Constraint Removal for MPC with Performance Preservation and a Hyperthermia Cancer Treatment Case Study

S.A.N. Nouwens¹, B. de Jager¹, M.M. Paulides²,³, W.P.M.H. Heemels¹

Abstract—Model predictive control (MPC) is an optimization-based control strategy with broad industrial adoption. Unfortunately, the required computation time to solve the receding-horizon MPC optimization problem can become prohibitively large for many applications with a large number of state constraints. This large number of state constraints can, for instance, originate from spatially discretizing a partial differential equation of which the solution has to satisfy constraints over the full spatial domain. This is particularly the case in MPC for RF-based hyperthermia cancer treatments, which forms a strong motivation for this study. To address this problem, we propose a novel constraint-adaptive MPC framework for linear discrete-time systems. In this framework, we select at each time-step a subset of the state constraints that are included in the optimization problem, thereby reducing the online computational burden. Critically, our framework guarantees the same closed-loop performance, recursive feasibility, and constraint satisfaction properties as the original (non-reduced) MPC scheme. We achieve this result by efficiently exploiting reachable set computations and the MPC cost function. We will demonstrate our novel method using a hyperthermia cancer treatment case study showing a two-orders of magnitude improvement in computation time, with identical closed-loop performance as the original (non-reduced) MPC scheme.

I. INTRODUCTION

Model predictive control (MPC) is a successful, optimization-based, control technology [1]. Generally, the control inputs are obtained by recursively optimizing a finite-horizon cost function subject to input and state constraints and system dynamics. The explicit inclusion of state and inputs constraints and the capability to handle multi-input multi-output systems are distinct advantages of MPC. However, solving an optimization problem at each time-step can prohibit the real-time feasibility of the controller for computationally complex scenarios. Particularly, in applications requiring the control of systems modeled by partial differential equations with constraints on the continuous spatial domain, such as the heat equation for hyperthermia cancer treatments, the real-time application of MPC can be problematic.

Efforts to improve the computational aspects of MPC are commonplace with among others, explicit MPC, model reduction, and tailored numerical solvers, see, e.g., [2]–[6], and the references therein. However, our focus in this paper is on systems with large-scale constraints. For such systems, the aforementioned techniques are typically not highly effective. In particular, interior-point methods are reported to scale linearly with the number of inequality constraints [5]. As a result, systems with many constraints motivate so-called constraint removal techniques, which are roughly separated into offline [7], and online methods [8]–[10]. Offline methods can be prohibitively complex to compute and possibly do not enable real-time MPC, as constraints can only be removed if they are redundant with respect to all feasible states. In contrast, online techniques can typically remove more constraints as they exploit the knowledge of the current state of the system. However, online methods use a so-called pre-solve step whose additional computation time cannot be ignored and should be limited. Although the existing online constraint removal techniques are effective in various scenarios, they still require extensive computational effort for large-scale problems, or do not provide performance guarantees.

In this paper, we propose a novel online constraint removal method for MPC of linear discrete-time systems that is computationally tractable and guarantees the same closed-loop performance, recursive feasibility, and constraint satisfaction as the original MPC scheme. To this end, we exploit properties inherent to the system by using fast reachability analysis and the optimal control sequence from the previous time step to systematically remove constraints from the optimization problem. We will show the strengths of our new framework using a hyperthermia cancer treatment case study [11]–[14], where we obtain a two-orders of magnitude reduction in computational time. Note that existing work on MPC for hyperthermia is largely focused on ultrasound heating and localized 2D models, which limits their application to small tumors [11], [14]. In contrast, radio frequency based (RF) hyperthermia, used for larger tumors, requires large-scale 3D models due to the non-localized heating from the actuators [12]. The non-local heating also introduces the need for temperature constraints on the whole spatial domain to ensure patient safety. For these reasons, MPC for RF-based hyperthermia is computationally challenging.

The remainder of this paper is structured as follows. We start by introducing the preliminaries in Section II. In Section III, we provide our ca-MPC scheme that exploits reachable sets and the cost function to remove constraints without changing the optimizer of the MPC problem, after which, we provide computationally cheap methods to remove state constraints in Section IV. The hyperthermia cancer
treatment case study is presented in Section V. Finally, conclusions are stated in Section VI.

II. PRELIMINARIES

In this section, we introduce the linear system to be controlled, a typical MPC setup, the forward and backward reachable sets and the cost function level sets. These sets form the basis of our constraint removal framework.

A. System

In this paper, we consider plants that are described by the discrete-time linear time-invariant (LTI) system

$$x_{k+1} = Ax_k + Bu_k.$$  \hfill (1)

Here, $x_k \in \mathbb{R}^n$ and $u_k \in \mathbb{R}^m$ denote the plant states and the inputs, respectively, at discrete time $k \in \mathbb{N}$. Furthermore, $A$ and $B$ are real matrices of appropriate dimensions. The system (1) is subject to polyhedral state and input constraints that are given by

$$x_k \in \{x \in \mathbb{R}^n \mid Cx \leq b\} := X, \quad \text{for some } C \in \mathbb{R}^{n \times n}, b \in \mathbb{R}^n.$$  \hfill (2a)
$$u_k \in \{u \in \mathbb{R}^m \mid Cu \leq b_u\} := U, \quad \text{for some } C \in \mathbb{R}^{m \times n}, b_u \in \mathbb{R}^m.$$  \hfill (2b)

Here, $X$ and $U$ are assumed to be non-empty polyhedral sets with $C \in \mathbb{R}^{n \times n}$ and $b \in \mathbb{R}^n$. Note that the definitions for the state and input constraints, i.e., $n_x > 1$.

B. MPC setup

Based on the system dynamics (1) and constraints (2), a common MPC setup, given state $x_k$ at time $k \in \mathbb{N}$, can be formulated as

$$\begin{align*}
\text{minimize} & \quad J(x_k, U_k), \\
\text{subject to} & \quad X_k = \Phi x_k + \Gamma U_k, \\
& \quad X_k \in \underbrace{X \times \cdots \times X}_{N \text{ times}} X_T := X_N, \\
& \quad U_k \in \underbrace{U \times \cdots \times U}_{N \text{ times}} := U_N.
\end{align*}$$  \hfill (3a)

where

$$J(x_k, U_k) := \ell_T(x_{N|k}) + \sum_{i=0}^{N-1} \ell(x_{i|k}, u_{i|k}),$$  \hfill (4a)
$$X_k := [x_{1|k}^T \cdots x_{N|k}^T]^T, \quad X_T := [x_{N|k}^T]^T,$$  \hfill (4b)
$$U_k := [u_{0|k}^T \cdots u_{N-1|k}^T]^T,$$  \hfill (4c)
$$\Phi := \begin{bmatrix} A & A^2 \\ \vdots & \vdots \\ A^N \end{bmatrix}, \quad \Gamma := \begin{bmatrix} B & 0 & \cdots & 0 \\ AB & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix}.$$  \hfill (4d)

Here, $\ell(\cdot)$, $\ell_T(\cdot)$, $x_{i|k}$, $u_{i|k}$, and $X_T$ denote the stage cost, the terminal cost, the predicted state, the predicted input, and the polyhedral terminal set, respectively. The terminal set satisfies $X_T := \{x \in \mathbb{R}^n \mid C_T x \leq b_T\} \subseteq X$, where $C_T \in \mathbb{R}^{n_T \times n}$ and $b_T \in \mathbb{R}^{n_T}$, which is positively invariant with respect to $U$ for some auxiliary control law $u = Kx$, i.e., $(A + BK)X_T \subseteq X_T$ and $KX_T \subseteq U$. The $i/k$ subscript is used to denote the $i$-th prediction at time $k$. For example, $x_{i|k}$ denotes the prediction of $x_{k+i}$ made at time $k$ for $i = 1, 2, \ldots, N$, where $N$ denotes the prediction horizon. Additionally, the stage and terminal cost quantify the performance in terms of the predicted states $x_{i|k}$ and inputs $u_{i-1|k}$ for $i = 1, 2, \ldots, N$.

For the optimization problem (3), we denote the set of feasible inputs parameterized by the state $x_k$ by

$$U_{f,N}(x_k) := \{U_k \in U_N \mid (3b) - (3d)\}.$$  \hfill (5)

Moreover, we denote the set of feasible states by

$$X_f := \{x \in X \mid U_{f,N}(x) \neq \emptyset\}.$$  \hfill (6)

Finally, note that under suitable assumptions on $\ell(\cdot)$ and $\ell_T(\cdot)$, for all $x_k \in X_f$ a minimizer of (3) exists and we denote by $U_k^* := [u^*_{0|k} \cdots u^*_{N-1|k}]^T$ a particular one at time $k \in \mathbb{N}$, i.e.,

$$U_k^* \in \arg \min_{U_k \in U_{f,N}(x_k)} J(x_k, U_k).$$  \hfill (7)

The optimal predicted state sequence corresponding to $U_k^*$ is denoted by $X_k^* := [x_{1|k}^T \cdots x_{N|k}^T]^T = \Phi x_k + \Gamma U_k^*$.

Using a receding horizon implementation, the MPC problem (3) is turned into a feedback law $K_{MPC} : X_f \rightarrow U$ by applying the first computed input in $U_k^*$ on the real plant (1), i.e., $u_k := K_{MPC}(x_k) := u_{0|k}^*$.

C. Forward and backward reachable sets

We define the forward reachable set $\overrightarrow{R}^k(x_0, U) := \{x^+ \in \mathbb{R}^n \mid x^+ = Ax + Bu, \text{ for some } x \in \overrightarrow{R}^k(x_0, U), \text{ and } u \in U\}, k \in \mathbb{N}$, where $\overrightarrow{R}^0(x_0, U) := x_0$. Similarly, the backward reachable set $\overleftarrow{R}^k(x_T, U)$ for a given terminal set at time $N \in \mathbb{N}$, $x_T \subseteq X$, is obtained through a backwards recursion $\overleftarrow{R}^{k-1}(x_T, U) := \{x \in \mathbb{R}^n \mid Ax + Bu \in \overrightarrow{R}^k(x_T, U)\}$ for some $u \in U$, $k \in \{1, 2, \ldots, N\}$, where $\overrightarrow{R}^N(x_T, U) := x_T$. Note that the definitions for the forward and backward reachable sets do not include the state constraints, but do include the input constraints.

Last, for ease of notation, we define forward and backward reachable tubes as

$$\overrightarrow{R}(x_0, U) := \overrightarrow{R}^1(x_0, U) \times \overrightarrow{R}^2(x_0, U) \times \cdots \times \overrightarrow{R}^N(x_0, U),$$  \hfill (10a)
$$\overrightleftarrows(x_T, U) := \overrightarrow{R}^1(x_T, U) \times \overrightarrow{R}^2(x_T, U) \times \cdots \times \overrightarrow{R}^{N-1}(x_T, U) \times \mathbb{R}^n.$$  \hfill (10b)

Note that, in (10b) we replace $\overrightarrow{R}^N(x_T, U)$ with $\mathbb{R}^n$ to ensure we do not remove terminal constraints with $x_T$ itself.

D. Cost function level set

Besides the forward and backward reachable sets, we will also exploit the cost function level set to remove constraints from (3c). The key observation is that for state $x_k \in X_f$ and any given input sequence $U_k \in U_{f,N}(x_k)$, an upper bound for the MPC cost function can be easily obtained as $J(x_k, U_k^*) \leq J(x_k, U)$, More precisely, this upper bound defines a cost function level set

$$J_u(x_k, U_k) := \{U \in \mathbb{R}^m \mid J(x_k, U) \leq J(x_k, U_k^*)\}.$$  \hfill (11)
For the well-known case where $J(x, U)$ is a quadratic cost function such that
\[ \ell(x, u) = x^T Q x + u^T R u, \quad \ell_T(x) = x^T P x, \]
with $Q$, $R$, and $P$ positive-definite matrices, the resulting level set (11) is an ellipse in $\mathbb{R}^{Nm}$ as a function of $x_k, \tilde{U}_k$.

Using (3b), it is also possible to define a cost function level set for state trajectories in $\mathbb{R}^{Nn}$.

\[ J_x(x_k, \tilde{U}_k) := \{ \Phi x_k + \Gamma U | U \in J_u(x_k, \tilde{U}_k) \}. \] (13)

### III. CONSTRAINT REMOVAL USING REACHABLE SETS AND COST FUNCTION LEVEL SETS

Generally, given a convex MPC problem, it is known that there exist exactly at most $Nm$ so-called support constraints whose removal changes the minimizer \([15].\) However, removing all but these state-dependent support constraints is time-consuming. Nevertheless, it is possible to improve the computational complexity of (3) by removing only a subset of the non-support constraints. Clearly, this speeds up the MPC problem without impacting the closed-loop performance, recursive feasibility, and constraint satisfaction.

We start with the observation that we can remove state constraints that are either, not forward- or backward-reachable, or correspond to state trajectories that do not lower the MPC cost below $J(x, \tilde{U}_k)$. To start exploiting this observation, we define the set

\[ \mathcal{H}(x_k, \tilde{U}_k) := \tilde{R}(x_k, \mathcal{U}) \cap \tilde{R}(\mathcal{X}_T, \mathcal{U}) \cap J_x(x_k, \tilde{U}_k). \] (14)

**Definition 1:** A constraint $c_j x \leq b_j$ is called $(x_k, \mathcal{U}, \tilde{X}_T, \tilde{U}_k)$-redundant (or redundant for short) at time $i \in \mathbb{N}[1, N]$ for system (1), if

\[ [x_{1,i}^T \ldots x_{N,i}^T] \subset \mathcal{H}(x_k, \tilde{U}_k) \implies c_j x_{i,j} \leq b_j. \] (15)

Loosely speaking, the constraint $c_j x_{i,j} \leq b_j$, where $C := [c_1^T, \ldots, c_{Nm}^T]$ and $b := [b_1, \ldots, b_{Nm}]^T$, is redundant at time $i$, if system dynamics, input constraints, terminal set, and MPC cost function already guarantee its satisfaction. Clearly, if $c_j x_{i,j} \leq b_j$ is redundant at time $i$, it can be removed from the MPC problem without changing the minimizer.

**Definition 2:** A set $\mathcal{X}_N^{red} := \mathbb{R}_{+1}^n \times \cdots \times \mathbb{R}_{+1}^n$ is called a reduced constraint set for $\mathcal{X}_N$, if there are index sets $\mathcal{A}_i \subseteq \mathbb{N}[1, n_x]$, for $i = 1, 2, \ldots, N - 1$, such that

\[ \mathcal{X}_N^{red} = \{ x \in \mathbb{R}^n | c_j x \leq b_j, \text{ for } j \in \mathcal{A}_i \}. \] (16a)

For $i = N$, we define index sets $\mathcal{A}_N \subseteq \mathbb{N}[1, n_T]$ such that

\[ \mathcal{X}_N^{red} = \{ x \in \mathbb{R}^n | c_j x \leq b_j, \text{ for } j \in \mathcal{A}_N \}, \] (16b)

where $C_T := [c_{1,T}^T, \ldots, c_{Nm,T}^T]$ and $b_T := [b_{1,T}, \ldots, b_{Nm,T}]^T$.

From the above reasoning the next result follows.

**Proposition 1:** Consider a reduced constraint set $\mathcal{X}_N^{red}$ for $\mathcal{X}_N$. If for initial state $x_k \in \mathcal{X}_f$ and $\tilde{U}_k \in \mathcal{U}_f, \mathcal{U}_N(x_k)$ it holds that

\[ \mathcal{H}(x_k, \tilde{U}_k) \cap \mathcal{X}_N^{red} = \mathcal{H}(x_k, \tilde{U}_k) \cap \mathcal{X}_N^{red}, \] (17)

then the MPC problem, where (3c) is replaced by $X_k \in \mathcal{X}_N^{red}$, has the same minimizer as the original MPC problem, i.e.,

\[ \arg \min_{U_k \in \mathcal{U}_f, \mathcal{U}_N} J(x_k, U_k) = \arg \min_{U_k \in \mathcal{U}_f, \mathcal{U}_N} J(x_k, U_k). \] (18)

### IV. NUMERICAL EFFICIENT CONSTRAINT REDUCTION

In this section, we provide numerically efficient procedures to remove state constraints based on the observations in Section III. While Section III describes conditions that can be used to remove redundant constraints, we need efficient and fast techniques to carry out the corresponding calculations online. To this end, we first show that outer approximations of the reachable sets and the cost function level set can be exploited in our ca-MPC framework. Second, we provide numerically efficient methods for computing such outer approximations of reachable sets. Finally, we present a method to efficiently construct suitable reduced constraints sets $\mathcal{X}_N^{red}$ based on these computational methods.

#### A. Outer approximations in constraint removal

In this section, we show that using outer approximations of $\mathcal{H}(x_k, \tilde{U}_k)$ can be exploited in our framework.

**Corollary 1:** Consider an outer approximation of $\mathcal{H}(x_k, \tilde{U}_k)$ in the sense that $\mathcal{H}(x_k, \tilde{U}_k) \subseteq \mathcal{H}_+(x_k, \tilde{U}_k)$. If

\[ \mathcal{H}_+(x_k, \tilde{U}_k) \cap \mathcal{X}_N = \mathcal{H}_+(x_k, \tilde{U}_k) \cap \mathcal{X}_N^{red} \] (19)

holds for a reduced constraint set $\mathcal{X}_N^{red}$ of $\mathcal{X}_N$, then (17) holds and thus (18).

Clearly, $\mathcal{X}_N^{red}$ obtained by removing constraints based on the outer approximation $\mathcal{H}_+(x_k, \tilde{U}_k)$ has an equal or larger number of constraints than a reduced constraint set that is computed without outer approximations. However, outer approximations can be exploited in our framework by providing a beneficial trade-off between the computational effort of removing redundant constraints and the associated speedup of the MPC problem.

#### B. Reachable set outer approximations

Computing reachable sets is known to be a computationally intensive task. However, we will exploit an important feature of reachable sets for LTI systems that will make these computations tractable. For ease of exposition, we limit ourselves to the forward reachable set in the explanation below, but the same properties apply to the backward reachable set, mutatis mutandis.

A key feature of reachable sets for LTI systems is the decomposition $\tilde{R}(x, \mathcal{U}) = \Phi x + \tilde{R}(0, \mathcal{U})$. Here, $\Phi x$ is the free state response with initial condition $x$ and zero input sequence, while $\tilde{R}(0, \mathcal{U})$ is the forward reachable tube starting at the origin. Crucially, $\tilde{R}(0, \mathcal{U})$ does not depend on the current state and can therefore be computed offline, thereby lowering the online computational burden drastically. Even though the forward reachable tube can be computed offline, it is not necessarily easy and the complexity of the set itself can be intractable in some cases. In such cases, the use of outer approximations of reachable sets can be useful and render the computations tractable while simultaneously lowering the complexity of the set itself. The reduced complexity of the sets themselves will be extensively used to adaptively remove redundant constraints in our ca-MPC framework detailed in Section IV-C.

In this paper, we limit ourselves to bounding boxes and outer ellipsoidal approximations of the reachable sets, although the main ideas also apply to other shapes. A bounding
box approximates the forward reachable tube such that
\[ B(P_1, q_1, l_1) \times \cdots \times B(P_N, q_N, l_N) \supseteq R(0, U), \]  
(20a)
\[ B(P, q, l) := \{ x \in R^n \mid -l \leq P(x - q) \leq l \}. \]  
(20b)

Here, \( P_1, \ldots, P_N \) are invertible matrices and \( l_1, \ldots, l_N \) non-negative vectors of appropriate dimension. Similarly, an outer ellipsoidal approximation satisfies
\[ E(L_1, q_1) \times \cdots \times E(L_N, q_N) \supseteq R(0, U), \]  
(21a)
\[ E(L, q) := \{ x \in R^n \mid \| L(x - q) \|_2 \leq 1 \}. \]  
(21b)

Here, \( L_1, \ldots, L_N \) are positive-definite matrices and \( q_1, \ldots, q_N \) vectors of appropriate dimension. These outer approximations allow for simple set descriptions that can be used to remove redundant constraints efficiently as we will show. In particular, recent developments in ellipsoidal methods allow for numerically efficient reachable set computations, even for large-scale systems [16].

C. Numerically efficient constraint removal

In this section, we use the previously introduced outer approximations to derive efficient methods to identify redundant state constraints. Recall, that redundant constraints can be identified by verifying if the reduced constraint set satisfies (19). One approach to identify and remove a redundant constraint is by solving an optimization problem for each inequality constraint [7]. However, these optimization problems are computationally expensive, therefore, we limit ourselves to numerically efficient, albeit more conservative tests. More specifically, we remove an inequality constraint given by \( cX \leq b \), when
\[ \{ x \in R^{Nn} \mid cX \leq b \} \cap \mathcal{H}(x_k, \hat{U}_k) = \mathcal{H}(x_k, \hat{U}_k) \quad (22) \]

Checking (22) is equivalent to verifying
\[ X \in \mathcal{H}(x_k, \hat{U}_k) \implies cX \leq b. \]  
(23)

Next, we elaborate on (23) for box-shaped and ellipsoidal sets, respectively. At the end of this section, we also discuss how a finite intersections and products of boxes and ellipses can be handled efficiently.

In the box case, we would like to test if
\[ X \in B(P, q, l) \implies cX \leq b, \]  
(24)
where \( B(P, q, l) \) is an \( Nn \)-dimensional box-shaped set. In case (24) holds, we set the test outcome of \( \mathcal{M}_B(c, b, B(P, q, l)) \) to true (the constraint can be removed), otherwise we set it to false. Elementary calculations show that (24) is true if and only if
\[ cP^{-1}(\text{sign}(P^{-1}c^\top)) \circ l \leq |b - cQ|. \]  
(25)
where \( \text{sign}(\cdot) \) denotes the element-wise sign and \( \circ \) the element wise product. Similar, for ellipses we have
\[ X \in E(L, q) \implies cX \leq b, \]  
(26)
and thus the constraint can be removed if (26) holds. Again, elementary calculations show that (26) holds if and only if
\[ \| cE^{-\top} \|_2 \leq |b - cQ|. \]  
(27)

Hence, we define \( \mathcal{M}_E(c, b, E(L, q)) \) \( \Longleftrightarrow \) (27).

Based on the definition of \( \mathcal{H}(x_k, \hat{U}_k) \) (14), it is convenient to extend our constraint removal framework to intersections of boxes and ellipses,
\[ \mathcal{H}(x_k, \hat{U}_k) = \bigcap_{p \in [1, N]} \mathcal{H}(p, x_k, \hat{U}_k), \]  
(28)
where \( \mathcal{H}(p, x_k, \hat{U}_k) \) is either a box or an ellipse for \( p \in [1, M] \). Observe that when a constraint \( cX \leq b \) is redundant for some \( p \in [1, M] \), i.e., \( X \in \mathcal{H}(p, x_k, \hat{U}_k) \implies cX \leq b \), then (23) also holds. Hence, the constraint can be removed.

Crucially, (28) enables the removal of redundant constraints using simpler and faster, sequential redundancy checks based on \( \mathcal{H}(p, x_k, \hat{U}_k) \). Additionally, when \( \mathcal{H}(x_k, \hat{U}_k) \) is given by
\[ \mathcal{H}(x_k, \hat{U}_k) = \mathcal{H}^1(x_k, \hat{U}_k) \times \cdots \times \mathcal{H}^N(x_k, \hat{U}_k), \]  
(29)
as is for the reachable tubes (10), then it is also possible to process \( \mathcal{H}^i(x_k, \hat{U}_k) \) and corresponding constraint set \( \mathcal{X}^i_{\text{red}} \) sequentially for each time \( i = 1, 2, \ldots, N \).

V. CASE STUDY: HYPEROTHERMIA TREATMENT

In this section, we demonstrate our proposed ca-MPC framework for numerically efficient constraint removal with a hyperthermia cancer treatment case study. A hyperthermia treatment involves heating a tumor thereby inducing a local fever. As a result, the tumor is sensitized to radio and chemo therapies, making these therapies more effective [11]-[14]. Critically, tissue temperature constraints ensure the safety and comfort of the patient [17]. The temperature constraint is defined on the continuous spatial domain and therefore results in many state constraints after spatial discretization, which is a typical scenario for MPC-based PDE control.

The remainder of this section is structured as follows. First, the system dynamics and MPC problem are introduced. Second, the ca-MPC relevant sets are computed. Last, we compare the results obtained using ca-MPC and the original non-reduced MPC setup.

A. System description

We model the system dynamics in the hyperthermia case study using the 1D heat equation with distributed control, convective boundary conditions, and additional convection on the spatial domain to model the body’s thermoregulatory response. This leads for the normalized spatial coordinates \( r \in [0, 1] \) and time \( t \in R_{\geq 0} \) to
\[ \dot{T}(r, t) = \alpha \nabla^2 T(r, t) - \beta T(r, t) + Q(r, t), \]  
(30a)
\[ Q(r, t) = B_1(r)u_1(t) + B_2(r)u_2(t), \]  
(30b)
with boundary conditions
\[ \nabla T(0, t) = \gamma T(0, t), \quad \nabla T(1, t) = -\gamma T(1, t). \]  
(31)
The parameters are given by \( \alpha = 2.5 \cdot 10^{-4}, \beta = 10^{-2}, \) and \( \gamma = 2.5 \cdot 10^{-3} \). Note that the 1D setup enables the comparison with a non-reduced MPC setup, which would become intractable for 2D and 3D cases.

To ensure patient safety and efficacy of the treatment, we constrain the healthy tissue temperature increase to five degrees with respect to the body temperature and seven degrees for the tumor on \( r \in [0.6, 0.9] \). This constraint is illustrated in Fig. 1 by the function \( T_{\text{max}}(r) \).

C. Heating is applied to the patient using RF-applicator [12], which is modeled through the actuator profiles \( B_1(r) \) and \( B_2(r) \), see Fig. 1. Observe that the heat deposition is not limited to the tumor region, this is also typically observed in real-RF-based hyperthermia treatments. Healthy tissue regions with high power deposition are typically referred to
as hot spots. These hot spots also motivate the use of control strategies that explicitly include state constraints to ensure patient safety. To ensure that the actuator applies no negative heat load and respects power limits, the input constraints are given by $0 \leq u_1 \leq 1$ and $0 \leq u_2 \leq 1$.

Next, we spatially and temporally discretize (30) and (31), using a central-difference scheme and zero-order hold, respectively, to obtain a model as in (1). Here, each element of $x_k \in \mathbb{R}^n$ denotes a nodal temperature defined on an equidistant grid of $n$ points, i.e., $\{0, \frac{1}{n-1}, \frac{2}{n-1}, \ldots, 1\}$. Note that the case study will use a varying number of discretization points $n$ to demonstrate our ca-MPC framework.

In order to heat the tumor, we use a quadratic cost function to track the reference $x_r$, illustrated in Fig. 1. The reference $x_r$ and $u_0$ are obtained through an offline optimization procedure to maximize the tumor temperature. The nominal MPC setup is now given as

$$\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{10} \| x_{i+k} - x_{r,i+k} \|^2 + \| u_{i+k} - u_{0,i+k} \|^2, \\
\text{subject to} & \quad X_k = \Phi x_k + \Gamma u_k, \\
& \quad x_{i,k} \leq T_{\text{max}}, \quad i = 1, \ldots, N - 1, \\
& \quad x_{N,k} \leq T_{\text{terminal}}, \\
& \quad 0 \leq u_{i,k} \leq 1, \quad i = 0, \ldots, N - 1.
\end{align*}$$  (32a)

Here, $T_{\text{max}} \in \mathbb{R}^n$ denotes the spatially discrete temperature constraint, and $T_{\text{terminal}} \leq T_{\text{max}}$ to ensure positive invariance with auxiliary input $u = 0$. For technical details we refer the interested reader to [18].

![Fig. 1. Temperature constraint $T_{\text{max}}(r)$ (---), terminal constraint $T_{\text{terminal}}(r)$ (-----), the state reference $x_{r,i}$ (---), and the distributed source $B_1(r)$ (---), and $B_2(r)$ (-----), for $r \in [0, 1]$.](image)

**B. ca-MPC setup**

We approximate the forward and backward reachable sets using axis-aligned boxes and exploit the observation that our model (30) is a positive system. As a result, the forward reachable sets are outer-approximated by

$$\overrightarrow{R}^i(0, U) := \{ x \in \mathbb{R}^n \mid 0 \leq x \leq \sum_{k=0}^{i-1} A^k B_1 \}.$$  (33)

Here, $1 \in \mathbb{R}^2$ denotes a column of ones. Moreover, note that (33) can be trivially transformed to (20b). Conceptually, (33) denotes a set based on the maximum temperature that can be achieved with both inputs. Next, the backward reachable set is approximated by

$$\overleftarrow{R}^i(X_T, 0) := \{ x \in \mathbb{R}^n \mid 0 \leq x_j \leq T_{j,\text{terminal}} + \delta_{i,j}, \quad \text{for} \quad j = 1, \ldots, n \},$$  (34)

where $\delta_{i,j} := A^{N-i} e_j, \delta \leq (I - A^{N-i}) T_{\text{terminal}}$ and $e_j$ denotes the $j$-th standard basis vector. Conceptually, (34) is the union of maximum single-state perturbations that satisfy the terminal set at $N - i$ time-steps with zero input, i.e., $u = 0$. Again, (34) can be trivially transformed to (20b).

To obtain the cost function level set, we substitute (3b) into the MPC cost function (32a), which yields

$$J(x_k, u_k) = \frac{1}{2} \sum_{i=1}^{N} G_k^T u_k + f(x_k) \in \mathbb{R}_{\geq 0},$$  (35)

where $G$, $f(x_k)$, and $c(x_k)$ denote the respective matrix, vector, and scalar obtained by rewriting (32). Given an initial state $x_k \in X_f$, and input sequence $U_k \in \mathbb{R}_{\geq 0}$, the cost function level set is given by the ellipse $J_u(x_k, U_k) := \mathcal{E}(L_k, q_k)$ with $L_k := G^T (U_k - q_k)^{-1}$, and $q_k := -(GG^T)^{-1} f(x_k)$.

Our framework allows for any feasible input sequence $\hat{U}_k \in \mathbb{R}^{n \times N}(x_k)$, but a natural choice is a shifted version of the previous optimal sequence and appended by the auxiliary control law, $\hat{U}_k = [u_{1,k} \ldots u_{N-1,k} 0^T]^T$. Crucially, note that $\hat{U}_k$ ensures feasibility as the terminal set was chosen to be positively invariant for $u = 0$.

For efficiency reasons, we sequentially remove state constraints for each time step and the forward-, backward-, and cost function level set, as mentioned in (28) and (29). More specifically, we compute each reduced constraint set $X_{i,r}$ for $i = 1, \ldots, N - 1$ using

$$A_i = \{ j \in \{1, n\} \mid [M_B(c_j, b_j, \overrightarrow{R}_i)] \}$$  (36a)

and for $i = N$,

$$A_N = \{ j \in \{1, n\} \mid [M_B(c_T,j, b_T,j, \overrightarrow{R}_N)] \}.$$  (36b)

Here $c_j, b_j, c_T,j$, and $b_T,j$ denote the state constraints $x \leq T_{\text{max}}$ and $x \leq T_{\text{terminal}}$ written as $C x \leq b$ and $C_T x \leq b_T$, respectively. The operator $\vee$, denotes the logical OR, and $\neg$ the logical NOT. For (36b) and (36d), $c_j, b_j$ and $c_T,j, b_T,j$ denote the state constraint $c_j x_{i,j} \leq b_j$ at time $i$ as $c_j, b_j \leq c_T,j, b_T,j$ using (3b). This allows for $\mathcal{M}_E(\cdot)$ to be evaluated using the $Nm$-dimensional $U_k$-space in contrast to the $Nn$-dimensional $x_k$-space, which is typically faster when $m < n$.

**C. Results**

To illustrate the obtained computational improvements we compare the maximum computation time of our proposed ca-MPC framework with respect to a standard MPC setup, with an increasing number of states $n \in [100, 2000]$. Additionally, for our proposed method, we differentiate between the computation time of the pre-solve and quadratic program to provide insight in the overhead of our method. In Fig. 2, the resulting computation time is shown. Clearly, our framework can achieve speedups of two-orders of magnitude. Moreover, the overhead from the pre-solve steps scales well with an increasing number of constraints.

Finally, in Fig. 3 the computation time as well as the percentage of total constraints are shown for a single ca-MPC simulation with $n = 2000$. Here, the adaptive nature of the method is clearly seen as the number of included constraints...
varies during the simulation. During the first 30 seconds, barely any constraints are included as the state cannot reach the constraints within the prediction horizon. Hereafter, the system approaches the temperature constraints, which results in the need to include more constraints as seen in Fig. 3. Healthy tissue temperatures approaching the temperature constraints are also typical for real hyperthermia treatments due to the presence of hot spots. After approximately 60 seconds, the cost function level set starts to shrink as the system approaches the steady-state reference. As result, more constraints can be removed (due to the cost level set) even though the constraints are approached even closer.

VI. CONCLUSION AND OUTLOOK

In this paper, we presented a novel constraint-adaptive MPC scheme that can enable real-time feasibility for systems with many state constraints. A hyperthermia cancer treatment case study demonstrated the benefits of ca-MPC showing up to two orders of magnitude reduction in computation time with respect to a standard MPC setup without loss of closed-loop performance.

This paper focused on computationally cheap methods in order to obtain low computational overhead for the constraint removal pre-solve step. Clearly, less conservative checks can be performed to remove more redundant constraints. For example, state constraints that are not removed are currently not used to identify more redundant constraints. Furthermore, specific solver interactions with, e.g., interior-point and active-set solvers are of interest. These are topics for future work next to applying our framework to a full (3D) application in simulation and experiments of a real hyperthermia treatment.

REFERENCES