

Constraint-adaptive MPC for large-scale systems: Satisfying state constraints without imposing them

S.A.N. Nouwens* B. de Jager* M. Paulides**,**
W.P.M.H. Heemels*

Eindhoven University of Technology, Mechanical Engineering - Control Systems Technology, Electrical Engineering - Electromagnetics for Care & Cure Lab**, Eindhoven, The Netherlands*
*Erasmus University Medical Center Cancer Institute, Department of Radiotherapy, Rotterdam, The Netherlands****

Abstract: Model Predictive Control (MPC) is a successful control methodology, which is applied to increasingly complex systems. However, real-time feasibility of MPC can be challenging for complex systems, certainly when an (extremely) large number of constraints have to be adhered to. For such scenarios with a large number of state constraints, this paper proposes two novel MPC schemes for general nonlinear systems, which we call constraint-adaptive MPC. These novel schemes dynamically select at each time step a (varying) set of constraints that are included in the on-line optimization problem. Carefully selecting the included constraints can significantly reduce, as we will demonstrate, the computational complexity with often only a slight impact on the closed-loop performance. Although not all (state) constraints are imposed in the on-line optimization, the schemes still guarantee recursive feasibility and constraint satisfaction. A numerical case study illustrates the proposed MPC schemes and demonstrates the achieved computation time improvements exceeding two orders of magnitude without loss of performance.

Copyright © 2021 The Authors. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0>)

Keywords: Model Predictive Control; Large-Scale Systems; Adaptive Constraints.

1. INTRODUCTION

1.1 Motivation

Model Predictive Control (MPC) is an advanced feedback control strategy with widespread industrial application (Mayne, 2014; Dotoli et al., 2015). While MPC exists in numerous forms tailored to the specific application, its basic structure remains the same. Indeed, in general, MPC is an optimization-based controller that minimizes a finite-horizon performance criterion subject to system dynamics and constraints over a future sequence of control inputs. At each point of time this optimization problem is solved based on the (estimated) state of the controlled plant. By selecting the first value of the optimal control sequence as the control input applied to the plant and using a receding-horizon principle to obtain the control input, the MPC feedback policy is created.

As mentioned, the control input in (implicit) MPC is obtained by solving an on-line optimization problem at each time step. As such, the real-time feasibility of MPC is determined to a large extent by the computational complexity of this optimization problem (and, of course, the available computational hardware). Certainly for complex systems with a high number of decision variables and con-

straints, the computational complexity of the optimization problem can prohibit real-time implementation. Therefore, numerous methods have been proposed that aim to reduce the computational complexity, such as explicit MPC, model reduction, and numerical solver optimizations, see, e.g., (Rawlings et al., 2019; Hovland et al., 2006; Jerez et al., 2011).

The computational complexity of MPC schemes depends on, among others, the number of decision variables, the model dimension, and the number of constraints (Boyd and Vandenberghe, 2009). In this paper, we are particularly interested in systems with an (extremely) large number of inequality (state) constraints, which can lead to computational intractability for real-time MPC. Dynamical systems with many state constraints are encountered in a diverse range of scenarios, including the case of controlling partial differential equations (PDEs). Certainly, in these cases MPC control is challenging (Altmüller and Grüne, 2012). Typically, the spatial discretization of the PDE results in a large-scale highly complex model. Furthermore, if there are constraints defined on the continuous spatial domain, spatial discretization will yield many constraints. This problem is, for instance, encountered in Magnetic Resonance (MR) guided hyperthermia in cancer treatments (Hendrikx et al., 2018; Paulides et al., 2013). Here, a temperature upper bound on the patients' tissue results in a large number of inequality constraints on the discrete spatial domain. As a result, both the large model and the high number of constraints complicate the real-time implementation of MPC strategies for PDE-based

* This research is supported by KWF Kankerbestrijding and NWO Domain AES, as part of their joint strategic research programme: Technology for Oncology II. The collaboration project is co-funded by the PPP Allowance made available by Health~Holland, Top Sector Life Sciences & Health, to stimulate public-private partnerships.

systems. While model reduction techniques for large-scale systems are available, they do not reduce the number of inequality constraints (Antoulas, A.C.; Sorensen, D.C.; Gugercin, 2000), and, hence, building MPC schemes for such applications remains a true challenge. This observation forms a strong motivation for the present work.

1.2 State-of-the-art

For systems with many inequality constraints, a natural idea is to reduce the number of inequality constraints to obtain real-time feasibility of the MPC scheme. For example, it is possible to identify and remove redundant constraints prior to solving the optimization problem. However, generally, this is intractable in cases with a large number of inequality constraints (Paulraj and Sumathi, 2010). Alternatively, one could aim for online constraint removal as proposed by (Jost and Mönnigmann, 2013). Unfortunately, this approach requires extensive offline computations that do not scale well to a (very) large number of constraints. Other recent techniques use machine learning to either approximate the MPC control law (Hertneck et al., 2018) or use a hybrid approach where machine learning is used to better warm start an active set algorithm (Klaučo et al., 2019). While these methods showed clear improvements in computation time, they require extensive offline training.

1.3 Contributions

As solving the MPC problem within the sample time for systems with a large number of inequality constraints remains to be impossible in many situations, novel MPC strategies are needed that ensure constraint satisfaction and good performance properties while being computationally tractable. In this research we propose two novel real-time feasible MPC schemes for models with many inequality constraints. The key idea behind our schemes is based on adaptively including only a (small) well-chosen and state-dependent subset of the state constraints in the on-line optimization problem, which then becomes tractable to solve. For this reason, we call our new schemes *constraint-adaptive MPC* (ca-MPC). Critically, our approach guarantees constraint satisfaction and recursive feasibility under appropriate conditions, as we will prove. So, although not all (state) constraints are imposed in the on-line optimization, we will still be able to prove that all constraints are satisfied under the proposed MPC laws.

The remainder of this paper is structured as follows. First, we introduce a regular MPC setup and the problem formulation in Section 2. In Section 3, we treat the case where the state constraint set is controlled invariant, propose the corresponding ca-MPC approach and establish recursive feasibility and constraint satisfaction. In Section 4, we consider the case where the state constraint set is not controlled invariant by including a suitably chosen terminal set. In Section 5, a simulation study is performed for a double integrator system to demonstrate the benefits of our new method. There we will observe a significant reduction in computation time (two orders of magnitude) without loss of performance. Last, in Section 6 we present the conclusions and recommendations.

2. MPC SETUP AND PROBLEM FORMULATION

In this section, we introduce the system to be controlled, the MPC setup, and the problem formulation.

2.1 System description

In this paper, we consider plants that are described by the discrete-time time-invariant nonlinear system

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k). \quad (1)$$

Here, $\mathbf{x}_k \in \mathbb{R}^n$ and $\mathbf{u}_k \in \mathbb{R}^m$ denote the plant states and the inputs, respectively at discrete time $k \in \mathbb{N}$. Furthermore, $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ denotes a possibly nonlinear state transition function. The system (1) is subject to state and input constraints that are given by

$$\mathbf{x}_k \in \mathbb{X}, \quad \mathbf{u}_k \in \mathbb{U}, \quad k \in \mathbb{N}, \quad (2)$$

where

$$\mathbb{X} := \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{C}\mathbf{x} \leq \mathbf{b}\}, \quad (3a)$$

$$\mathbb{U} := \{\mathbf{u} \in \mathbb{R}^m \mid \mathbf{C}_u\mathbf{u} \leq \mathbf{b}_u\}, \quad (3b)$$

are assumed to be non-empty sets with $\mathbf{C} \in \mathbb{R}^{n_x \times n}$ and $\mathbf{C}_u \in \mathbb{R}^{n_u \times m}$. Here, n_x and n_u denote the number of state and input constraints, respectively.

2.2 MPC setup

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

$$\underset{\mathcal{U}_k := \{\mathbf{u}_{0|k}, \dots, \mathbf{u}_{N-1|k}\}}{\text{minimize}} \quad \ell_T(\mathbf{x}_{N|k}) + \sum_{i=0}^{N-1} \ell(\mathbf{x}_{i|k}, \mathbf{u}_{i|k}), \quad (4a)$$

$$\text{subject to} \quad \mathbf{x}_{i+1|k} = f(\mathbf{x}_{i|k}, \mathbf{u}_{i|k}), \quad (4b)$$

$$\mathbf{x}_{0|k} = \mathbf{x}_k, \quad (4c)$$

$$\mathbf{x}_{i|k} \in \mathbb{X}, \quad i = 1, \dots, N, \quad (4d)$$

$$\mathbf{u}_{i|k} \in \mathbb{U}, \quad i = 0, \dots, N-1. \quad (4e)$$

Here, $\ell(\cdot)$, $\ell_T(\cdot)$, $\mathbf{x}_{i|k}$, and $\mathbf{u}_{i|k}$ denote the stage cost, the terminal cost, the predicted state, and the predicted input, respectively. In particular, the $i|k$ subscript is used to denote the i -th prediction at time k . For example, $\mathbf{x}_{i|k}$ denotes the prediction of \mathbf{x}_{k+i} made at time k . Additionally, the stage and terminal cost quantify the performance in terms of the predicted states $\mathbf{x}_{i|k}$ and inputs $\mathbf{u}_{i|k}$, $i = 0, 1, 2, \dots, N$. For the optimization problem (4) we denote the set of feasible inputs for state \mathbf{x}_k by

$$\mathbb{U}_F(\mathbf{x}_k) := \{\{\mathbf{u}_{0|k}, \dots, \mathbf{u}_{N-1|k}\} \in \mathbb{U}^N \mid (4b) - (4e)\}. \quad (5)$$

Moreover, we denote the set of feasible states of (4) by

$$\mathbb{X}_F := \{\mathbf{x} \in \mathbb{X} \mid \mathbb{U}_F(\mathbf{x}) \neq \emptyset\}. \quad (6)$$

Lastly, under suitable assumptions on $\ell(\cdot)$, $\ell_T(\cdot)$, \mathbb{X} , and \mathbb{U} , a minimizer of this optimization problem exists for all $\mathbf{x}_k \in \mathbb{X}_F$ and we denote by $\mathcal{U}_k^* := \{\mathbf{u}_{0|k}^*, \dots, \mathbf{u}_{N-1|k}^*\}$, a particular one at time $k \in \mathbb{N}$, i.e.,

$$\mathcal{U}_k^* \in \arg \min_{\mathcal{U}_k \in \mathbb{U}_F(\mathbf{x}_k)} \ell_T(\mathbf{x}_{N|k}) + \sum_{i=0}^{N-1} \ell(\mathbf{x}_{i|k}, \mathbf{u}_{i|k}). \quad (7)$$

The optimal predicted state sequence corresponding to \mathcal{U}_k^* is denoted by $\mathcal{X}_k^* = \{\mathbf{x}_{1|k}^*, \dots, \mathbf{x}_{N|k}^*\}$.

Using a receding horizon implementation, the MPC problem (4) is turned into a feedback law by applying the first computed input in \mathcal{U}_k^* on the real plant (1), i.e.,

$$\mathbf{u}_k := K_{MPC}(\mathbf{x}_k) := \mathbf{u}_{0|k}^*. \quad (8)$$

2.3 Problem statement

As discussed in the introduction, to enable real-time feasibility of the MPC scheme (8), we need to be able to compute a minimizer to the optimization problem (4) within the sampling period that corresponds to the discrete-time plant (1). This becomes particularly challenging if the number of constraints, n_x , in the state constraint set \mathbb{X} becomes very large. The goal of this paper is to provide novel MPC schemes that are computationally more tractable by replacing the state constraint set in the online optimization problem by one with a significantly lower number of (state) constraints, while still leading to good performance properties and guaranteeing constraint satisfaction (2).

3. CA-MPC UNDER CONTROLLED INVARIANCE OF ORIGINAL CONSTRAINT SET

As already hinted upon in the previous section, we will replace the constraint set \mathbb{X} in the on-line optimization problem (4) by a “simpler” one that contains a small subset of the constraints in \mathbb{X} , thereby making the on-line optimization problem computationally tractable. Regardless, constraint satisfaction must be assured for all times. In particular, we are going to replace \mathbb{X} by a well-crafted combination of a reduced constraint set

$$\mathbb{X}_r(\mathbf{x}_k) := \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{C}_j \mathbf{x} \leq \mathbf{b}_j, \quad j \in \mathcal{J}(\mathbf{x}_k)\}, \quad (9)$$

where $\mathcal{J}(\mathbf{x}_k) \subset \mathbb{N}_{[1, n_x]}$ is the index set that captures the selected constraints, and a so-called delta set $\Delta(\mathbf{x}_k)$ based on the current state \mathbf{x}_k (and possibly other information available at time $k \in \mathbb{N}$). The exact construction will be explained in detail below. In this section, we first consider the case where \mathbb{X} is controlled invariant.

Definition 1. The set \mathbb{X} is controlled invariant for (1) with the input set \mathbb{U} , if for all $\mathbf{x} \in \mathbb{X}$ there exists an $\mathbf{u} \in \mathbb{U}$ such that $f(\mathbf{x}, \mathbf{u}) \in \mathbb{X}$.

The reduced MPC problem at time $k \in \mathbb{N}$ for state \mathbf{x}_k , where \mathbb{X} is replaced by a combination of $\mathbb{X}_r(\mathbf{x}_k)$ and $\Delta(\mathbf{x}_k)$, is given by

$$\begin{aligned} & \text{minimize} && \ell_T(\mathbf{x}_{N|k}) + \sum_{i=0}^{N-1} \ell(\mathbf{x}_{i|k}, \mathbf{u}_{i|k}), & (10a) \\ & \text{subject to} && (4b), (4c), (4e), \\ & && \mathbf{x}_{1|k} \in \mathbb{X}_r(\mathbf{x}_k) \cap \Delta(\mathbf{x}_k), & (10b) \\ & && \mathbf{x}_{i|k} \in \mathbb{X}_r(\mathbf{x}_k), \quad i = 2, \dots, N. & (10c) \end{aligned}$$

A few comments are in order. First, only the first predicted state, $\mathbf{x}_{1|k}$, is constrained by both the reduced constraint set $\mathbb{X}_r(\mathbf{x}_k)$ and the delta set $\Delta(\mathbf{x}_k)$, while the other states are constrained only by $\mathbb{X}_r(\mathbf{x}_k)$. Second, the reduced constraint set depends on \mathbf{x}_k , which leads to the adaptive nature of the set. Last, we define the feasible set for (10) as $\mathbb{X}_F^r := \{\mathbf{x}_k \in \mathbb{X} \mid \mathbb{U}_F^r(\mathbf{x}_k) \neq \emptyset\}$, where $\mathbb{U}_F^r(\mathbf{x}_k) := \{\mathbf{u}_k \in \mathbb{U}^N \mid (4b), (4c), (4e), (10b), (10c)\}$. Moreover, the corresponding control law is denoted by $K_{MPC}^r : \mathbb{X}_F^r \rightarrow \mathbb{U}$ and defined similarly as in (8).

We now proceed with proposing a construction for the reduced constraint set $\mathbb{X}_r : \mathbb{X} \rightrightarrows \mathbb{R}^n$ according to (9) and guarantee recursive feasibility and constraint satisfaction by providing suitable conditions on $\Delta : \mathbb{X} \rightrightarrows \mathbb{R}^n$.

To select the relevant constraints in (9) via $\mathcal{J}(\mathbf{x}_k)$ given a state \mathbf{x}_k , we introduce the concept of a safety set. Here, a safety set $\mathbb{X}_s \subseteq \mathbb{X}$ is defined as a tightened constraint set based on non-negative vector $\boldsymbol{\alpha} \geq \mathbf{0}$, which can be seen as a “safety margin” or a “margin to constraint violation.” In particular, $\mathbb{X}_s := \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{C}\mathbf{x} \leq \mathbf{b} - \boldsymbol{\alpha}\}$. Given \mathbf{x}_k and \mathbb{X}_s , we include constraints in $\mathbb{X}_r(\mathbf{x}_k)$ based on the constraint violations of the safety set:

$$\mathcal{J}(\mathbf{x}_k) := \{j \in \mathbb{N}_{[1, n_x]} \mid \mathbf{C}_j \mathbf{x}_k > b_j - \alpha_j\}. \quad (11)$$

This ensures that once the state approaches the boundary of \mathbb{X} , the relevant constraints are included in $\mathbb{X}_r(\mathbf{x}_k)$ as, loosely speaking, for these constraints the safety margin has become too small. Observe that the safety set itself is not included in the MPC problem. Furthermore, note that the computation of (11) and thus computing $\mathbb{X}_r(\mathbf{x}_k)$ can be performed efficiently and only has to be done once at each time step. We illustrate the MPC setup from (10) with the reduced constraint set, delta set, and predicted state trajectory in Fig. 1.

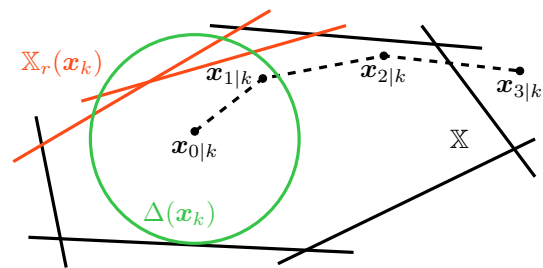


Fig. 1. Illustrating ca-MPC when \mathbb{X} is controlled invariant. Clearly, $\mathbf{x}_{1|k}$ cannot be outside of \mathbb{X} while $\mathbf{x}_{2|k}, \dots, \mathbf{x}_{N|k}$ are not necessarily constrained to \mathbb{X} as illustrated by $\mathbf{x}_{3|k}$.

While the safety set tries to include relevant constraints in $\mathbb{X}_r(\mathbf{x}_k)$, it does not guarantee recursive feasibility and constraint satisfaction automatically. To obtain these properties, conditions have to be imposed on $\Delta(\mathbf{x}_k)$.

Theorem 2. Consider the ca-MPC scheme (10) with \mathbb{X} controlled invariant for (1) with the input set \mathbb{U} . If for all states $\mathbf{x} \in \mathbb{X}$ and corresponding reduced constraint set $\mathbb{X}_r(\mathbf{x})$, the delta set $\Delta(\mathbf{x})$ satisfies the properties

- (i) $\mathbb{X}_r(\mathbf{x}) \cap \Delta(\mathbf{x}) \subseteq \mathbb{X}$,
- (ii) $\exists \bar{\mathbf{u}} \in \mathbb{U}$ such that $f(\mathbf{x}, \bar{\mathbf{u}}) \in \mathbb{X}_r(\mathbf{x}) \cap \Delta(\mathbf{x})$,

then the feasible set is equal to \mathbb{X} , i.e., $\mathbb{X}_F^r = \mathbb{X}$ and (10) is recursively feasible in the sense that if $\mathbf{x} \in \mathbb{X}_F^r$, then $f(\mathbf{x}, K_{MPC}^r(\mathbf{x})) \in \mathbb{X}_F^r$. Moreover, constraint satisfaction is guaranteed for the closed-loop system

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k), \quad \mathbf{u}_k = K_{MPC}^r(\mathbf{x}_k) \quad (12)$$

in the sense that for all trajectories of this system with $\mathbf{x}_0 \in \mathbb{X}_F^r = \mathbb{X}$ the constraints in (2) are satisfied.

Due to space constraints, all proofs are omitted.

We now have established sufficient conditions on $\Delta : \mathbb{X} \rightrightarrows \mathbb{R}^n$ that guarantee recursive feasibility and constraint satisfaction. However, it is still unclear how to construct a $\Delta(\mathbf{x})$ that satisfies Conditions (i)-(ii) in Theorem 2. To this end, we propose a construction for $\Delta(\mathbf{x})$ that satisfies both Conditions (i) and (ii). An example of a simple set $\Delta(\mathbf{x})$ for $\mathbf{x} \in \mathbb{X}$ given an admissible input $\bar{\mathbf{u}}$ such that $f(\mathbf{x}, \bar{\mathbf{u}}) \in \mathbb{X}$

is a sphere: $\Delta(\mathbf{x}) = \{\tilde{\mathbf{x}} \in \mathbb{R}^n \mid \|f(\mathbf{x}, \tilde{\mathbf{u}}) - \tilde{\mathbf{x}}\|_p \leq c(\mathbf{x})\}$, $p \in \{\mathbb{N}, \infty\}$. Here, the radius $c(\mathbf{x})$ depends on \mathbf{x} and is chosen such that $\mathbb{X}_r(\mathbf{x}) \cap \Delta(\mathbf{x}) \subseteq \mathbb{X}$ holds. Note that a $\Delta(\mathbf{x})$ satisfying the required conditions can always be obtained by selecting $c(\mathbf{x}) = 0$ (although that is a very restrictive choice). Depending on the choice of the norm $\|\cdot\|_p$ different state constraint sets are obtained in the on-line MPC optimization problem, e.g., for $p = 1, \infty$ we obtain polyhedral sets, while for $p = 2$, we obtain quadratic-type constraints as seen in Fig. 1.

4. CA-MPC WITH TERMINAL SETS

In the previous section, we assumed that \mathbb{X} was controlled invariant. While this resulted in rather basic conditions regarding recursive feasibility and constraint satisfaction, it is often not the case that the original constraint set \mathbb{X} is controlled invariant, or that there is a natural (sufficiently large) and easy-to-compute controlled invariant set inside \mathbb{X} to replace \mathbb{X} . Alternatively, simple terminal sets are more commonplace in guaranteeing recursive feasibility for MPC schemes (Mayne et al., 2000). Therefore, it is of interest to extend the previous results to ca-MPC schemes based on terminal sets, for which we assume the following:

Assumption 3. The set $\mathbb{X}_T \subset \mathbb{X}$ is a nonempty and controlled invariant set for (1) with input set \mathbb{U} .

We start by introducing a standard MPC setup with terminal set as

$$\begin{aligned} & \underset{\mathbf{u}_{0|k}, \dots, \mathbf{u}_{N-1|k}}{\text{minimize}} && \ell_T(\mathbf{x}_{N|k}) + \sum_{i=0}^{N-1} \ell(\mathbf{x}_{i|k}, \mathbf{u}_{i|k}), \quad (13a) \\ & \text{subject to} && \mathbf{x}_{i+1|k} = f(\mathbf{x}_{i|k}, \mathbf{u}_{i|k}), \quad (13b) \\ & && \mathbf{x}_{0|k} = \mathbf{x}_{1|k-1}^* = \mathbf{x}_k, \quad (13c) \\ & && \mathbf{x}_{i|k} \in \mathbb{X}, \quad i = 1, \dots, N-1, \quad (13d) \\ & && \mathbf{x}_{N|k} \in \mathbb{X}_T, \quad (13e) \\ & && \mathbf{u}_{i|k} \in \mathbb{U}, \quad i = 0, \dots, N-1. \quad (13f) \end{aligned}$$

Observe that the final predicted state $\mathbf{x}_{N|k}$ is constrained to be contained in the terminal set \mathbb{X}_T .

Now we propose a modification of (13) to obtain a reduced MPC problem with terminal set at time $k \in \mathbb{N}$ with state $\mathbf{x}_k = \mathbf{x}_{1|k-1}^*$. This new MPC is parameterized by a state sequence $\mathcal{X}_{k-1}^* = (\mathbf{x}_{1|k-1}^*, \dots, \mathbf{x}_{N|k-1}^*)$ and given by

$$\begin{aligned} & \underset{\mathbf{u}_{0|k}, \dots, \mathbf{u}_{N-1|k}}{\text{minimize}} && \ell_T(\mathbf{x}_{N|k}) + \sum_{i=0}^{N-1} \ell(\mathbf{x}_{i|k}, \mathbf{u}_{i|k}), \quad (14a) \\ & \text{subject to} && (13b), (13c), (13e), (13f), \\ & && \mathbf{x}_{i|k} \in \mathbb{X}_r(\mathbf{x}_{i+1|k-1}^*) \cap \Delta(\mathbf{x}_{i+1|k-1}^*), \\ & && i = 1, \dots, N-1. \quad (14b) \end{aligned}$$

First of all, let us note that the reduced constraint set $\mathbb{X}_r(\cdot)$ is based on using the optimal predicted state sequence \mathcal{X}_{k-1}^* at the previous time step (and not only based on \mathbf{x}_k). Second, note that the construction of $\mathbb{X}_r(\cdot)$ can remain unchanged and, hence, is given by (9) by now exploiting the optimal predicted sequence \mathcal{X}_{k-1}^* . Third, the delta set $\Delta(\mathbf{x}_{i+1|k-1}^*)$, $i = 1, \dots, N-1$, is applied over the complete prediction horizon (except the last step). Conceptually, the sequence of delta sets can be interpreted as a tube $\Delta^{N-1}(\mathcal{X}_{k-1}^*) := (\Delta(\mathbf{x}_{2|k-1}^*), \dots, \Delta(\mathbf{x}_{N|k-1}^*))$

that constrains the predicted state sequence (which will make sure in the end that we have constraint satisfaction, as we will prove in Theorem 5). Note that the role of the tube is different from tube-based MPC (Mayne et al., 2005). In Fig. 2 we provide an illustrative example of the ca-MPC scheme from (14).

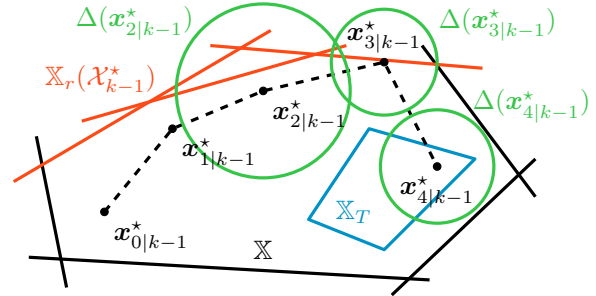


Fig. 2. Illustrating ca-MPC with a terminal set and parameterized by \mathcal{X}_{k-1}^* . Note that $\mathbb{X}_r(\mathcal{X}_{k-1}^*)$ is constructed as detailed in Remark 6 for illustration purposes. Critically, the reduced constraint set and delta sets ensure $\mathbf{x}_{i|k} \in \mathbb{X}$ for $i = 1, \dots, N$.

Remark 4. We assume the terminal set is sufficiently simple to be included in (14) without computational issues. We note that reduction of the terminal set constraints is possible as well, but it is not detailed for ease of exposition.

In particular, the constraints in Problem (14) are well-defined (and parameterized) for all $\mathcal{X}_{k-1} = (\mathbf{x}_{1|k-1}, \dots, \mathbf{x}_{N|k-1})$ contained in the set \mathbb{S} defined as

$$\begin{aligned} \mathbb{S} := \{(\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_N) \in \mathbb{X}^N \mid \text{there exists} \\ (\tilde{\mathbf{u}}_1, \dots, \tilde{\mathbf{u}}_{N-1}) \in \mathbb{U}^{N-1} \text{ s.t. } \tilde{\mathbf{x}}_{i+1} = f(\tilde{\mathbf{x}}_i, \tilde{\mathbf{u}}_i), \\ i = 1, \dots, N-1, \text{ and } \tilde{\mathbf{x}}_N \in \mathbb{X}_T\}, \quad (15) \end{aligned}$$

which are all state trajectories of length N that satisfy the system dynamics (1), the input and state constraints (2), and end in the terminal set \mathbb{X}_T .

For $k = 0$, the sets $\mathbb{X}_r(\mathbf{x}_{i+1|k-1})$ for $i = 1, \dots, N-1$ are not explicitly defined yet as the state sequence \mathcal{X}_{-1}^* is not available. However, to overcome this, the scheme is started up at $k = 0$ and state \mathbf{x}_0 by finding an $\mathcal{X}_{-1}^* \in \mathbb{S}$, where \mathbb{S} as in (15) with the additional constraint that $\tilde{\mathbf{x}}_1 = \mathbf{x}_0$. With such a sequence $\mathcal{X}_{-1}^* \in \mathbb{S}$ the optimization problem (14) can be formulated and solved, and this can be considered as the initial condition to our ca-MPC scheme. Therefore, we introduce

$$\mathbb{S}_0(\mathbf{x}_0) := \{(\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \dots, \tilde{\mathbf{x}}_N) \in \mathbb{S} \mid \tilde{\mathbf{x}}_1 = \mathbf{x}_0\}. \quad (16)$$

We denote the set of feasible input sequences to (14) given $\mathcal{X}_{k-1}^* \in \mathbb{S}$, by $\mathbb{U}_F^T(\mathcal{X}_{k-1}^*) := \{(\mathbf{u}_{0|k}, \dots, \mathbf{u}_{N-1|k}) \in \mathbb{U}^N \mid (13b), (13c), (13e), (13f), (14b)\}$, and corresponding state trajectories by

$$\begin{aligned} \mathbb{X}_F^T(\mathcal{X}_{k-1}^*) := \{(\mathbf{x}_{1|k}, \dots, \mathbf{x}_{N|k}) \in (\mathbb{R}^n)^N \mid (13b), (13c) \\ \text{for some } (\mathbf{u}_{0|k}, \dots, \mathbf{u}_{N-1|k}) \in \mathbb{U}_F^T(\mathcal{X}_{k-1}^*)\}. \quad (17) \end{aligned}$$

The resulting MPC law based on the receding-horizon principle is denoted by $K_{MPC}^{T,r} : \mathbb{S} \rightarrow \mathbb{U}$, i.e.,

$$\mathbf{u}_k = K_{MPC}^{T,r}(\mathcal{X}_{k-1}^*) = \mathbf{u}_{0|k}^*, \quad k \in \mathbb{N} \quad (18)$$

where $\mathbf{u}_k^* := (\mathbf{u}_{0|k}^*, \dots, \mathbf{u}_{N-1|k}^*)$ is a minimizer in (14), i.e.,

$$\mathcal{U}_k^* \in \arg \min_{\mathcal{U}_k \in \mathbb{U}_F^T(\mathcal{X}_{k-1}^*)} \ell_T(\mathbf{x}_{N|k}) + \sum_{i=0}^{N-1} \ell(\mathbf{x}_{i|k}, \mathbf{u}_{i|k}). \quad (19)$$

For $k = 0$, we select $\mathcal{X}_{-1}^* \in \mathbb{S}_0(\mathbf{x}_0) \subset \mathbb{S}$. Note that \mathbf{u}_k in this MPC scheme depends on \mathcal{X}_{k-1}^* , which includes \mathbf{x}_k , but thus also depends on the future predicted states at the previous time step.

Similar to the previous section, recursive feasibility and constraint satisfaction are guaranteed by imposing suitable conditions on $\Delta^{N-1}(\mathcal{X})$.

Theorem 5. Consider system (1) with the reduced MPC scheme (10) and let the terminal set $\mathbb{X}_T \subseteq \mathbb{X}$ satisfy Assumption 3. Suppose that for all $\mathcal{X} = (\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_N) \in \mathbb{S}$, the tube $\Delta^{N-1}(\mathcal{X})$ satisfies

- (i) $\mathbb{X}_r(\tilde{\mathbf{x}}_i) \cap \Delta(\tilde{\mathbf{x}}_i) \subseteq \mathbb{X}$ for $i = 2, 3, \dots, N$,
- (ii) $\tilde{\mathbf{x}}_i \in \Delta(\tilde{\mathbf{x}}_i)$ for $i = 2, 3, \dots, N$.

Then, the reduced MPC scheme (14) satisfies the following properties:

Recursive feasibility : Given $\mathcal{X}_{k-1} \in \mathbb{S}$, then $\mathbb{U}_F^T(\mathcal{X}_{k-1}) \neq \emptyset$ and any $\mathcal{X}_k \in \mathbb{X}_F^T(\mathcal{X}_{k-1})$ (including any optimal $\mathcal{X}_k^* \in \mathbb{X}_F^T(\mathcal{X}_{k-1})$) satisfies $\mathcal{X}_k \in \mathbb{S}$.

Constraint satisfaction : For all $\mathbf{x}_0 \in \mathbb{X}$ with $\mathcal{X}_{-1}^* \in \mathbb{S}_0(\mathbf{x}_0)$, the trajectories of the closed-loop system

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k), \quad \mathbf{u}_k = K_{MPC}^{T,r}(\mathcal{X}_{k-1}^*) \quad (20)$$

are well-defined for all $k \in \mathbb{N}$ and the constraints in (2) are satisfied for all $k \in \mathbb{N}$.

We now have sufficient conditions on the structure of $\Delta^{N-1}(\cdot)$ that guarantee recursive feasibility and constraint satisfaction. Again, an example for a simple tube $\Delta^{N-1}(\mathcal{X}_{k-1})$ that satisfies Conditions (i) and (ii) given $\mathcal{X}_{k-1} \in \mathbb{S}$, is a sphere centered around \mathcal{X}_{k-1} : $\Delta(\mathbf{x}_{i+1|k-1}) = \{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x}_{i+1|k-1} - \mathbf{x}\|_2 \leq c(\mathbf{x}_{i+1|k-1})\}$. Here, we increase the radius $c(\mathbf{x}_{i+1|k-1})$ subject to $\mathbb{X}_r(\mathbf{x}_{i+1|k-1}) \cap \Delta(\mathbf{x}_{i+1|k-1}) \subseteq \mathbb{X}$. Observe that the conditions in Theorem 5 are trivially satisfied by these choices and that $c(\mathbf{x}_{i+1|k-1}) = 0$ is always a valid choice, although, clearly, large values of $c(\mathbf{x}_{i+1|k-1})$ are desirable as it gives more freedom in the MPC optimization problem. This particular choice of $\Delta^{N-1}(\mathcal{X}_{k-1})$ is illustrated in Fig. 2.

Remark 6. If a reduced constraint set that varies over the horizon is undesired, it is also possible to replace $\mathbb{X}_r(\mathbf{x}_{i+1|k-1})$ by $\mathbb{X}_r(\mathcal{X}_{k-1}) := \bigcap_{i=2}^N \mathbb{X}_r(\mathbf{x}_{i|k-1})$.

Remark 7. Interestingly, it can be established that if the terminal set and cost are chosen to guarantee asymptotic stability of the origin for the closed-loop system under the common MPC setup (13) using the typical terminal conditions, see, e.g. (A1-A4) in Mayne et al. (2000), then this also guarantees asymptotic stability of the origin under the ca-MPC control strategy (14).

5. DOUBLE INTEGRATOR CASE STUDY

To demonstrate ca-MPC we apply the proposed methodology in Section 4 to a double integrator system. We opted to use this two-dimensional system as it is more easy to visualize and we can, just as for large-scale systems, impose many (non-redundant) state constraints, which makes this a good example for illustration. Additionally,

the non-reduced MPC solution (13) is tractable for this system, which allows for a comparison to our ca-MPC scheme. Finally, our implementation uses the inputs as decision variables (so-called ‘‘dense’’ formulation) which makes the model order less relevant. Hence, the double integrator system with many state constraints is a good representative for more complex systems.

5.1 System and constraint definition

The double integrator system is described by

$$\mathbf{x}_{k+1} = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} 0.005 \\ 0.1 \end{bmatrix} u_k. \quad (21)$$

Next, the state constraints are defined as piecewise linear approximations of two quadratic constraints resulting in $n_x = 1000$ constraints, see Fig. 3. The input is constrained by $\mathbb{U} = [-0.5, 0.5]$ and the prediction horizon is chosen as $N = 15$. Last, the terminal set is shown in Fig. 3. The resulting quadratic program given by (13) has approximately 14,000 linear inequality constraints.

We implement the ca-MPC scheme from Section 4 as \mathbb{X} is not controlled invariant. Here, we choose $\ell(\mathbf{x}, u) = \|\mathbf{x}\|_2^2 + u^2$ and $\ell_T(\mathbf{x}) = \|\mathbf{x}\|_2^2$. Next, $\boldsymbol{\alpha}$, which defines the safety set and in turn the reduced constraint set, is given by

$$\boldsymbol{\alpha} = 0.1 [\|\mathbf{C}_{1,:}\|_2 \dots \|\mathbf{C}_{n_x,:}\|_2]^\top. \quad (22)$$

This has the interpretation that \mathbb{X} is tightened by a uniform distance. Recall that $\mathbf{C} = [\mathbf{C}_{1,:}^\top \dots \mathbf{C}_{n_x,:}^\top]^\top$ is the matrix defining the state constraints (2). Additionally, for ease of implementation, we use a reduced constraint set that does not vary over the horizon, see Remark 6. Next, we define the delta set as $\Delta(\mathbf{x}) := \{\hat{\mathbf{x}} \in \mathbb{R}^2 \mid \|\hat{\mathbf{x}} - \mathbf{x}\|_\infty \leq c(\mathbf{x})\}$. The radius $c(\mathbf{x})$ is chosen such that $\Delta(\mathbf{x}_{i+1|k-1}) \subseteq \mathbb{X}_c(\mathcal{X}_k)$, where

$$\mathbb{X}_c(\mathcal{X}_k) := \{\mathbf{x} \in \mathbb{R}^2 \mid \mathbf{C}_j \mathbf{x} \leq b_j, j \in \mathcal{J}_c(\mathcal{X}_k)\}, \quad (23a)$$

$$\mathcal{J}_c(\mathcal{X}_k) := \mathbb{N}_{[1, n_x]} \setminus \left(\bigcap_{i=2}^N \mathcal{J}(\mathbf{x}_{i|k}) \right). \quad (23b)$$

Observe that $\mathbb{X}_c(\mathcal{X}_k)$ contains the constraints not included in $\mathbb{X}_r(\mathcal{X}_k)$. The radius $c(\mathbf{x})$ is chosen by checking the vertices of $\Delta(\mathbf{x})$ against $\mathbb{X}_c(\mathcal{X})$. Note that this choice for $\mathbb{X}_c(\mathcal{X}_k)$ is conservative in the light of Condition (i) from Theorem 5. However, the additional conservatism is considered acceptable given the resulting trivial construction of the delta sets. Of course, all guarantees regarding constraint satisfaction and recursive feasibility remain to hold.

5.2 Results

The ca-MPC closed-loop trajectory is compared with the (unreduced) MPC solution from (4), see Fig. 3. Here, (21) is initialized at $\mathbf{x}_0 = [-4.2 \ -0.3]^\top$ and steered towards the origin. As seen in Fig. 3, the ca-MPC solution is indistinguishable from the MPC solution. Nevertheless, ca-MPC uses only a fraction of the state constraints and has significantly lower computation time. The percentage of state constraints and the computational time, both with respect to traditional MPC, are shown in Fig. 4. Note that the computation of the delta sets and reduced

constraint set is included in the computation time for ca-MPC. From Fig. 4 we observe an improvement in the computation time of two orders of magnitude. This result clearly demonstrates the potential of ca-MPC for systems with many state constraints.

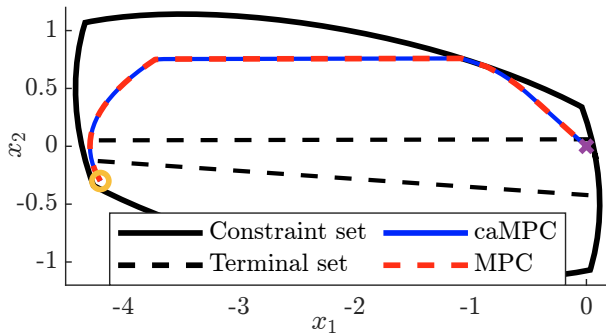


Fig. 3. The closed-loop state trajectory using the regular MPC solution and the ca-MPC solution.

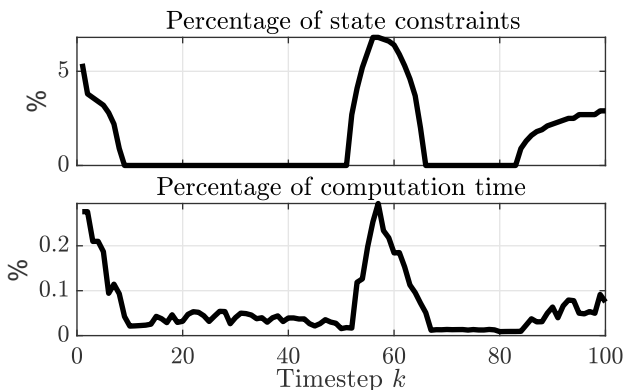


Fig. 4. The percentage of constraints and the computation time with respect to the full MPC solution over time. When the state approaches the boundary of \mathbb{X} the number of constraints and computation time increase.

6. CONCLUSION

In this paper, we presented *constraint-adaptive* MPC (ca-MPC) schemes that enable real-time feasibility for complex nonlinear systems with a large number of state inequality constraint. The two novel ca-MPC designs were based on the idea of dynamically varying the set of imposed state constraints in the MPC optimization problem in such a way that recursive feasibility and constraint satisfaction is guaranteed. The first scheme exploited the controlled invariance of the original state constraint set, while the second relaxed this assumption and just required the availability of a basic terminal set. A simulation study with a benchmark double integrator system showed the potential of our new framework. Indeed, the implemented ca-MPC lead to a significant reduction of the computation time with a negligible impact on closed-loop performance.

This work can be extended in various directions. For instance, as indicated in Remark 7, it is possible to preserve stability of the closed-loop ca-MPC system using standard conditions on the terminal ingredients. Interesting future work in this light is to extend this preservation of closed-loop stability also to performance guarantees by proper construction of safety and delta sets. Also the extension

of our results to ca-MPC setups with large prediction horizons, without the use of controlled invariance of the state constraint set or the use terminal sets, is of interest.

REFERENCES

- Altmüller, N. and Grüne, L. (2012). Distributed and boundary model predictive control for the heat equation. *GAMM-Mitteilungen*, 35(2), 131–145.
- Antoulas, A.C.; Sorensen, D.C.; Gugercin, S. (2000). A Survey of Model Reduction Methods for Large-Scale Systems. Technical report.
- Boyd, S. and Vandenberghe, L. (2009). *Convex optimization*. Cambridge University Press, Cambridge, 7th edition.
- Dotoli, M., Fay, A., Miśkiewicz, M., and Seatzu, C. (2015). A Survey on Advanced Control Approaches in Factory Automation. *IFAC-PapersOnLine*, 48(3), 394–399.
- Hendriks, R., Curto, S., De Jager, B., Maljaars, E., Van Rhoon, G., Paulides, M., and Heemels, W. (2018). Pod-based recursive temperature estimation for mr-guided rf hyperthermia cancer treatment: A pilot study. In *IEEE Conf. on Decision and Control*, 5201–5208.
- Hertneck, M., Köhler, J., Trimpe, S., and Allgöwer, F. (2018). Learning an Approximate Model Predictive Controller With Guarantees. *IEEE Control Systems Letters*, 2(3), 543–548.
- Hovland, S., Willcox, K., and Gravdahl, J.T. (2006). MPC for Large-Scale Systems via Model Reduction and Multiparametric Quadratic Programming. In *45th IEEE Conf. on Decision and Control*, 3418–3423.
- Jerez, J.L., Kerrigan, E.C., and Constantinides, G.A. (2011). A condensed and sparse QP formulation for predictive control. In *50th IEEE Conf. on Decision and Control and European Control Conf.*, 5217–5222.
- Jost, M. and Mönnigmann, M. (2013). Accelerating model predictive control by online constraint removal. In *52nd IEEE Conf. on Decision and Control*, 5764–5769.
- Klaučo, M., Kalúz, M., and Kvasnica, M. (2019). Machine learning-based warm starting of active set methods in embedded model predictive control. *Engineering Applications of Artificial Intelligence*, 77, 1–8.
- Mayne, D.Q. (2014). Model predictive control: Recent developments and future promise. *Automatica*, 50(12), 2967–2986.
- Mayne, D.Q., Rawlings, J., Rao, C., and Scaekaert, P. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6), 789–814.
- Mayne, D.Q., Seron, M., and Raković, S. (2005). Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41(2), 219–224.
- Paulides, M.M., Stauffer, P.R., Neufeld, E., Maccarini, P.F., Kyriakou, A., Canters, R.A., Diederich, C.J., Bakker, J.F., and Van Rhoon, G.C. (2013). Simulation techniques in hyperthermia treatment planning. *International Journal of Hyperthermia*, 29(4), 346–357.
- Paulraj, S. and Sumathi, P. (2010). A comparative study of redundant constraints identification methods in linear programming problems. *Mathematical Problems in Engineering*, 2010.
- Rawlings, J.B., Mayne, D.Q., and Diehl, M.M. (2019). *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, 2nd edition.