Automatica 83 (2017) 309–316

Contents lists available at ScienceDirect

Automatica

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

Brief paper

Semi-explicit MPC based on subspace clustering[☆]

Gregor Goebel, Frank Allgöwer

Institute for Systems Theory and Automatic Control, University of Stuttgart, Pfaffenwaldring 9, 70550 Stuttgart, Germany

ARTICLE INFO

Article history:
Received 20 April 2016
Received in revised form 28 December 2016
Accepted 16 May 2017
Available online 5 July 2017

ABSTRACT

This paper presents a new strategy of simplifying the online computations in linear model predictive control (MPC). Employing a specific type of state dependent parameterization for the optimization variable in MPC, advantages of explicit MPC are combined with those of online optimization based MPC into an efficient MPC scheme. The parameterization is computed offline applying a tailored subspace clustering algorithm to training data consisting of states and corresponding solutions to the MPC optimization problem. It is then refined to guarantee feasibility of the parameterized optimization. During the offline design phase, complexity of the parameterization can be adjusted and control performance can be traded off against online computational effort and storage requirements. Numerical examples evaluate the presented methods and illustrate their benefits.

1. Introduction

Model predictive control (MPC) is a successful control strategy due to its capability to optimize control performance and take constraints on system states and controlled inputs explicitly into account. Yet, its computational load classically caused by solving a finite horizon open-loop optimal control problem online in each time step poses demanding requirements for the computational hardware used and restricts applicability of MPC. Many results aiming at alleviating this drawback are available, which can be grouped into a few main paradigms. In this paper we present a new approach which joins ideas of two of the existing paradigms of fast MPC schemes.

A first class of simplified MPC algorithms adheres to the online optimization problem and simplifies it such that it can be solved faster numerically. An important branch therein is based on parameterizing the decision variable, yielding a smaller optimization problem which can be solved faster. Many practically applied MPC schemes fit into this framework and consist of simple move-blocking strategies where the predicted input trajectory is just blocked over some time steps (Qin & Badgwell, 2003), mostly applied disregarding theoretical feasibility as well as closed-loop stability considerations. More recent results explicitly address feasibility and closed-loop stability (Cagienard, Grieder, Kerrigan, & Morari, 2007; Gondhalekar & Imura, 2010; Shekhar & Manzie, 2015) as well as more elaborate ways of determining parameterizations aiming at reducing the performance loss incurred due to the parameterization and/or extending their feasible set (Khan, Valencia-Palomo, Rossiter, Jones, & Gondhalekar, 2014; Li, Xi, & Lin, 2013; Longo, Kerrigan, Ling, & Constantinides, 2011; Valencia-Palomo, Rossiter, Jones, Gondhalekar, & Khan, 2011).

A second type of fast MPC algorithms alleviates online computations by shifting computational load offline. As is the strategy in explicit MPC, a state dependent solution to the open-loop optimal control problem is precomputed offline and stored so that online it only has to be evaluated for the current system state, see Bemporad, Morari, Dua, and Pistikopoulos (2002), Tøndel, Johansen, and Bemporad (2003) and many subsequent results. A downside of such approaches is that the complexity of the solution grows rapidly with increasing problem size, rendering the approach quickly impractical. As a remedy, simplified sub-optimal explicit MPC schemes as e.g. in Johansen and Grancharova (2003), Jones and Morari (2009), Summers, Jones, Lygeros, and Morari (2011) have been proposed.

Obviously, both paradigms have a complementing nature. Applying parameterizations no pre-computation or storage of relevant amounts of data is required as all information needed for control is generated online during runtime. Yet, state dependence of the solutions is not accounted for or exploited at all. In contrast, the second approach is completely based on exploiting this dependence. No optimization is required online but large and possibly obstructive amounts of data have to be dealt with offline and online. The results presented in the current paper make use of this situation and join ideas from both paradigms to exploit and combine their individual advantages: A particular type of state

[☆] The material in this paper was partially presented at the 52nd IEEE Conference on Decision and Control, December 10–13, 2013, Florence, Italy and at the 2014 American Control Conference, June 4–6, 2014, Portland, OR, USA. This paper was recommended for publication in revised form by Associate Editor Giancarlo Ferrari-Trecate under the direction of Editor Ian R. Petersen.

E-mail addresses: gregor.goebel@ist.uni-stuttgart.de (G. Goebel), allgower@ist.uni-stuttgart.de (F. Allgöwer).

http://dx.doi.org/10.1016/j.automatica.2017.06.036

dependent parameterization is proposed. In contrast to existing results, this parameterization explicitly accounts for the state dependence of the optimization it is applied for.

In more detail, we propose a parameterization which is a piecewise linear function in the system state and in the parameter, i.e., the new decision variable. We show how to determine the parameterization offline applying a tailored subspace clustering algorithm to training data and to refine the obtained preliminary result to guarantee feasibility of the final parameterized optimization. During the offline design phase, the complexity of the parameterization and the dimension of the corresponding online optimization variable can be selected. Increasing either of the values improves accuracy of the approximations of optimal input trajectories. This allows to achieve feasibility of the parameterized optimization problem for a given set of states and to improve control performance. Thus, trading off complexity versus control performance of the resulting MPC scheme is possible. In contrast to most existing parameterizations, the proposed parameterization explicitly depends on the state to be applied for and it is designed to guarantee feasibility of the parameterized optimization for a given set of states.

Summarizing, the presented method exploits – in the spirit of explicit MPC – state dependent information on optimal input trajectories encoded in the parameterization and combines it with online optimization based MPC. Thus, we call the method *semi-explicit MPC*.

Further results combining the explicit approach with online optimization are Zeilinger, Jones, and Morari (2011) where a simplified suboptimal explicit solution is used to warm-start an online optimization, and Jost and Mönnigmann (2013) where state-dependent information about inactive constraints is stored and exploited online to simplify the optimization. Beyond that, in Borrelli, Baotić, Pekar, and Stewart (2010) and Ferreau, Bock, and Diehl (2008) fast online optimization based MPC algorithms are proposed which are inspired by the theory behind explicit MPC. In contrast, in Lazar and Heemels (2003) the term *semi-explicit MPC* has been used in a slightly different meaning.

The paper at hand extends our previous work (Goebel & Allgöwer, 2013, 2014). A simplified and improved way of computing the parameterization is presented which considerably extends applicability of the method to systems with increased state space dimension. Comprehensive and more general theoretical considerations as well as larger numerical examples are presented and evaluated in detail. The examples highlight the effect of the mentioned tuning knobs as well as the general benefits of the proposed method. An extended version of the current paper is available (Goebel & Allgöwer, 2017a).

The remainder of this paper is organized as follows. The control problem considered and the proposed parameterization are introduced in Section 2. A tailored clustering algorithm to compute the parameterization is presented in Section 3. Section 4 contains the offline procedure to determine the parameterization, whereas in Section 5 the online procedure applying the parameterization is presented. We evaluate the algorithms and show numerical examples in Section 6 and conclude in Section 7.

2. Preliminaries and problem formulation

2.1. Control problem and underlying MPC scheme

Throughout this paper control of a linear time-invariant discrete-time system of the form

$$x^+ = Ax + Bu, \tag{1}$$

with constrained state $x \in \mathcal{X} \subseteq \mathbb{R}^n$ and constrained input $u \in \mathcal{U} \subseteq \mathbb{R}^m$ is considered, where $\mathcal{X}$ and $\mathcal{U}$ are the constraint sets. The control problem considered is to render the origin of this system asymptotically stable, i.e. to drive initial conditions in a given set $\mathcal{X}_f \subseteq \mathcal{X}$ towards the origin subject to a given performance criterion $\sum_{k=0}^{\infty} \ell(x_k, u_k) \to \min$, where $\ell : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is a given positive definite stage cost function satisfying $\ell(x, u) \geq \underline{\alpha}(\|x\|)$ for a $\mathcal{K}$-function $\underline{\alpha}$. The presented algorithm takes a classical MPC scheme as starting point based on the following optimization problem

$$\begin{aligned}\mathbf{P1}(x) : \min_{U \in \mathbb{R}^{mN}} &J(x, U)\\ \text{s.t. } &x_{k+1} = Ax_k + Bu_k,\\ &x_k \in \mathcal{X},\ u_k \in \mathcal{U}\ \forall k = 1, \ldots, N-1,\\ &x_0 = x,\ u_0 \in \mathcal{U},\ x_N \in \mathcal{X}_T,\end{aligned}$$

where $J(x, U) = \sum_{k=0}^{N-1} \ell(x_k, u_k) + F(x_N)$ and $U = (u_0^\top, \ldots, u_{N-1}^\top)^\top$ is the stacked vector of predicted inputs and $x_1, \ldots, x_N$ is the resulting predicted state trajectory, $F : \mathbb{R}^n \to \mathbb{R}$ and $\mathcal{X}_T \subset \mathbb{R}^n$ are a terminal cost function and a terminal set, respectively. Let $J^*(x)$ and $U^*(x)$ denote the optimal value and optimizer of $\mathbf{P1}(x)$. As is well known, in MPC this optimization is solved for the current system state, the first part of the obtained input trajectory is applied to the system and the process is repeated over a shifted horizon. Following standard assumptions are kept throughout the paper.

Assumption 1 (*Mayne, Rawlings, Rao, & Scokaert, 2000*)**.** Let the following hold

- The set $\mathcal{X}_T$ fulfills $\mathcal{X}_T \subset \mathcal{X}$, $\mathcal{X}_T$ is closed and $0 \in \mathcal{X}_T$.
- A local control law $\kappa(\cdot)$, defined on $\mathcal{X}_T$ is know with $\kappa(x) \in \mathcal{U}$ for all $x \in \mathcal{X}_T$ such that the following holds:
- For all $x \in \mathcal{X}_T$ it holds that $Ax + B\kappa(x) \in \mathcal{X}_T$.
- For $x \in \mathcal{X}_T$ the terminal cost fulfills $F(Ax + B\kappa(x)) - F(x) + \ell(x, \kappa(x)) \leq 0$.

Furthermore, we assume the following.

Assumption 2. Let $\mathcal{X}$, $\mathcal{U}$, $\mathcal{X}_T$ and $\mathcal{X}_f$ be polytopes with 0 in their respective interiors.

Based thereon, **P1** can be rewritten in a condensed form

$$\mathbf{P1}(x) : \qquad \min_U J(x, U), \quad \text{s.t. } GU \leq W + Ex$$

using suitable matrices G, W and E. We assume the set $\mathcal{X}_f$ to be given in the control problem. For the problem to be feasible $\mathcal{X}_f$ has to be a subset of the N step controllable set to the terminal set $\mathcal{X}_T$. We will assume this implicitly throughout and call $\mathcal{X}_f$ desired feasible set.

2.2. The parameterization

The proposed parameterization has the form

$$p(x, \tilde{U}) = p_i(x, \tilde{U}) = M_i\tilde{U} + \mathcal{K}_i x, \qquad \text{for } x \in \mathcal{D}_i,\ i \in \{1, \ldots, K\} \tag{2}$$

with sets $\mathcal{D}_i \subset \mathcal{X}$ and matrices $M_i \in \mathbb{R}^{mN\times q}$ and $\mathcal{K}_i \in \mathbb{R}^{mN\times n}$ where $q < mN$ holds. For the parameterization to be well-defined we require $\mathcal{D}_i \cap \mathcal{D}_j = \varnothing$ for $i \neq j$ and discuss in the sequel of the paper how this can be relaxed. Using this parameterization, K parameterized versions of the original optimization problem **P1** are formulated

$$\mathbf{P2}_i(x) : \quad \min_{\tilde{U}} J(x, p_i(x, \tilde{U})) \text{ s.t. } Gp_i(x, \tilde{U}) \leq W + Ex,$$

where we denote the optimal value by $J^*_{P_i}(x)$. The parameterization is required to be such that $\mathbf{P2}_i(x)$ is feasible for all $x \in \mathcal{D}_i$. In order to recover overall at least the feasible set $\mathcal{X}_f$ for the parameterized optimization, the union of the sets $\mathcal{D}_i$ has to contain $\mathcal{X}_f$, i.e., $\mathcal{X}_f \subseteq \cup_i\mathcal{D}_i$. A second goal is to have a parameterization such that $J^*_{P_i}(x)$ is close to $J^*(x)$, i.e. performance is maintained. The main challenge