

# Real-time suboptimal Model Predictive Control using a combination of Explicit MPC and Online Optimization

Melanie N. Zeilinger, Colin N. Jones, *Member, IEEE* and Manfred Morari, *Fellow, IEEE*

**Abstract**—Limits on the storage space or the computation time restrict the applicability of model predictive controllers (MPC) in many real problems. Currently available methods either compute the optimal controller online or derive an explicit control law. In this paper we introduce a new approach combining the two paradigms of explicit and online MPC to overcome their individual limitations. The algorithm computes a piecewise affine approximation of the optimal solution that is used to warm-start an active set linear programming procedure. A preprocessing method is introduced that provides hard real-time execution, stability and performance guarantees for the proposed controller. By choosing a combination of the quality of the approximation and the number of online active set iterations the presented procedure offers a tradeoff between the warm-start and online computational effort. We show how the problem of identifying the optimal combination for a given set of requirements on online computation time, storage and performance can be solved. Finally, we demonstrate the potential of the proposed warm-start procedure on numerical examples.

## I. INTRODUCTION

It is well-known that computation of a model predictive controller (MPC), under certain assumptions on the problem structure, amounts to the solution of a linear or a quadratic program at each sampling instant. Whereas classically, these optimization problems are solved online, it has been shown in recent years that the optimal solution to this type of problem is a piecewise affine function (PWA) defined over a polyhedral partition of the feasible states. This so-called explicit solution can then be used as a control look-up table online, enabling MPC to be used for fast sampled systems (see e.g. [1], [2], [3] and the references therein).

However, both the offline and online method have limitations. The main disadvantage of the online method is that it is in general only applicable for controlling slow processes. In the case of the explicit solution, the number of state-space regions over which the control law is defined, the so-called complexity of the partition, grows in the worst case exponentially due to the combinatorial nature of the problem [1]. This has given rise to an increasing interest in the development of new methods to either improve online optimization or to approximate explicit solutions (e.g. [4], [5], [6], [7], [8]). Depending on the particular problem properties and implementation restrictions, the user then has to decide for one of the two approaches.

This work aims at enlarging the possibilities to tradeoff solution properties through the combination of these two methods. It aims at an existent gap of problem sizes and types, which are either intractable for explicit MPC or the complexity of the explicit solution exceeds the storage capacity, but where an online MPC solution cannot meet the required online computation times. Specifically, ideas from approximation are combined with warm-start techniques. In this paper we use a PWA approximation of the optimal control law, which has been computed offline, to warm-start the online optimization. The optimization executes a finite number of active set iterations before returning a feasible, suboptimal control action, which is then applied to the system. The goal is to choose a good tradeoff between the complexity of the PWA approximation and the number of active set iterations required in order to satisfy system constraints in terms of online computation, storage and performance. Conditions are derived which guarantee that the suboptimal solution is closed-loop stabilizing, feasible and has a bounded performance deterioration. The provided analysis has the important benefit that it is not based on the optimal parametric solution to the MPC problem which might be prohibitively complex.

We also raise the question of an optimal combination of warm-start and online computational effort, with respect to certain requirements on the solution. Considering computation time and performance as two exemplifying requirements, this can be informally stated in the form of the following optimization problems:

1. Minimize online computation time while respecting a bound on the performance deterioration
2. Maximize the performance within available computation time

This paper extends the main ideas that have been introduced in [9] with both theoretical and numerical results. The outline of the paper is as follows: In Section IV we present the main idea of using an offline approximation to warm-start an active set linear programming procedure. An explicit representation of the proposed control law is derived in Section V. In Section VI a preprocessing method is introduced that allows the analysis of the properties of the control input that will be applied online. The question of an optimal combination between warm-start and online computation is discussed in Section VII. Finally, numerical examples illustrating the pro-

posed method and ideas are given in Section VIII and we briefly conclude in Section IX .

## II. NOTATION

If  $G \in \mathbb{R}^{m \times n_z}$  and  $\mathcal{I} \subseteq \{1, \dots, m\}$ , then  $G_{\mathcal{I}} \in \mathbb{R}^{|\mathcal{I}| \times n_z}$  is the matrix formed by the rows of  $G$  indexed by  $\mathcal{I}$ . If  $c \in \mathbb{R}^m$  is a vector then  $c_{\mathcal{I}}$  is the vector formed by the elements of  $c$  indexed by  $\mathcal{I}$ . A *polyhedron* is the intersection of a finite number of halfspaces  $P = \{x \mid Gx \leq f\}$  and a *polytope* is a bounded polyhedron.  $\mathcal{P}^N := \{P_j\}_{j=1}^N$  with  $N \in \mathbb{N}$  is called a polyhedral partition of  $\mathcal{X} \subseteq \mathbb{R}^{n_x}$  if all  $P_j$  are full-dimensional polyhedra,  $\cup_{j=1}^N P_j = \mathcal{X}$  and  $\text{int } P_i \cap \text{int } P_j = \emptyset \ \forall i \neq j$  and  $i, j \in \{1, \dots, N\}$ . A PWA function  $f(x)$  defined over the polyhedral partition  $\mathcal{P}^N$  is denoted as  $f(x) := C^j x + D^j$  if  $x \in P_j \ \forall P_j \in \mathcal{P}^N$ .

A function  $\gamma : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$  is of class  $\mathcal{K}$  if it is continuous, strictly increasing and  $\gamma(0) = 0$  [10].

## III. PRELIMINARIES

Consider the discrete-time linear system

$$x^+ = Ax + Bu, \quad (1)$$

where  $x \in \mathbb{R}^{n_x}$  is the state,  $u \in \mathbb{R}^{n_u}$  is the control input and  $A$  and  $B$  are the dynamic matrices. If system (1) is controlled by the control law  $u = \kappa(x)$  then the closed-loop system is denoted by

$$x^+ = Ax + B\kappa(x) = A_{\kappa}(x) \quad (2)$$

**Definition III.1 (Positively invariant (PI) set).** A set  $S \subseteq \mathbb{R}^n$  is a positively invariant (PI) set of system (2), if  $A_{\kappa}(x) \in S$  for all  $x \in S$ .

**Theorem III.2 (Asymptotic stability, [10]).** Let  $\mathcal{X}$  be a PI set for system (2) with  $0 \in \text{int } \mathcal{X}$  and let  $\underline{\alpha}(\cdot)$ ,  $\overline{\alpha}(\cdot)$ , and  $\beta(\cdot)$  be  $\mathcal{K}$ -class functions. If there exists a function  $V : \mathcal{X} \rightarrow \mathbb{R}_{\geq 0}$  with  $V(0) = 0$ , such that

$$\begin{aligned} \underline{\alpha}(\|x\|) &\leq V(x) \leq \overline{\alpha}(\|x\|), \ \forall x \in \mathcal{X}, \\ V(x^+) - V(x) &\leq -\beta(\|x\|), \ \forall x \in \mathcal{X}, \end{aligned}$$

then the equilibrium at  $x = 0$  is asymptotically stable in  $\mathcal{X}$ .

In the following, (asymptotic) stability of a system is used to mean that the system has an (asymptotically) stable equilibrium at the origin.

### Model Predictive Control

In this paper, we consider the following formulation of the MPC problem:

$$J^*(x) = \min_{\{u_0, \dots, u_{N-1}\}} V_f(x_N) + \sum_{i=0}^{N-1} l(x_i, u_i) \quad (4a)$$

$$\begin{aligned} \text{subject to } x_{i+1} &= Ax_i + Bu_i, \ i = 0, \dots, N-1, \\ (x_i, u_i) &\in \mathbb{X} \times \mathbb{U}, \ i = 0, \dots, N-1, \\ x_N &\in \mathcal{X}_f, \\ x_0 &= x, \end{aligned}$$

(4b)

where  $N$  is the control horizon,  $\mathbb{X}, \mathbb{U}$  and  $\mathcal{X}_f$  are polytopic constraints on the states and inputs and the stage cost is defined as  $l(x_i, u_i) := \|Qx_i\|_p + \|Ru_i\|_p$  with  $p \in \{1, \infty\}$ , where  $Q$  and  $R$  are positive definite weights penalizing the states and inputs.  $V_f$  is the terminal penalty function and  $\mathcal{X}_f \subseteq \mathbb{X}$  is a compact terminal target set. The MPC problem implicitly defines the set of feasible initial states  $\mathcal{X}$ .

**Definition III.3.** [Feasibility of a control law] A control law  $u(x)$  is called feasible for  $x$  if the input sequence  $u_0(x_0), \dots, u_{N-1}(x_{N-1})$  and the corresponding state sequence  $x_0, \dots, x_N$ , with  $x_0 = x$  and  $x_{i+1} = Ax_i + Bu_i(x_i)$ ,  $i = 1, \dots, N$ , generated by applying the control law over  $N$  time steps, satisfy all state and input constraints in (4b).

If the norm  $p$  is taken to be the 1- or the  $\infty$ -norm, we can write (4) as a parametric Linear Program (pLP) of the form:

$$\begin{aligned} z^*(x) &= \underset{z}{\text{argmin}} \ c^T z \\ &\text{subject to } G_{\mathcal{I}} z \leq f_{\mathcal{I}}, \\ &\quad G_{\mathcal{E}} z = F_{\mathcal{E}} x, \end{aligned} \quad (5)$$

where  $z \in \mathbb{R}^{n_z}$  is a vector containing the sequence of control inputs  $\{u_0, \dots, u_{N-1}\}$ , states  $\{x_1, \dots, x_N\}$  and appropriate slack variables introduced to rewrite the state and input penalties as a linear cost. The current state  $x \in \mathbb{R}^{n_x}$  is the parameter,  $G \in \mathbb{R}^{m \times n_z}$ ,  $\mathcal{E} \subset \{1, \dots, m\}$  and  $\mathcal{I} = \{1, \dots, m\} \setminus \mathcal{E}$ . For a description of how the optimal control problem in (4) is transformed into the pLP (5), i.e. how the state and input constraints, the dynamics and the cost are converted into  $G_{\mathcal{I}}, G_{\mathcal{E}}, f_{\mathcal{I}}, F_{\mathcal{E}}$  and  $c$ , see e.g. [3]. (Note that for simplicity we use the same indexing for  $f$  and  $F$  as for  $G$  although we distinguish the vector  $f$  from the matrix  $F$  in order to account for the different dimension).

**Definition III.4.** Let  $z$  be the vector of decision variables in (5). We define  $\pi : \mathbb{R}^{n_z} \rightarrow \mathbb{R}^{n_u}$  to be a linear mapping that returns the first control input  $u_0$  contained as a component in  $z$ .

By solving the pLP (5) the optimal solution  $z^*(x)$  can be computed for each feasible value of the state. The implicit optimal MPC control law is then given in a receding horizon fashion by  $u^*(x) = \pi(z^*(x))$ .

**Assumption III.5.** In the following it is assumed that  $V_f(\cdot)$  is a Lyapunov function in  $\mathcal{X}_f$  and  $\mathcal{X}_f$  is a PI set under the control law  $\kappa_f(x)$ , given by the following conditions:

$$\begin{aligned} \text{A1: } & A + B\kappa_f(x) \in \mathcal{X}_f \text{ and } \kappa_f(x) \in \mathbb{U} \ \forall x \in \mathcal{X}_f \\ \text{A2: } & V_f((Ax + B\kappa_f(x)) - V_f(x) \leq -l(x, \kappa_f(x)) \ \forall x \in \mathcal{X}_f. \end{aligned}$$

**Remark III.6.** Note that in this work the 1- or  $\infty$ -norm is chosen to penalize the states and inputs in the cost function (4a) instead of the commonly used 2-norm, as the resulting pLP (5) allows for an exact analysis of the control law obtained by the proposed procedure, as will be shown in Section V.

**Theorem III.7 (Solution to the MPC problem, [1], [11]).** Consider Problem (4) fulfilling Assumption III.5. The optimal MPC control law  $u^*(x)$  is a continuous PWA function and the optimal value function  $J^*(x)$  is a continuous PWA Lyapunov function of the state  $x$  defined over a polyhedral partition of the feasible set  $\mathcal{X}$ . The closed-loop system  $x^+ = Ax + Bu^*(x)$  is asymptotically stable with region of attraction  $\mathcal{X}$ .

**Remark III.8.** Note that even in the case of dual degeneracy, there always exists a polyhedral partition such that the optimizer  $z^*(x)$  in (5) and thereby the optimal control law  $u^*(x)$  is unique and continuous in  $\mathcal{X}$  [1]. Continuity of  $u^*(x)$  is however not a required assumption for the proposed method.

#### A. Approximation of the MPC problem

We first define an approximation of the MPC problem (4) and some useful properties that will be used in Section VI in order to give guarantees on the control law proposed in this paper. Let  $z^*(x)$  be the optimizer of the optimal control problem (5) and  $J^*(x) = c^T z^*(x)$  be the corresponding optimal cost and a Lyapunov function.

**Definition III.9 (Approximation error).** Let  $\tilde{z}(x)$  be a sub-optimal solution to (5). A function  $\tilde{u}(x) = \pi(\tilde{z}(x))$  is called an *approximate control law* for (4) if  $\tilde{u}(x)$  is feasible for all  $x \in \mathcal{X}$ . The approximate control law  $\tilde{u}(x)$  is an  $\epsilon$ -approximation if for all  $x \in \mathcal{X}$  the condition  $\tilde{J}(x) - J^*(x) \leq \epsilon$  is satisfied, where  $\tilde{J}(x) := c^T \tilde{z}(x)$  and  $\epsilon$  is the smallest value satisfying this inequality.

**Remark III.10.** Note that the approximation error  $\epsilon$  denotes the worst accuracy over the feasible set  $\mathcal{X}$  and is hence uniform while the accuracy of the approximate cost  $\tilde{J}(x)$  is varying with  $x$ .

A standard condition to test if an approximate solution is *stabilizing* is given by the following theorem:

**Theorem III.11 (Stability under  $\tilde{u}(x)$ , [12], [13]).**

Let  $\tilde{u}(x)$  be an approximate control law of (4). The approximate value function  $\tilde{J}(x)$  is a Lyapunov function if  $\tilde{J}(x) - J^*(x) < l(x, \tilde{u}(x))$  for all  $x \in \mathcal{X}$ , where  $l: \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}$  is the stage cost (4a) and the closed-loop system  $x^+ = Ax + B\tilde{u}(x)$  is asymptotically stable with region of attraction  $\mathcal{X}$ .

## IV. PROPOSED CONTROL LAW

In order to overcome the limitations of the offline and online methods mentioned in the introduction, several authors recently proposed new approaches to speed up online optimization or to reduce the complexity of explicit solutions by means of approximation. The authors in [14], [8] for example utilize new developments in interior-point methods and show how these can be applied to efficiently solve the optimal control problem. Another paradigm that is frequently applied to improve online optimization is warm-starting

(see e.g. [5], [7]). In explicit MPC, approximation methods have been proposed that either modify the original MPC problem (4), retrieve a suboptimal solution or postprocess the computed optimal solution, with the goal of reducing the complexity of the explicit controller, cf. e.g. [15], [6], [12], [16]. All of the cited explicit approximation methods provide a feasible control law and allow verification of closed-loop stability by means of Theorem III.11 for some minimally required complexity. All currently available online MPC methods however lack the possibility of giving guarantees on the suboptimal solution, e.g. closed-loop stability or feasibility, in a real-time setting.

The strategy proposed in this paper combines the idea of offline approximation with warm-start techniques from online optimization with the goal of providing hard real-time, stability and performance guarantees. Warm-start techniques aim at identifying advanced starting points for the optimization in order to reduce the number of iterations required to reach the optimum. They often try to make use of the information gained during the solution of one problem to solve the next one in a sequence of closely related problems. When solving MPC problems in a receding horizon fashion an LP is computed for every measured state. However in practice, the optimal control input from a previous measurement might be an infeasible solution to (4) at the current instance, due to disturbances acting on the system. We therefore propose a warm-start strategy that utilizes a PWA approximate control law of (4) to provide a good and feasible starting point. The pre-knowledge of the initial solution for all feasible values of  $x$  allows us to analyze the solution obtained by the online optimization.

The following two parameters are used to classify the warm-start solution: the complexity  $N_P$  (number of regions) and its approximation error  $\epsilon$ , given by Definition III.9.

**Definition IV.1 (Warm-start Solution).** A function  $\nu(x, N_P)$  is called a warm-start solution of (4) if  $\pi(\nu(x, N_P))$  is a feasible, PWA approximate control law of (5) defined over  $N_P$  polytopic regions.

**Lemma IV.2 (Convergence of  $\nu(\cdot, \cdot)$ ).** There exists a function  $\nu(x, N_{P_{opt}})$  of finite complexity  $N_{P_{opt}} \in \mathbb{N}$ , such that  $z^*(x) = \nu(x, N_{P_{opt}})$  for all  $x \in \mathcal{X}$ , where  $z^*(x)$  is the optimal solution to (5).

*Proof:* Result follows directly from the fact that the approximation is PWA and Theorem III.7. ■

By means of the parameter  $N_P$  requirements can be set on the complexity of the warm-start solution  $\nu(\cdot, N_P)$  that is computed and stored in an offline preprocessing step.

**Remark IV.3.** As was shown in [12] there exists an analytical relation between the approximation error  $\epsilon$  and the complexity  $N_P$  of a PWA offline approximation. Requirements on the approximation error can therefore also be imposed using the parameter  $N_P$ .

In the online control procedure, the warm-start solution is evaluated for the measured state and used to initialize the online optimization. A standard active set method (ASM) is applied to compute the control action since it allows us to take advantage of a feasible starting point that is not necessarily located at a vertex. Active set methods generate a sequence of feasible iterates that converge to the optimal solution. At each iterate  $z$ , the active set is given by  $\mathcal{I}_A = \mathcal{E} \cup \{i \in \mathcal{I} | G_i z = f_i\}$ . In an active set iteration, a subset of the active set is chosen as the working set  $W \subseteq \mathcal{I}_A$  using standard heuristics. From the current iterate  $z$ , the maximal step in the steepest descent direction is then computed, which is the direction that minimizes the objective in (5) while keeping the constraints in  $W$  active, defining the next iterate. See e.g. [17], [18] for more details on active set methods.

**Assumption IV.4 (Non-degeneracy).** It is assumed that the active set is non-degenerate, i.e. the active constraints are linearly independent.

Note that it is possible to extend the approach to degenerate cases by using one of the standard approaches for anti-cycling (e.g. [19], [20], [21]) or lexicographic perturbation (e.g. [22]) in active set methods.

Whereas in standard active set methods iterations are performed until the optimality conditions are met, the online optimization procedure is stopped early after exactly  $K$  active set iterations and the current suboptimal control input is applied to the system.

**Definition IV.5 (Warm-start optimization).** Let  $z$  be a feasible solution of (5) for the parameter  $x$ . We define  $\kappa(z, K)$  to be the decision variable of (5) after  $K$  iterations of the linear programming active set method starting from the feasible point  $z$ .

**Definition IV.6 (Proposed control law).** Let  $\nu(\cdot, N_P)$  be a warm-start solution to (5) and  $\kappa(\cdot, K)$  be as defined in IV.5. The proposed control law is

$$u_{\text{on}}(x) = \pi(\kappa(\nu(x, N_P), K)), \text{ for } x \in \mathcal{X} . \quad (6)$$

**Lemma IV.7 (Properties of  $\kappa(\cdot, \cdot)$ ).** *The proposed control law (6) is feasible for all  $x \in \mathcal{X}$ , and for each  $N_P$  there exists a finite  $K_{\text{opt}} \in \mathbb{N}$ , such that  $u^*(x) = \pi(\kappa(\nu(x, N_P), K_{\text{opt}}))$ .*

*Proof:* Feasibility is ensured by the procedure of the ASM and the fact that  $\nu(x, N_P)$  is feasible for all  $x \in \mathcal{X}$ . The existence of a finite  $K_{\text{opt}}$  is guaranteed by the convergence of the ASM in finite time [18]. ■

The warm-start linear programming procedure for a fixed complexity  $N_P$  and number of iterations  $K$  is summarized in Algorithm 1. In Section VI an offline analysis is introduced providing guarantees for the proposed control law  $u_{\text{on}}(x)$  in (6) to be closed-loop stabilizing, feasible and to have a bounded performance deterioration compared to the optimal solution.

---

**Algorithm 1** Warm-start linear programming procedure

---

Input: warm-start solution  $\nu(\cdot, N_P)$  and current measured state

$x_{\text{meas}}$

Output: approximate control input  $u_{\text{on}}$

- 1: run point location algorithm:  $z = \nu(x_{\text{meas}}, N_P)$  [23], [24], [25], [26]
  - 2: **for**  $k = 1, \dots, K$  **do**
  - 3:   perform an active set iteration [17], [18]
  - 4:   update iterate  $z$
  - 5: **end for**
  - 6:  $u_{\text{on}} = \pi(z)$
- 

The above described procedure of using an approximation to warm-start an online optimization offers the possibility to decide on the complexity and approximation error of the warm-start solution  $\nu(\cdot, \cdot)$ . A tradeoff can be made between the degree of approximation realized by the warm-start and the effort expended in online optimization. The goal is to identify a good if not optimal combination that achieves the best properties of the online control input applied to the system for given requirements on the approximation error and/or limitations on the online computation time or storage.

The proposed procedure and algorithms are detailed in the following sections.

## V. PARAMETRIC CALCULATION OF THE ONLINE CONTROL LAW

Our goal is to give guarantees on the proposed suboptimal control law (6). Apart from feasibility, which is guaranteed by Lemma IV.7, we want to ensure stability and a certain bound on the approximation error. In order to analyze the solution properties, we need an explicit representation of the approximate control input  $\kappa(\cdot, \cdot)$  for the entire feasible set  $\mathcal{X}$ . We will show that starting from the warm-start solution, the iterative path taken by the active set method is a function of  $x$ , defined over a polyhedral subdivision of  $\mathcal{X}$ .

**Remark V.1 (Offline Complexity).** Note that the complexity (number of regions) of this subdivision does not affect the actual optimization carried out online, since the parametric solution is only used for offline analysis.

The operations performed during an active set iteration can be formulated as functions of the parameter  $x$ . Let  $z^k(x)$  be the iterate and  $W$  the working set at the  $k$ -th iteration step. A search direction  $\Delta z$  from  $z^k(x)$  is computed that maintains activity of the constraints in  $W$ . In this work, we use the steepest descent direction corresponding to the projection of  $c$  onto the subspace defined by the constraints in the working set given by:

$$\Delta z = \underset{\Delta z}{\text{argmin}} \{ c^T \Delta z \mid G_W \Delta z = 0, \Delta z^T \Delta z = \delta \} \quad (7)$$

where  $\delta \in \mathbb{R}_{\geq 0}$  is a scaling parameter.

The steps performed during one active set iteration are outlined in the following (cf. e.g. [17], [18]):

1. Compute step direction by solving the KKT conditions of (7)

$$\begin{bmatrix} I & G_W^T \\ G_W & 0 \end{bmatrix} \begin{bmatrix} \Delta z \\ \lambda \end{bmatrix} = \begin{bmatrix} -c \\ 0 \end{bmatrix}, \quad (8)$$

where  $I$  denotes the identity matrix of size  $n_z \times n_z$  and  $\lambda$  are the Lagrange multipliers.

If  $\Delta z = 0$  and  $\lambda \geq 0$ : Stop with optimal solution  $z^*(x) = z^k(x)$ .

If  $\Delta z = 0$  and at least one  $\lambda_i < 0$ : Remove constraint  $a$  from  $W$ , i.e.  $W = W \setminus \{a\}$ , where  $a = \operatorname{argmin}_i \{\lambda_i \mid i \in W \cap \mathcal{I}\}$ . Recompute (8).

2. Compute maximum step length for which all constraints are satisfied:

$$\tau(x) = \min_{i \in \mathcal{I} \setminus W} \left\{ \frac{f_i - G_i z^k(x)}{G_i \Delta z} \mid G_i \Delta z > 0 \right\}. \quad (9)$$

3. Update  $W$ :  $W = W \cup e$ , where the new active constraint  $e$  is the optimizer of (9).
4. Update iterate:

$$z^{k+1}(x) = z^k(x) + \tau(x) \Delta z. \quad (10)$$

**Remark V.2 (ASM).** A similar parametric formulation can also be obtained using a different search direction or any of the other standard implementations found in the literature; the simplex-type algorithm in [27] for instance. For simplicity, the strategy of adding or removing at most one inequality constraint in each iteration to or from the working set is considered. Note that  $\mathcal{E} \subseteq W$ , since the equality constraints  $\mathcal{E}$  always have to remain in the working set.

**Theorem V.3.** *At every iteration  $k \in \{1, \dots, K\}$ , the step length  $\tau(x)$  in (9) and the current iterate  $z^k(x)$  are PWA functions of  $x$  defined over a polyhedral partition  $\mathcal{P}^{N_k}$  of the feasible set  $\mathcal{X}$ .*

*Proof:* Assume that the statement is true for  $k$  and that  $z^k(x) := C^j x + D^j$  if  $x \in G_j \forall G_j \in \mathcal{P}^{N_k}$ . For one region  $\mathcal{P}_j \in \mathcal{P}^{N_k}$  the line search (9) determining the next constraint to become active is given by the pLP  $\tau(x) = \min_i \{\alpha_i(x)\}$ , with  $\alpha_i(x) = \frac{-G_i C^j x + f_i - A_i D^j}{A_i \Delta z}$ ,  $i \in \mathcal{I}_{cand}$  and  $\mathcal{I}_{cand} = \{i \in \mathcal{I} \setminus \mathcal{I}_A \mid G_i \Delta z > 0\}$ . This shows that  $\tau(x)$  is a PWA function of  $x$ , since the optimal cost of a pLP is PWA [28], [1]. With  $\tau(x)$  and  $z^k(x)$  being PWA, the  $k+1$ -th iterate (10) is as well a PWA function of  $x$ . Since the initial solution is  $z^0(x) = \nu(x, N_P)$ , the statement is true for  $k=1$  and hence for all  $k \in \{1, \dots, K\}$ . ■

With  $z^k(x)$  being a PWA function, equation (9) results in a parametric LP. The approximate control law at iteration  $K$  is obtained by solving (8) and (9) iteratively for all  $k \in \{1, \dots, K\}$ . With each iteration, the parametric solution of (9) causes a further refinement of the polyhedral partition  $\mathcal{P}^{N_k}$ .

**Remark V.4.** Note that the computation of the PWA function  $\tau(x)$  in (9) can be reduced to redundancy elimination of the halfspaces  $z(x) \geq \frac{f_i - G_i z^k(x)}{G_i \Delta z}$ . All halfspaces  $j$  with  $\frac{f_j - G_j z^k(x)}{G_j \Delta z} \geq \frac{f_i - G_i z^k(x)}{G_i \Delta z} \forall i \neq j, i \in \mathcal{I} \setminus W, G_i \Delta z > 0$  are

redundant and can be disregarded, the irredundant halfspaces form the PWA parametric solution of (9). Redundancy elimination problems can be easily solved by computing one linear program per constraint.

**Corollary V.5.** *The proposed control law  $u_{on}(x)$  is a PWA function of  $x$  defined over a polyhedral partition  $\mathcal{P}^{N_K}$  of the feasible set  $\mathcal{X}$ .*

Theorem V.3 enables us to compute an explicit PWA representation of the approximate control law  $\kappa(\nu(x, N_P), K)$  for a fixed complexity  $N_P$  and number of iterations  $K$  using Algorithm 2.

---

#### Algorithm 2 Offline Analysis

---

Input: warm-start solution

$\nu(x, N_P) = \nu_j(x)$  if  $x \in P_j \forall P_j \in \mathcal{P}^{N_P}$

Output: Explicit representation of the proposed control law  $u_{on}(x)$

```

1: initialize stack  $S = \emptyset$ 
2: for all  $P_j \in \mathcal{P}^{N_P}$  do push  $(\nu_j(x), P_j)$  onto  $S$ 
3: end for
4: for all  $k \in \{1, \dots, K\}$  do           [ $K$  active set iterations]
5:   initialize stack  $\widehat{S} = \emptyset$ 
6:   while  $S \neq \emptyset$  do           [subdivide each region]
7:     pop  $(z_P(x), P)$  from  $S$ 
8:     compute  $\Delta z$ 
9:     compute  $a_j, b_j$  for  $z_P(x)$ ,  $x \in P$ ,           (8),(9)
       such that  $\tau(x) = a_j x + b_j$  if  $x \in R_j \forall R_j \in \mathcal{R}$ 
       [ $\mathcal{R}$  is a polyhedral partition of  $P$ ]
10:    for all  $R_j \in \mathcal{R}$  do
11:      push  $(z_P(x) + (a_j x + b_j) \Delta z, R_j)$  onto  $\widehat{S}$ 
12:    end for
13:  end while
14:   $z^k(x) := z_j(x)$  if  $x \in P_j \forall (z_j(x), P_j) \in \widehat{S}$ 
15:   $\mathcal{P}^{N_k} := \{P_j\}_{j=1}^{N_k}$  [ $P_j$  form polyhedral partition  $\mathcal{P}^{N_k}$ ]
16:   $S = \widehat{S}$ 
17: end for
18:  $u_{on}(x) = \pi(z^K(x))$ 

```

---

**Remark V.6.** The parametric calculation of the proposed control law in (6) using Algorithm 2 requires the iterative solution of a parametric program. Although the computational complexity depends on the properties of the considered control problem it grows significantly with the problem size. This currently limits the execution of the offline analysis in practice, where in our experience a system of order 8 was the biggest problem to be computed. Note that the warm-start linear programming procedure is however not affected by this complexity and could still be applied without the rigorous guarantees provided by the analysis.

## VI. ANALYSIS OF THE PROPOSED CONTROL LAW ACCORDING TO STABILITY, SUBOPTIMALITY, STORAGE SPACE AND COMPUTATION TIME

We introduce a preprocessing analysis that investigates the following properties of the approximate control law  $u_{on}(x)$  in

(6): stability, approximation error, storage space and online computation time.

**Lemma VI.1 (Approximation error of  $u_{on}(x)$ ).** *If  $\kappa(\cdot, K) := C^j x + D^j$  if  $x \in P_j \forall P_j \in \mathcal{P}^{N_K}$  and  $u_{on}(x) = \pi(\kappa(\cdot, K))$  is the proposed control law in (6), then the approximation error defined in Definition III.9 is given by*

$$\epsilon_K = \max_{j \in \{1, \dots, N_K\}} d_j, \text{ with} \quad (11)$$

$$d_j = \max_{z, x} c^T(C^j x + D^j) - c^T z \quad (12)$$

s.t.  $G_{\mathcal{I}} z \leq f_{\mathcal{I}}, G_{\mathcal{E}} z = F_{\mathcal{E}} x, x \in P_j$ .

*Proof:* We first proof that equation (12) computes the largest distance between the approximate and the optimal cost. We take  $x = \bar{x}$  fixed and show by contradiction that equation (12) computes the distance to the optimal cost at  $\bar{x}$ . Assume  $z^*(\bar{x})$  is the optimal solution to (5) and  $z^o$  the optimal solution to (12) for  $x = \bar{x}$ . If  $z^o(\bar{x})$  is not the optimal solution to (5), i.e.  $c^T z^o \geq c^T z^*(\bar{x})$ , then it follows that  $c^T(C^j \bar{x} + D^j) - c^T z^o \leq c^T(C^j \bar{x} + D^j) - c^T z^*(\bar{x})$ . Therefore  $z^o$  cannot be the optimal solution to (12) at  $\bar{x}$  which proves that  $z^o$  also has to be the optimal solution to (5) at the state  $\bar{x}$ . Now letting  $x$  vary and simultaneously taking the maximum over all  $x \in P_j$  gives the worst case distance in  $P_j$ . Finally, the biggest error over all  $x \in \mathcal{X}$  and hence over all the regions in  $\mathcal{P}^{N_K}$  is the smallest  $\epsilon$  that fulfills the condition in Definition III.9 for all  $x \in \mathcal{X}$ . ■

Stability can be easily tested using the conditions of Theorem III.11:

**Theorem VI.2 (Stability of  $u_{on}(x)$ ).** *If  $\kappa(\cdot, K) := C^j x + D^j$  if  $x \in P_j \forall P_j \in \mathcal{P}^{N_K}$  and  $u_{on}(x) = \pi(\kappa(\cdot, K))$  is the proposed control law in (6) at iteration  $K$ , then the closed-loop system  $x_{i+1} = A_d x_i + B_d u_{on}(x_i)$  is asymptotically stable with region of attraction  $\mathcal{X}$  if*

$$\max_{j \in \{1, \dots, N_K\}} s_j < 0, \text{ with} \quad (13)$$

$$s_j = \max_{z, x} c^T(C^j x + D^j) - c^T z - l(x, u_{on}(x))$$

s.t.  $G_{\mathcal{I}} z \leq f_{\mathcal{I}}, G_{\mathcal{E}} z = F_{\mathcal{E}} x, x \in P_j$ .

*Proof:* Condition (13) follows directly from Theorem III.11. If the difference between the approximate cost  $\tilde{J}(x) := c^T(C^j x + D^j)$  and the optimal cost  $J^*(x) := c^T u$  is less than the stage cost  $l(x, u_{on}(x))$  for all  $x \in P_j$  and for all the regions  $P_j$  in  $\mathcal{P}^{N_K}$ , then the condition in Theorem III.11 is fulfilled for all  $x \in \mathcal{X}$ . ■

Whereas condition (13) is sufficient to prove stability of the proposed control law at iteration  $K$ , it does not guarantee stability of the control laws at iterations  $k \geq K$ . However, by using a more conservative condition instead of (13) we can modify Theorem VI.2 in order to ensure stability not only at iteration  $K$  but also at all subsequent active set iterations.

**Corollary VI.3.** *Let  $\kappa(\cdot, K) := C^j x + D^j$  if  $x \in P_j \forall P_j \in \mathcal{P}^{N_K}$  and  $u_{on}(x) = \pi(\kappa(\cdot, K))$  be the proposed control law in (6) at iteration  $K$ . The closed-loop system  $x_{i+1} = A_d x_i +$*

*$B_d u_{on}(x_i)$  is asymptotically stable with region of attraction  $\mathcal{X}$  for all control laws  $u_{on}(x) = \pi(\kappa(\cdot, k))$  for  $k \geq K$  if*

$$\max_{j \in \{1, \dots, N_K\}} s_j < 0, \text{ with} \quad (14)$$

$$s_j = \max_{z, x} c^T(C^j x + D^j) - c^T z - l_{min}(x)$$

s.t.  $G_{\mathcal{I}} z \leq f_{\mathcal{I}}, G_{\mathcal{E}} z = F_{\mathcal{E}} x, x \in P_j$ ,

where  $l_{min}(x) = \min_u \{l(x, u) \mid u \in \mathbb{U}\}$ .

*Proof:* Follows from Theorem VI.2 and the fact that  $l_{min}(x) \leq l(x, u_{on}(x))$ . ■

**Remark VI.4.** Note that if the origin is contained in  $\mathbb{U}$ , then  $l_{min}(x) = \|Qx\|_p$ .

**Remark VI.5 (Stability/Performance test).** Note that equations (13) and (14) compute the maximum distance to the optimal cost plus the stage cost without computation of the parametric optimal solution to problem (4), which can be easily shown following the first part of the proof of Lemma VI.1. Using Theorem VI.2 or Corollary VI.3 stability of the proposed control law (6) can hence be tested without the need to compute the optimal and potentially complex parametric solution to problem (4).

Storage space is determined by the complexity (number of regions)  $N_P$  of the warm-start solution since only the warm-start has to be stored.

Online computation time will be estimated in terms of floating point operations (flops) for the calculations that have to be performed online. First the region of the current state is identified using a point location algorithm (e.g. [23], [24], [25], [26]), then the corresponding affine control law is evaluated and finally  $K$  online iterations are executed.

**Remark VI.6 (Sparsity of the MPC problem).** In the case of the MPC problem (4), the matrices  $G$  and  $F$  in (5) have a special structure, resulting from the particular problem setup. The matrices are extremely sparse and by reordering can be shown to be in fact block diagonal or banded. We can exploit the banded structure of the matrices to solve equations (8) and (9), achieving significant computational savings [29].

**Theorem VI.7 (Flop count).** *If the input and state dimensions are  $n_u$  and  $n_x$  respectively, the number of constraints on each state-input pair is  $m_c$ ,  $N$  is the horizon in (4a) and the number of slack variables introduced for each state-input pair to write problem (4) as pLP (5) is  $n_s$  then the number of flops to calculate the control action  $u_{on}(x_{meas})$  for a measured state  $x_{meas}$  can be bounded by:*

$$f_{on} = N_P f_{ws} + f_{eval} + K f_{ASM}, \quad (15)$$

$$\text{where } f_{eval} = 2n_x(Nn_x + n_u + n_s),$$

$$f_{ASM} = N(n_x(82n_x + 26m_c + 68n_s + 11) + n_u(40n_u + 116n_x + 56n_s + 20m_c + 9) + n_s(12m_c + 3) - 3m_c)$$

$f_{ws}$  denotes the flop number for point location per region,  $f_{eval}$  the flops for evaluation of a PWA warm-start solution and  $f_{ASM}$  the flops per active set iteration.

*Proof:* The flop counts for active set iterations use the fact that the matrices in (5) have banded structure. An  $LU$  factorization and  $LU$  updates as described in [29] and [30] is considered, where  $L$  and  $U$  have half-bandwidth  $5n_x + 4n_u + m_c - 1$ . The worst case is taken in form of the maximal number of active constraints. The flop counts for calculation of the steepest descent direction, step length and the next iterate follow directly from the equations (8), (9) and (10). ■

**Remark VI.8.** The number of flops for point location  $f_{ws}$  in Theorem VI.7 depends on the particular point location algorithm used in Algorithm 1, e.g. the one described in [23] requires  $2n_x$  flops.

The worst-case estimates for the properties of the proposed control law  $\kappa(\cdot, K)$  in terms of stability, approximation error, storage space and online computation time can be calculated. For fixed parameters  $N_P$  and  $K$  this allows us to give guarantees on the properties of the control law that is applied to the system.

**Remark VI.9.** In the case that one only wants to prove stability for a fixed  $K$ , the analysis can be stopped early after stability is guaranteed for a certain iteration  $K_s \leq K$  using Corollary VI.3. In the actual warm-start procedure, the online optimization can then be carried out to  $K$  iterations while still guaranteeing stability of the proposed control law.

**Remark VI.10 (Computational effort).** Note that the computational effort for Algorithm 2 can be reduced by applying ideas from tree search. A node in the tree represents a single region and each depth level corresponds to an active set iteration. Using depth-first-search, for instance, an error bound for the  $K$ -th active set iteration can be derived without calculating the full parametric solution. Since in this case there are regions for which we infer the solution at the  $K$ -th level from an earlier iteration, we have to employ the more conservative stability test using Corollary VI.3.

## VII. OPTIMIZATION OVER THE PARAMETERS

In this section we will now try to optimize over the parameters that determine the applied control input: the complexity  $N_P$  of the warm-start solution and the number of iterations  $K$ . The choice of the warm-start solution  $\nu(\cdot, N_P)$  determines the computational effort for point location  $f_{ws}$  in (15) on the one hand and the quality of the warm-start on the other and therefore the number of active set iterations. This offers the possibility to trade off the amount of online computation time spent on the warm-start with that spent on online optimization. The challenge is to identify an optimal combination of explicit approximation and online optimization that achieves the best properties of the applied control input.

We consider the problem of optimizing the online time to compute a control law that guarantees stability, a certain performance bound  $\epsilon_{max}$  and a limit on the storage space  $N_{P,max}$ . Computation time is again measured in the form of flops (15), resulting in the following optimization problem

$$\begin{aligned} F_{min} &= \min_{N_P, K} N_P f_{ws} + K f_{ASM} & (16) \\ \text{subject to} & \quad \epsilon_K & \leq \epsilon_{max} , \\ & \quad N_P & \leq N_{P,max} \\ & & (13)/(14) . \end{aligned}$$

While the exact solution of this optimization problem is not possible with currently available methods, it can be used as a quality measure to evaluate different combination possibilities. We demonstrate in the following how the identification of a good combination can be pursued for a particular offline approximation method.

There are several approximation methods that can be used to create a PWA warm-start solution (e.g. [16], [6], [4]). In this work the method introduced in [12] was chosen, which is based on the beneath/beyond (B/B) algorithm, a common approach for convex hull calculation [31], [32]. An approximation  $\tilde{J}(x)$  of  $J^*(x)$  in (5) is constructed by computing the convex hull of a subset of vertices of the epigraph of  $J^*(x)$ . The approximation can be iteratively improved by adding one vertex at a time and updating the convex hull. When all vertices of the polytope are included, the optimal solution of (5) is reached. The approximate control law is obtained by interpolating between the optimal control inputs at the vertices. The main advantage of the beneath/beyond method is that it is an incremental approach, allowing one to set requirements on either the complexity or the error of the approximation  $\tilde{J}(x)$ . In addition, it is based on an implicit rather than on an explicit representation of the optimal solution and is hence not dependent on the computability of the optimal parametric solution to pLP (5).

**Theorem VII.1 (B/B warm-start solution [12]).** *Given a parameter  $N_P \in \mathbb{N}$  the B/B method returns a feasible PWA approximation  $\nu(\cdot, N_P)$  of (5).*

**Remark VII.2.** The approximate control law generated by the B/B method is not necessarily defined over the entire feasible set. For small dimensional problems the B/B algorithm can be initialized with all the vertices of the boundary of  $\mathcal{X}$ , which can be computed by projection of the feasible set defined by the constraints in (5) onto the  $x$ -space, in order to provide a feasible control law for all  $x \in \mathcal{X}$ . For higher dimensional problems the method can be extended so that one can reduce the complexity of the approximation by considering only a subset of  $\mathcal{X}$  [12].

The error of a B/B approximation is related to its complexity by the following Lemma VII.3.

**Lemma VII.3 (Complexity/Approx. Error, [33], [34]).**

*Let  $\nu(\cdot, N_P)$  be an approximation to (5) generated by the B/B method,  $\epsilon_{BB}$  the approximation error as defined in*

**Definition III.9** and  $N_P$  its complexity. For every  $\epsilon_{BB}$  there exists an  $N_{BB} \in \mathbb{N}$  such that the approximation error of  $\nu(\cdot, N_{BB})$  is less than  $\epsilon_{BB}$ .

**Remark VII.4.** Note that whereas the approximation error of a B/B approximation is monotonically decreasing with every B/B improvement, the complexity might not be monotonic (see [12]).

Using a B/B approach, problem (16) is a function of only the complexity  $N_P$  of the warm-start solution. This follows from the fact that for each complexity  $N_P$  of the B/B warm-start there exists exactly one minimal number of iterations  $K$  to achieve a certain approximation error  $\epsilon_{max}$ . For each  $N_P \leq N_{P,max}$  the smallest number of optimization steps  $K$  that satisfies the constraint on the approximation error can be computed using Algorithm 2. This is the case not only for the B/B method, but for all offline approximation methods for which there exists a one-to-one relationship between the approximation error  $\epsilon$  and the complexity  $N_P$  of the PWA warm-start solution. Since the calculation of B/B approximations and the solution of Algorithm 2 can be computationally expensive, we propose to solve a subproblem of (16), in order to identify a good combination instead of the optimal one. We use a subset of values for  $N_P \leq N_{P,max}$  and compute the minimal  $K$  for each of them. The best combination of explicit approximation and online optimization is represented by the solution with the minimum cost value. The samples can then be iteratively refined to improve the obtained result and approach the optimal solution to problem (16).

**Remark VII.5.** The problem of identifying the optimal combination of explicit approximation and online optimization to minimize the approximation error subject to constraints on the online computation time and the storage space can be approached following the same procedure described for problem (16).

### VIII. NUMERICAL EXAMPLES

In this section we will illustrate the proposed warm-start linear programming procedure and demonstrate its advantages using three numerical examples. The point location algorithm in [23] was used for Step 1 of Algorithm 1.

**Example 1.** We first exemplify the main procedure for a small 1D randomly generated toy system:

$$x_{i+1} = -0.969x_i + 0.4940u_i, \quad (17)$$

with a prediction horizon  $N = 10$  and the constraints  $\|u\|_\infty \leq 1$ ,  $\|x\|_\infty \leq 5$  on the input and state respectively. The norm  $p$  for the stage cost is taken as the 1-norm and the weight matrices are taken as  $Q = 1$  and  $R = 1$ .

A warm-start solution with  $N_P = 3$  regions is computed by means of the B/B approximation algorithm [12]. We investigate the proposed control law after each active set iteration until the optimum solution is reached. The cost of the warm-start B/B solution and the proposed control

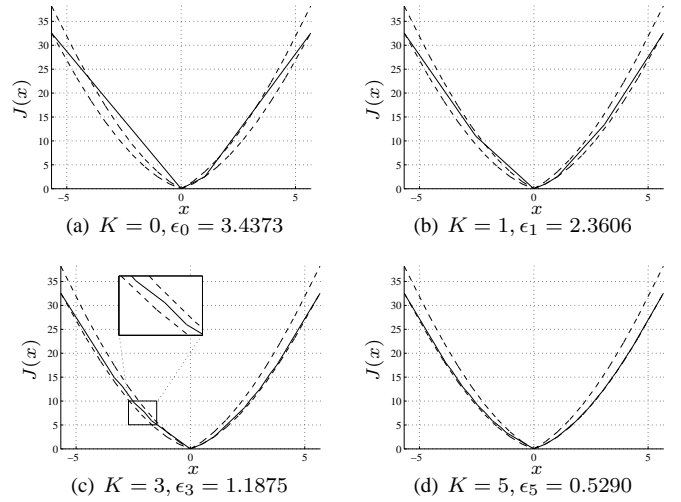


Figure 1. Warm-start linear programming procedure for Example 1. The solid line represents the cost after  $K$  online iterations starting from the B/B approximation in (a) at  $K = 0$  and  $\epsilon$  denotes the corresponding approximation error. The lower dashed line is the optimal cost that together with the upper dashed line represents the stability bound.

laws at  $K = 1, 3$  and  $5$  are shown in Fig. 1, as well as the stability bound given by Theorem III.11. Once the approximate cost lies within this bound, stability of the proposed control law is guaranteed. It can be seen in Fig. 1 (d) that stability for this example is achieved after  $K = 5$  iterations. The procedure converges to the optimal solution after  $K = 8$  iterations. Whereas the warm-start solution has an approximation error of  $\epsilon = 3.4373$ , the stable control law at  $K = 5$  with an approximation error of  $\epsilon = 0.5290$  is already very close to optimality. In Fig. 1 (c) we can observe another characteristic of the warm-start procedure, namely the fact that the approximate cost function for a proposed control law is in general not convex although it is still a Lyapunov function. Note that this small example problem can be solved explicitly with an optimal partition of  $N_P = 18$  regions suggesting a pure offline solution using explicit MPC rather than a combined procedure.

**Example 2.** After illustrating the procedure on a small problem we now exemplify the approach for assessing the optimal combination problem (16) as discussed in Section VII. Consider the randomly generated 3D-system:

$$x_{i+1} = \begin{bmatrix} -0.5 & 0.3 & -1.0 \\ 0.2 & -0.5 & 0.6 \\ 1.0 & 0.6 & -0.6 \end{bmatrix} x_i + \begin{bmatrix} -0.601 & -0.890 \\ 0.955 & -0.715 \\ 0.246 & -0.184 \end{bmatrix} u_i,$$

with a prediction horizon  $N = 5$  and the constraints  $\|u\|_\infty \leq 1$ ,  $\|x\|_\infty \leq 5$  on the input and state respectively. The norm  $p$  for the stage cost is taken as the  $\infty$ -norm and the weight matrices  $Q$  and  $R$  are taken as identity matrices.

We first try to solve the problem with the two classic approaches using online and explicit MPC. The explicit solution could not be fully computed due to the high complexity of the example problem and the solution was terminated after 24 hours at a complexity of  $14 \times 10^5$  regions which therefore represents a lower bound on the complexity or computation



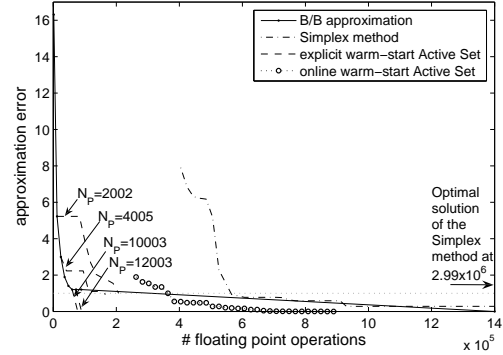
time for the explicit solution. For the online solution, the worst case in the number of iterations and the error was taken over a large number of sample points. A cold start simplex method as well as a warm-start active set method is considered that uses the solution computed at the previous measurement as an initial guess. In order to compare the online warm-start approach, a worst-case additive disturbance  $x^+ = Ax + Bu + w$  with  $\|w\| \leq 0.5$  keeping the state inside the feasible set is considered. The restricted optimal combination problem (16) was then solved for a set of warm-start solutions with  $N_P \in \{2002, 4005, 10003, 12003\}$  and a maximum approximation error  $\epsilon_{max} = 1$  that corresponds to a performance deterioration of about 0.03%, taken over a large number of sample points. The performance deterioration is measured as the relative difference between the cost of the closed loop trajectory using the optimal control input and the one using the suboptimal control input, given by

$$\left[ \sum_{i=0}^{\infty} (l(x_i, u_{on}(x_i)) - l(x_i, u^*(x_i))) \right] / \sum_{i=0}^{\infty} l(x_i, u^*(x_i)) . \quad (18)$$

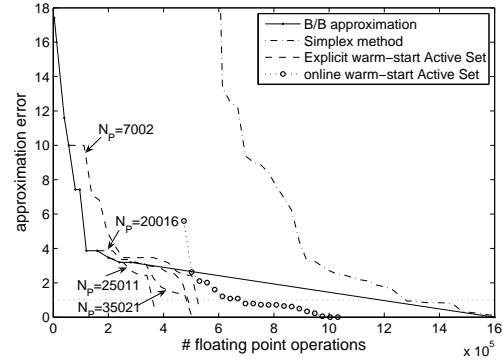
The results are shown in Fig. 2(a). The proposed control laws at an approximation error of  $\epsilon \leq 1$  are additionally guaranteed to be stabilizing (by Corollary VI.3).

If there are no storage limitations, the best combination of approximation and online iterations is given by the lower envelope of the curves, as it represents the best online computation time for a certain approximation error (or the other way round). Additional limitations on the storage space would restrict the possible warm-start solutions by a maximum number of regions  $N_P$ . The online solution using a simplex method starts at a comparably low error after the first feasible point is encountered, but as often mentioned in the literature, phase I already takes up a large amount of computation time. In the worst case, the initial solution when using an online warm-start active set method is infeasible requiring several iterations to reach feasibility. It provides a significant improvement when compared to the cold start method but always requires more computation time than the combined approach. The first two warm-start approximations with  $N_P = 2002$  and  $N_P = 4005$  take significantly more time to achieve a certain approximation error in comparison with the pure B/B approximation. In contrast, the two solutions starting from  $N_P = 10003$  and  $N_P = 12003$  achieve a clear improvement over the offline approximation.

For any error above 1.2 a pure approximation by the B/B method results in the fastest computation times. The best combination for any error below 1.2 is given by a combination of a warm-start solution of complexity  $N_P = 10003$  with active set iterations. Note that a further refinement of the warm-start solution does not improve the results and hence this particular combination of warm-start and online solution represents the best combination to achieve an approximation error below 1.2.



(a) Warm-start procedure for Example 2 starting from four different PWA B/B approximations with  $N_P \in \{2002, 4005, 10003, 12003\}$ .



(b) Warm-start procedure for Example 3 starting from four different PWA B/B approximations with  $N_P \in \{7002, 20016, 25011, 35021\}$ .

Figure 2. The solid line represents the offline B/B approximations. The dashed lines show the improvement by the active set method warm-started from the B/B approximations. The dash-dotted line is a sampled worst-case estimate of a pure online solution using the Simplex method and the dotted line using the online warm-start Active Set method, shown after the first feasible solution is found. The flop number for zero approximation error of the B/B approximation was interpolated, since the optimal solution could not be computed due to its excessive complexity.

**Example 3.** We will now investigate the optimal combination problem for the 4D randomly generated system:

$$x_{i+1} = \begin{bmatrix} -0.251 & 0.114 & 0.123 & -0.433 \\ 0.319 & -0.658 & 0.905 & 0.118 \\ 0.459 & -0.484 & -0.175 & -0.709 \\ 0.016 & 0.116 & -0.002 & -0.505 \end{bmatrix} x_i + \begin{bmatrix} -0.873 & 0.879 \\ -0.669 & 0.936 \\ -0.353 & 0.777 \\ 0.268 & -0.336 \end{bmatrix} u_i$$

with a prediction horizon  $N = 5$  and the constraints  $\|u\|_{\infty} \leq 5$ ,  $\|x\|_{\infty} \leq 5$  on the input and state respectively. The norm  $p$  for the stage cost is taken as the  $\infty$ -norm and the weight matrices  $Q$  and  $R$  are taken as the identity matrix and two times the identity matrix.

As in the previous example, the explicit solution for this problem is highly complex and could not be fully computed within 24 hours, when it was terminated with a complexity of  $16 \times 10^5$  regions. The online solution was computed as described in Example 2 with  $\|w\| \leq 2$ . The restricted optimal combination problem (16) was solved for a set of warm-start solutions with  $N_P \in \{7002, 20016, 25011, 35021\}$  and a maximum approximation error  $\epsilon_{max} = 1$  that corresponds to a performance deterioration (18) of about 0.03%, taken over

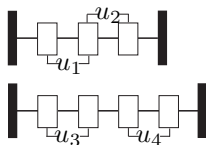


Figure 3. Two systems of 3 and 4 oscillating masses. The bold lines represent the spring-damper system, the dark blocks on the side represent the walls.

a large number of sample points. The results are shown in Fig. 2. The proposed control laws at an approximation error of  $\epsilon \leq 1$  are again guaranteed to be stabilizing (by Corollary VI.3).

For an approximation error up to 3.2, a pure approximation by the B/B method results in the fastest computation times, but the solution is not guaranteed to be stabilizing. For any error below 3.2 a combination of a warm-start solution of complexity  $N_P = 25011$  with active set iterations represents the best combination. Note that a further improvement of the warm-start solution does not provide any benefit. In comparison with a pure online solution, the warm-start procedure is again always superior.

**Example 4.** After illustrating the proposed procedures on smaller examples we will now demonstrate their application to bigger problem dimensions. We consider the problem of regulating a system of oscillating masses as described in [8], one consisting of 3 and the other of 4 masses which are interconnected by spring-damper systems and connected to walls on the side, as shown in Fig. 3. The masses have value 1, the springs constant 1 and the damping constant is 0.5. The MPC problem for the 3-masses example has 6 states and 2 inputs, resulting in an LP with 50 optimization variables. The MPC problem for the 4 masses example consists of 8 states and 2 inputs and the LP has 60 optimization variables. The state and input constraints are  $\|u\|_\infty \leq 1$ ,  $\|x\|_\infty \leq 4$ , the horizon is chosen to  $N = 5$ , the norm  $p$  for the stage cost is taken as the  $\infty$ -norm and the weight matrices  $Q$  and  $R$  are taken as identity matrices.

We identify the best combination of explicit approximation and online optimization for providing a hard real-time stability guarantee for these two example problems out of a selection of warm-start solutions by computing the minimal number of optimization steps  $K_{\min}$  as well as the number of floating point operations required to provide a suboptimal control law that is guaranteed to be stabilizing. While the optimal explicit solution is not computable for the considered problem dimensions, we compare the flop numbers for the warm-start combinations to the computational effort for a pure online MPC solution. A lower bound on the flop number for computing the optimal solution using a simplex method is estimated by taking the worst case number of iterations over a large number of sampling points. For the 6-dimensional example a worst case number of 51 iterations including phase I and for the 8-dimensional example of 61 iterations was observed. The average closed

loop performance deterioration taken over a large number of sampling points is calculated for all warm-start combinations using (18). The results are shown in Tables I(a) and I(b).

For the 6-dimensional problem we choose  $N_P \in \{3061, 5080, 7152\}$ . The number of required optimization steps to reach stability decreases with the refinement of the warm-start solution, resulting in a decreasing number of flops. The best combination is therefore represented by the warm-start solution consisting of 7152 regions, where stability can be guaranteed after only 5 online steps. The average closed loop performance deterioration for this combination is 0.63%. For the 8-dimensional problem a set of warm-start solutions  $N_P \in \{3202, 5312, 7164\}$  is computed. While the first two warm-start combinations require at least 17 optimization steps to guarantee stability, the number of iterations is reduced to 15 online iterations for the more complex warm-start solution. Identifying the best combination among these 3 options, we see that it is given by the warm-start solution of 7164 regions combined with 15 online iterations having the lowest flop number, which is additionally supported by the low average performance deterioration of 0.82% observed for this combination.

The offline analysis hence provides a real-time stability guarantee in both test cases for a comparably small number of regions that need to be stored and a small number of optimization steps. The warm-start procedure is always superior to a pure online optimization approach, which takes more than 7 or 4 times the computation time of the best combination for the 6-dimensional or the 8-dimensional example, respectively. An explicit approximation with a stability guarantee cannot be computed in both cases within 24 hours due to its high complexity.

While Examples 2 and 3 are small randomly generated example systems which are generally observed to result in problems of high complexity, Example 4 represents a physical system model of higher dimension and average complexity that is related to many applications involving spring-damper systems (e.g. active suspension). We hereby cover test cases reflecting the scope of the presented approach. Examples 2 and 3 show that the optimal strategy to achieve a certain set of solution properties is often not to compute the best warm-start solution, but a particular combination of warm-start and online optimization. Example 4 demonstrates that even for a higher dimensional example the combined procedure clearly outperforms an online solution when minimizing the online computation time and a hard real-time stability guarantee can be provided. The optimal combination of explicit approximation and online optimization does however highly depend on the particular problem structure and the given requirements on storage space and performance. For certain problems the best solution procedure will be a particular combination of the two methods whereas for others it will as well be a pure offline or online approximation, e.g. in the case of extremely small and simple or large problems which can be identified by means of

Table I  
SIMULATION RESULTS FOR EXAMPLE 4

(a) 6D Oscillating Masses Example			
# Regions for B/B approximation $N_P$	3064	5080	7152
# Pivots to stability guarantee $K_{\min}$	15	8	5
Online computation time in kilo flops	789	462	337
Closed loop performance deterioration	6.94%	3.30%	0.63%
Flops for simplex method in kilo flops	2558		

(b) 8D Oscillating Masses Example			
# Regions for B/B approximation $N_P$	3202	5312	7164
# Pivots to stability guarantee $K_{\min}$	17	17	15
Online computation time in kilo flops	1384	1418	1291
Closed loop performance deterioration	5.56%	1.85%	0.82%
Flops for simplex method in kilo flops	4779		

the presented analysis.

## IX. CONCLUSIONS

We presented a new approach that combines the two paradigms of online and explicit MPC and hereby offers new possibilities in the applicability of MPC to real problems that often have limits on the storage space or the available computation time. The proposed method computes a piecewise affine approximation of the optimal solution offline that is stored and used to warm-start an active set method. By means of a preprocessing analysis hard real-time, stability and performance guarantees for the proposed controller are provided. The analysis does not require the calculation of the optimal parametric solution to the MPC problem, since it may be prohibitively complex and could restrict the applicability of the method.

The warm-start procedure enlarges the possibilities to trade-off solution properties in order to satisfy constraints in terms of online computation time, storage and performance. The best solution method is dependent on the particular system as well as the given hardware and performance restrictions. We show how the offline analysis can be utilized to compare different MPC methods and identify the best approach for a considered application and set of requirements. In addition to the discussed aspects, desired implementation properties may affect the choice of the MPC method. By using a combination of offline approximation and online MPC, the method inherits the properties of both paradigms and therefore also the numerical and computational challenges of online MPC, such as the need for floating point computations or software verification and maintenance.

The presented numerical examples illustrate the proposed procedures and confirm the fact that a warm-start solution can often outperform either a pure offline or online method. The warm-start procedure provides hard real-time guarantees on the applied suboptimal controller where an approximate explicit or online approach is either intractable or can not meet the given requirements.

## REFERENCES

- [1] A. Bemporad, F. Borrelli, and M. Morari, "Model Predictive Control Based on Linear Programming - The Explicit Solution," *IEEE Transactions on Automatic Control*, vol. 47, no. 12, pp. 1974–1985, 2002.
- [2] A. Bemporad, M. Morari, V. Dua, and E. Pistikopoulos, "The Explicit Linear Quadratic Regulator for Constrained Systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.
- [3] F. Borrelli, *Constrained Optimal Control of Linear and Hybrid Systems*, ser. Lecture Notes in Control and Information Science. Springer Verlag, 2003, vol. 290.
- [4] B. Lincoln and A. Rantzer, "Relaxing dynamic programming," *IEEE Transactions on Automatic Control*, vol. 51, no. 8, pp. 1249–1260, 2006.
- [5] E. A. Yildirim and S. J. Wright, "Warm-Start Strategies In Interior-Point Methods For Linear Programming," *SIAM Journal on Optimization*, vol. 12, no. 3, pp. 782–810, 2002.
- [6] A. Bemporad and C. Filippi, "An Algorithm for Approximate Multiparametric Convex Programming," *Computational Optimization and Applications*, vol. 35, no. 1, pp. 87–108, 2006.
- [7] H. J. Ferreau, H. G. Bock, and M. Diehl, "An online active set strategy to overcome the limitations of explicit MPC," *International Journal of Robust and Nonlinear Control*, vol. 18, no. 8, pp. 816–830, 2008.
- [8] Y. Wang and S. Boyd, "Fast Model Predictive Control Using Online Optimization," in *IFAC World Congress*, 2008.
- [9] M. Zeilinger, C. Jones, and M. Morari, "Real-time suboptimal Model Predictive Control using a combination of Explicit MPC and Online Computation," in *Proc. 47th IEEE Conf. on Decision and Control*, Cancun, Mexico, Dec. 2008.
- [10] M. Vidyasagar, *Nonlinear Systems Analysis*, 2nd ed. Prentice Hall, 1993.
- [11] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.
- [12] C. N. Jones, M. Baric, and M. Morari, "Multiparametric Linear Programming with Applications to Control," *European Journal of Control*, vol. 13, no. 2-3, pp. 152–170, 2007.
- [13] P. O. M. Scokaert, D. Q. Mayne, and J. B. Rawlings, "Suboptimal model predictive control (feasibility implies stability)," *IEEE Transactions on Automatic Control*, vol. 44, no. 3, pp. 648–654, 1999.
- [14] C. V. Rao, S. J. Wright, and J. B. Rawlings, "Application of interior-point methods to model predictive control," *Journal of Optimization Theory and Applications*, vol. 99, pp. 723–757, 1998.
- [15] A. Bemporad and C. Filippi, "Suboptimal explicit MPC via approximate multiparametric quadratic programming," in *Proc. 40th IEEE Conf. on Decision and Control*, vol. 5, 2001, pp. 4851–4856.
- [16] T. A. Johansen and A. Grancharova, "Approximate explicit constrained linear model predictive control via orthogonal search tree," *IEEE Transactions on Automatic Control*, vol. 48, pp. 810–815, 2003.
- [17] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*. Academic Press, 1982.
- [18] R. Fletcher, *Practical Methods of Optimization*, 2nd ed. John Wiley and Sons, New York, 1987.
- [19] P. Wolfe, "A technique for resolving degeneracy in linear programming," *SIAM Journal*, vol. 11, pp. 205–211, 1963.
- [20] R. Bland, "New finite pivoting rules for the simplex method," *Mathematics of Operations Research*, vol. 2, no. 2, pp. 103–107, 1977.
- [21] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, "A practical anti-cycling procedure for linearly constrained optimization," *Mathematical Programming*, vol. 45, no. 1-3, pp. 437–474, 1989.
- [22] K. G. Murty, *Linear Programming*. John Wiley & Sons, 1983.
- [23] M. Baotić, F. Borrelli, A. Bemporad, and M. Morari, "Efficient On-Line Computation of Constrained Optimal Control," *SIAM Journal on Control and Optimization*, vol. 47, no. 5, pp. 2470–2489, 2008.
- [24] P. Tøndel, T. Johansen, and A. Bemporad, "Evaluation of Piecewise Affine Control via Binary Search Tree," *Automatica*, vol. 39, no. 5, pp. 945–950, May 2003.
- [25] C. N. Jones, P. Grieder, and S. V. Rakovic, "A logarithmic-time solution to the point location problem for parametric linear programming," *Automatica*, vol. 42, pp. 2215–2218, 2006.
- [26] Y. Wang, C. Jones, and J. M. Maciejowski, "Efficient point location via subdivision walking with application to explicit MPC," in *European Control Conf.*, Kos, Greece, Jul. 2007.
- [27] M. J. Best and K. Ritter, *Linear Programming: Active Set Analysis and Computer Programs*. Prentice-Hall, Inc., 1985.

- [28] T. Gal and J. Nedoma, "Multiparametric Linear Programming," *Management Science*, vol. 18, no. 7, pp. 406–422, 1972.
- [29] S. J. Wright, "Applying New Optimization Algorithms To Model Predictive Control," in *5th Int. Conf. on Chemical Process Control - CPC V*, vol. 93. CACHE Publications, 1997, pp. 147–155.
- [30] J. J. H. Forrest and J. A. Tomlin, "Updated triangular factors of the basis to maintain sparsity in the product form simplex method," *Mathematical Programming*, vol. 2, no. 1, pp. 263–278, 1972.
- [31] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The Quickhull Algorithm for Convex Hulls," *ACM Trans. Math. Software*, vol. 22, no. 4, pp. 469–483, 1996.
- [32] B. Grünbaum, "Measures of symmetry for convex sets," *Proc. 7th Symp. in Pure Math., American Math. Soc., Symp. on Convexity*, pp. 233–270, 1961.
- [33] R. Schneider and J. A. Wieacker, "Approximation of Convex Bodies by Polytopes," *Bull. London Math. Soc.*, vol. 13, no. 2, pp. 149–156, 1981.
- [34] E. M. Bronshteyn and L. D. Ivanov, "The approximation of convex sets by polyhedra," *Siberian Math. J.*, vol. 16, no. 5, pp. 852–853, 1975.



**Melanie N. Zeilinger** received the diploma in Engineering Cybernetics from the University of Stuttgart, Germany, in 2006. She conducted her diploma thesis research at the Department of Chemical Engineering, University of California at Santa Barbara, USA, in 2005–2006. Since 2006 she has been pursuing a PhD at the ETH in Zurich. Her research interests are in high-speed model predictive control.



**Colin N. Jones** Colin Jones is a senior researcher (Oberassistent) at the Automatic Control Laboratory at the ETH in Zurich. He obtained a PhD in 2005 from the University of Cambridge for his work on polyhedral computational methods for constrained control. Prior to that, he was at the University of British Columbia in Canada, where he took a BAsC and MASc in Electrical Engineering and Mathematics. Colin has worked in a variety of industrial roles, ranging from control of heating, ventilation and air conditioning to ballistic missile interception. He is

co-founder of Apex Optimization; a custom optimization house that focuses on human resource scheduling. His current research interests are in the areas of high-speed predictive control, optimization, energy management and optimal scheduling.



**Manfred Morari** Manfred Morari was appointed head of the Department of Information Technology and Electrical Engineering at ETH Zurich in 2009. He was head of the Automatic Control Laboratory from 1994 to 2008. Before that he was the McCollum-Corcoran Professor of Chemical Engineering and Executive Officer for Control and Dynamical Systems at the California Institute of Technology. He obtained the diploma from ETH Zurich and the Ph.D. from the University of Minnesota, both in chemical engineering. His interests are in

hybrid systems and the control of biomedical systems. In recognition of his research contributions he received numerous awards, among them the Donald P. Eckman Award and the John R. Ragazzini Award of the Automatic Control Council, the Allan P. Colburn Award and the Professional Progress Award of the AIChE, the Curtis W. McGraw Research Award of the ASEE, Doctor Honoris Causa from Babes-Bolyai University, Fellow of IEEE, the IEEE Control Systems Field Award, and was elected to the National Academy of Engineering (U.S.). Manfred Morari has held appointments with Exxon and ICI plc and serves on the technical advisory boards of several major corporations.