with Constraints*. Prentice Hall, London.
- MATLAB Optimization Toolbox, 2019. the MathWorks. Natick, MA, USA.
- Mayne, D.Q., 2014. Model predictive control: recent developments and future promise. *Automatica* 50 (12), 2967–2986. <https://doi.org/10.1016/j.automatica.2014.10.128>.
- Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.M., 2000. Constrained model predictive control: stability and optimality. *Automatica* 36, 786–814. [https://doi.org/10.1016/S0005-1098\(99\)00214-9](https://doi.org/10.1016/S0005-1098(99)00214-9).
- Mayne, D.Q., Seron, M.M., Raković, S.V., 2005. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica* 41 (2), 219–224. <https://doi.org/10.1016/j.automatica.2004.08.019>.
- Ndje, M., Bitjoka, L., Boum, A.T., Fotsa Mbogne, D.J., Busoniu, L., Kamgang, J.C., Tchane Djogdom, G.V., 2021. Fast constrained nonlinear model predictive control for implementation on microcontrollers. *IFAC-PapersOnLine* 54 (4), 19–24. <https://doi.org/10.1016/j.ifacol.2021.10.004>. 4th IFAC Conference on Embedded Systems, Computational Intelligence and Telematics in Control CESCIT 2021

- Oravec, J., Kvasnica, M., Bakošová, M., 2017. Quasi-non-symmetric input and output constraints in LMI-based robust MPC. Preprints of the 20th IFAC World Congress, Toulouse, France, Vol. 20, pp. 11829–11834. <https://doi.org/10.1016/j.ifacol.2017.08.1671>.
- Piovesan, J.L., Tanner, H.G., 2009. Randomized model predictive control for robot navigation. 2009 IEEE International Conference on Robotics and Automation, pp. 94–99. <https://doi.org/10.1109/ROBOT.2009.5152468>.
- Qin, S.J., Badgwell, T.A., 2003. A survey of industrial model predictive control technology. *Control Eng. Pract.* 11 (7), 733–764. [https://doi.org/10.1016/S0967-0661\(02\)00186-7](https://doi.org/10.1016/S0967-0661(02)00186-7).
- Shoukry, Y., El-Kharashi, M.W., Hamma, S., 2010. MPC-On-Chip: an embedded GPC coprocessor for automotive active suspension systems. *IEEE Embed. Syst. Lett.* 2 (2), 31–34. <https://doi.org/10.1109/LES.2010.2051794>.
- Tempo, R., Calafiore, G., Dabbene, F., 2013. *Randomized Algorithms for Analysis and Control of Uncertain Systems: With Applications*. Springer. <https://doi.org/10.1007/978-1-4471-4610-0>.
- Vidyasagar, M., 2001. Randomized algorithms for robust controller synthesis using statistical learning theory. *Automatica* 37 (10), 1515–1528. [https://doi.org/10.1016/S0005-1098\(01\)00122-4](https://doi.org/10.1016/S0005-1098(01)00122-4).
- Vidyasagar, M., 2001. Randomized algorithms for robust controller synthesis using statistical learning theory: a tutorial overview. *Eur. J. Control* 7 (2), 287–310. <https://doi.org/10.3166/ejc.7.287-310>.
- Youness, H., Moness, M., Khaled, M., 2014. MPSoCs and multicore microcontrollers for embedded PID control: a detailed study. *IEEE Trans. Ind. Inf.* 10 (4), 2122–2134. <https://doi.org/10.1109/TII.2014.2355036>.



Robust model predictive control and PID control of shell-and-tube heat exchangers

Juraj Oravec^{a,*}, Monika Bakošová^a, Marian Trafczynski^b, Anna Vasičkaninová^a, Alajos Mészáros^a, Mariusz Markowski^b

^a Slovak University of Technology in Bratislava, Faculty of Chemical and Food Technology, Institute of Information Engineering, Automation, and Mathematics, Radlinského 9, SK-812 37 Bratislava, Slovak Republic

^b Warsaw University of Technology, Faculty of Civil Engineering, Mechanics and Petrochemistry, Institute of Mechanical Engineering, Department of Process Equipment, Lukasiewiczza 17, 09-400 Plock, Poland

ARTICLE INFO

Article history:

Received 19 January 2018

Received in revised form

13 June 2018

Accepted 16 June 2018

Available online 20 June 2018

Keywords:

Energy savings

Shell-and-tube heat exchangers

PID control

Robust control

Model predictive control

Convex optimization

ABSTRACT

Robust model predictive control (MPC) with integral action is designed for the shell-and-tube heat exchangers that are a part of an industrial heat-exchanger network. The advanced control strategy is used for optimizing the control performance as fouling influences operation of the heat exchangers and causes changes of the heat exchangers' parameters. The time-varying parameters of the heat exchangers are considered as parametric uncertainties and robust MPC is used as it is able to handle processes with uncertainties. Integral action is implemented in the robust MPC to assure offset-free control responses. The extensive simulation case study of the robust MPC and proportional-integral-derivative (PID) control of the shell-and-tube heat exchangers confirms significantly improved control performance and energy savings when the newly designed robust MPC with integral action is used.

© 2018 Elsevier Ltd. All rights reserved.

1. Introduction

Optimal supply and efficient use of energy are important for sustainable development of economies [1]. Recently, advanced strategies have been developed and applied to ensure efficient operation and energy savings in energy-intensive processes. In Ref. [2], the sequential and simultaneous approaches were presented to solve the problem of heat exchanger network (HEN) synthesis. A new methodology combining economic and control system design is proposed for HENs in Ref. [3]. The closed-loop optimized control saved more energy in air-cooled steam condenser units [4]. In Ref. [5], the adaptive controller was designed for the liquid-liquid heat exchangers and implemented in the PLC system.

Model predictive control (MPC) is one of the most developed advanced model-based control strategies. The idea of receding horizon is accepted and the optimal control input is calculated in each control step subject to various input, state or output

constraints and control requirements [6]. MPC is widely used for optimal energy utilization in various fields of industry, and various energy-intensive plants were controlled by MPC as it enabled simultaneously to optimize both, control performance and energy consumption. Multi-layer perception based MPC was used for the thermal power of nuclear superheated-steam supply systems [7]. The results showed the satisfactory improvement in optimizing the thermal power response. MPC for temperature in an industrial electric heating furnace was designed in Ref. [8]. The design is based on the fractional order modelling and the control signals are the results of minimization of the fractional order performance index. The MPC-based scheme of the reference governor was proposed in Ref. [9], where a boiler-turbine system was controlled by the set of interconnected PI controllers. The simulation results confirmed that the proposed strategy improved safety and economic performance. A dynamic simulator was developed for a cycle plant combined with integrated solar collectors and the simulation results in Ref. [10] showed that, in general, fuel consumption was lower under the supervision of the MPC strategy in comparison with the PI controller. Despite intensive research in MPC of thermal processes, there is a lack of applications of this advanced control strategy to optimize the control performance of heat exchangers

* Corresponding author.

E-mail address: juraj.oravec@stuba.sk (J. Oravec).

(HEs) or heat exchanger networks (HENs). Among the reasons, the complex non-linear and non-symmetric behaviour affected by disturbances, measurement noise and other uncertainties play important role [11]. Moreover, the operation of heat exchangers is deteriorated by fouling [11] that is the accumulation of foreign matter on the solid internal or external heat transfer surfaces of processing units. Fouling is determined by the material of the unit as well as hydraulic characteristics of the flows and physical properties of heat-exchanging media [12]. Fouling is also a very important source of uncertainties in the heat exchanger operation as it causes changes of various heat exchangers' parameters. Mathematical description of these phenomena in heat exchangers brings significant complications in control system design and represents a challenging field of research [13]. As HEs have a lot of uncertain parameters, a robust control strategy can be used to optimize their control performance and to ensure energy savings. Robust control represents promising control strategy as it is able to take into account process uncertainties in the controller design and to assure stability and performance for the whole range of uncertain parameters. The robust control of HEs was compared with conventional control in Ref. [14]. Here, the unstructured uncertainty in the controlled process was considered and robust H_∞ approach was used for robust controller design. Improvement of operating efficiency of waste heat energy conversion systems with organic Rankine cycle was achieved by robust H_2 control [15]. Robust MPC combines advantages of robust control treating various types of uncertainties in the controlled process with the benefits of MPC. The process model that is used for prediction of future controlled outputs includes the description of uncertainties and the optimized control inputs are calculated to ensure the robust stability and robust performance for the whole range of uncertain parameters. Robustness analysis with model predictive control for a hybrid ground-coupled heat pump system was studied in Ref. [16]. The robustness analysis is done with respect to state estimation uncertainty. Explicit computations to assure robustness are computationally demanding, therefore, not widely applied. A key problem in robust MPC design is a formulation of the robust MPC problem in terms of a solvable convex optimisation. Linear matrix inequalities (LMIs) represent an effective tool that can be used to reformulate the non-convex optimization problem into a convex one, e.g., in the form of a semidefinite programming (SDP). Intensive research in this field was done in last years. LMI-based robust MPC for the heat exchanger network was designed in Ref. [17]. Possibility to ensure energy savings was demonstrated by simulation results of the closed-loop control. Various alternative robust MPC strategies were studied and energy savings using an alternative robust MPC design for a laboratory heat exchanger were experimentally investigated in Ref. [18].

Although intensive research in robust MPC and its applications has been done, there are still open problems that have to be solved. Robust MPC is a state-feedback-based strategy and the offset-free control response is not ensured when only the proportional (P) state-feedback controller is used. Robust MPC design supposes symmetric constraints on control inputs and controlled outputs and this requirement does not correspond to the situation in practice where non-symmetric constraints are more typical. It is also very important to study robust MPC and its applications from three points of view: control optimization, energy saving and economic operation. This paper aims to optimize control of heat exchangers with uncertainties caused by fouling and to improve the control performance in comparison with conventional PID control. The second goal is to improve the design strategy by adding the integral action to the state-feedback controller and to design the state-feedback proportional-integral (PI) controller instead of only P controller. And the third goal is to find relations between

factors interesting for heat exchangers' operation, i.e., control performance, energy saving and economic operation.

The paper extends the previous works [19] and [20]. In Ref. [19], a detailed mathematical model of the heat-exchanger unit was developed. The model was validated using the data that were experimentally recorded during three-year operation of the plant. Proportional-integral-derivative (PID) controllers were designed for four heat exchangers to reduce the impact of fouling. In Ref. [20], robust MPC for the heat exchanger network is introduced. As heat-exchanger fouling leads to burning of extra fuel, reduction in heat recovery, increasing costs caused by cleaning interventions, as well as non-satisfactory PID control, the robust MPC was developed to overcome these drawbacks and to improve control performance in comparison with conventional PID control. One of the original results is the development of the advanced robust MPC with integral action leading to the design of the robust state-feedback PI controller. This controller was used for control of the shell-and-tube heat exchangers. The other benefit is the extensive simulation case study of the PID control and the robust MPC of the heat exchangers. Compared to the previous work [20], the theoretical derivations and the case study have been significantly extended. Control performance of the heat-exchanger units in the set-point tracking was analysed in detail. The problem of disturbance rejection in the heat-exchanger units was also investigated and the control performance was evaluated. The simulation results were used for comparison of the performance assured using robust MPC and the conventional PID control. Moreover, energy consumed during control and economy of heat exchangers' operation were compared.

The next sections of the paper are organised as follows. Section 2 presents the considered shell-and-tube heat exchangers from the heat exchanger network coupled with a crude-oil distillation unit (CDU). The advanced robust MPC control strategy is presented in Section 3. The obtained results are discussed in Section 4. The main conclusions are summarized in Section 5.

2. Controlled shell-and-tube heat exchangers

The considered industrial network of heat exchangers is coupled with CDU that processes 220 kg s^{-1} of crude oil. The model parameters were identified using the data measured during three-year operation of the real plant in the Polish Oil Company ORLEN. The technical details of the network, determination of the thermal resistance of fouling and mathematical model of the heat exchanger operation were described in Ref. [21].

The controlled process is represented by four selected units from the network [19]. Two shell-and-tube heat exchangers in series create each unit, see Fig. 1. Here, \dot{M}_t , \dot{M}_s are the tube-side and the shell-side stream flow rates, T_{ti} , T_{si} are the tube-side and the shell-side inlet temperatures and T_{to} , T_{so} are the tube-side and the shell-side outlet temperatures.

The scheme of the heat exchanger control is depicted in Fig. 2, where all streams are process streams, i.e., no utility stream is involved. The tube-side stream is a crude oil feed and the shell-side stream is a pumparound, i.e., a circulating reflux from the distillation tower. The pumparound is not a product stream, it is used to remove heat from the distillation column. The stream at a higher temperature that is taken out of the column exchanges heat with the crude oil feed and heats it. Then, it returns back to the column at some higher position in the column with a lower temperature. In this case, there are no upstream or downstream heat exchangers in the pumparound stream. The flow rate of the pumparound changes in a certain flow range by the control valve with no worries that production in a distillation tower will be disturbed.

Assuming no phase change in both liquids, physical properties dependence only on the temperature, negligible heat losses into

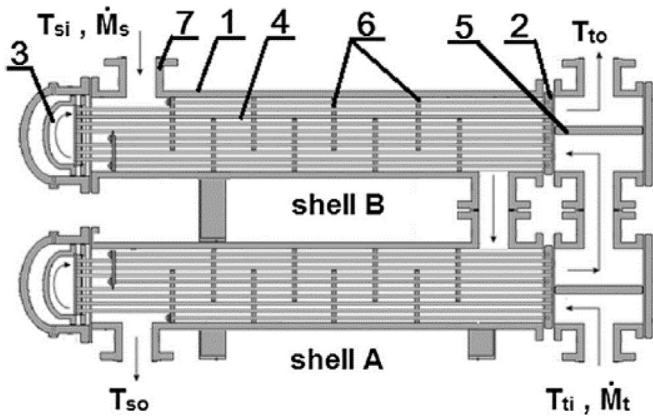


Fig. 1. Flow arrangement in two series-connected heat exchangers, TEMA type AES with floating head: (1) shell, (2) tube sheet, (3) floating head, (4) tubes, (5) pass divider, (6) baffles, and (7) nozzle [19].

the environment and negligible thermal resistance of the tube walls, the lumped-parameter approach was used to model the heat exchange in each unit [21]. The steady-state energy balance of a volume considering the changes in the state of the shell-side fluid, tube walls and tube-side fluid was applied to describe the heat exchangers:

$$\dot{M}_t c_{p,t} (T_{ti} - T_{to}) = n_b \pi d_{in} l h_t (T_{to} - T_{tw}), \tag{1a}$$

$$\dot{M}_s c_{p,s} (T_{si} - T_{so}) = n_b \pi d_{out} l h_s (T_{sw} - T_{so}), \tag{1b}$$

where c_p is the specific heat capacity, h is the surface film conductance, n_b is the number of tubes in one exchanger pass, l is the length of the heat exchanger, d_{in} , d_{out} respectively are inner and outer tube diameters. The subscripts t, s, w denote tube-side, shell-side and wall, respectively.

In the above described control configuration, each heat exchanger cell has four inputs and two outputs. The inputs are the tube-side inlet temperature T_{ti} , the shell-side inlet temperatures T_{si} , the tube-side mass flow \dot{M}_t and the shell-side mass flow \dot{M}_s . The outputs are tube-side and shell-side outlet temperatures T_{to} , T_{so} . Therefore, the mathematical model of each heat-exchanger cell can be represented by eight transfer functions.

For controller design, the models in the form of transfer functions were identified so that the controlled variable was the tube-side outlet temperature, T_{to} . The manipulated variable was the shell-side stream flow rate, \dot{M}_s . This modelling approach had an advantage of yielding simple analytical relationships between the control inputs and the controlled outputs. The models in the form of transfer functions were validated using the data measured during three-year operation of the real plant. The identification is described in detail in Ref. [19]. Disturbances affecting the heat-exchanger unit may be represented by changes of the tube-side inlet temperature T_{ti} , the shell-side inlet temperature T_{si} , and the tube-side mass flow \dot{M}_t . If the shell-side outlet temperature T_{so} changes, it represents a disturbance to the distillation unit.

The PID control strategy was applied to the heat exchangers at

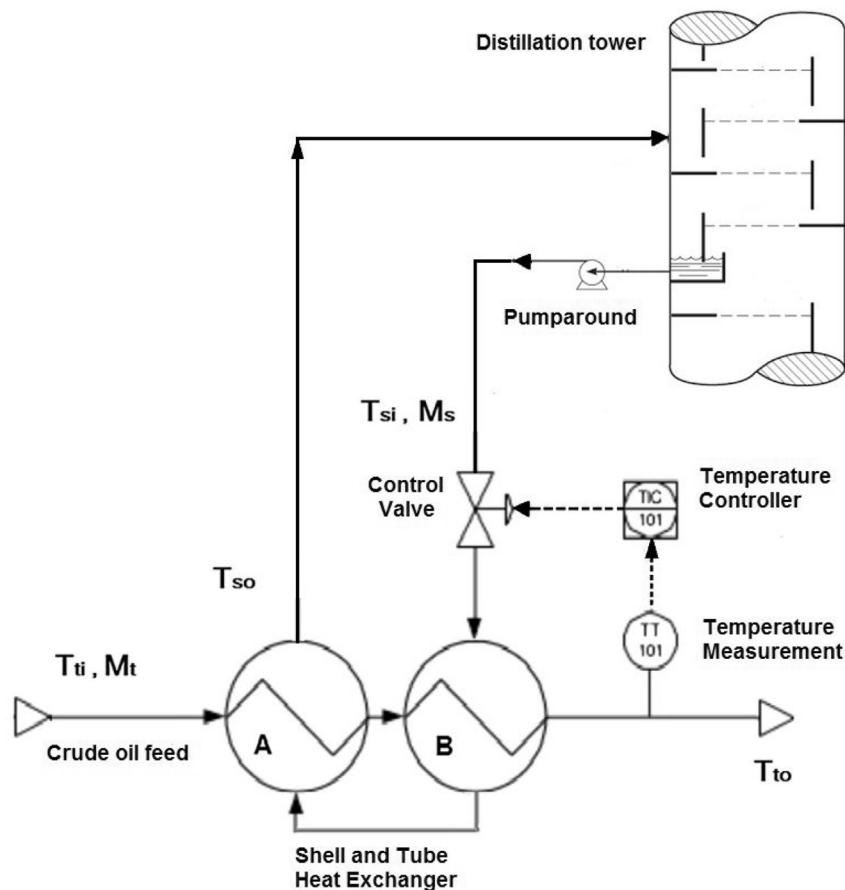


Fig. 2. Scheme of heat exchanger control [19].

first. As the heat exchangers have non-linear behaviour and their gains change with respect to the operating conditions [22], the PID controllers were not able to reflect this phenomenon explicitly during the tuning procedure and an advanced control strategy that was able to perform self-tuning adaptation of the controller gain was needed.

Other serious problems in the industrial operation and control of heat exchangers are caused by fouling build-up. Because of fouling, the operational costs and energy losses increase, the efficacy of the heat exchangers decreases [23], the parameters change in time and an advanced robust control strategy that is able to optimize processing and to handle time-varying parameters is required.

3. Design of the offset-free robust model predictive control

The alternative robust MPC strategy that is able to optimize the operation of process units, to handle time-varying process parameters and to ensure offset-free control was developed.

3.1. State-space model of the controlled process

The robust MPC design procedure requires the linear time-invariant state-space mathematical model of the controlled process in the discrete-time domain that is able to include uncertainties of the controlled process. Such model has the form

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k), \quad \mathbf{x}(0) = \mathbf{x}_0, \quad (2a)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k), \quad (2b)$$

$$[\mathbf{A}, \mathbf{B}] \in \mathbb{A}, \quad (2c)$$

$$\mathbb{A} = \text{convhull}\left\{\left[\mathbf{A}^{(v)}, \mathbf{B}^{(v)}\right], \forall v\right\}, \quad (2d)$$

where $k \geq 0$ is a discrete-time instant, $\mathbf{x}(k) \in \mathbb{R}^{n_x}$ are system states, $\mathbf{u}(k) \in \mathbb{R}^{n_u}$ are manipulated variables, $\mathbf{y}(k) \in \mathbb{R}^{n_y}$ are outputs of the system, \mathbf{x}_0 are initial conditions, $\mathbf{A} \in \mathbb{R}^{n_x \times n_x}$ denotes a system matrix, $\mathbf{B} \in \mathbb{R}^{n_x \times n_u}$ is an input matrix, $\mathbf{C} \in \mathbb{R}^{n_y \times n_x}$ is an output matrix, and $v = 1, \dots, n_v$. Parameter n_v stands for the number of the vertices of the uncertain system. The matrix superscript (v) denotes the v -th vertex system of \mathbb{A} and $\mathbf{A}^{(v)}, \mathbf{B}^{(v)}, v = 1, \dots, n_v$ are the matrices from the linear state-space models of the vertices of the uncertain polytopic system (2).

The model (2) of the controlled process was derived with respect to three years of fouling and four vertex systems represent the heat exchanger (1) at the beginning of operation; (2) after a one-year operation; (3) after a two-year operation; and (4) after a three-year operation.

The task is to design the robust feedback control law

$$\mathbf{u}(k) = \mathbf{F}(k)\mathbf{x}(k), \quad (3)$$

that robustly stabilizes the uncertain system in (2). The matrix $\mathbf{F}(k)$ in (3) is the gain matrix of the robust controller in the k -th control step.

Control performance can be quantified using an analytical integral quality criterion

$$J_{0 \rightarrow n_k} = \sum_{k=0}^{n_k} J(k) = \sum_{k=0}^{n_k} \left(\mathbf{x}(k)^\top \mathbf{Q}\mathbf{x}(k) + \mathbf{u}(k)^\top \mathbf{R}\mathbf{u}(k) \right), \quad (4)$$

where n_k is the number of the evaluated control steps. Analogous to the linear-quadratic (LQ) optimal control approach, the considered

robust MPC strategy assumes the infinity control horizon, see Ref. [24]. The matrices \mathbf{Q}, \mathbf{R} are the square positive-definite weightings of the states $\mathbf{x}(k)$ and the control inputs $\mathbf{u}(k)$, respectively. The goal of robust MPC design is to find the gain matrix $\mathbf{F}(k)$ assuring robust stability for all considered system vertices. Simultaneously, the quadratic criterion J (4) has to be minimized.

The quality of control can be improved when symmetric constraints on the system outputs $\mathbf{y}(k)$ and inputs $\mathbf{u}(k)$ are taken into account in the form

$$-\mathbf{u}_{\text{sat}} < \mathbf{u}(k) < \mathbf{u}_{\text{sat}}, \quad -\mathbf{y}_{\text{sat}} < \mathbf{y}(k) < \mathbf{y}_{\text{sat}}, \quad \forall k \geq 0, \quad (5)$$

where $\mathbf{u}_{\text{sat}} \in \mathbb{R}^{n_u}$ and $\mathbf{y}_{\text{sat}} \in \mathbb{R}^{n_y}$ are the symmetric limit values.

3.2. Augmented state-space model

To assure offset-free set-point tracking, the robust MPC was supplemented by integral action. To implement the robust MPC with integral action, the augmented state-space model was considered with the augmented vector of system states $\mathbf{z} \in \mathbb{R}^{(n_x+n_y)}$ having the form:

$$\mathbf{z}(k) = \begin{bmatrix} \mathbf{x}(k) \\ \sum_{i=0}^k \mathbf{e}(i) \end{bmatrix}, \quad (6)$$

where \mathbf{e} is the control error given as: $\mathbf{e} = \mathbf{w} - \mathbf{y}$, and $\mathbf{w} \in \mathbb{R}^{n_y}$ is the reference value.

The control law in (3) was augmented with respect to the integral action:

$$\mathbf{u}(k) = [\mathbf{F}_p(k) \quad \mathbf{F}_i(k)]\mathbf{z}(k) = \tilde{\mathbf{F}}(k)\mathbf{z}(k), \quad (7)$$

where $\mathbf{F}_p, \mathbf{F}_i$ are the proportional and the integral parts of the linear control law, respectively, i.e. the proportional and the integral gains of the PI controller. Hereafter, the proportional and the integral gains are joined into one gain $\tilde{\mathbf{F}} \in \mathbb{R}^{n_u \times (n_x+n_y)}$.

To ensure offset-free control and to add integral action to the robust MPC, the augmented state-space system was used in the robust control design. This state-space system is described as follows (8)

$$\mathbf{z}(k+1) = \tilde{\mathbf{A}}\mathbf{z}(k) + \tilde{\mathbf{B}}\mathbf{u}(k), \quad \mathbf{z}(0) = \mathbf{z}_0, \quad (8a)$$

$$\mathbf{y}(k) = \tilde{\mathbf{C}}\mathbf{z}(k), \quad (8b)$$

$$[\tilde{\mathbf{A}}, \tilde{\mathbf{B}}] \in \tilde{\mathbb{A}}, \quad (8c)$$

$$\tilde{\mathbb{A}} = \text{convhull}\left(\left\{\left[\tilde{\mathbf{A}}^{(v)}, \tilde{\mathbf{B}}^{(v)}\right], \forall v\right\}\right), \quad (8d)$$

where:

$$\tilde{\mathbf{A}}^{(v)} = \begin{bmatrix} \mathbf{A}^{(v)} & \mathbf{0} \\ -t_s \mathbf{C} & \mathbf{I} \end{bmatrix}, \quad \tilde{\mathbf{B}}^{(v)} = \begin{bmatrix} \mathbf{B}^{(v)} \\ \mathbf{0} \end{bmatrix}, \quad \tilde{\mathbf{C}} = [\mathbf{C} \quad \mathbf{0}]. \quad (9)$$

The quality criterion (4) was modified subject to the augmented system in (8) to design the robust MPC with integral action

$$\tilde{J}_{0 \rightarrow n_k} = \sum_{k=0}^{n_k} \left(\mathbf{z}(k)^\top \tilde{\mathbf{Q}}\mathbf{z}(k) + \mathbf{u}(k)^\top \mathbf{R}\mathbf{u}(k) \right), \quad (10)$$

where $\tilde{\mathbf{Q}} \in \mathbb{R}^{2 \times 2 n_x \times 2 n_x} > \mathbf{0}$ is the weighting matrix for the augmented vector of states \mathbf{z} (6), and has the form:

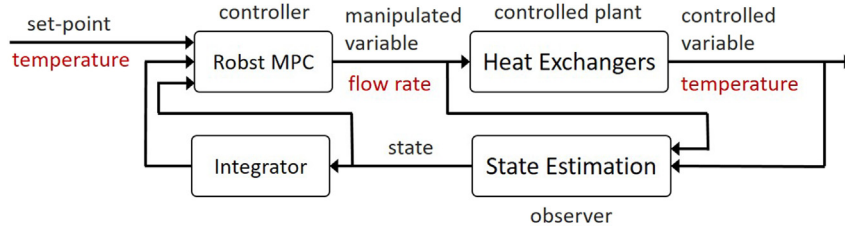


Fig. 3. Control scheme of robust MPC with integral action.

$$\tilde{\mathbf{Q}} = \begin{bmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_I \end{bmatrix}. \quad (11)$$

where \mathbf{Q}_I is the weighting matrix of integral action.

3.3. Design of robust MPC with integral action

The advantage of the proposed design of the robust MPC with integral action is the possibility to reformulate originally non-convex optimization problem to the convex optimization one and to solve it using the convex programming. The optimal solution is obtained as a result of the convex optimization. Moreover, the robust MPC enables computing the manipulated variables so that the constraints on inputs and outputs are respected and the uncertain parameters of the controlled process are taken into account. The receding horizon principle requires solving the optimization problem in each control step and so the process-model mismatch is minimized. In such a way, the robust MPC is able to ensure optimal performance of the controlled process with minimum energy consumption.

The following conditions hold for the controller $\mathbf{F}(k)$ and the symmetric positively defined Lyapunov matrix $\mathbf{P}(k)$

$$\mathbf{P} = \gamma(k)\mathbf{X}(k)^{-1}, \quad \mathbf{Y}(k) = \mathbf{F}(k)\mathbf{X}(k) \Rightarrow \mathbf{F}(k) = \mathbf{Y}(k)\mathbf{X}(k)^{-1}, \quad (12)$$

where $\mathbf{X} = \mathbf{X}^T \in \mathbb{R}^{n_x \times n_x}$ is the weighted inverse-matrix of the quadratic Lyapunov matrix \mathbf{P} , $\gamma \in \mathbb{R}$ is the weighting parameter of \mathbf{X} , and $\mathbf{Y} \in \mathbb{R}^{n_u \times n_x}$ is the auxiliary matrix of the robust feedback controller $\mathbf{F}(k)$ design.

The newly formulated design problem of the robust MPC with integral action was solved as the convex optimization problem formulated in the form of the following SDP:

$$\min_{\gamma, \mathbf{X}, \mathbf{Y}, \mathbf{U}} \gamma \quad (13a)$$

$$\text{s.t. : } \begin{bmatrix} 1 & \star \\ \mathbf{z}(k) & \mathbf{X} \end{bmatrix} > 0, \quad (13b)$$

$$\begin{bmatrix} \mathbf{X} & \star & \star & \star \\ \tilde{\mathbf{A}}^{(0)}\mathbf{X} + \tilde{\mathbf{B}}^{(0)}(\mathbf{E}^{(m)}\mathbf{Y} + \tilde{\mathbf{E}}^{(m)}\mathbf{U}) & \mathbf{X} & \star & \star \\ \tilde{\mathbf{Q}}^{\frac{1}{2}}\mathbf{X} & \mathbf{0} & \gamma\mathbf{I} & \star \\ \mathbf{R}^{\frac{1}{2}}(\mathbf{E}^{(m)}\mathbf{Y} + \tilde{\mathbf{E}}^{(m)}\mathbf{U}) & \mathbf{0} & \mathbf{0} & \gamma\mathbf{I} \end{bmatrix} > 0, \quad (13c)$$

$$\begin{bmatrix} \tilde{\mathbf{A}}^{(v)}\mathbf{X} + \tilde{\mathbf{B}}^{(v)}(\mathbf{E}^{(m)}\mathbf{Y} + \tilde{\mathbf{E}}^{(m)}\mathbf{U}) & \star \\ \mathbf{X} & \mathbf{X} \end{bmatrix} > 0, \quad (13d)$$

$$\begin{bmatrix} \mathbf{X} & \star \\ \mathbf{u}_{\text{sat}}^2\mathbf{I} & \mathbf{U} \end{bmatrix} > 0, \quad (13e)$$

$$\begin{bmatrix} \tilde{\mathbf{C}}(\tilde{\mathbf{A}}^{(v)}\mathbf{X} + \tilde{\mathbf{B}}^{(v)}(\mathbf{E}^{(m)}\mathbf{Y} + \tilde{\mathbf{E}}^{(m)}\mathbf{U})) & \star \\ \mathbf{y}_{\text{sat}}^2\mathbf{I} & \mathbf{U} \end{bmatrix} > 0, \quad (13f)$$

where $v = 1, \dots, n_v$ denotes vertices and $\gamma \in \mathbb{R}$, $\mathbf{X}, \mathbf{Y}, \mathbf{U} \in \mathbb{R}^{n_u \times n_x}$, $\mathbf{E}^{(m)}$, $\tilde{\mathbf{E}}^{(m)}$ are the decision variables. Between them, $\mathbf{E}^{(m)}$ are the diagonal matrices with all variations of 1 and 0 on the principal diagonal and zeroes elsewhere. The complement matrices $\tilde{\mathbf{E}}^{(m)}$ are calculated from $\mathbf{E}^{(m)} = \mathbf{I} - \tilde{\mathbf{E}}^{(m)}$. The symbol \star in (13) denotes symmetric structure of the matrix. \mathbf{I} is the identity matrix and $\mathbf{0}$ is the zero matrix with appropriate dimensions. $\mathbf{u}_{\text{sat}}^2$, $\mathbf{y}_{\text{sat}}^2$ stand for element-wise power of two. Note, the LMI-based optimization problem in (13) has a simplified notation, i.e., all the decision variables are discrete-time-dependent: $\mathbf{X}(k)$, $\mathbf{Y}(k)$, $\mathbf{U}(k)$, and $\gamma(k)$.

The design procedure of the robust MPC strategy is based on optimization of the control performance with respect to a nominal system. Although the theoretical nominal system is an idealized system without an influence of uncertainties, it is possible to create it using mean values of the uncertainties' intervals. The nominal-system-based optimization is represented by (13c). Despite the nominal-system-based optimisation in the robust MPC design, the robust stability is guaranteed for the whole family of uncertain systems represented as the convex hull of all vertex systems, i.e., all combinations of the boundary values of the interval uncertainties. This guaranty is represented by (13d). LMIs in (13e)–(13f) represent optimization of the manipulated variable subject to all combinations of constrained and unconstrained control inputs and controlled outputs. Further details related to the solution of such optimization problem are discussed in Ref. [25]. One of the benefits of this paper is extending the design of the robust MPC with integral action.

Finally, the procedure for the alternative robust MPC with integral action is formulated in Algorithm 1. The scheme of robust MPC with integral action is depicted in Fig. 3, where the state estimation is realised using the model in (2).

Algorithm 1

Design of the alternative robust MPC with integral action.

Require: measured/estimated $\mathbf{x}(k)$, cost function weight matrices $\tilde{\mathbf{Q}}, \mathbf{R}$, constraints $\mathbf{u}_{\text{sat}}, \mathbf{y}_{\text{sat}}$

Ensure: control action \mathbf{u}

1: solve optimization problem in (13)

2: $\tilde{\mathbf{F}} \leftarrow \mathbf{Y}\mathbf{X}^{-1}$ using (3)

3: $\mathbf{u} \leftarrow \tilde{\mathbf{F}}\mathbf{z}$

4. Results and discussion

4.1. Control objectives and methods

The paper extends and analyses the results of the PID control of heat exchangers proposed in Ref. [19] and robust MPC designed in Ref. [20]. Particularly, the novelty of this extensive case study lies in the proposal of the advanced robust MPC with integral action for the heat exchangers affected by fouling and analysis of the closed-loop control performance, energy saving and economic operation in the set-point tracking and disturbance rejection. To investigate the control performance of the proposed robust MPC strategy and PID control, the control setup was considered as it is shown in Fig. 2 to obtain comparable results. 4 heat exchangers were considered, each of them with 4 levels of fouling, i.e., clean and after 1, 2, and 3 years of operation. In total, this case study investigated 16 controlled systems in both: set-point tracking (servo) and disturbance rejection (regulatory) problems. Robust MPC with integral action is compared with PID control based on the control performance, energy consumption represented by hot fluid consumption during control and energy needed for preparing it, and economy of operation represented by the price of energy consumed during control. The controlled variable in each heat exchanger was the tube-side outlet temperature, T_{to} . The manipulated variable was the shell-side stream flow rate, \dot{M}_s . The objective of advanced robust MPC was to optimize the control performance, to minimize energy consumption and to minimize the operation costs.

4.2. Model of heat exchangers

Four heat exchanger units denoted E11AB, E15AB, E30AB, E35AB represent the controlled system. The mathematical model of the heat exchangers was derived in Ref. [21] in the form of transfer functions and had to be transformed into the form of (2) to design the robust MPC. The sampling period $t_s = 5$ s was considered. Four components of the system-state vector \mathbf{x} in (2) corresponded to the states of the heat exchangers: E11AB, E15AB, E30AB, E35AB. The system matrix $\mathbf{A} \in \mathbb{R}^{4 \times 4}$ was a diagonal matrix with the elements $\mathbf{A}_{i,i}$ on the main diagonal, where $i = 1, 2, 3, 4$ represented the heat exchanger E11AB, E15AB, E30AB, and E35AB, respectively. Analogous, the input matrix $\mathbf{B} \in \mathbb{R}^{4 \times 4}$ and the output matrix $\mathbf{C} \in \mathbb{R}^{4 \times 4}$ were the diagonal matrices with elements $\mathbf{B}_{i,i}$, $\mathbf{C}_{i,i}$, $i = 1, \dots, 4$, on the main diagonals and zeroes elsewhere.

Modelling of each of the controlled heat exchangers with three-year fouling led to four vertex systems that represent the heat exchanger (1) at the beginning of operation; (2) after a one-year operation; (3) after a two-year operation; and (4) after a three-year operation. Parameters of the vertex systems were different as fouling caused changes of the heat exchanger parameters in time and the matrices of system in (2) had 4 vertices $\mathbf{A}^{(v)}$, $\mathbf{B}^{(v)}$, $\mathbf{C}^{(v)}$, $v = 1, 2, 3, 4$, representing the considered levels of fouling. The numerical values of these matrices are given in Table 1. The advanced robust MPC design needed also the nominal models of the heat exchangers. Parameters of the nominal model of each heat exchanger were calculated as the mean values of the parameters of corresponding 4 vertex systems. From the robust control viewpoint, the fouling led to the time-varying parametric uncertainty. To remove the steady state error, integral action was added to the robust model predictive controller. Therefore, the matrices $\tilde{\mathbf{A}}$, $\tilde{\mathbf{B}}$, $\tilde{\mathbf{C}}$ of the augmented system in (8) were computed according to (9).

4.3. Control setup

The closed-loop control performance was investigated using MATLAB/Simulink R2014b environment, CPU i5 1.7 GHz and 6 GB

Table 1
Parameters of the mathematical model (2) of the heat exchangers.

Fouling	Model parameters	E11AB	E15AB	E30AB	E35AB
clean	$\mathbf{A}_{i,i}^{(1)}$	0.9248	0.9498	0.9460	0.9083
	$\mathbf{B}_{i,i}^{(1)}$	0.1250	0.1250	0.1250	0.2500
	$\mathbf{C}_{i,i}^{(1)}$	0.1804	0.0924	0.0865	0.1760
1 year	$\mathbf{A}_{i,i}^{(2)}$	0.8991	0.9281	0.9311	0.8926
	$\mathbf{B}_{i,i}^{(2)}$	0.1250	0.1250	0.1250	0.2500
	$\mathbf{C}_{i,i}^{(2)}$	0.1453	0.0978	0.0082	0.2148
2 years	$\mathbf{A}_{i,i}^{(3)}$	0.9237	0.9270	0.9200	0.9260
	$\mathbf{B}_{i,i}^{(3)}$	0.2500	0.2500	0.2500	0.2500
	$\mathbf{C}_{i,i}^{(3)}$	0.1556	0.1459	0.1535	0.1658
3 years	$\mathbf{A}_{i,i}^{(4)}$	0.9116	0.9160	0.8991	0.9131
	$\mathbf{B}_{i,i}^{(4)}$	0.1250	0.1250	0.1250	0.1250
	$\mathbf{C}_{i,i}^{(4)}$	0.1839	0.1680	0.1776	0.2086
mean	$\mathbf{A}_{i,i}^{(0)}$	0.9148	0.9302	0.9241	0.9100
	$\mathbf{B}_{i,i}^{(0)}$	0.1563	0.1563	0.1563	0.2188
	$\mathbf{C}_{i,i}^{(0)}$	0.1663	0.1260	0.1064	0.1913

RAM. MUP toolbox [26] was used to design robust MPC with integral action, the optimization problems were parsed using the YALMIP toolbox [27] and solved by the means of the solver MOSEK [28].

The PID control was designed using models in the form of transfer functions identified for 4 heat exchangers and for 4 above mentioned levels of fouling. The PID controllers were tuned using the Ziegler Nichols method [29] and the transfer functions of the PID controllers had the form

$$G_C(s) = K_P + \frac{K_I}{s} + K_D s, \quad (14)$$

where s is the operator of Laplace domain, K_P is the proportional, K_I is the integral and K_D is the derivative gain. The parameters of the PID controllers designed in Ref. [19] are presented in Table 2. These parameters were used for PID control of the heat exchangers also in this case study.

Tuning of PID controllers was simple, implementation of PID controllers was not computationally demanding, and the well-tuned PID controllers ensured satisfying control performance. But, the control inputs were not optimized with respect to the input and output constraints, the process-model mismatch, the system uncertainties, and minimization of energy consumption. All these requirements can be ensured using robust MPC.

The robust MPC with integral action was designed considering

Table 2
PID controller parameters designed by the Ziegler-Nichols method [19].

Fouling	PID Parameters	E11AB	E15AB	E30AB	E35AB
clean	K_P	69.8	2.4	38.2	32.5
	K_I	4.6	0.1	1.9	2.4
	K_D	255.5	13.8	182.3	106.7
1-year operation	K_P	70.7	3.4	41.5	35.8
	K_I	3.6	0.1	2.0	1.9
	K_D	334.2	19.2	209.3	158.5
2-year operation	K_P	73.9	3.5	44.1	51.1
	K_I	3.7	0.1	2.0	2.4
	K_D	358.5	20.4	230.1	261.8
3-year operation	K_P	88.7	3.9	49.7	62.2
	K_I	4.2	0.2	2.2	2.7
	K_D	451.7	22.9	275.2	342.1

the quality criterion J in (10), where the weighting matrices were tuned as $\mathbf{Q} = 100 \times \mathbf{I}$, $\mathbf{Q}_1 = 0.936 \times \mathbf{I}$, $\mathbf{R} = 10 \times \mathbf{I}$. Here, \mathbf{I} is the 4×4 identity matrix.

The closed-loop control performance of the designed PID controllers and the advanced robust MPC were investigated in the set-point tracking problem and the disturbance rejection problem. The step change of the set-point $0 \rightarrow 1$ °C in time $t = 0$ s was considered, and the disturbance $0 \rightarrow 0.2$ °C in time $t = 150$ s represented increasing of the temperature of the tube-side flow coming from the upstream unit.

4.4. Closed-loop control performance

The closed-loop system responses obtained using both, the PID control and the robust MPC (RMPC) are shown in Figs. 4–7. The control trajectories of the temperature are depicted in the normalized form, i.e., variable Δ temperature represents the change of the temperature from its steady-state value.

Two types of uncertain parameters were considered in the presented case study. The first uncertainty was fouling that changed parameters of HES and represented the parametric uncertainty included in the system model. As 3 years of fouling were considered, four vertex systems were assumed in (2). The second uncertainty was the external disturbance represented by an unexpected change of the temperature of the tube-side flow coming from the upstream unit. Robustness of the designed controllers was investigated subject to external disturbance in the disturbance rejection problem. As fouling was taken into account in the process model as the parametric uncertainty, the controller robustness was verified in both, the set-point tracking and disturbance rejection.

In comparison with the results of PID control presented in Ref. [19], this case study considered the simplified linear models of the heat exchangers. The heat exchangers were described using the set of transfer functions for simulation of PID control and in the form of the linear state-space models (2) for robust MPC.

The control performance was evaluated also using various analytical quality criteria. The integral criterion ISE is defined as follows:

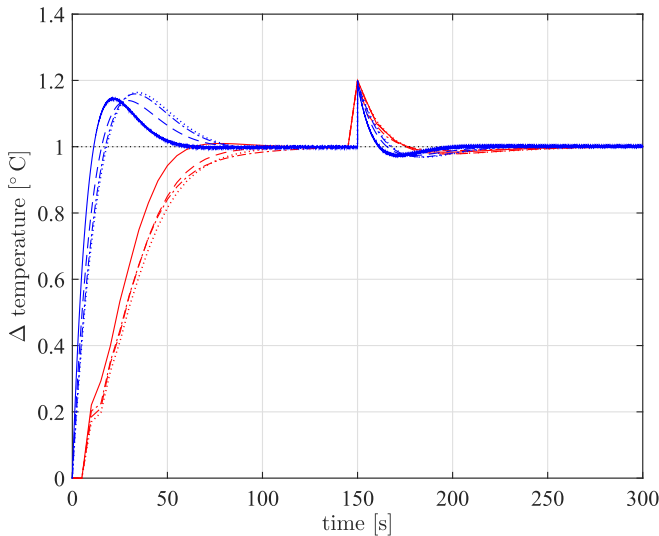


Fig. 4. Closed-loop step response of E11AB model ensured by PID controllers (blue) and robust MPC (red) for clean E11AB (solid) and for E11AB after 1 year (dashed), after 2 years (dash-dotted), and after 3 years (dotted) of fouling. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

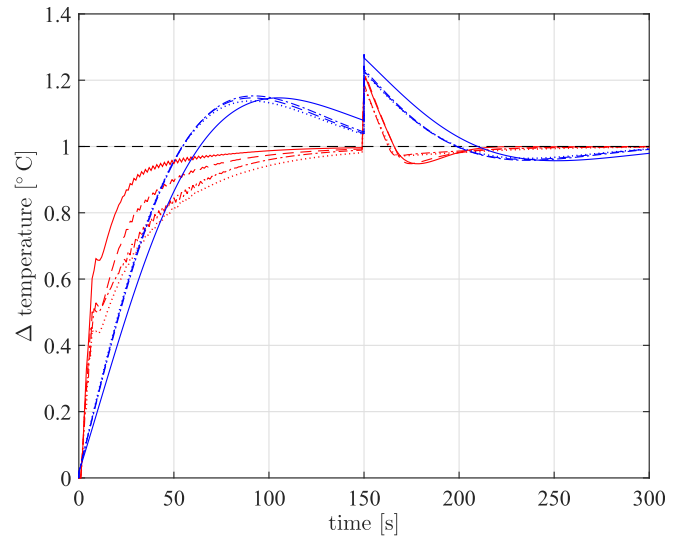


Fig. 5. Closed-loop step response of E15AB model ensured by PID controllers (blue) and robust MPC (red) for clean E15AB (solid) and for E15AB after 1 year (dashed), after 2 years (dash-dotted), and after 3 years (dotted) of fouling. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

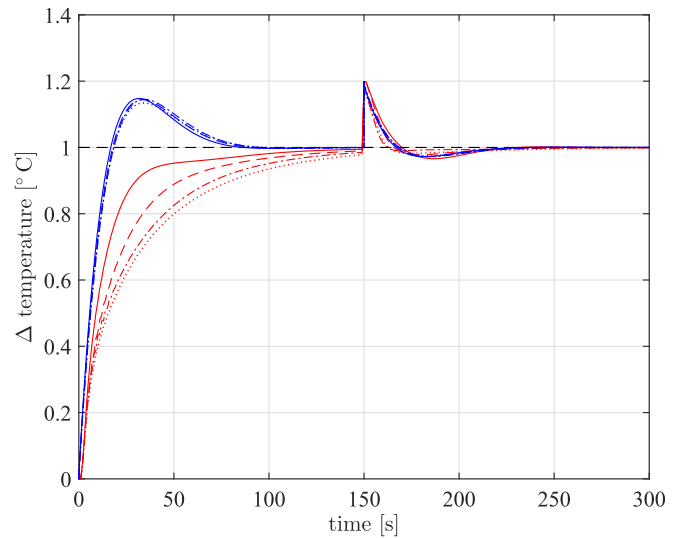


Fig. 6. Closed-loop step response of E30AB model ensured by PID controllers (blue) and robust MPC (red) for clean E30AB (solid) and for E30AB after 1 year (dashed), after 2 years (dash-dotted), and after 3 years (dotted) of fouling. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

$$ISE = \int_{t=0}^{\infty} e(t)^2 dt \approx t_s \sum_{k=0}^{n_k} e(k)^2, \quad (15)$$

where $e = w - y$ is the control error given as the difference between the set-point w and the controlled variable y , t_s is the sampling time, and n_k is the number of control steps. The lower the ISE value is, the better closed loop control performance is achieved.

Table 3 shows the computed values of ISE in the set-point tracking using the PID control and RMPC. The data are presented for the clean heat exchangers, and for the heat exchangers after 1, 2 and 3 years of fouling. The last row summarizes the mean values of

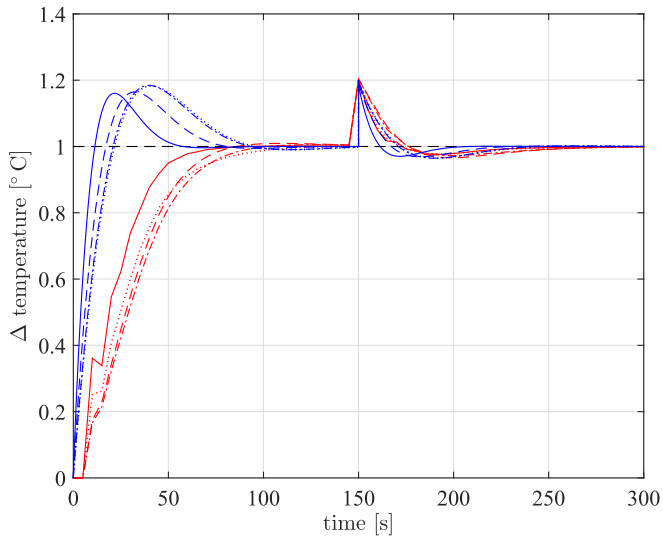


Fig. 7. Closed-loop step response of E35AB model ensured by PID controllers (blue) and robust MPC (red) for clean E35AB (solid) and for E35AB after 1 year (dashed), after 2 years (dash-dotted), and after 3 years (dotted) of fouling. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

Table 3
ISE values in the set-point tracking.

ISE ($^{\circ}\text{C}^2\text{ s}$)								
Heat Exchanger Fouling/Strategy	E11AB		E15AB		E30AB		E35AB	
	PID	RMPC	PID	RMPC	PID	RMPC	PID	RMPC
clean	2.9	4.0	18.4	6.0	4.4	7.5	3.0	3.4
1 year	3.9	4.5	15.4	8.2	4.6	9.5	4.7	4.1
2 years	4.8	4.6	15.7	8.9	4.8	10.4	6.2	4.5
3 years	5.0	4.8	15.1	10.0	4.6	11.1	6.4	4.8
mean	4.1	4.5	16.1	8.3	4.6	9.6	5.1	4.2

ISE. The evaluated values of ISE in the disturbance rejection are summarized in Table 4.

Based on the results presented in Table 3, the PID controllers ensured better closed-loop control of the heat exchanger E30AB during the whole time of operation. PID control was also better for E11AB and E35AB at the beginning of their operation, when fouling was small, i.e. clean E11AB and 1-year operation of E11AB and clean E35AB. PID control was worse in E15AB control. The RMPC assured better control performance subject to ISE in E15AB control, in E35AB control with the exception of the clean heat exchanger and in E11AB with fouling after 2 and 3 years of operation. In disturbance rejection, the RMPC approach outperforms the PID controllers in all the investigated cases except clean E30AB and E30AB after the 1-year operation, see Table 4. This behaviour is typical for

Table 4
ISE values in the disturbance rejection.

ISE ($^{\circ}\text{C}^2\text{ s}$)								
Heat Exchanger Fouling/Strategy	E11AB		E15AB		E30AB		E35AB	
	PID	RMPC	PID	RMPC	PID	RMPC	PID	RMPC
clean	0.10	0.10	1.30	0.22	0.16	0.25	0.11	0.09
1 year	0.15	0.10	0.76	0.17	0.17	0.19	0.18	0.08
2 years	0.18	0.10	0.81	0.10	0.18	0.10	0.23	0.06
3 years	0.19	0.10	0.73	0.05	0.17	0.04	0.24	0.06
mean	0.16	0.10	0.90	0.13	0.17	0.15	0.19	0.07

the robust controllers that are designed to minimize the influence of uncertain parameters. Uncertainty in the controlled heat exchangers is caused by fouling and disturbances.

The control performance was judged also using the maximal overshoot that is given by:

$$\sigma_{\max} = \frac{y_{\max} - y(\infty)}{y(\infty) - y(0)} \times 100\%, \quad (16)$$

where y_{\max} is the maximal value of the controlled variable, $y(0)$ is the initial value of controlled variable, and y_{∞} is its steady-state value. The results were obtained considering again the clean heat exchangers and the heat exchangers after 1, 2, and 3 years of fouling, respectively. The overshoot of the temperature response decreases the quality of control and in the worst case, it may lead to the thermal degradation of the final product. Therefore one of the goals of control was to minimize the overshoots.

Table 5 shows the obtained values of σ_{\max} in the set-point tracking considering both strategies, i.e., the PID control and RMPC. The values of σ_{\max} ensured in the disturbance rejection are in Table 6. As it can be seen in Figs. 4–7, significant overshoots are presented in the PID control responses. The implementation of RMPC removed or significantly decreased the overshoots for each heat exchanger in set-point tracking, see Table 5. The control performance in disturbance rejection led to the overshoots or undershoots in both considered control strategies, see Table 6. RMPC ensured lower values of σ_{\max} in all control scenarios, except control of the clean E15AB, E15AB after 1 year of fouling and clean E35AB. On the other hand, under the same control conditions, the PID controllers ensured rather poor control performances in both investigated control conditions, i.e., the set-point tracking and the disturbance rejection problems, cf. Figs. 5 and 7 (blue).

Finally, the amount of energy needed for control is very important in practice. The energy has to be minimized and can be measured by the control input or some variables calculated based on the control input. Therefore, the control performance of the heat exchangers was analysed based on the consumption of the heating medium. Particularly, the overall consumption of the shell-side medium $\Delta M_{s,\text{total}}$ was evaluated as the integral of the normalized shell-side flow rate given by:

Table 5
Maximum overshoot in the set-point tracking.

Maximum overshoot (%)								
Heat Exchanger Fouling/Strategy	E11AB		E15AB		E30AB		E35AB	
	PID	RMPC	PID	RMPC	PID	RMPC	PID	RMPC
clean	14.6	1.0	6.3	1.7	14.8	0.0	16.0	0.5
1 year	13.9	0.0	10.4	0.0	14.4	0.0	16.4	0.1
2 years	15.9	0.0	10.5	0.0	14.6	0.0	18.7	0.0
3 years	16.4	0.0	9.8	0.0	13.5	0.0	18.6	0.0
mean	15.2	0.2	9.2	0.3	14.3	0.3	17.4	0.1

Table 6
Maximum overshoot in the disturbance rejection.

Maximum overshoot (%)								
Heat Exchanger Fouling/Strategy	E11AB		E15AB		E30AB		E35AB	
	PID	RMPC	PID	RMPC	PID	RMPC	PID	RMPC
clean	13.8	10.5	7.5	20.0	13.9	6.6	14.8	16.2
1 year	13.3	9.9	12.8	19.1	13.7	4.2	15.6	12.9
2 years	15.4	8.0	13.0	6.5	13.9	2.9	18.1	11.8
3 years	15.8	5.6	11.6	6.4	12.7	2.8	18.0	11.1
mean	15.2	6.8	11.2	13.0	13.6	4.1	16.6	10.4

Table 7
Normalized consumption of shell-side medium.

Normalized consumption of shell-side medium $\times 10^3$ (kg)								
Heat Exchanger	E11AB		E15AB		E30AB		E35AB	
Fouling/Strategy	PID	RMPC	PID	RMPC	PID	RMPC	PID	RMPC
clean	16.2	8.6	15.5	7.9	16.0	8.2	15.6	8.5
1 year	16.7	8.5	15.6	7.8	16.2	8.2	16.1	8.5
2 years	16.9	8.5	15.6	7.8	16.3	8.1	16.5	8.4
3 years	17.0	8.3	15.6	7.7	16.4	8.1	16.7	8.0
mean	16.7	8.5	15.6	7.8	16.2	8.1	16.2	8.3

Table 8
Total energy consumption of control.

Total energy consumption of control $\times 10^3$ (kJ)								
Heat Exchanger	E11AB		E15AB		E30AB		E35AB	
Fouling/Strategy	PID	RMPC	PID	RMPC	PID	RMPC	PID	RMPC
clean	648	344	620	316	640	328	624	340
1 year	668	340	624	312	648	328	644	340
2 years	676	340	624	312	652	324	660	336
3 years	680	332	624	308	656	324	668	320
mean	668	340	624	312	648	324	648	332

$$\Delta M_{s,\text{total}} = \int_0^{\infty} \Delta \dot{M}_s(t) dt \approx t_s \sum_{k=0}^{n_k} \dot{M}_s(k). \quad (17)$$

where $\Delta \dot{M}_s$ is the normalized shell-side flow rate represented as the change of the shell-side flow rate from its steady-state value. The aim was to minimize the overall consumption of the shell-side medium $\Delta M_{s,\text{total}}$. The values of $\Delta M_{s,\text{total}}$ ensured by PID controllers and RMPC are summarized in Table 7.

As the advanced controller ensured lower consumption of the shell-side medium, it led to the reduced energy consumption in the distillation column. Analysis of the energy consumption in the distillation column is a very complex problem, so energy consumption was evaluated in a simplified way, as in the case of heating the medium in the furnace. The considered heating value of heavy fuel oil burned in the process furnaces to produce hot medium was approximately 40×10^3 kJ kg⁻¹, see Ref. [21]. Then, the total energy consumption needed to preheat the calculated amount of the hot medium given by $\Delta M_{s,\text{total}}$ is summarised in Table 8. From the energy consumption viewpoint, as it can be seen in Table 8, the control performance of the RMPC strategy for each heat exchanger outperforms twice the performances of the PID controllers for all considered cases of fouling. The economy of HEs operation is also important. Based on the total energy consumption evaluated in Table 8 and using the approximated price of heavy fuel oil 0.4 EUR/kg (see Ref. [21]), RMPC ensured significant operation cost savings from 113 600 EUR in the worst-case, up to 139 200 EUR in the best-case.

5. Conclusions

Robust MPC of shell-and-tube heat exchangers with fouling was studied and compared with conventional PID control. The advanced robust MPC strategy with integral action was developed that enabled the design of the robust state-feedback PI controller. This controller was used for control of the shell-and-tube heat exchangers and the simulation results confirmed that robust MPC with integral actions ensured offset-free control performance. Extensive simulation case study of PID control and robust MPC of

the heat exchangers was done to study not only control performance but also energy saving and operational economy. The results of 32 simulation experiments are presented to compare the robust and PID controller properties in the set-point tracking (servo problem) and the results of the same number of experiments are presented for the disturbance rejection (regulatory problem). According to the maximum overshoot, robust MPC assured smooth control responses in the set-point tracking and smaller overshoots almost in all experiments in the disturbance rejection. According to the analytical quality criterion ISE, the robust MPC outperformed conventional PID control in most of the experiments in both, the set-point tracking and disturbance rejection. PID control was able to assure better results only for some clear HEs and HEs with low fouling. Energy consumption was measured by consumption of the shell-side medium during control as the flow rate of this medium was the manipulated variable. Energy consumption was calculated also as the energy needed for preparing the hot shell-side medium. Robust MPC outperformed almost twice conventional PID control in all experiments according to these two criteria and assured energy savings. The price of consumed energy served for evaluating economy of HEs operation. RMPC ensured operation cost savings from 113 600 EUR in the worst-case, up to 139 200 EUR in the best-case.

Robust MPC with integral action is a very promising strategy for the control of energy-intensive processes. It is able to ensure optimal control performance, minimum energy consumption and economic operation of these processes. The disadvantages of this strategy are much more complicated controller design, much higher computational load and more complex implementations on convection control systems.

Acknowledgement

J. Oravec, M. Bakošová, A. Vasičkaninová, and A. Mészáros highly acknowledge the contribution of the Scientific Grant Agency of the Slovak Republic under the grants 1/0112/16, 1/0403/15, and the Slovak Research and Development Agency under the project APVV-15-0007.

References

- [1] Liu ZY, Varbanov PS, Klemeš JJ, Yong JY. Recent developments in applied thermal engineering: process integration, heat exchangers, enhanced heat transfer, solar thermal energy, combustion and high temperature processes and thermal process modelling. *Appl Therm Eng* 2016;75:5762. <https://doi.org/10.1016/j.applthermaleng.2016.06.183>.
- [2] Escobar M, Trierweiler J. Optimal heat exchanger network synthesis: a case study comparison. *Appl Therm Eng* 2013;51:801826. <https://doi.org/10.1016/j.applthermaleng.2012.10.022>.
- [3] Masoud I, Abdel-Jabbar N, Qasim M, Chebbi R. Methodological framework for economical and controllable design of heat exchanger networks: steady-state analysis, dynamic simulation, and optimization. *Appl Therm Eng* 2016;104:439449. <https://doi.org/10.1016/j.applthermaleng.2016.05.026>.
- [4] Yang T, Wang W, Zeng D, Liu J, Cui C. Closed-loop optimization control on fan speed of air-cooled steam condenser units for energy saving and rapid load regulation. *Energy* 2017;135:394404. <https://doi.org/10.1016/j.energy.2017.06.142>.
- [5] Fratzczak M, Czczot J, Nowak P, Metzger M. Practical validation of the effective control of liquid-liquid heat exchangers by distributed parameter balance-based adaptive controller. *Appl Therm Eng* 2018;129:549556. <https://doi.org/10.1016/j.applthermaleng.2017.10.056>.
- [6] Mayne DQ. Model predictive control: recent developments and future promise. *Automatica* 2014;12:2967–86. <https://doi.org/10.1016/j.automatica.2014.10.128>.
- [7] Dong Z, Zhang Z, Dong Y, Huang X. Multi-layer perception based model predictive control for the thermal power of nuclear superheated-steam supply systems. *Energy* 2018;151:116125. <https://doi.org/10.1016/j.energy.2018.03.046>.
- [8] Zhanga R, Zoua Q, Caob Z, Gaob F. Design of fractional order modeling based extended non-minimalstate space MPC for temperature in an industrial electric heating furnace. *J Process Contr* 2017;56:13–22. <https://doi.org/10.1016/j.jprocont.2017.05.003>.

- [9] Klaučo M, Kvasnica M. Control of a boiler-turbine unit using MPC-based reference governors. *Appl Therm Eng* 2017;110:1437–47. <https://doi.org/10.1016/j.applthermaleng.2016.09.041>.
- [10] Ponce CV, Sáez D, Bordons C, Núñez A. Dynamic simulator and model predictive control of an integrated solar combined cycle plant. *Energy* 2016;109:974–86. <https://doi.org/10.1016/j.energy.2016.04.129>.
- [11] Fraczkak M, Nowak P, Czczot J, Metzger M. Simplified dynamical input-output modeling of plate heat exchangers – case study. *Appl Therm Eng* 2016;98:880893. <https://doi.org/10.1016/j.applthermaleng.2016.01.004>.
- [12] Demirskyy OV, Kapustenko PO, Khavin GL, Arsenyeva OP, Matsegora OI, Kusakov SK, Bocharnikova IO, Tovazhnianskiy VI. Investigation of fouling in plate heat exchangers at sugar factory. *Chem Eng Trans* 2016;52:583–8. <https://doi.org/10.3303/CET1652098>.
- [13] Zahid K, Patel R, Mujtaba I. Development of a dynamic fouling model for a heat exchanger. *Chem Eng Trans* 2016;52:1135–40. <https://doi.org/10.3303/CET1652190>.
- [14] Dulau M, Oltean S, Gligor A. Conventional vs. robust control on heat-exchangers. *Proced Technol* 2015;19:534–40. <https://doi.org/10.1016/j.jprotcy.2015.02.076>.
- [15] Zhang J, Lin M, Chen J, Xu J, Li K. PLS-based multi-loop robust H2 control for improvement of operating efficiency of waste heat energy conversion systems with organic rankine cycle. *Energy* 2017;123:460–72. <https://doi.org/10.1016/j.energy.2017.01.131>.
- [16] Antonov S, Helsen L. Robustness analysis of a hybrid ground coupled heat pump system with model predictive control. *J Process Contr* 2016;47:191–200. <https://doi.org/10.1016/j.procont.2016.08.009>.
- [17] Bakošová M, Mészáros A, Klemes J, Oravec J. Robust and optimal control approach for exothermic reactor stabilization. *Theor Found Chem Eng* 2012;46:740–6. <https://doi.org/10.1134/S0040579512060036>.
- [18] Oravec J, Bakošová M, Mészáros A. Robust model predictive control of heat exchangers in series. *Chem Eng Trans* 2016;52:253–8. <https://doi.org/10.3303/CET1652043>.
- [19] Trafczynski M, Markowski M, Alabrudzinski S, Urbaniec K. The influence of fouling on the dynamic behaviour of PID-controlled heat exchangers. *Appl Therm Eng* 2016;109:727–38. <https://doi.org/10.1016/j.applthermaleng.2016.08.142>.
- [20] Oravec J, Trafczynski JM, Bakošová M, Markowski M, Mészáros A, Urbaniec K. Robust model predictive control of heat exchanger network in the presence of fouling. *Chem Eng Trans* 2017;61:334–42. <https://doi.org/10.3303/CET1761054>.
- [21] Trafczynski M, Markowski M, Urbaniec K. Validation of the method for determination of the thermal resistance of fouling in shell and tube heat exchangers. *Energy Convers Manag* 2013;76:307–13. <https://doi.org/10.1016/j.enconman.2013.07.052>.
- [22] Lipták BG. *Instrument engineers' handbook, Vol. 2: process control and optimization [Chapter 8]. 29 Heat Exchanger Control and Optimization*. fourth ed. USA: CDC Press; 2006.
- [23] Jelemenský M, Sharma A, Paulen R, Fikar M. Time-optimal control of diafiltration processes in the presence of membrane fouling. *Comput Chem Eng* 2016;91:343–51. <https://doi.org/10.1016/j.compchemeng.2016.04.018>.
- [24] Mikleš J, Fikar M. *Process modelling, identification, and control*. Berlin Heidelberg: Springer Verlag; 2007.
- [25] Oravec J, Bakošová M. Alternative LMI-based robust MPC design approaches. In: *Proceedings of the 8th IFAC symposium on robust control design*, vol. 8. Bratislava, Slovak Republic: Elsevier; 2015. p. 180–4. <https://doi.org/10.1016/j.ifacol.2015.09.454>.
- [26] Oravec J, Bakošová M. MUP toolbox for MATLAB. 2014. <https://bitbucket.org/oravec/mup/wiki/Home>.
- [27] Löfberg J. YALMIP : a toolbox for modeling and optimization in MATLAB. In: *Proc. Of the CACSD conf.*, Taipei, Taiwan; 2004. <http://users.isy.liu.se/johanl/yalmip>.
- [28] ApS M. MOSEK. 2015. <https://mosek.com/>.
- [29] Ziegler J, Nichols NB. Optimum settings for automatic controllers. *Trans A.S.M.E.* 1942;11:759–65.

C: output matrix of the state space system
C̄: output matrix of the extended state space system
 c_p : specific heat capacity, J kg⁻¹ K⁻¹
 d_{in} : inner tube diameter, m
 d_{out} : outer tube diameter, m
 e : control error, °C
E: auxiliary matrix of controller design
Ē: complementary auxiliary matrix of controller design
F: gain of the state-feedback control law
F̄: proportional and integral gains of the state-feedback control law
F_I: integral gain of the state-feedback control law
F_P: proportional gain of the state-feedback control law
 G_C : transfer function of PID controller
 h : surface film conductance, W m⁻² K⁻¹
I: identity matrix
 J : quadratic quality criterion
 \bar{J} : extended quadratic quality criterion
 k : sample of discrete time domain, s
 K_D : derivative gain of PID controller
 K_I : integral gain of PID controller
 K_P : proportional gain of PID controller
 l : length of heat exchanger, m
 M_S : shell-side mass flow, kg s⁻¹
 M_T : tube-side mass flow, kg s⁻¹
 n_b : number of tubes in one exchanger pass
 n_c : number of control steps
 n_u : total number of system inputs
 n_v : total number of system vertices
 n_x : total number of system states
 n_y : total number of system outputs
P: Lyapunov matrix
Q: weighting matrix of system states
Q̄: weighting matrix of the extended system states
Q_I: weighting matrix of integral action
R: weighting matrix of system inputs
 \mathbb{R} : Euclidean space of real numbers
 s : operator of Laplace domain
 t : time, s
 t_s : sampling time, s
 T_{ti} : tube-side inlet temperature, °C
 T_{to} : tube-side outlet temperature, °C
u: inputs of the state space system, kg s⁻¹
 u_{sat} : constraints on the manipulated variables, kg s⁻¹
U: auxiliary matrix of controller design
 v : vertex system
 w : set-point temperature, °C
x: states of the state space system, °C
 x_0 : initial conditions of the state space system, °C
X: weighted inverted Lyapunov matrix
y: outputs of the state space system, °C
 y_{max} : maximal value of output variable, °C
 y_{sat} : constraints on the controlled variables, °C
Y: auxiliary matrix of controller design
 z : states of the extended state space system, °C
0: zero matrix

Greek letters

γ : weighting parameter of Lyapunov matrix
 ΔM_s : normalized shell-side mass flow, kg s⁻¹
 $\Delta M_{s,total}$: normalized mass of the shell-side medium, kg
 σ_{max} : maximal overshoot, %

Abbreviations

CDU: crude-oil distillation unit
HE: heat exchanger
HEN: heat exchanger network
ISE: integral square error
LMI: linear matrix inequality
MPC: model predictive control
PID: proportional-integral-derivative
RMPC: robust model predictive control
SDP: semidefinite programming

Nomenclature

Symbols

A: system matrix of the state space system
Ā: system matrix of the extended state space system
 \mathbb{A} : set of all state space systems
B: input matrix of the state space system
B̄: input matrix of the extended state space system

Slovenská technická univerzita v Bratislave
Fakulta chemickej a potravinárskej technológie
Ústav informatizácie, automatizácie a matematiky

SÚBOR 5 VYBRANÝCH NAJVÝZNAMNEJŠÍCH CITÁCIÍ

doc. Ing. Juraj Oravec, PhD.

Bratislava, 2024

Citovaná práca: ADC12 Jiang, Y. [25%] – Oravec, J. [25%] – Houska, B. [25%] – Kvasnica, M. [25%]: Parallel MPC for Linear Systems with Input Constraints. *IEEE Transactions on Automatic Control*, č. 7, zv. 66, str. 3401–3408, 2021. [Citované: 13] IF: 6,6

Citujúca práca: Särkkä, S. - García-Fernández, AF: Temporal Parallelization of Dynamic Programming and Linear Quadratic. *IEEE Transactions on Automatic Control*, č. 2, zv. 68, str. 851-866, 2023.

Temporal Parallelization of Dynamic Programming and Linear Quadratic Control

Publisher: **IEEE**

[Cite This](#)


[PDF](#)

Simo Särkkä  ; Ángel F. García-Fernández  [All Authors](#)

1
Cites in
Paper

1212
Full
Text Views



 Open Access

Under a [Creative Commons License](#)

- Abstract**
- Document
- Sections
 - I. Introduction
 - II. Background
 - III. Parallel Optimal Control
 - IV. Parallel Linear Quadratic Tracker
 - V. Implementation and Computational Complexity
- Show Full Outline ▾
- Authors
- Figures
- References


Abstract:

This article proposes a general formulation for temporal parallelization of dynamic programming for optimal control problems. We derive the elements and associative operators to be able to use parallel scans to solve these problems with logarithmic time complexity rather than linear time complexity. We apply this methodology to problems with finite state and control spaces, linear quadratic tracking control problems, and to a class of nonlinear control problems. The computational benefits of the parallel methods are demonstrated via numerical simulations run on a graphics processing unit.

Published in: [IEEE Transactions on Automatic Control](#) (Volume: 68 , Issue: 2, February 2023)

Page(s): 851 - 866

DOI: [10.1109/TAC.2022.3147017](https://doi.org/10.1109/TAC.2022.3147017)

Date of Publication: 31 January 2022  **Publisher:** IEEE

ISSN Information:

Print ISSN: 0018-9286

Electronic ISSN: 1558-2523

CD: 2334-3303

Funding Agency:

[10.13039/501100002341-Academy of Finland](#)

requires positive definite matrices in the cost function and may require regularization. An algorithm to approximately solve an optimal control problem by solving different subproblems with partially overlapping time windows is provided in [19], and an approximate parallel algorithm for linear MPC is given in [20]. Calvet *et al.* [21] provided combination rules to separate the dynamic programming algorithm into different subproblems across the temporal domain. These combination rules are the foundation for temporal parallelization.

20. Y. Jiang, J. Oravec, B. Houska and M. Kvasnica, "Parallel MPC for linear systems with input constraints", *IEEE Trans. Autom. Control*, vol. 66, no. 7, pp. 3401-3408, Jul. 2021.

[Show in Context](#)

[View Article](#)



[Google Scholar](#)



Citovaná práca: ADC12 Jiang, Y. [25%] – Oravec, J. [25%] – Houska, B. [25%] – Kvasnica, M. [25%]: Parallel MPC for Linear Systems with Input Constraints. *IEEE Transactions on Automatic Control*, č. 7, zv. 66, str. 3401–3408, 2021. [Citované: 13] IF: 6,6

Citujúca práca: Cao, W.K. - Liu, S. - Li, J.W. - Zhang, Z.L. - He, H.W.: Analysis and Design of Adaptive Cruise Control for Smart Electric Vehicle With Domain-Based Poly-Service Loop Delay. *IEEE transactions on industrial electronics*. č. 1, zv. 70, str. 866-877, 2023.

Analysis and Design of Adaptive Cruise Control for Smart Electric Vehicle With Domain-Based Poly-Service Loop Delay

Publisher: IEEE

Cite This

PDF

Wanke Cao  ; Shao Liu  ; Jianwei Li  ; Zhaolong Zhang ; Hongwen He  [All Authors](#)

16

Cites in
Papers

1012

Full
Text Views



Abstract

Document Sections

I. Introduction

II. Problem Formulation

III. Analysis of
Heterogeneous-Topology
Loop Delays

IV. Control Synthesis

V. Results Analysis

Show Full Outline ▾

Authors

Figures

References

Citations

Keywords

Metrics

Abstract:

Domain-based electronic and electrical (E/E) architectures have been regarded as a possible upgrade to distributed E/E architectures currently used in electric vehicles. In a distributed E/E design, E/E components are directly connected to the automobile bus. Domain-based architectures split E/E components into distinct domains depending on their functions, which clearly benefits software upgrading and wire harness reduction. However, due to its heterogeneous topology with multiple network protocols, domain-based E/E architecture introduces complicated multilink and multinode delays into the control loop. The delays may degrade and even deteriorate the stability of adaptive cruise control (ACC) employing domain-based E/E architecture. To this end, this article proposes a heterogeneous-topology loop delay analysis by introducing a notion of poly-service loop delay. With a graphical pattern, the analytical process is presented in depth. The worst-case loop delay is calculated using an upper-boundary mathematic equation. Then, a hierarchical cyber-physical control method for ACC is designed. The upper level is intended to achieve desired acceleration based on vehicle and intervehicle motion states. And the lower level is intended to mitigate the negative impact of loop delays and provide reliable acceleration tracking. The results of cosimulation and hardware-in-loop experiment verify effectiveness of proposed approaches.

Published in: [IEEE Transactions on Industrial Electronics](#) (Volume: 70 , Issue: 1, January 2023)

Page(s): 866 - 877

DOI: [10.1109/TIE.2022.3148732](#)

Date of Publication: 21 February 2022 

Publisher: IEEE


ISSN Information:

Print ISSN: 0278-0046

Electronic ISSN: 1557-9948

The fundamental goal of MPC is to minimize cost function [48], [49], thus, a predictive optimization problem can be expressed as

$$\min J_M \quad (22)$$

49. Y. Jiang, J. Oravec, B. Houska and M. Kvasnica, "Parallel MPC for linear systems with input constraints", *IEEE Trans. Autom. Control*, vol. 66, no. 7, pp. 3401-3408, Jul. 2020. 

► Show in Context

View Article



Google Scholar 

Citovaná práca: ADC14 Galčíková, L. [60%] – Oravec, J. [40%]: Fixed complexity solution of partial explicit MPC. *Computers & Chemical Engineering*, zv. 157, str. 107606, 2022. [Citované: 5] IF: 3,8

Citujúca práca: Guo, L.X. - Nie, S.D. - Liu, H. - Han, L.J. - Ruan, S.M. - Liu, R. - Yang, N.K. - Xiang, C.L.: Bi-Level Optimization of Speed Trajectory and Power Management for Autonomous HEVs in Off-Road Scenarios. *IEEE Transactions on intelligent vehicles*, č. 2, zv. 9, str. 4191-4205, 2024.

Bi-Level Optimization of Speed Trajectory and Power Management for Autonomous HEVs in Off-Road Scenarios

Publisher: IEEE

Cite This

PDF

Lingxiong Guo  ; Shida Nie  ; Hui Liu  ; Lijin Han  ; Shumin Ruan  ; Rui Liu  ; Ningkang Yang  ; Changle Xiang  [All Authors](#)

1

Cites in
Paper

264

Full
Text Views



Abstract

Document Sections

I. Introduction

II. Modelling for Hierarchical Strategy

III. Optimal Control System Framework

IV. Simulation Validations

V. Conclusion

Authors

Figures

References

Citations

Keywords

Metrics

Abstract:

With the advent of intelligent and connected vehicle technology, the simultaneous optimization of speed trajectory and power management for autonomous hybrid electric vehicles (HEVs) has become a significant research hotspot to enhance energy efficiency further. Especially for off-road scenarios, restricted by complex road conditions, vehicle nonlinear dynamics, and different time scales between speed planning and power split optimization, finding these control variables in one optimal problem is quite challenging. Thus, this article proposes a bi-level optimal control strategy for autonomous HEV to reduce computational burden by separating simultaneous optimization into two subproblems. In the upper layer, a vehicle stability constraints system is designed with consideration of both external road characteristics and the vehicle's lateral stability to determine the longitudinal speed boundary to ensure the vehicle's driving safety in off-road scenarios. Then, speed planning is formulated as a MPC problem with time-varying constraints, and solved by roll optimization. In the bottom layer, an explicit MPC (EMPC) is developed to obtain explicit solutions to achieve real-time energy management. Finally, simulation results show that the proposed method decreases equivalent fuel by 4.06% compared with ECMS, significantly improving computational efficiency with a slight sacrifice on fuel economy, compared with MPC and DP.

Published in: *IEEE Transactions on Intelligent Vehicles* (Volume: 9 , Issue: 2, February 2024)

Page(s): 4191 - 4205

DOI: 10.1109/TIV.2023.3313123

Date of Publication: 08 September 2023 

Publisher: IEEE

ISSN Information:

Electronic ISSN: 2379-8904

Print ISSN: 2379-8858

The powertrain control model is highly nonlinear with many variables, which puts forward a high requirement for computational efficiency for EMS. Since the EMPC is made to cope with linear or hybrid systems, it is important to simply the nonlinear model to a linear one before its application [36]. Piecewise-affine

36. L. Galčíková and J. Oravec, "Fixed complexity solution of partial explicit MPC", *Comput. Chem. Eng.*, vol. 157, 2022.

[Show in Context](#)

[CrossRef](#) 

[Google Scholar](#) 

Citovaná práca: ADC10 Oravec, J. [34%] – Horváthová, M. [33%] – Bakošová, M. [33%]: Energy efficient convex-lifting-based robust control of a heat exchanger. Energy, č. 201, str. 117566, 2020. [Citované: 5] IF: 8,2

Citujúca práca: Stebel, K. - Fratzczak, M. - Grelewicz, P. - Czczot, J. - Klopot, T.: Adaptive predictive controller for energy-efficient batch heating process. Applied Thermal Engineering, ISSN 1359-4311, zv. 192, 116954, 2021.



Applied Thermal Engineering



Volume 192, 25 June 2021, 116954



Adaptive predictive controller for energy-efficient batch heating process

Krzysztof Stebel, Michal Fratzczak, Patryk Grelewicz, Jacek Czczot  , Tomasz Klopot

Show more 

 Add to Mendeley  Share  Cite

<https://doi.org/10.1016/j.applthermaleng.2021.116954> 

[Get rights and content](#) 

network in the presence of fouling phenomenon is proposed in [8]. In [9], the authors compare experimental results for three convex-lifting-based robust MPC controllers derived for the plate heat exchanger. The results clearly show that the control strategies they suggest yield significant energy savings that can only be achieved by applying different control strategies. There are also some examples of advanced (e.g. predictive) control of chemical exothermic batch reactors [10], [11], but these studies

- [9] J. Oravec, M. Horváthová, M. Bakošová
Energy efficient convex-lifting-based robust control of a heat exchanger
Energy, 201 (2020)

Citovaná práca: ADC13 Drgoňa, J. [20%] – Bastida, J. A. [10%] – Figueroa, I. C. [10%] – Blum, D. [10%] – Arendt, K. [10%] – Kim, D. [10%] – Ollé, E. P. [5%] – Oravec, J. [10%] – Wetter, M. [5%] – Vrabie, D. [5%] – Helsen, L. [5%]: All you need to know about model predictive control for buildings. *Annual Reviews in Control*, zv. 50, str. 190–232, 2020. [Citované: 355], IF: 10,5

Citujúca práca: Sen S. - Kumar, M: Distributed-MPC Type Optimal EMS for Renewables and EVs Based Grid-Connected Building Integrated Microgrid. *IEEE Transactions on Industry Applications*, č. 2, zv. 60, 2390-2408, 2024.

Distributed-MPC Type Optimal EMS for Renewables and EVs Based Grid-Connected Building Integrated Microgrid

Publisher: IEEE

Cite This

PDF



Sachidananda Sen  ; Maneesh Kumar  All Authors

288

Full

Text Views



Abstract	Abstract: The vision of smart cities are being realized gradually by converting each building into a smart building. These are cyber-physical systems meant to provide best comfort to its habitant at most economical and environmentally sustainable way. There can be multiple loads for heating, cooling, lighting purposes. Renewables such as rooftop solar photovoltaic and small-scale wind turbines are also integrated for generating electricity that reduces the grid-dependency. This forms a Building Integrated Microgrid (BIMG) system consisting of its own sources, storage, and loads. Handling intermittent generation, variable load conditions, and external grid dynamics are the three essential factors to increase buildings' smartness. In this work, an adaptive Distributed Model Predictive Control (DMPC) principle for energy management of renewables based grid-tied BIMG is presented. All possible arbitrages, viz., generation intermittency of renewables, time-variable load profiles, storage unit's charging and discharging cycles, interruption of non-critical loads, and dynamic pricing of grid energy are considered. The feasibility of the given strategy is validated by realizing extensive test cases over 24 hrs duration using variable profiles of solar irradiance, continuous dynamic load variation at small and large scale, interruption of non-critical loads, two-tier tariff, and battery longevity by accounting for smooth charging/discharging cycles on the MATLAB software and Real-Time Digital Simulator (RTDS) based Hardware-in-Loop (HIL) setup.	
Document Sections		
I. Introduction		
II. Overview and Modeling of BIMG System		
III. Problem Formulation and Adaptive DMPC Strategy		
IV. Simulation Results and Discussion		
V. Comparative Studies With an Existing DMPC		
Show Full Outline ▾	Published in: <i>IEEE Transactions on Industry Applications</i> (Volume: 60 , Issue: 2, March-April 2024)	
Authors	Page(s): 2390 - 2408	DOI: 10.1109/TIA.2023.3332055
Figures	Date of Publication: 13 November 2023 	Publisher: IEEE
References	 ISSN Information:	

carbon footprint in the environment. The goal of any smart building is to achieve the Nearly Zero Emissions (nZE) by increasing usage of RESs and penalizing consumption fossil fuel based generation. Reduced energy consumption and minimizing electricity bill by demand response under dynamic pricing are also important [36]. Taking all these factors into account, the literature lacks any such

36. J. Drgona et al., "All you need to know about model predictive control for buildings", *Ann. Rev. Control*, vol. 48, pp. 118-151, Jun. 2018.

► [Show in Context](#) [CrossRef](#)  [Google Scholar](#) 