Heat exchanger control using model predictive control with constraint removal

Raphael Dyrska[◇a], 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*

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

[⋆] Support by the Alexander von Humboldt Foundation research group linkage cooperation program is gratefully acknowledged. This paper is funded by the European Union's Horizon Europe under 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) under grant MO 1086/15-1.

[◇] These authors contributed equally to this work.

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.

Keywords: Heat Exchanger, Model Predictive Control, Constraint Removal, Energy Consumption, 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. 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, 21, 22, 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 Sec. 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

$$x(k+1) = Ax(k) + Bu(k), \tag{1a}$$

$$y(k) = Cx(k), \tag{1b}$$

$k \geq 0$, 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

$$x_{i,\min} \leq x_i(k) \leq x_{i,\max}, \tag{2a}$$

$$u_{j,\min} \leq u_j(k) \leq u_{j,\max}, \tag{2b}$$

for all k and with $i = 1, ..., n,\ j = 1, ..., m$. The objective of the MPC design is to regulate the system state (1a) to the origin by periodically solving the optimization problem

$$
\begin{aligned}
&\min_{X,U}\; x(N)^\top P x(N) + \sum_{k=0}^{N-1} x(k)^\top Q x(k) + u(k)^\top R u(k) && \text{(3a)}\\
&\text{s.t. } x(0) = x_0, && \text{(3b)}\\
&\qquad x(k+1) = Ax(k) + Bu(k),\; k = 0, ..., N-1, && \text{(3c)}\\
&\qquad x(k) \in \mathcal{X},\; k = 0, ..., N-1, && \text{(3d)}\\
&\qquad u(k) \in \mathcal{U},\; k = 0, ..., N-1, && \text{(3e)}\\
&\qquad x(N) \in \mathcal{T}, && \text{(3f)}
\end{aligned}
$$

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 $X = (x(1)^\top, ..., x(N)^\top)^\top$ and $U = (u(0)^\top, ..., u(N-1)^\top)^\top$, 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

$$
\begin{aligned}
&\min_{U}\; V(x(0), U)\\
&\text{s.t. } GU \leq w + Ex(0),
\end{aligned}
\tag{4}
$$

where $V(x(0), U) = \frac{1}{2}x(0)^\top Y x(0) + x(0)^\top F U + \frac{1}{2}U^\top H U$, $Y \in \mathbb{R}^{n\times n}$, $F \in \mathbb{R}^{n\times mN}$, $H \in \mathbb{R}^{mN\times mN}$, $H \succ 0$, 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 $H \succ 0$, the solution to (4) is unique, if it exists.

After solving (4) for an optimal input trajectory $U^\star$, we apply the first element, i.e., $u^\star(0)$ 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 equations (1a-b) serves as the prediction model used for the MPC design and the optimization problem described by equations (3a-f) does not implement a robust variant of MPC, uncertainties arising in the real plant are not considered in equations (1a-b) explicitly.

Notation

Let $\mathcal{F}$ refer to the set of states $x \in \mathcal{X}$ for which (4) has a solution, and let $U^\star(x(0))$, for any $x(0) \in \mathcal{F}$, refer to the sequence that optimizes (4). We often write $U^\star$, $U^\star(x)$, $V(x, U^\star)$, $V(x, U^\star(x))$, etc., as short for $U^\star(x(0))$, $V(x(0), U^\star(x(0)))$, 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 $G_i U^\star - w_i - E_i x(0) = 0$, and inactive if $G_i U^\star - w_i - E_i x(0) < 0$. 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 $U^\star(x)$ 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 $\widetilde{\mathcal{I}} \subset \mathcal{I}$ be an arbitrary subset of the inactive constraints. Consider the reduced optimization problem*

$$\begin{aligned} &\min_{\hat{U}} V(x_0, \hat{U}) \\ &s.t.\ G_{\mathcal{Q}\setminus\widetilde{\mathcal{I}}} \hat{U} \leqslant w_{\mathcal{Q}\setminus\widetilde{\mathcal{I}}} + E_{\mathcal{Q}\setminus\widetilde{\mathcal{I}}} x_0. \end{aligned} \tag{5}$$

Then (5) *has a unique solution, which we denote by* $\hat{U}^\star$*. This solution is equal to the solution obtained from* (4)*, i.e.,* $\hat{U}^\star = U^\star$ *and* $V(x_0, \hat{U}^\star) = V(x_0, U^\star)$*.*

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*

$$\begin{aligned} &\min_{x,U} \; V(x,U) \\ &\text{s.t. } G_i U - w_i - E_i x = 0, \\ &\qquad G_{\mathcal{Q}\setminus i} U - w_{\mathcal{Q}\setminus i} - E_{\mathcal{Q}\setminus i} x \leqslant 0 \end{aligned} \tag{6}$$

has a solution, set σ_i *to the minimum that results for* (6)*, i.e.,* $\sigma_i := V(x^\star, U^\star)$*. 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 $V^\star$, 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 $V(x, U^\star(x)) < \sigma_i$, then constraint i is inactive at the optimal solution of* (4) *for $x(0) = x$. Furthermore, $V(x(k_0), U^\star(x(k_0))) < \sigma_i$ 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 Figure 1 to explain the steps of Algorithm 1. Let the optimal cost function value $V(x(k), U^\star(x(k)))$ be denoted by $V^\star$ for short, and let $\mathcal{R}$ 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 Alg. 1). Therefore, this constraint is removed from the QP (4) (step 7 in Alg. 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 $\sigma_i, i \in \mathcal{Q} = \{1, ..., q\}$ results from the order of the constraints in (4). This order can be chosen arbitrarily but has to be fixed.

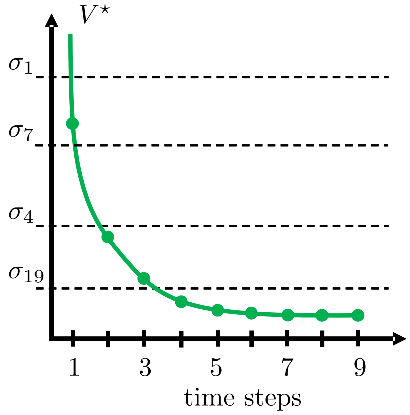


Figure 1: Optimal cost function trajectory $V^\star$ over discrete time steps for a hypothetical example.

Algorithm 1 MPC with constraint removal (see [23]).

```
**Input:** $V^\star$, $x^+$, $\mathcal{R} = \mathcal{Q} \setminus \widetilde{\mathcal{I}}$, $\sigma_i, i \in \mathcal{R}$.
**for all** $i \in \mathcal{R}$ **do**
    **if** $V^\star < \sigma_i$ **then**
        $i$th constraint will remain inactive: $\widetilde{\mathcal{I}} \leftarrow \widetilde{\mathcal{I}} \cup \{i\}$.
    **end if**
**end for**
Remove inactive constraints: $\mathcal{R} \leftarrow \mathcal{R} \setminus \widetilde{\mathcal{I}}$
Solve reduced optimization problem in (4) for $x(0) = x^+$ and reduced set of constraints $\mathcal{R}$.
**Output:** Updated $U^\star$, $V^\star$, $x^+$, $\mathcal{R}$.
```

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 Figure 2.

The three-stage counter-current liquid-liquid plate heat exchanger (Figure 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.

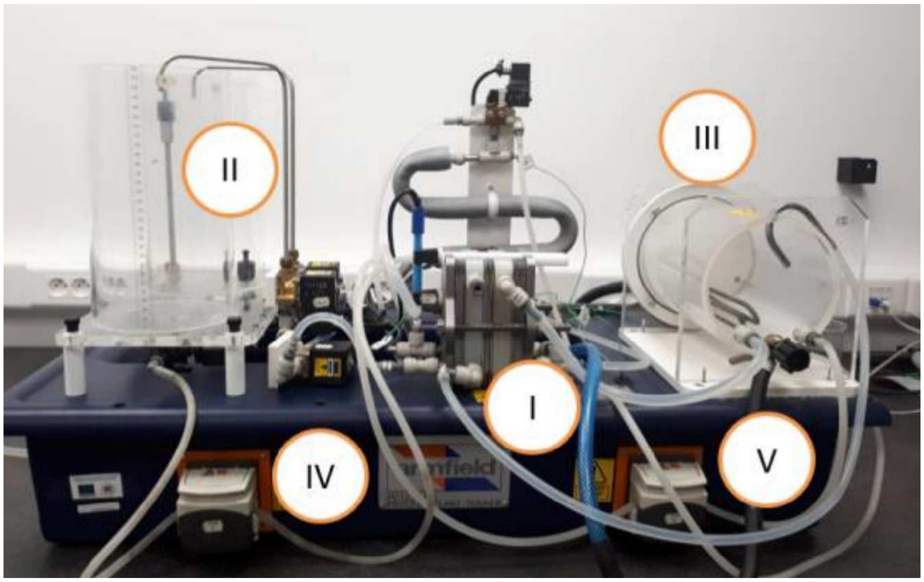


Figure 2: The controlled plant – plate heat exchanger (I), tanks for cold medium (II), tank for hot medium with heating system (III), peristaltic pump used for cold medium (IV), peristaltic pump used for hot medium (V).

The 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 (Figure 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 $T_{\mathrm{hot}} = 70^\circ$ C using a heating coil inside a retention tank (Figure 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 (Figure 2, devices (IV) and (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_{s}. The control output is the outlet temperature of the cold medium T. The value of the temperature mainly depends on the volumetric flow rate

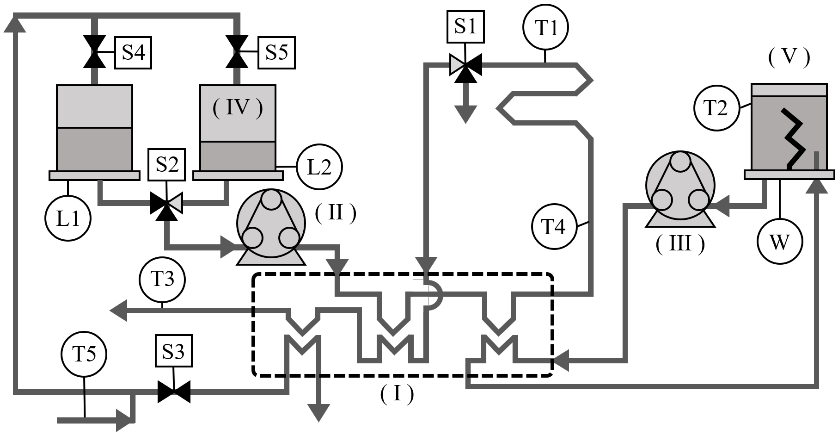


Figure 3: The detailed scheme of the controlled heat exchanger [27].

of the hot medium $\dot{V}$, as the temperature of the hot medium is constant with $T_{\text{hot}} = 70^{\circ}$ 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 (Figure 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 $T_0 = 20^{\circ}$ C. The control output is measured with a K-type thermocouple with operation range $0 - 150^{\circ}$C. The detailed scheme of the controlled heat exchanger is presented in Figure 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]. Figure 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].

For the MPC design, the input, state, and output variables of the system in (1) were defined in the form of deviation variables

$$u(k) = \dot{V}(k) - \dot{V}_{\mathrm{s}}, \tag{7a}$$

$$x(k) = T(k) - T_{\mathrm{s}}, \tag{7b}$$

$$y(k) = T(k) - T_{\mathrm{s}}, \tag{7c}$$

where T_{s} is the operating point of controlled temperature and $\dot{V}_{\mathrm{s}}$ 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

$$\hat{x}(k) = \begin{bmatrix} x(k) \\ \sum_{j=0}^{k} e(j) \end{bmatrix}, \tag{8}$$

with $e(j) = T_{\mathrm{s}} - T(j) = -x(j)$ the control error, and the negative sign resulting from the laboratory setup.

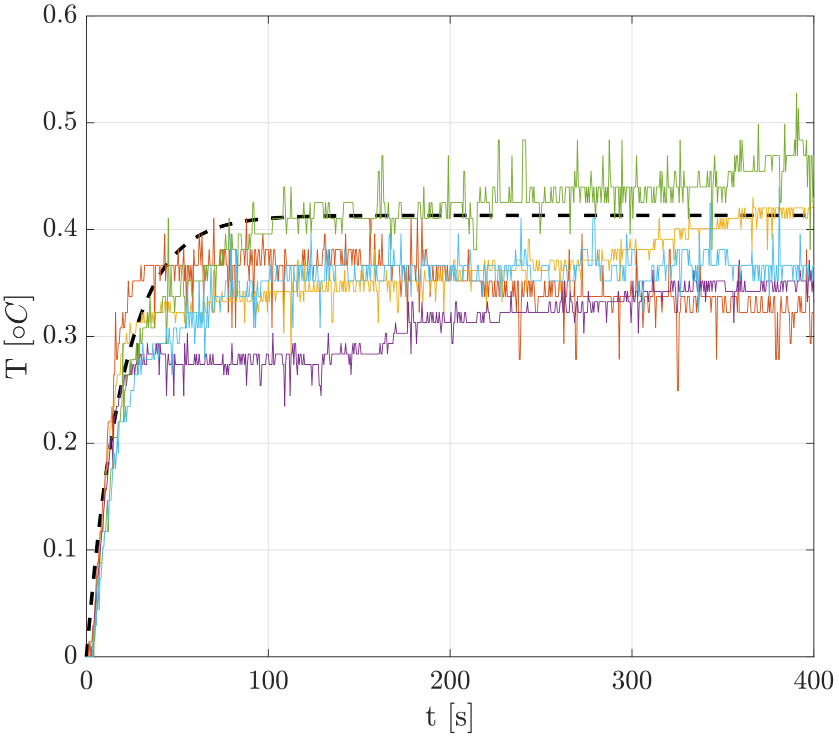


Figure 4: The set of measured and normalized step responses serves for the identification of the parameters of the mathematical model. The step response of the nominal mathematical model is depicted in the black dashed line.

The matrices of the state-space model considering the augmented vector of states from (8) are defined as

$$\hat{A} = \begin{bmatrix} A & 0 \\ -t_s C & I \end{bmatrix}, \hat{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}, \hat{C} = \begin{bmatrix} C & 0 \end{bmatrix}, \tag{9}$$

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)

$$\hat{x}(k+1) = \hat{A}\hat{x}(k) + \hat{B}u(k), \tag{10a}$$

$$y(k) = \hat{C}\hat{x}(k). \tag{10b}$$

State, input, and output matrices read

$$\hat{A} = \begin{bmatrix} 0.8736 & 0 \\ -2 & 1 \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} 0.5562 \\ 0 \end{bmatrix}, \quad \hat{C} = \begin{bmatrix} 1 & 0 \end{bmatrix}, \tag{11}$$

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

$$P = \begin{bmatrix} 0.1479 & -0.0217 \\ -0.0217 & 0.0053 \end{bmatrix}, \quad Q = \begin{bmatrix} 0.001 & 0 \\ 0 & 0.001 \end{bmatrix}, \quad R = 0.1. \tag{12}$$

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 $t_\mathrm{s} = 2\,\mathrm{s}$. 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 $[0, 11.5]\,\mathrm{ml\,s^{-1}}$ representing the physical constraints. This range was normalized so that the operating point corresponded to $\dot{V}_{\mathrm{s}}$. Therefore, the constraints considered on the control input in the deviation form were chosen as

$$-5.75\,\mathrm{ml\,s^{-1}} \leq u(k) \leq 5.75\ \mathrm{ml\,s^{-1}}. \tag{13}$$

The outlet temperature T of the cold medium was constrained to the interval $[25, 65]\,^{\circ}\mathrm{C}$ (i.e. $[298.15, 338.15]$ 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

$$\begin{bmatrix} -20 \\ -100 \end{bmatrix} \leq \hat{x}(k) \leq \begin{bmatrix} 20 \\ 100 \end{bmatrix}. \tag{14}$$

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.

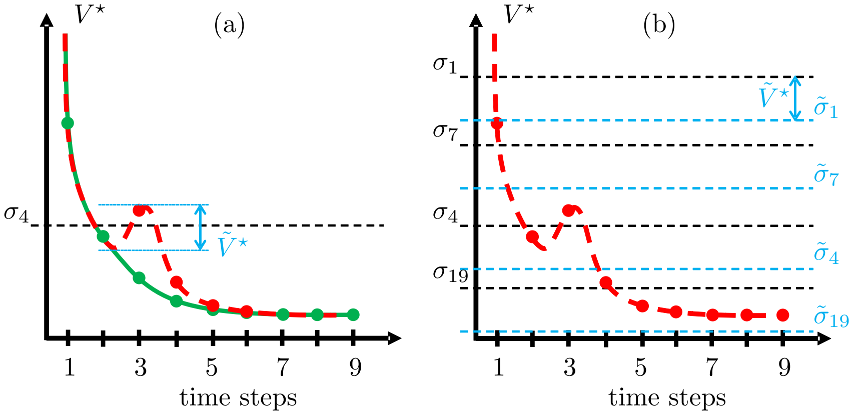


Figure 5: Modification of the applied constraint removal approach for the hypothetical example from Sec. 3. (a) MPC with increasing cost (red) and maximum increase $\tilde{V}^\star$. If $V^\star$ increases due to the integrator state after a setpoint change, constraints previously detected to be inactive can become active again. (b) Modification of the hypothetical bounds to more conservative values (blue) based on the maximum increase in $V^\star$.

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 $V^\star$ 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 $x_2(k)$ 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 $V^\star$, in Figure 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 Figure 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 Figure 5, red curve, i.e., the integrator state will lead to a temporary increase of the optimal cost function value $V^\star$ for some time steps k. For bound σ_4 corresponding to a hypothetical constraint, e.g., the value of $V^\star$ 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 $V^\star$, 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 $V^\star$ caused by the setpoint changes ($\widetilde{V}^\star$ in Figure 5 (a)). Then we determine sufficiently conservative bounds by reducing their value by the maximum increase in the optimal cost function value $V^\star$, i.e., the value $\widetilde{V}^\star$. This is illustrated in Figure 5 (b), where the reduced bounds $\widetilde{\sigma}_i$ 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 $\widetilde{\sigma}_4$ takes into account the level of the maximum increase in the original cost function $V^\star$, and thus constraint $i = 4$ is not removed too early. This prevents removing a constraint that could become active 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., $\widetilde{\sigma}_{19}$ in Figure 5 (b). However, constraint removal is efficient also for a subset $\Sigma \subseteq \mathcal{Q}$ 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 $V^\star$ 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 Figure 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

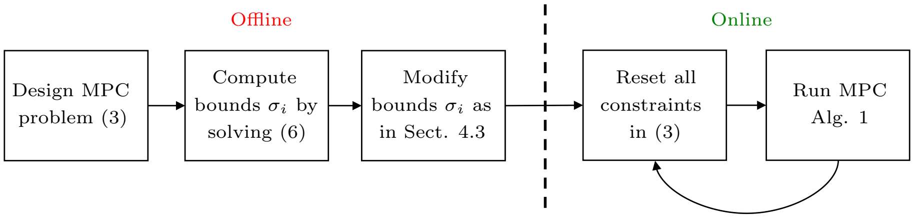
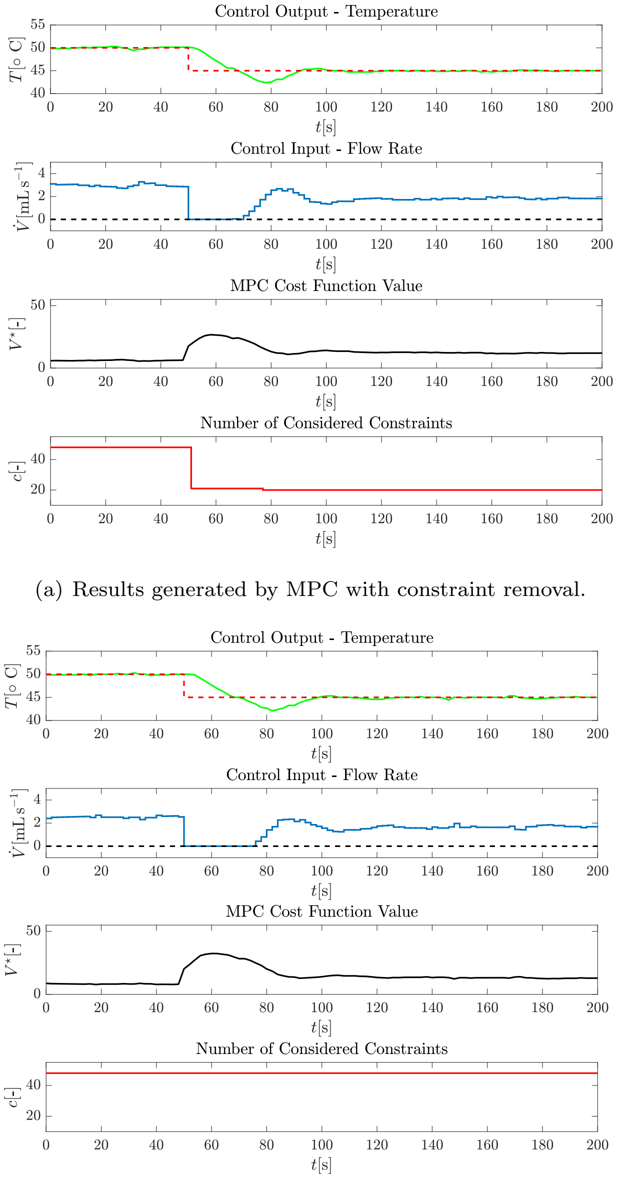

Offline
Online
Design MPC problem (3)
Compute bounds σ_i by solving (6)
Modify bounds σ_i as in Sect. 4.3
Reset all constraints in (3)
Run MPC Alg. 1

Figure 6: Workflow of operating MPC with constraint removal on a real plant.

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.

5.1. Experimental validation using the plate heat exchanger

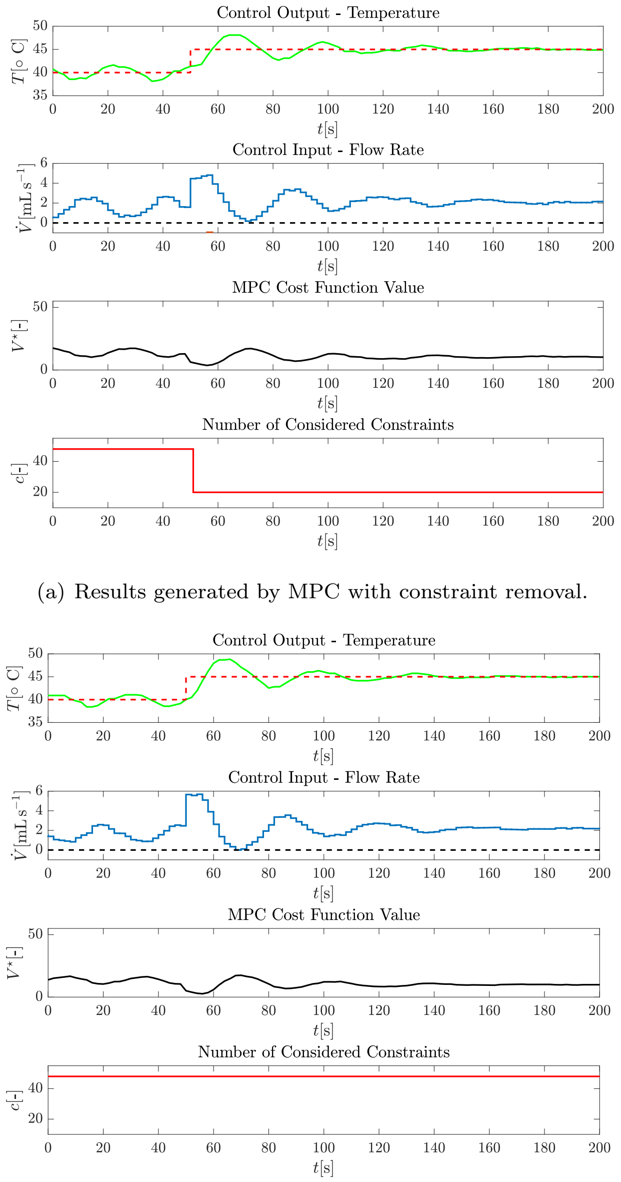
Multiple setpoint step changes were performed and analyzed to demonstrate the efficiency of the proposed acceleration by the constraint removal approach. The closed-loop control trajectories, cost function value $V^\star$, and number of considered constraints c for MPC with constraint removal are depicted in Figure 7(a) and Figure 8(a) for two representative setpoint changes, respectively.



(a) Results generated by MPC with constraint removal.

(b) Results generated by MPC without constraint removal.

Figure 7: Experimental results for MPC with (see (a)) and without (see (b)) constraint removal and setpoint step change 50°C → 45°C (323.15K → 318.15K): 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.

Figure 8: Experimental results for MPC with (see (a)) and without (see (b)) constraint removal and setpoint step change 40°C → 45°C (313.15K → 318.15K): 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).

The constraint removal strategy (and thus Alg. 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 Figure 7(b)–Figure 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 Figure 7(a)–Figure 8(a) can slightly differ from the performance depicted in Figure 7(b)–Figure 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 $\widetilde{V}^\star$ were evaluated based on the results shown in Figure 7(b)– Figure 8(b). By analyzing these sequences, we determined the maximum increase $\widetilde{V}^\star$ to the value $\widetilde{V}^\star = 19.06$. The corresponding value of each modified bound σ_i, $i \in \{1, \dots, q\}$ is depicted in Figure 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 Figure 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 Figure 7(a)–Figure 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

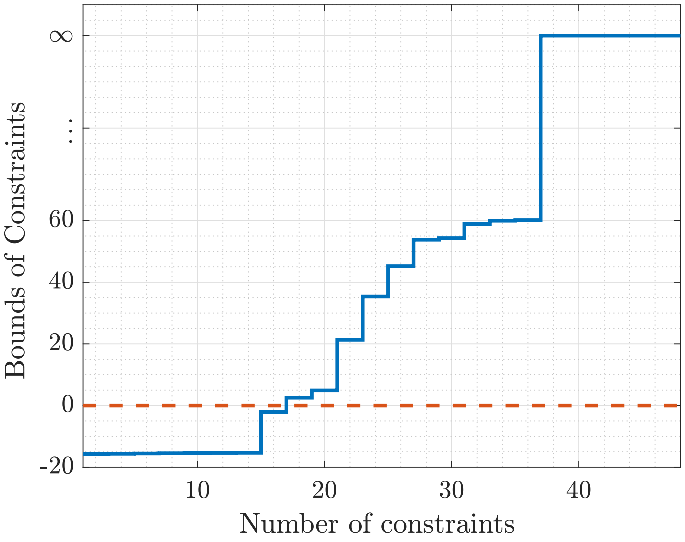


Figure 9: Values of the modified bounds evaluated for the 48 constraints of the MPC problem, sorted for increasing values. Note that the enumeration here does not reflect the order of the constraints.

Figure 7(a) and Figure 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 Figures 7 − 8 is not decreasing to zero, but stays slightly above. Measurement noise and especially the value of the integrator state 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 $\widetilde{\sigma}_i$, between around 58% and 62.5% of all constraints were removed from the original OCP without affecting the control performance negatively.

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.

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 Figure 8(b), and (ii) control using MPC with constraint removal, depicted in Figure 8(a). Both data sets refer to the step change from 40°C (313.15 K) to 45°C (318.15 K). Figure 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 Figure 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 $V^\star$ 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 sampling time, the microcontroller compared the bounds σ_i for each considered constraint with the current cost function value $V^\star$.

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

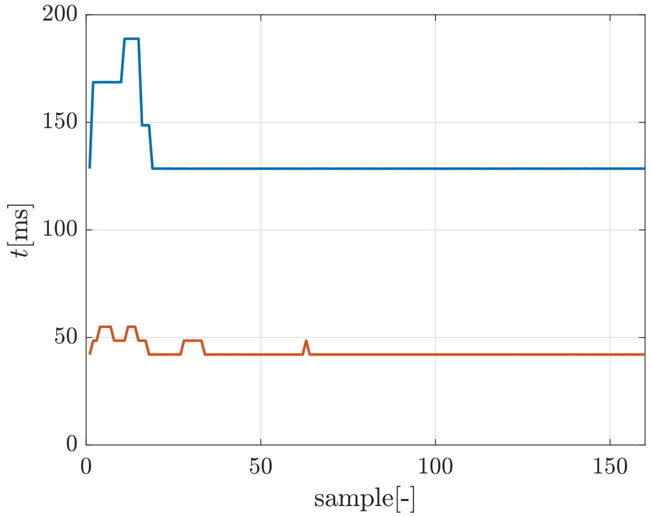


Figure 10: Time necessary for solving a QP at each sampling instance using the modified constraint removal method (orange) and conventional MPC (blue).

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

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.

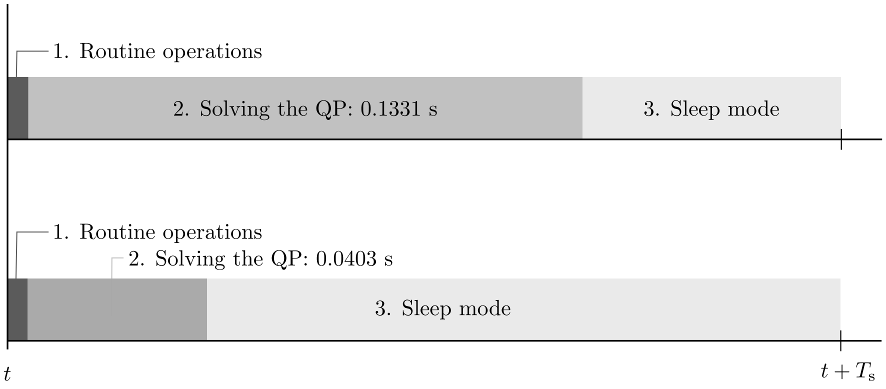


Figure 11: The timeline of the operations within one sampling instance.

Analogously to the case study presented in Section 5.1, we also considered a sampling period of 2 s here. Figure 11 illustrates the individual stages of the operation within one sampling instance.

The measured results shown in Figure 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 $I_{\mathrm{comp}} = 61.3$ mA when a QP is solved. On the other hand, during the sleep mode, the value is $I_{\mathrm{sleep}} = 3.92$ 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

$$E_{\mathrm{t_s}} = (I_{\mathrm{comp}} - I_{\mathrm{sleep}})\, U_{\mathrm{s}}\, t_{\mathrm{avg}}, \tag{15}$$

where $E_{\mathrm{t_s}}$ is the energy consumed within one sampling period, and U_{s} is the supply voltage of the control unit set to $U_{\mathrm{s}} = 5\,\mathrm{V}$. 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.

Table 2: Comparison of the energy consumption for conventional MPC and MPC with constraint removal within one sampling period.

MPC method	$E_{\mathrm{t_s}}$ [$\times 10^{-3}$ J]
Conventional MPC	38
MPC with constraint removal	12

The energy consumption saved per hour of operation is then computed as

$$E_{\mathrm{h}} = \frac{3\,600}{t_{\mathrm{s}}}\,\Delta E_{\mathrm{t_s}} = 41.222\,\mathrm{J}\,, \tag{16}$$

where $\Delta E_{\mathrm{t_s}}$ is the difference between energy consumption $E_{\mathrm{t_s}}$ 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, $E_{\mathrm{y}} = 361.1\,\mathrm{kJ}$. 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.

Nomenclature

Symbols

A, B, C	state-space system
$\hat{A}, \hat{B}, \hat{C}$	augmented state-space system
$\mathcal{A}, \mathcal{I}, \tilde{\mathcal{I}}$	active set, inactive set, subset
e	control error
$E_\mathrm{h}, E_\mathrm{y}$	energy consumption per hour, year
$E_{\mathrm{t_s}}, \Delta E_{\mathrm{t_s}}$	energy consumed within one sampling period, difference
$\mathcal{F}, \mathcal{T}, \mathcal{R}$	feasible set, terminal set, set of remaining constraints
G, w, E	constraint matrices of the quadratic program
$I_\mathrm{comp}, I_\mathrm{sleep}$	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
$\mathcal{Q}, \Sigma$	constraint set, subset
$T, T_\mathrm{hot}, T_\mathrm{s}$	temperature of the cold medium, hot medium, setpoint
$t_\mathrm{sol}, t_\mathrm{avg}$	total, average solver time
u, y	system input, output
$u_\mathrm{min}, u_\mathrm{max}$	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
$V, \widetilde{V}^\star$	cost function, maximum increase in $V^\star$
$\dot{V}, \dot{V}_\mathrm{s}$	volumetric flow rate of the hot medium, operating point
$x, x^+, x_0, \hat{x}$	system state, subsequent state, initial state, augmented system state
$x_\mathrm{min}, x_\mathrm{max}$	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
OCP	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, Applied Thermal Engineering 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, Applied Thermal Engineering 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, Industrial & Engineering Chemistry Research 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, Industrial & Engineering Chemistry Research 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, Chemical Engineering and Processing: Process Intensification 45 (2006) 661–671.

[10] A. Vasičkaninová, M. Bakošová, Control of a heat exchanger using neural network predictive controller combined with auxiliary fuzzy controller, Applied Thermal Engineering 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, Journal of 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 Transactions on Automatic Control 59 (1) (2014) 18–33.

[14] Y. Wang, S. Boyd, Fast model predictive control using online optimization, IEEE Transactions on Control Systems Technology 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 Journal of Control 5 (2) (2003) 271–286.

[17] M. Kvasnica, M. Fikar, Clipping-based complexity reduction in explicit MPC, IEEE Transactions on Automatic Control 57 (7) (2012) 1878–1883.

[18] C. N. Jones, M. Morari, Polytopic approximation of explicit model predictive controllers, IEEE Transactions on Automatic Control 55 (11) (2010) 2542–2553.

[19] M. Kvasnica, P. Bakaráč, M. Klaučo, Complexity reduction in explicit mpc: A reachability approach, Systems & Control Letters 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, European Journal of Control 35 (2017) 42–49.

[24] R. Dyrska, 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] Armfiled, 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. doi: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. Čirka, 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, Optimization and Engineering 13 (1) (2012) 1–27.