

**SÚBOR 10 VYBRANÝCH NAJVÝZNAMNEJŠÍCH
VEDECKÝCH PRÁC**

doc. Ing. Michal Kvasnica, Dr.sc.

Bratislava, 2021

1. [ADC] **Kvasnica, M.** [60%] – Löfberg, J. [20%] – Fikar, M. [20%]: Stabilizing polynomial approximation of explicit MPC. *Automatica*, č. 10, zv. 47, str. 2292–2297, 2011. IF: 5,541
2. [ADC] **Kvasnica, M.** [50%] – Hledík, J. [20%] – Rauová, I. [10%] – Fikar, M. [20%]: Complexity reduction of explicit model predictive control via separation. *Automatica*, č. 6, zv. 49, str. 1776–1781, 2013. IF: 5,541
3. [ADC] Grieder, P. [25%] – **Kvasnica, M.** [25%] – Baotic, M. [25%] – Morari, M. [25%]: Stabilizing low complexity feedback control of constrained piecewise affine systems. *Automatica*, vol. 41, issue 10, str. 1683–1694, 2005. IF: 5,541
4. [ADC] Herceg, M. [25%] – Jones, C. [25%] – **Kvasnica, M.** [25%] – Morari, M. [25%]: Enumeration-based approach to solving parametric linear complementarity problems. *Automatica*, č. 62, str. 243–248, 2015. IF: 5,541
5. [ADC] Nguyen, N. A. [25%] – Olaru, S. [25%] – Rodríguez-Ayerbe, P. [25%] – **Kvasnica, M.** [25%]: Convex liftings-based robust control design. *Automatica*, č. March 2017, zv. 77, str. 206–213, 2017. IF: 5,541
6. [ADC] **Kvasnica, M.** [80%] – Fikar, M. [20%]: Clipping-Based Complexity Reduction in Explicit MPC. *IEEE Transactions on Automatic Control*, č. 7, zv. 57, str. 1878–1883, 2012. IF: 5,625
7. [ADC] Jiang, Y. [25%] – Oravec, J. [25%] – Houska, B. [25%] – **Kvasnica, M.** [25%]: Parallel MPC for Linear Systems with Input Constraints. *IEEE Transactions on Automatic Control*, str. 1–8, 2020. DOI: 10.1109/TAC.2020.3020827. IF: 5,625
8. [ADC] Cseko, L. [40%] – **Kvasnica, M.** [40%] – Lantos, B. [20%]: Explicit MPC-Based RBF Neural Network Controller Design With Discrete-Time Actual Kalman Filter for Semiactive Suspension. *IEEE Transactions on Control Systems Technology*, č. 5, zv. 23, str. 1736–1753, 2015. IF: 5,312
9. [ADC] Theunissen, J. [15%] – Sorniotti, A. [15%] – Gruber, P. [15%] – Fallah, S. [15%] – Ricco, M. [15%] – **Kvasnica, M.** [15%] – Dhaens, M. [10%]: Regionless Explicit Model Predictive Control of Active Suspension Systems With Preview. *IEEE Transactions on Industrial Electronics*, č. 6, zv. 67, str. 4877–4888, 2020. IF: 7,515
10. [ADC] **Kvasnica, M.** [70%] – Bakaráč, P. [10%] – Klaučo, M. [20%]: Complexity reduction in explicit MPC: A reachability approach. *Systems & Control Letters*, zv. 124, str. 19–26, 2019. IF: 2,762



Brief paper

Stabilizing polynomial approximation of explicit MPC[☆]Michal Kvasnica^{a,1}, Johan Löfberg^b, Miroslav Fikar^a^a Faculty of Chemical and Food Technology, Slovak University of Technology in Bratislava, 812 37 Bratislava, Slovakia^b Division of Automatic Control, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden

ARTICLE INFO

Article history:

Received 23 August 2010

Received in revised form

9 February 2011

Accepted 14 April 2011

Available online 1 September 2011

Keywords:

Asymptotic stability

Predictive control

Function approximation

ABSTRACT

A given explicit piecewise affine representation of an MPC feedback law is approximated by a single polynomial, computed using linear programming. This polynomial state feedback control law guarantees closed-loop stability and constraint satisfaction. The polynomial feedback can be implemented in real time even on very simple devices with severe limitations on memory storage.

© 2011 Elsevier Ltd. All rights reserved.

1. Introduction

In explicit Model Predictive Control (Bemporad, Morari, Dua, & Pistikopoulos, 2002), parametric programming (Borrelli, 2003) is used to construct a function μ which maps state measurements x onto the optimal control inputs. Provided such a μ exists, real-time implementation of MPC in Receding Horizon fashion (RH MPC) boils down to a mere function evaluation. For a rich class of MPC setups, μ can be shown to be a piecewise affine (PWA) function defined over $N_{\mathcal{R}}$ polytopic regions. The main practical limitation, however, is that the number of regions grows quickly with problem size, having negative impact on the required memory storage and processing power. The number of regions of μ can be reduced e.g. by move blocking (Tøndel & Johansen, 2002), by model reduction techniques (Hovland, Willcox, & Gravdahl, 2008), or by relaxing optimality (Bemporad & Filippi, 2003). Another direction is to a-posteriori simplify the regions either by merging (Geyer, Torrisi, & Morari, 2008; Kvasnica & Fikar, 2010), by replacing them by hyperboxes (Johansen & Grancharova, 2003) or by simplices (Hovd, Scibilia, Maciejowski, & Oлару, 2009). Evaluation of μ for a

particular value of x can be simplified by organizing the regions into a binary search tree (Tøndel, Johansen, & Bemporad, 2003), or by building a lattice representation (Wen, Ma, & Ydstie, 2009) of the PWA function μ . A common denominator of all referenced approaches is that they lead to a simpler (sub)optimal RH MPC feedback $\tilde{\mu}$, which still is a PWA function. As a consequence, although a remarkable reduction of complexity can be achieved in certain cases, the memory footprint of the approximation $\tilde{\mu}$ still typically exceeds ten kilobytes. In this work, we aim at simplifying the RH MPC in such a way that it can easily be implemented on typical industrial hardware platforms, such as programmable logic controllers, which usually only provide 2–8 kB of memory.

We propose to remove all regions of μ completely and to approximate it by a single polynomial feedback $\tilde{\mu}(x)$ of an *a-priori fixed* degree such that closed-loop stability and constraint satisfaction are preserved. The approach is applicable not only to linear systems, but also covers switched affine systems which belong to the class of hybrid systems (Bemporad & Morari, 1999). Building upon our previous work (Kvasnica, Christophersen, Herceg, & Fikar, 2008; Kvasnica, Löfberg, Herceg, Čirka, & Fikar, 2010), the approximation is performed in two steps. First, a parameterization of a set of stabilizing controllers, referred to as the *stability tube* (Christophersen, 2007), is obtained using basic computational geometry tools. Subsequently, we show how to search for the coefficients of $\tilde{\mu}$ by solving a single linear program (LP). If the LP is feasible, the polynomial control law is guaranteed to reside in the stability tube, and hence it is closed-loop stabilizing and satisfies constraints for all time.

The key advantage is that the memory footprint of the approximate feedback $\tilde{\mu}$ is minute compared to the storage of μ . In particular, for the type of problems considered here, the total

[☆] The authors are pleased to acknowledge the financial support of the Scientific Grant Agency of the Slovak Republic under the grants 1/0079/09 and 1/0095/11 and of the Slovak Research and Development Agency under the contracts No. VV-0029-07 and No. LPP-0092-07. This paper was not presented at any IFAC meeting. This paper was recommended for publication in revised form by Associate Editor Martin Guay under the direction of Editor Frank Allgöwer.

E-mail addresses: michal.kvasnica@stuba.sk (M. Kvasnica), johanl@isy.liu.se (J. Löfberg), miroslav.fikar@stuba.sk (M. Fikar).

¹ Tel.: +421 2 59325352; fax: +421 2 59325340.

storage for $\tilde{\mu}$ is roughly equal to the footprint of a single region of μ . It follows that the overall memory requirements are reduced $N_{\mathcal{R}}$ times. The price to be paid is the inherent loss of optimality. Moreover, certain assumptions on the shape of the stability tube have to be imposed in order to formulate the search for coefficients of $\tilde{\mu}$ as a single LP. Because of this, and since the LP-based search is based on sufficient conditions, it does not have to be always successful.

Compared to our previous work (Kvasnica et al., 2008, 2010), we report a detailed complexity analysis of the overall design procedure. A large case study is provided to illustrate how the computation scales with increasing problem size and to assess the overall success rate. More importantly, new ideas for reducing the size of the LP problem are presented which extend the applicability of our approach to larger problems.

2. Preliminaries

The set of non-negative real numbers is denoted by $\mathbb{R}_{\geq 0}$. Interior of a set Ω is $\text{int}(\Omega)$. We call a collection of polytopes $\{\mathcal{R}_i\}_{i=1}^{N_{\mathcal{R}}}$ the *partition* of a set Ω if $\Omega = \bigcup_{i=1}^{N_{\mathcal{R}}} \mathcal{R}_i$, and $\text{int}(\mathcal{R}_i) \cap \text{int}(\mathcal{R}_j) = \emptyset$ for all $i \neq j$. Each polytope \mathcal{R}_i will be referred to as a *region* of the partition. A function $\mu : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_u}$ with domain $\Omega \subseteq \mathbb{R}^{n_x}$ is called Piecewise Affine (PWA) over polytopes if $\{\mathcal{R}_i\}_{i=1}^{N_{\mathcal{R}}}$ is a partition of Ω and $\mu(x) := K_i x + L_i \forall x \in \mathcal{R}_i$, $i = 1, \dots, N_{\mathcal{R}}$.

We consider well-posed (Bemporad & Morari, 1999), stabilizable PWA systems in discrete time $x_{t+1} = f_{\text{PWA}}(x_t, u_t)$, composed of finitely many local affine dynamics, each valid in a polytope $\mathcal{D}_d \subseteq \mathbb{R}^{n_x}$:

$$f_{\text{PWA}}(x_t, u_t) := A_d x_t + B_d u_t + a_d \quad \text{if } x_t \in \mathcal{D}_d, \quad (1)$$

where $x_t \in \mathbb{R}^{n_x}$ are the states and $u_t \in \mathbb{R}^{n_u}$ the inputs. The task is to control the PWA system (1) toward the origin (which is assumed to be an equilibrium of (1)) while fulfilling state and input constraints for all time, i.e. $x_t \in \mathcal{X}$, $u_t \in \mathcal{U}$, $\forall t \geq 0$, where $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ and $\mathcal{U} \subseteq \mathbb{R}^{n_u}$ are assumed to be non-empty polytopic sets containing the origin in their respective interiors.

We define for the PWA system (1) the *constrained finite time optimal control* (CFTOC) problem

$$J_N^*(x_t) = \min_{U_N} \ell_N(x_{t+N}) + \sum_{k=0}^{N-1} \ell(x_{t+k}, u_{t+k}) \quad (2a)$$

$$\text{s.t. } \begin{cases} x_{t+k+1} = f_{\text{PWA}}(x_{t+k}, u_{t+k}), \\ u_{t+k} \in \mathcal{U}, \quad x_{t+k} \in \mathcal{X}, \quad x_{t+N} \in \mathcal{X}_f, \end{cases} \quad (2b)$$

where x_{t+k} is the future evolution of (1) over a prediction horizon N , given the initial condition x_t and the vector of future control inputs $U_N := [u_t^T, \dots, u_{t+N-1}^T]^T$. $\mathcal{X}_f \subseteq \mathcal{X}$ is a polytopic terminal set with $0_{n_x} \in \mathcal{X}_f$, $\ell_N(x_{t+N}) = \|Q_N x_{t+N}\|_p$ is the terminal penalty, and $\ell(x_{t+k}, u_{t+k}) = \|Q_x x_{t+k}\|_p + \|Q_u u_{t+k}\|_p$ is the stage cost. It is assumed that $p \in \{1, \infty\}$ in (2a). For problems of modest size it is possible to characterize the RHMPC feedback law $\mu : \Omega \rightarrow \mathcal{U}$ and the optimal value function $J_N^* : \Omega \rightarrow \mathbb{R}_{\geq 0}$ explicitly as PWA functions of x_t (Bemporad et al., 2002; Borrelli, 2003). Here, $\Omega := \{x_t \mid \exists u_t, \dots, u_{t+N-1} \text{ s.t. (2b) hold}\}$, and it is partitioned into $N_{\mathcal{R}}$ polytopic regions \mathcal{R}_i .

Assumption 2.1. The RHMPC feedback $\mu(x_t)$ is closed-loop stabilizing, feasible for all time (Christophersen, 2007) and a PWA Lyapunov function $V : \Omega \rightarrow \mathbb{R}_{\geq 0}$ for the closed-loop system $f^{\text{CL}} := f_{\text{PWA}}(x_t, \mu(x_t))$ exists $\forall x_t \in \Omega$ and is given.

This is not a restricting requirement but rather the aim of most (if not all) control strategies. We remark that if N , Q_x , Q_u , Q_N , \mathcal{X}_f are chosen as in Baotić, Christophersen, and Morari (2006), then $\mu(\cdot)$ satisfies Assumption 2.1 and $V := J_N^*$ is a Lyapunov function.

Theorem 2.2 (Lazar, Munoz de la Pena, Heemels, & Alamo, 2008). Let Ω be a bounded positively invariant set with $0_{n_x} \in \text{int}(\Omega)$ and let $\underline{\beta}(\cdot)$ and $\overline{\beta}(\cdot)$ be \mathcal{K}_{∞} -class functions. Then if there exists function $V : \Omega \rightarrow \mathbb{R}_{\geq 0}$ with $V(0_{n_x}) = 0$, bounded by $\underline{\beta}(\|x\|) \leq V(x) \leq \overline{\beta}(\|x\|)$, and satisfying $V(f^{\text{CL}}(x)) \leq \gamma V(x)$ for some $\gamma \in [0, 1)$ and for all $x \in \Omega$, then the closed-loop system f^{CL} is asymptotically stable in Ω .

The freedom in γ allows one to find a set of stabilizing controllers which render the function V a control Lyapunov function. Such sets are denoted as *stability tubes* (Christophersen, 2007):

$$\mathcal{S}(V, \gamma) := \left\{ \begin{bmatrix} x \\ u \end{bmatrix} \mid u \in \mathcal{U}, x \in \Omega, f(x, u) \in \Omega, \right. \\ \left. V(f(x, u)) \leq \gamma V(x) \right\}. \quad (3)$$

For the type of PWA systems (1), PWA Lyapunov functions V , and fixed γ , the tube can be computed explicitly using reachability analysis (Christophersen, 2007, Ch. 10.4) and represented as a (possibly non-convex) union of polytopes. To see this, note that for each feasible transition from region \mathcal{R}_i to region \mathcal{R}_j for which the value of V decreases, (3) is a polytope $\mathcal{S}_{i,j}$ in the $x - u$ space. The whole stability tube is then given by $\mathcal{S}(V, \gamma) := \bigcup_{i=1}^{N_{\mathcal{R}}} \mathcal{S}_i$, where $\mathcal{S}_i := \bigcup_{j=1}^{N_{\mathcal{R}}} \mathcal{S}_{i,j}$, $i = 1, \dots, N_{\mathcal{R}}$.

3. Main results

We aim at approximating a given RHMPC control law μ by a single multivariate polynomial $\tilde{\mu}$ of pre-specified degree δ :

$$\tilde{\mu}(x) = \alpha_1 x + \alpha_2 x^2 + \dots + \alpha_{\delta} x^{\delta}. \quad (4)$$

Here, $\alpha_i \in \mathbb{R}^{n_u \times n_x^i}$, $i = 1, \dots, \delta$, are the coefficients to be determined, and x^i is the element-wise i -th power of a vector $x \in \mathbb{R}^{n_x}$, i.e. $x^i = [x_1^i, x_2^i, \dots, x_{n_x}^i]^T$. Note that in a multi-input case with $n_u > 1$, (4) is a vector-valued polynomial. The constant offset α_0 is not considered in (4) since $\tilde{\mu}(0_{n_x}) = 0_{n_u}$ must hold to attain stability. Formally, we aim at solving the following problem.

Problem 3.1. Find the coefficients $\alpha = \{\alpha_1, \dots, \alpha_{\delta}\}$ of the polynomial state-feedback law (4) of fixed degree δ such that $\tilde{\mu} \approx \mu$ asymptotically stabilizes the PWA system (1) to the origin while fulfilling state and input constraints for all time.

To solve this problem, we exploit the inherent freedom of the Lyapunov function V , captured by its stability tube.

Theorem 3.2 (Christophersen, 2007). Let the stability tube $\mathcal{S}(V, \gamma)$ be given. Then every control law $\tilde{\mu}(x)$ fulfilling $\begin{bmatrix} x \\ \tilde{\mu}(x) \end{bmatrix} \in \mathcal{S}(V, \gamma)$ asymptotically stabilizes the system $x^+ = f_{\text{PWA}}(x, \tilde{\mu}(x))$ for all $x \in \Omega$ to the origin.

Remark 3.3. The concept of stability tubes does not require that the function V originates as a solution of the MPC problem (2). In fact, the tube can be constructed for an arbitrary feedback law which admits a PWA² Lyapunov function on Ω . It follows that the presented procedure can be applied to approximate arbitrary feedback laws with this property.

In the sequel we show that, given a stability tube $\mathcal{S}(V, \gamma)$, the polynomial $\tilde{\mu}$ satisfying $\begin{bmatrix} x \\ \tilde{\mu}(x) \end{bmatrix} \in \mathcal{S}(V, \gamma)$, $\forall x \in \Omega$ can be found by solving a single linear program under the following assumption.

² For piecewise quadratic Lyapunov functions the tube can no longer be represented as a union of polytopes, in general.

Assumption 3.4. For a given Lyapunov function V there exists a $\gamma \in [0, 1)$ for which:

- A1: a full-dimensional stability tube $\mathcal{S}(V, \gamma) := \bigcup_{i=1}^{N_{\mathcal{R}}} \mathcal{S}_i$ exists;
 A2: for each $i = 1, \dots, N_{\mathcal{R}}$ either $\mathcal{S}_i := \bigcup_j \mathcal{S}_{i,j}$ is a convex polytope, or an inner polytopic approximation $\mathcal{S}_i \subseteq \bigcup_j \mathcal{S}_{i,j}$ exists such that $\text{proj}_x \mathcal{S}_i = \mathcal{R}_i$;
 A3: the union $\bigcup_i \mathcal{S}_i$ is connected.

The existence of $\mathcal{S}(V, \gamma)$ hints at the existence of control laws, other than μ , which would provide closed-loop stability and constraint satisfaction for all time. Connectivity is implied by the objective of approximating μ by a single continuous polynomial valid over the whole domain $\text{dom}(\tilde{\mu}) = \Omega$. Finally, convexity (and hence uniqueness) is dictated by the desire of being able to perform the approximation in a computationally efficient manner. If A2 does not hold, $\tilde{\mu}$ can still be found by solving a combinatorial problem. If A3 is violated, the remedy would be to approximate independently each connected part of the tube, giving rise to a piecewise polynomial type of approximation.

Under this assumption, the tube consists of $N_{\mathcal{R}}$ polytopes in the state-input space:

$$\mathcal{S}_i := \left\{ \begin{bmatrix} x \\ u \end{bmatrix} \mid \begin{bmatrix} S_i^x & S_i^u \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} \leq S_i^0 \right\}. \quad (5)$$

We remark that the whole tube $\mathcal{S}(V, \gamma) := \bigcup_i \mathcal{S}_i$ is not required to be convex. Define, for each $i = 1, \dots, N_{\mathcal{R}}$, a set of polynomials

$$p_i(\alpha, x) := S_i^0 - S_i^x x - S_i^u \tilde{\mu}(x), \quad (6)$$

where the cardinality of $p_i(\cdot)$ is equal to the number of constraints of the i -th element of the stability tube, i.e. the number of rows of S_i^0 . Then we get the following straightforward result.

Lemma 3.5. Let a stability tube $\mathcal{S}(V, \gamma)$ satisfying Assumption 3.4 be given. If there exist coefficients $\alpha_1, \dots, \alpha_\delta$ of $\tilde{\mu}$ as in (4) such that

$$p_i(\alpha, x) \geq 0, \quad \forall x \in \mathcal{R}_i, \quad i = 1, \dots, N_{\mathcal{R}}, \quad (7)$$

then $\tilde{\mu}$ solves Problem 3.1.

Proof. First note that (7) with $p_i(\alpha, x)$ as in (6) is equivalent, for a fixed i , to (5) with $u = \tilde{\mu}(x)$. Therefore, if (7) admits a solution, then $\begin{bmatrix} x \\ \tilde{\mu}(x) \end{bmatrix} \in \mathcal{S}_i \forall x \in \mathcal{R}_i$. Hence if (7) holds for all $i = 1, \dots, N_{\mathcal{R}}$, it follows from Theorem 3.2 that $\tilde{\mu}$ provides closed-loop stability and constraint satisfaction for all $x \in \Omega$. \square

Lemma 3.5 suggests that finding $\tilde{\mu}$ of the form (4) as a solution to Problem 3.1 can be cast as finding the coefficients $\alpha_1, \dots, \alpha_\delta$ such that polynomials $p_i(\alpha, x)$ are non-negative over corresponding regions. There is a subtle, yet very important issue which makes solving problem (7) far from straightforward: even for a fixed i , all polynomials $p_i(\cdot)$ associated to region \mathcal{R}_i must be non-negative for all points $x \in \mathcal{R}_i$, not just for some of them (e.g. for the vertices of \mathcal{R}_i). One approach is to employ the Positivstellensatz and show positivity of polynomials by solving a sum-of-squares problem, as suggested in Kvasnica et al. (2008). However, as documented in Kvasnica et al. (2010), such a procedure is, from a practical point of view, limited to small-scale problems only. A different direction is therefore persuaded here, which is based on the following theorem, originally formulated by Hardy, Littlewood, and Pólya (1952) to show strict positivity of polynomials and later extended to the non-strict case by Mok and To (2008).

Theorem 3.6 (Pólya's Theorem). If a homogeneous polynomial $p_i(\alpha, x)$ is non-negative over a unit simplex, then all the coefficients of the extended polynomial $p_i^M(\alpha, x) = p_i(\alpha, x) \cdot \left(\sum_{j=1}^{n_x} x_j\right)^M$ are non-negative for a sufficiently large Pólya degree M .

Remark 3.7. Search for the coefficients α , such that $p_i^M(\alpha, x)$ is non-negative over a simplex can be performed by using the more obvious reverse of Pólya's theorem, i.e. non-negative coefficients of the extended polynomial imply its non-negativity over the whole simplex.

Corollary 3.8. Given a symbolic representation of coefficients of $p_i^M(\alpha, x)$, the coefficients α of $\tilde{\mu}$ can be found by solving a linear program. To see this, observe that α enters (6) in a linear fashion per definition of $\tilde{\mu}$ as in (4). All constraints in (7) are therefore linear in α .

Note, however, that Theorem 3.6 is not directly applicable to find α from (7) as \mathcal{R}_i are not unit simplices with $0_{n_x} \in \mathcal{R}_i$, in general. Therefore, we propose to represent the polytopic regions in their equivalent vertex representation, i.e. by

$$\mathcal{R}_i = \left\{ x \mid x = \sum_{j=1}^{|\mathcal{V}_i|} \lambda_j [\mathcal{V}_i]_j, \lambda \in \Lambda_i \right\}, \quad (8a)$$

$$\Lambda_i = \left\{ \lambda \mid 0 \leq \lambda_j \leq 1, \sum_{j=1}^{|\mathcal{V}_i|} \lambda_j = 1 \right\}. \quad (8b)$$

Here, \mathcal{V}_i are the vertices the i -th region, $|\mathcal{V}_i|$ denotes their cardinality, $[\mathcal{V}_i]_j$ is the j -th vertex of \mathcal{R}_i , and $\lambda = [\lambda_1, \dots, \lambda_{|\mathcal{V}_i|}]$. By substituting for $x = \sum_j \lambda_j [\mathcal{V}_i]_j$ into (6) and (7), we get

$$p_i(\alpha, \lambda) \geq 0, \quad \forall \lambda \in \Lambda_i, \quad i = 1, \dots, N_{\mathcal{R}}. \quad (9)$$

Note that Λ_i in (9) are now $|\mathcal{V}_i|$ -dimensional unit simplices and Theorem 3.6 can therefore be applied to find α such that $p_i(\alpha, \lambda)$ is non-negative $\forall \lambda \in \Lambda_i$, $i = 1, \dots, N_{\mathcal{R}}$. Also note that such change of variables is needed even if all \mathcal{R}_i originally were simplices, since the Pólya's Theorem only applies if $0_{n_x} \in \mathcal{R}_i$.

We can now state the main result of the paper, which is Theorem 3.9 and Algorithm 1 for calculating values of the coefficients $\alpha_1, \dots, \alpha_\delta$ of the polynomial feedback law $\tilde{\mu}$ which is an admissible solution to Problem 3.1.

Algorithm 1 Polynomial approximation

INPUT: Optimal RHMPC feedback law μ , PWA Lyapunov function V , scalar $\gamma \in [0, 1)$, degree of the approximation polynomial δ , Pólya degree M .

OUTPUT: Coefficients $\alpha_1, \dots, \alpha_\delta$ of the polynomial feedback law (4).

- 1: Obtain the stability tube $\mathcal{S}(V, \gamma)$ per (5).
- 2: Calculate extremal vertices \mathcal{V}_i of all regions \mathcal{R}_i .
- 3: Formulate polynomials $p_i(\alpha, \lambda)$ per (8)–(9).
- 4: Homogenize $p_i(\alpha, \lambda)$ by multiplying single monomials by $\left(\sum_{j=1}^{|\mathcal{V}_i|} \lambda_j\right)$ until all monomials have the same degree.
- 5: Obtain symbolic representation of coefficients c_i^M of Pólya's polynomials $p_i^M(\alpha, \lambda) = p_i(\alpha, \lambda) \cdot \left(\sum_{j=1}^{|\mathcal{V}_i|} \lambda_j\right)^M$.
- 6: Search for α by solving a linear program:

$$\text{find } \alpha_1, \dots, \alpha_\delta, \quad (10a)$$

$$\text{s.t. } c_i^M \geq 0, \quad i = 1, \dots, N_{\mathcal{R}}. \quad (10b)$$

Theorem 3.9. Let the input arguments of Algorithm 1 satisfy Assumption 2.1 and assume that the tube $\mathcal{S}(V, \gamma)$ computed in Step 1 satisfies Assumption 3.4. If the LP (10a) is feasible, the polynomial feedback law $\tilde{\mu}$ of the form (4) calculated by Algorithm 1 is a solution to Problem 3.1.

Proof. If (10) is feasible, then, according to [Theorem 3.6](#), polynomials $p_i(\alpha, \lambda)$ are non-negative over corresponding regions \mathcal{R}_i . This in turn implies that (7) is satisfied, which, according to [Lemma 3.5](#), shows that $\tilde{\mu}(x)$ belongs to the stability tube $\mathcal{S}(V, \gamma)$, $\forall x \in \Omega$. Therefore, by [Theorem 3.2](#), $\tilde{\mu}$ is guaranteed to be closed-loop stabilizing and feasible for all time. \square

Remark 3.10. Algorithm 1 is a non-iterative procedure and therefore it always terminates in a single pass, provided that all of its steps are successful. However, since [Theorems 3.2](#) and [3.6](#) are only sufficient conditions for the existence of a stabilizing feedback $\tilde{\mu}$, the algorithm may fail to find it even if one exists.

Instead of a pure feasibility objective in (10a), an alternative is to minimize the point-wise distance $|\mu(x_j) - \tilde{\mu}(x)(x_j)|_q$ with $q \in \{1, \infty\}$ over some points x_j (e.g. over the vertices of each \mathcal{R}_i). Doing so will let $\tilde{\mu}$ to follow the shape of $\mu(x)$ more tightly, hence mitigating the induced loss of optimality. Another approach is to aim for low-order polynomials. This can be done in three ways: (i) minimize the ℓ_1 norm of α , which tends to give sparse solutions; (ii) use bisection in conjunction with Algorithm 1; or (iii) minimize the number of non-zero coefficients to a global minimum by solving a mixed-integer version of (10).

Example 3.11. Consider the following open-loop unstable PWA system ([Kvasnica et al., 2008](#)):

$$x_{t+1} = \begin{cases} 6/5x_t - 2u_t & \text{if } x_t \geq 0, \\ -4/5x_t + u_t & \text{otherwise,} \end{cases} \quad (11)$$

with $u_t \in [-1, 1]$ and $x_t \in [-4, 4]$. With the choice of $p = 1$, $Q_x = 1$, $Q_u = 1$, $N = \infty$ in (2) we obtain a stabilizing feedback μ and a Lyapunov function J_N^* defined over 6 regions. The stability tube $\mathcal{S}(V, \gamma)$ for $V := J_N^*$ and $\gamma = 0.7$, the optimal RHMPc feedback $\mu(x_t)$, and its polynomial approximations $\tilde{\mu}$ of degrees $\delta = 3, 5, 7$ computed by Algorithm 1 are shown in [Fig. 1](#).

4. Complexity analysis

4.1. Complexity of Algorithm 1

Computation of stability tubes $\mathcal{S}(V, \gamma)$ in Step 1 can be done in $\mathcal{O}(N_{\mathcal{R}}^2)$ time, since all possible transitions between various regions have to be investigated. The LP in Step 6 has $n_u n_x \delta$ variables (the coefficients $\alpha_1, \dots, \alpha_\delta$) and $\mathcal{O}(N_c)$ constraints. Here, $N_c = \sum_{i=1}^{N_{\mathcal{R}}} N_{c,i}$ is the total number of coefficients of Pólya polynomials $p_i^M(\alpha, \lambda)$, with $N_{c,i} = \binom{\delta_p + |\mathcal{V}_i| - 1}{\delta_p}$ and $\delta_p = \delta + M$, where $|\mathcal{V}_i|$ is the number of vertices of the i -th region.

Remark 4.1. The number of Pólya's coefficients, and hence the number of constraints of the LP (10), grows quickly with the number of states n_x . In the most general case, $|\mathcal{V}_i| = \mathcal{O}(2^{n_x})$. This is the main bottleneck of the presented procedure. One way to mitigate such a quick growth is to triangulate the regions \mathcal{R}_i . Although this will increase the total number of regions to, at most, $\mathcal{O}\left(\sum_{i=1}^{N_{\mathcal{R}}} |\mathcal{V}_i|^{\lceil n_x/2 \rceil}\right)$, the gained advantage is that $|\mathcal{V}_i|$ stays fixed at $n_x + 1$, $\forall i$. From numerical experiments, we have observed that employing triangulation reduces the total number of constraints in (10) by a factor of 5, on average.

Remark 4.2. Another option to reduce the size of the linear program (10) is to eliminate the redundant constraints. Full redundancy elimination would require solving $\mathcal{O}(N_c)$ copies of the LP (10), which clearly is not an option. Note, however, that a valid solution to (10) has to be non-negative due to [Theorem 3.6](#) and

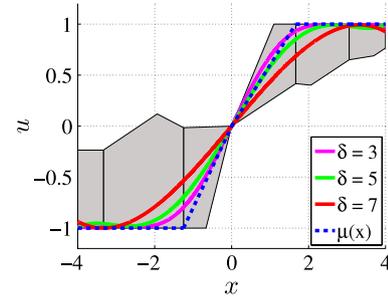


Fig. 1. Stability tube $\mathcal{S}(V, \gamma)$ for $\gamma = 0.7$ (gray sets), optimal control law μ (blue dashed line), and stabilizing polynomials $\tilde{\mu}$ of different degrees δ . (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Remark 3.7. Therefore, eliminating from (10b) those constraints where all constant multipliers in c_i^M are non-negative will not affect feasibility. Numerical examples suggest that such a simple elimination reduces the number of constraints of (10) by a factor of 2, on average.

4.2. On-line complexity

Implementing $\tilde{\mu}$ in a feedback arrangement reduces to a mere evaluation of the polynomial for a given x . Since the polynomial continuously covers the whole state-space of interest, no region search is necessary. Using *Horner's* scheme ([Eve, 1964](#)), $\tilde{\mu}$ can be evaluated³ by at most $1/2 n_u n_x (3\delta + 5)$ FLOPS. Storing the coefficients $\alpha_1, \dots, \alpha_\delta$ consumes $\delta n_u n_x$ floating point numbers. On the other hand, evaluating the optimal feedback law μ via a binary search tree ([Tøndel et al., 2003](#)) requires $\mathcal{O}(\log_2 N_{\mathcal{R}})$ FLOPS and the tree consumes $\mathcal{O}(N_{\mathcal{R}}(n_x + n_u))$ memory elements. Complexity of the lattice representation ([Wen et al., 2009](#)), both in terms of runtime and memory, is $\mathcal{O}(N_{\mathcal{U}}^2)$ where $N_{\mathcal{U}}$ denotes the number of unique feedback laws.

5. Examples

5.1. Standard PWA benchmark

Consider the following PWA system with 2 states and one input, introduced in [Bemporad and Morari \(1999\)](#): $x^+ = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$, where the value of θ switches depending on the value of the first element of the state vector: $\theta = -\pi/3$ if $x_1 \leq 0$, and $\theta = \pi/3$, otherwise. State constraints $|x_i| \leq 5$, $i = 1, 2$ are assumed, along with input bounds $|u| \leq 1$. Even though the system is open-loop stable, a controller is needed to guarantee constraint satisfaction for all time. The explicit RHMPc feedback law μ was constructed by solving (2) with $Q_x = \mathbb{1}$, $Q_u = 1$, $p = \infty$ and $N = \infty$, and consists of 112 regions shown in [Fig. 2\(a\)](#). We have then applied Algorithm 1 to find approximate feedbacks $\tilde{\mu}$ of degrees $\delta = 1, \dots, 7$. The stability tube in Step 1 was computed for $\gamma = 0.99$ and it satisfied [Assumption 3.4](#). The polynomial of degree 7 is shown in [Fig. 2\(b\)](#). To assess the induced loss of optimality, we have analyzed closed-loop profiles of states and inputs. The performance degradation is given by $\Delta_J := (J^* - \tilde{J})/J^*$, where J^* is the value of (2a) for a closed-loop profile obtained by applying the optimal feedback $\mu(x)$, while \tilde{J} is the cost of the closed-loop evolution driven by $u = \tilde{\mu}(x)$. The average values of Δ_J over 1000 equidistantly spaced initial conditions are reported in [Table 1](#).

³ If only fixed-point arithmetics is available, evaluation can be done as in [Brisebarre, Chevillard, Ercegovac, Muller, and Torres \(2008\)](#).

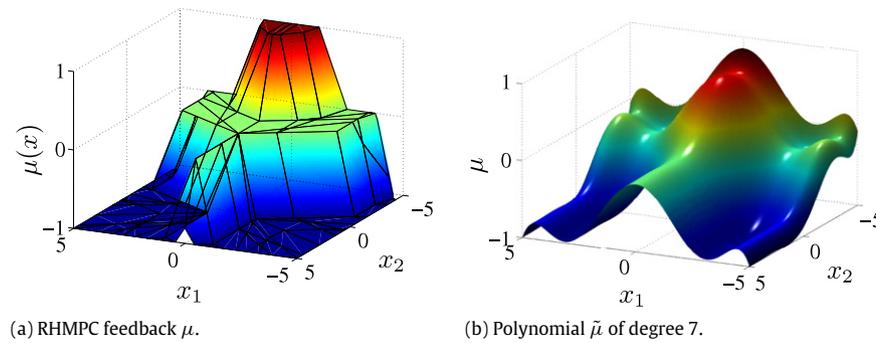


Fig. 2. Optimal RHMPC feedback law and its approximation.

Table 1
Performance degradation for various degrees of the polynomial approximation $\tilde{\mu}$.

δ	1	2	3	4	5	6	7
Δ_J (%)	48	46	43	37	36	28	28

5.2. Linear system

Consider the following linear system with 4 states and 1 input:

$$x^+ = \begin{bmatrix} 0.7 & -0.1 & 0.0 & 0.0 \\ 0.2 & -0.5 & 0.1 & 0.0 \\ 0.0 & 0.1 & 0.1 & 0.0 \\ 0.5 & 0.0 & 0.5 & 0.5 \end{bmatrix} x + \begin{bmatrix} 0.1 \\ 1.0 \\ 0.0 \\ 0.0 \end{bmatrix} u,$$

which is subject to constraints $|x_i| \leq 5$, $i = 1, \dots, 4$, and $|u| \leq 5$. The optimal RHMPC feedback law for $Q_x = 10 \cdot \mathbb{1}_4$, $Q_u = 0.1$, and $N = 3$ has 230 regions in 4D state space. The lowest feasible degree of $\tilde{\mu}$ was $\delta = 1$, leading to the linear feedback $\tilde{\mu}(x) = -0.0715x_1$. It follows that while the on-line implementation of the optimal RHMPC controller would require the storage of 9290 floating point numbers (8140 for describing the 230 regions, and 1628 for the associated feedback laws), the polynomial feedback requires storing exactly one floating point number at the price of a 35% worst-case drop of performance.

This case also illustrates practical consequences of Remarks 4.1 and 4.2. Without any of them applied, the LP (10) for $\delta = 3$ would have $5.8 \cdot 10^6$ constraints, which is above the limit of most LP solvers.⁴ Performing triangulation per Remark 4.1 led to $1.8 \cdot 10^6$ inequalities. Further elimination of trivially redundant constraints per Remark 4.2 decreased this figure to $0.9 \cdot 10^6$.

5.3. Random systems

To assess versatility of the presented approach, we have analyzed random PWA systems with 2 dynamics under state constraints $|x| \leq 5$ and input bounds $|u| \leq 1$. Three batches of random systems of various dimensions were considered, with 100 systems in each batch. For each system the optimal RHMPC feedback law μ was computed⁵ by solving (2) with $Q_x = \mathbb{1}_{n_x}$, $Q_u = \mathbb{1}_{n_u}$, and $N = 5$. Q_N and X_f were designed as in Baotić et al. (2006). Subsequently, the stability tubes $\mathcal{S}(V, \gamma)$ were constructed for $V := J_N^*$ and $\gamma = 0.99$. In 62% of the 300 investigated problems the respective stability tubes satisfied Assumption 3.4. Important to notice is that the success rate was 97% when investigating a supplemental batch of 100 random linear systems.

Table 2
Data for random systems.

n_x/n_u	$N_{\mathcal{R}}$	$N_{\mathcal{T}}$	Step 1 (s)	Step 6 (s)	δ_{\min}	Δ_J (%)
2/1	117	215	28	1	1	77
	180	329	54	2	5	23
	251	485	103	1	3	51
	288	552	152	2	6	37
2/2	115	206	62	3	2	81
	197	379	135	1	1	65
	277	522	164	1	5	35
	376	738	328	7	4	42
3/1	270	1258	293	3	3	12
	450	1755	551	1	1	72
	606	2963	829	12	6	41
	834	4278	1453	3	4	65

The tubes were then triangulated according to Remark 4.1 and further processed by Algorithm 1. The runtime of triangulation never exceeded 20 s for any of the investigated examples. Enumeration of vertices in Step 2 never took more than 1 s using MPT (Kvasnica, Grieder, & Baotić, 2004). The LP in Step 6 was formulated by YALMIP (Löfberg, 2004) and solved by CPLEX 12.1 (ILOG, 0000). Only degrees up to 7 were investigated due to practical reasons. The success rate of the LP-based procedure was 81%. No obvious correlation between the number of regions of μ and the required degree δ in (10) was observed. Around 25% of all feasible cases admitted the existence of a linear approximation $\tilde{\mu}$, regardless of n_x and n_u . Higher order approximations with minimal feasible degrees $\delta = 2, \dots, 6$ appeared with a roughly equal distribution.

A representative selection of the results is reported in Table 2 which shows how the computation scales with increasing problem size. Columns of the tables denote, respectively, state and input dimensions, number of regions $N_{\mathcal{R}}$, number of triangulated regions $N_{\mathcal{T}}$, runtime of construction of the stability tube in Step 1, runtime of the LP in Step 6, minimum degree δ_{\min} for which the LP was feasible, and the average performance degradation induced by using $\tilde{\mu}$ of the minimal degree. Even though the average performance drop Δ_J might sound large, one has to take into account three facts. First, as discussed previously, performance usually improves if δ is enlarged. Second, and more importantly, design of any stabilizing feedback controller for PWA systems is a non-trivial task, even putting optimality aside. Finally, magnitudes of the reported performance drops are similar to what can be achieved by other techniques; see e.g. Bemporad, Oliveri, Poggi, and Storace (2010) and Hovd et al. (2009).

6. Conclusions

We have presented a novel way of deriving simple stabilizing feedback laws for the class of constrained linear and PWA systems.

⁴ For instance, the 32-bit version of CPLEX only allows around $2 \cdot 10^6$ constraints.

⁵ On a 2.5 GHz CPU and 2 GB of RAM using Matlab R2009a and MPT 2.6.3.

Stability and feasibility of the approximate polynomial controllers are guaranteed by employing the concept of stability tubes, which can be viewed as a parameterization of stabilizing feedback laws. It was illustrated that coefficients of the polynomials can be found by solving a single linear program. Triangulation and a cheap redundancy elimination were proposed as a way to significantly mitigate the size of the LP, hence allowing to process even large problems. Although the presented procedure inherently induces sub-optimality, the synthesized polynomial feedback not only guarantees stability and constraint satisfaction, but also puts very low requirements on its implementation in real time.

Certain restrictions have to be imposed on the shape of stability tubes in order to be able to find the approximation by solving a single LP. Investigation of a large number of random cases showed that a suitable tube was found in 60% of PWA systems, while the success rate is close to 100% when considering linear systems. If the tube has “unfavorable” shape, one would need to resort to a piecewise polynomial nature of the approximation. Although no obvious correlation between the number of elements of the tube and the degree of the approximate polynomial was observed, it cannot be ruled out that higher order polynomials might be necessary to approximate more complex tubes.

References

- Baotić, M., Christophersen, F. J., & Morari, M. (2006). constrained optimal control of hybrid systems with a linear performance index. *IEEE Transactions on Automatic Control*, 51(12), 1903–1919.
- Bemporad, A., & Filippi, C. (2003). Suboptimal explicit RHC via approximate multiparametric quadratic programming. *Journal of Optimization Theory and Applications*, 117(1), 9–38.
- Bemporad, A., & Morari, M. (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3), 407–427.
- Bemporad, A., Morari, M., Dua, V., & Pistikopoulos, E. N. (2002). The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1), 3–20.
- Bemporad, A., Oliveri, A., Poggi, T., & Storaice, M. (2010). Synthesis of stabilizing model predictive controllers via canonical piecewise affine approximations. In *Conference on decision and control. CDC. Atlanta, USA* (pp. 5296–5301).
- Borrelli, F. (2003). *Lecture notes in control and information sciences: Vol. 290. Constrained optimal control of linear and hybrid systems*. Springer-Verlag.
- Brisebarre, N., Chevillard, S., Ercegovic, M.D., Muller, J.-M., & Torres, S. (2008). An efficient method for evaluating polynomial and rational function approximations. In *Proceedings of the ASAP 2008 conference* (pp. 233–238).
- Christophersen, F. J. (2007). *Lecture notes in control and information sciences: Vol. 359. Optimal control of constrained piecewise affine systems*. Springer Verlag.
- Eve, J. (1964). The evaluation of polynomials. *Numerische Mathematik*, 6, 17–21.
- Geyer, T., Torrisi, F. D., & Morari, M. (2008). Optimal complexity reduction of polyhedral piecewise affine systems. *Automatica*, 44(7), 1728–1740.
- Hardy, G. H., Littlewood, J. E., & Pólya, G. (1952). *Inequalities* (second ed.) Cambridge University Press.
- Hovd, M., Scibilia, F., Maciejowski, J., & Oлару, S. (2009). Verifying stability of approximate explicit MPC. In *IEEE conference on decision and control. Shanghai, China* (pp. 6345–6635).
- Hovland, S., Willcox, K. E., & Gravdahl, J. T. (2008). Explicit MPC for large-scale systems via model reduction. *AIAA Journal of Guidance, Control and Dynamics*, 31(4).
- ILOG, Inc. CPLEX user manual. Gentilly Cedex, France. <http://www.ilog.fr/products/cplex/>.
- Johansen, T. A., & Grancharova, A. (2003). Approximate explicit constrained linear model predictive control via orthogonal search tree. *IEEE Transactions on Automatic Control*, 48, 810–815.
- Kvasnica, M., Christophersen, F.J., Hecceg, M., & Fikar, M. (2008). Polynomial approximation of closed-form MPC for piecewise affine systems. In *Proceedings of the 17th IFAC world congress. Seoul, Korea* (pp. 3877–3882).
- Kvasnica, M., & Fikar, M. (2010). Performance-lossless complexity reduction in explicit MPC. In *Conference on decision and control. CDC. Atlanta, USA* (pp. 5270–5275).
- Kvasnica, M., Grieder, P., & Baotić, M. (2004). Multi-parametric toolbox (MPT). Available from: <http://control.ee.ethz.ch/~mpt/>.
- Kvasnica, M., Löfberg, J., Hecceg, M., Čirka, L., & Fikar, M. (2010). Low-complexity polynomial approximation of explicit MPC via linear programming. In *Proceedings of the American control conference. Baltimore, USA* (pp. 4713–4718).
- Lazar, M., de la Pena, D., Munoz, Heemels, W. P. M. H., & Alamo, T. (2008). On input-to-state stability of min-max nonlinear model predictive control. *Systems & Control Letters*, 57, 39–48.
- Löfberg, J. (2004). YALMIP: a toolbox for modeling and optimization in MATLAB. In *Proc. of the CACSD conference. Taipei, Taiwan*. Available from: <http://users.isy.liu.se/johanl/yalmip/>.
- Mok, H., & To, W. (2008). Effective Pólya semi-positivity for non-negative polynomials on the simplex. *Journal of Complexity*, 24(4), 524–544.
- Tøndel, P., & Johansen, T.A. (2002). Complexity reduction in explicit linear model predictive control. In *Proc. of 15-th IFAC world congress*.
- Tøndel, P., Johansen, T. A., & Bemporad, A. (2003). Evaluation of piecewise affine control via binary search tree. *Automatica*, 39(5), 945–950.
- Wen, Ch., Ma, X., & Ydstie, B. E. (2009). Analytical expression of explicit MPC solution via lattice piecewise-affine function. *Automatica*, 45(4), 910–917.



Michal Kvasnica was born in 1977. He received his diploma in chemical engineering from the Slovak University of Technology in Bratislava, Slovakia and the Ph.D. in electrical engineering from the Swiss Federal Institute of Technology in Zurich, Switzerland. Since 2006 he is an Assistant Professor at the Slovak University of Technology in Bratislava. His research interests are in model predictive control, modeling of hybrid systems, and development of software tools for control. He is the co-author and developer of the MPT Toolbox for explicit model predictive control.



language YALMIP.

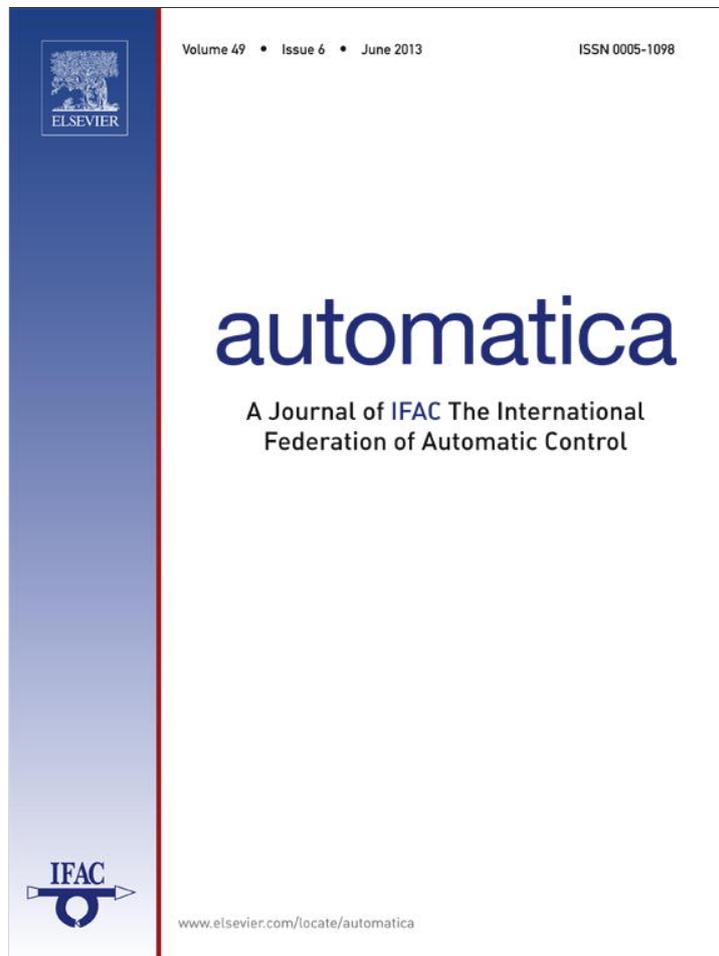
Johan Löfberg was born in Jönköping, Sweden in 1974. He received his M.Sc. degree in Mechanical Engineering in 1998 and the Ph.D. degree in Automatic Control in 2003, both from Linköping University, Linköping, Sweden. He held a post-doctoral position at the Automatic Control Laboratory at ETH Zurich, Switzerland 2003–2006. From 2006, he works as an Assistant Professor at Linköping University. His research interests include optimal and robust control in a model predictive control framework, and general aspects related to the use of optimization in control. He is the author and developer of the modeling



chemical process control.

Miroslav Fikar was born in Bratislava, Slovakia, in 1966 and received his ME and Ph.D. degrees in Chemical Engineering from the Slovak University of Technology in Bratislava in 1989 and 1994, respectively. He has stayed with the Faculty of Chemical and Food Technology STU where he is currently a professor and institute director. He was Postdoc Fellow in Nancy, France (Elf Aquitaine, Ministère de la Recherche), Alexander von Humboldt Fellow in Bochum, Germany and has spent several stays in Denmark, Germany, France, Switzerland. His current research interests include optimal control, MPC, and

Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.



This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

<http://www.elsevier.com/authorsrights>



Contents lists available at SciVerse ScienceDirect

Automatica

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

Brief paper

Complexity reduction of explicit model predictive control via separation[☆]Michal Kvasnica^{a,b,1}, Juraj Hledík^c, Ivana Rauová^a, Miroslav Fikar^a^a Faculty of Chemical and Food Technology, Slovak University of Technology in Bratislava, Slovakia^b Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic^c Vienna University of Economics and Business, Vienna, Austria

ARTICLE INFO

Article history:

Received 21 February 2012

Received in revised form

15 August 2012

Accepted 18 January 2013

Available online 14 March 2013

Keywords:

Model predictive control

Constrained control

Parametric optimization

ABSTRACT

The problem of reducing complexity of explicit MPC feedback laws for linear systems is considered. We propose to simplify controllers defined by continuous Piecewise Affine (PWA) functions by employing separating functions. If a state resides in a region where the optimal control action attains a saturated value, the optimal control move is determined from the sign of the separator. Thus, instead of storing all regions, only the unconstrained regions and the separator are needed. We propose several approaches to construct separators with different efficacy and properties.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

Implementation of MPC in the Receding Horizon fashion (RHMP) boils down to repetitively solving, at each sampling instance, an optimization problem initialized by the current state measurements x . As shown in Bemporad, Morari, Dua, and Pistikopoulos (2002), for MPC problems of modest size one can precompute the explicit RHMP optimizer $u^* = \kappa(x)$ as a PWA function κ which is defined over a set of polytopic regions. Computing u^* on-line then reduces to a mere function evaluation, which can be done quickly on embedded hardware. However, the number of regions of κ , which is problem-dependent, tends to be large, easily exceeding the storage capacity of the hardware. Therefore it is important to keep the number of regions as low as possible. One way to reduce the complexity of κ is to construct a sub-optimal replacement function $\tilde{\kappa}$ such that $\tilde{\kappa}(x) \approx \kappa(x)$, see e.g. Bemporad, Oliveri, Poggi, and Storace (2011), Johansen and Grancharova (2003), Jones and Morari (2010) and Scibilia, Oлару, and

Hovd (2009). Another option is to find a replacement $\tilde{\kappa}$ which is not only simpler than the original function, but also maintains the equivalence $\kappa(x) = \tilde{\kappa}(x)$ for all $x \in \text{dom}(\kappa)$. Employing $\tilde{\kappa}$ instead of κ therefore does not incur any loss of optimality and performance. In Geyer, Torrisi, and Morari (2008) such a replacement is constructed by merging together regions which share the same expression of the optimal feedback law. The downside being that merging regions optimally is of combinatorial complexity. If κ is a continuous PWA function, its lattice representation (Wen, Ma, & Ydstie, 2009) can be built, leading to a replacement $\tilde{\kappa}$ of smaller complexity. Evaluation of κ on-line for a given value of x can be accelerated by constructing a search tree (Tøndel, Johansen, & Bemporad, 2003). However, creating such trees can be prohibitive for large numbers of regions. In our related work (Kvasnica & Fikar, 2012), the equivalent replacement $\tilde{\kappa}$ was constructed by removing some of the so-called *saturated regions* from the definition of κ , followed by applying a clipping filter to restore equivalence.

Following our preliminary work (Kvasnica, Rauová, & Fikar, 2011), in this paper we present a novel approach to reducing complexity of explicit RHMP feedback laws described by continuous PWA functions. The central idea is that typical explicit RHMP feedbacks usually contain many regions where the optimal control action is either constantly on the upper limit or constantly on the lower limit. These two sets of regions will be denoted by \mathcal{R} and $\underline{\mathcal{R}}$, respectively. We show that if there exists a function ξ which strictly separates $\overline{\mathcal{R}}$ and $\underline{\mathcal{R}}$, then these two sets of regions can be completely removed from the definition of κ . Finding such a strict separator, however, is not trivial, since the unions of polytopes $\overline{\mathcal{R}}$ and $\underline{\mathcal{R}}$ are non-convex, in general. In Kvasnica et al. (2011) we have

[☆] The authors are pleased to acknowledge the financial support of the Scientific Grant Agency of the Slovak Republic under grant 1/0095/11 and the contribution of the Slovak Research and Development Agency under project APVV 0551-11. M. Kvasnica was also supported by the Ministry of Education of the Czech Republic under the Centralized Project for University Development CSM 100 TALENT. The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Lalo Magni under the direction of Editor Frank Allgöwer.

E-mail addresses: michal.kvasnica@stuba.sk (M. Kvasnica), juraj.hledik@vgsf.ac.at (J. Hledík), iva.rauova@gmail.com (I. Rauová), miroslav.fikar@stuba.sk (M. Fikar).

¹ Tel.: +421 2 59325352; fax: +421 2 59325340.

shown how to find, off-line, a strict separator in the form of a multivariate polynomial by employing an iterative procedure with two considerable drawbacks: (1) at each iteration a nonlinear programming problem had to be solved, and (2) no guarantees of finite-time convergence could be given. In this work we address these limitations and show how to find the separating polynomial by solving a single linear program (LP). Moreover, we also present a procedure which implicitly defines the value of $\xi(x)$ for a given point x . The advantage of this method is twofold. First, the off-line preprocessing is virtually zero. More important, though, is the fact that such an implicit separation guarantees complexity reduction even when no analytic form of the separator could be found.

Notation and definitions

A finite set of n elements $\mathcal{I} := \{\mathcal{I}_1, \dots, \mathcal{I}_n\}$ will be denoted as $\{\mathcal{I}_i\}_{i=1}^n$ and its cardinality by $|\mathcal{I}|$. The interior of a set \mathcal{R} is denoted by $\text{int}(\mathcal{R})$. Given a function κ , $\text{dom}(\kappa)$ denotes its domain. A polytope is the bounded convex intersection of finitely many closed affine half-spaces, i.e., $\mathcal{R} := \{x \in \mathbb{R}^{n_x} \mid Fx \leq g\}$. We call the collection of polytopes $\{\mathcal{R}_i\}_{i=1}^R$ the *partition* of a polytope \mathcal{R} if $\mathcal{R} = \bigcup_{i=1}^R \mathcal{R}_i$, and $\text{int}(\mathcal{R}_i) \cap \text{int}(\mathcal{R}_j) = \emptyset$ for all $i \neq j$. Each polytope \mathcal{R}_i will be referred to as the *region* of the partition. The function $\kappa : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_u}$ with $x \in \mathcal{R} \subset \mathbb{R}^{n_x}$, \mathcal{R} being a polytope, is called piecewise affine over polytopes if $\{\mathcal{R}_i\}_{i=1}^R$ is the partition of \mathcal{R} and $\kappa(x) := K_i x + L_i$ if $x \in \mathcal{R}_i$, with $K_i \in \mathbb{R}^{n_u \times n_x}$, $L_i \in \mathbb{R}^{n_u}$, and $i = 1, \dots, R$. The PWA function κ is continuous if $K_i x + L_i = K_j x + L_j$ holds $\forall x \in \mathcal{R}_i \cap \mathcal{R}_j$, $i \neq j$.

2. Preliminaries and problem statement

2.1. Explicit model predictive control

We consider the class of discrete-time, stabilizable linear time-invariant systems

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

which are subject to polytopic constraints $x \in \mathbb{X} \subset \mathbb{R}^{n_x}$ and $u \in \mathbb{U} \subset \mathbb{R}^{n_u}$. Assume the following constrained finite-time optimal control problem:

$$\min_{u_N} \sum_{k=0}^{N-1} x_{k+1}^T Q_x x_{k+1} + u_k^T Q_u u_k \quad (2a)$$

$$\text{s.t. } x_{k+1} = Ax_k + Bu_k, \quad x_k \in \mathbb{X}, \quad u_k \in \mathbb{U}, \quad (2b)$$

where x_k and u_k denote, respectively, the k -th step state and input predictions over a finite horizon N , given the initial condition $x_0 = x(t)$ where $x(t)$ is the state measured at time t . It is assumed that $Q_x = Q_x^T \geq 0$, $Q_u = Q_u^T > 0$ in (2a), i.e., that (2) is a strictly convex QP. The receding horizon MPC feedback then becomes $u^* = [\mathbf{1} \ 0 \ \dots \ 0] U_N^*$, where $U_N^* := [u_0^T, \dots, u_{N-1}^T]^T$ is the optimal solution to (2). For problems of modest size, it is possible to characterize the optimal feedback explicitly as a PWA function of x by solving (2) as a *parametric quadratic program* (pQP):

Theorem 2.1 (Bemporad et al., 2002). *The RHMPC feedback for problem (2) is given by $u^* = \kappa(x)$ where:*

1. The set of feasible initial conditions Ω is a polytope.
2. $\kappa : \Omega \rightarrow \mathbb{R}^{n_u}$ is a continuous PWA function defined over R regions \mathcal{R}_i , $i = 1, \dots, R$:

$$\kappa(x) = K_i x + L_i \quad \text{if } x \in \mathcal{R}_i. \quad (3)$$
3. Regions \mathcal{R}_i are full-dimensional polytopes.
4. $\{\mathcal{R}_i\}_{i=1}^R$ is a partition of Ω .

² To simplify the notation, we will henceforth abbreviate $x(t)$, the initial condition of (2), by x .

2.2. Problem statement

We aim at replacing the RHMPC feedback function κ in (3) by a different function $\tilde{\kappa}$ which meets two requirements:

1. $\tilde{\kappa}$ is equivalent to κ in the sense that $\tilde{\kappa}(x) = \kappa(x)$ for all $x \in \Omega$,
2. $\tilde{\kappa}$ is simpler than κ , i.e., it consists of fewer regions.

If such a replacement function $\tilde{\kappa}$ can be found, one can implement the explicit RHMPC feedback using smaller amounts of memory and using fewer computations.

3. Complexity reduction via separation

Denote by $\bar{\kappa}$ and $\underline{\kappa}$ the maximal and minimal values which κ attains over its domain $\Omega = \text{dom}(\kappa)$ (Kvasnica & Fikar, 2012)

$$\bar{\kappa} = \max\{\kappa(x) \mid x \in \Omega\}, \quad \underline{\kappa} = \min\{\kappa(x) \mid x \in \Omega\}. \quad (4)$$

Then each region \mathcal{R}_i of the domain $\Omega = \{\mathcal{R}_i\}_{i=1}^R$ can be classified as follows. If $K_i = 0$ and $L_i = \bar{\kappa}$, then region \mathcal{R}_i is *saturated at the maximum*. If $K_i = 0$ and $L_i = \underline{\kappa}$, then region \mathcal{R}_i is *saturated at the minimum*. Otherwise the region is called an *unsaturated* region. Denote by \mathcal{I}_{\max} and \mathcal{I}_{\min} the index lists of regions saturated at the maximum and minimum, respectively, and by $\mathcal{I}_{\text{unsat}}$ the index list of unsaturated regions. Let $\mathcal{U} = \{\mathcal{R}_i\}_{i \in \mathcal{I}_{\text{unsat}}}$, $\bar{\mathcal{R}} = \{\mathcal{R}_i\}_{i \in \mathcal{I}_{\max}}$, and $\underline{\mathcal{R}} = \{\mathcal{R}_i\}_{i \in \mathcal{I}_{\min}}$. With this classification, we can rewrite (3) as

$$\kappa(x) = \begin{cases} K_i x + L_i & \text{if } x \in \mathcal{U}, \quad i \in \mathcal{I}_{\text{unsat}} \\ \bar{\kappa} & \text{if } x \in \bar{\mathcal{R}}, \\ \underline{\kappa} & \text{if } x \in \underline{\mathcal{R}}. \end{cases} \quad (5)$$

Lemma 3.1. *Let a function $\xi : \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ which satisfies*

$$\xi(x) > 0, \quad \forall x \in \bar{\mathcal{R}}, \quad (6a)$$

$$\xi(x) < 0, \quad \forall x \in \underline{\mathcal{R}}. \quad (6b)$$

be given. Define

$$\tilde{\kappa}(x) = \begin{cases} K_i x + L_i & \text{if } x \in \mathcal{U}, \quad i \in \mathcal{I}_{\text{unsat}} \\ \bar{\kappa} & \text{if } x \notin \mathcal{U}, \quad \xi(x) > 0, \\ \underline{\kappa} & \text{if } x \notin \mathcal{U}, \quad \xi(x) < 0. \end{cases} \quad (7)$$

Then $\tilde{\kappa}(x) = \kappa(x)$ for all $x \in \text{dom}(\kappa)$.

Proof. Follows directly from (5) and from the definition of ξ in (6). We remark that, since polytopes \mathcal{R}_i do not overlap due to Theorem 2.1, we have $\text{dom}(\kappa) = \bar{\mathcal{R}} \cup \mathcal{U} \cup \underline{\mathcal{R}}$, $\text{int}(\bar{\mathcal{R}}) \cap \text{int}(\mathcal{U}) = \emptyset$, $\text{int}(\mathcal{U}) \cap \text{int}(\underline{\mathcal{R}}) = \emptyset$. Moreover, due to continuity of κ we have $\bar{\mathcal{R}} \cap \underline{\mathcal{R}} = \emptyset$. \square

Lemma 3.1 indicates that if we are able to find the function ξ that strictly separates sets $\bar{\mathcal{R}}$ and $\underline{\mathcal{R}}$, then κ can be evaluated by only looking at the unsaturated regions \mathcal{U} . If $\exists r \in \mathcal{I}_{\text{unsat}}$ such that $x \in \mathcal{R}_r$, then $\kappa(x) = K_r x + L_r$. Otherwise, based on the sign of $\xi(x)$, one either takes $\kappa(x) = \bar{\kappa}$ or $\kappa(x) = \underline{\kappa}$. As will be evidenced later, a typical explicit RHMPC feedback law κ contains a significantly smaller number of unsaturated regions as compared to the number of saturated ones, i.e., $|\mathcal{I}_{\text{unsat}}| \ll |\mathcal{I}_{\max}| + |\mathcal{I}_{\min}|$. Therefore $\tilde{\kappa}$ will require significantly less memory than κ , and will be faster to evaluate too, if ξ is a “simple” separator of the two sets $\bar{\mathcal{R}}$ and $\underline{\mathcal{R}}$.

Efficiency of the presented procedure depends on the ratio of unsaturated regions to the total number of regions. If κ does not contain any saturated regions, then no simplification can be achieved. As observed e.g. in Grieder and Morari (2003) and Kvasnica and Fikar (2012), the number of unsaturated regions depends mainly on two factors: tightness of input constraints \mathbb{U} and selection of the input penalty Q_u in (2a). The tighter the constraints and/or the lower Q_u , the more regions will become saturated, hence enabling our approach to be more efficient.

All procedures of this paper are applicable to generic PWA functions $\kappa : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_u}$ as long as they are continuous and all their regions \mathcal{R}_i are full-dimensional polytopes. The scope of this work therefore extends to cases where 1- or ∞ -norms are used in (2a), or when tracking of a non-zero reference is achieved by a suitable augmentation of the state vector. If $n_u > 1$, then κ can be decomposed (Kvasnica & Fikar, 2012) into n_u scalar-valued functions $\kappa_j : \mathbb{R}^{n_x} \rightarrow \mathbb{R}, j = 1, \dots, n_u$, where each κ_j is defined over the original partition $\{\mathcal{R}_i\}_{i=1}^R$. Denote by $\bar{\kappa}_j$ and $\underline{\kappa}_j$ the maximum and minimum of κ_j per (4), and let $\bar{\kappa} = [\bar{\kappa}_1, \dots, \bar{\kappa}_{n_u}]^T, \underline{\kappa} = [\underline{\kappa}_1, \dots, \underline{\kappa}_{n_u}]^T$. Then one option to select saturated regions is to pick regions where all scalar-valued functions are jointly saturated either at maximum (i.e., $\kappa(x) = \bar{\kappa}$) or at minimum (i.e., $\kappa(x) = \underline{\kappa}$). Another option is to apply Lemma 3.1 to each component κ_j individually. The latter approach is typically more effective (Kvasnica & Fikar, 2012).

4. Finding the separator

Problem 4.1. Given are two collections of polytopes $\bar{\mathcal{R}} = \{\mathcal{R}_i\}_{i \in \mathcal{I}_{\max}}$ and $\underline{\mathcal{R}} = \{\mathcal{R}_i\}_{i \in \mathcal{I}_{\min}}$ with $\bar{\mathcal{R}} \cap \underline{\mathcal{R}} = \emptyset$. Find the separating function $\xi : \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ which satisfies (6).

If ξ is chosen as a linear function, then its parameters can be found using standard support vector machine (SVM) approaches (Cortes & Vapnik, 1995). However, as pointed out in Section 6, many practical cases require nonlinear types of separators. The difficulty of devising a suitable nonlinear separator then stems from the fact that (6) has to hold for all points from the (in general non-convex) sets $\bar{\mathcal{R}}$ and $\underline{\mathcal{R}}$, not just for some points (e.g., for the vertices). Hence SVM-like approaches are not directly applicable in such cases.

4.1. Polynomial separation

In this section we show how to find coefficients of a multivariate polynomial

$$\xi(x) := \sum_{i_1+\dots+i_n \leq \delta} \alpha_{i_1, \dots, i_n} x_1^{i_1} \dots x_n^{i_n} \quad (8)$$

of pre-specified maximal degree δ such that ξ of (8) satisfies (6). The proposed approach is based on the fundamental result of Pólya:

Theorem 4.2 (Pólya's Theorem Hardy, Littlewood, & Pólya, 1952). Let Δ be a n_λ -dimensional unit simplex

$$\Delta = \left\{ \lambda \in \mathbb{R}^{n_\lambda} \mid \lambda \geq 0, \sum_{k=1}^{n_\lambda} \lambda_k = 1 \right\}, \quad (9)$$

and let ξ be a homogeneous polynomial. Then $\xi(\lambda) > 0 \forall \lambda \in \Delta$ if all coefficients of the extended polynomial

$$\tilde{\xi}(\lambda) := \xi(\lambda) \cdot \left(\sum_{k=1}^{n_\lambda} \lambda_k \right)^P \quad (10)$$

are positive for a sufficiently large Pólya degree P . \square

Despite being only a sufficient condition for positivity of a polynomial over a simplex, the advantage of Pólya's theorem is that coefficients of ξ can be found by solving a linear programming problem. However, Theorem 4.2 cannot be directly applied to solve Problem 4.1 since polytopes \mathcal{R}_i are not unit simplices with $0 \in \mathcal{R}_i$, in general. To work around this issue we represent each polytope as a convex hull of its vertices, i.e.,

$$\mathcal{R}_i = \left\{ x \mid x = \sum_{k=1}^{|\mathcal{V}_i|} \lambda_k [\mathcal{V}_i]_k, \lambda \in \Delta_i \right\}, \quad (11)$$

where \mathcal{V}_i are vertices of the i -th polytope, $[\mathcal{V}_i]_k$ is the k -th vertex, $|\mathcal{V}_i|$ is the number of vertices, and Δ_i is a unit simplex as in (9).

Lemma 4.3. Let $\xi_i(\lambda)$ are obtained by substituting for $x = \sum_k \lambda_k [\mathcal{V}_i]_k$ into (8) for each $i \in \mathcal{I}_{\max} \cup \mathcal{I}_{\min}$. If there $\exists \alpha$ such that

$$\xi_i(\lambda) > 0, \quad \forall \lambda \in \Delta_i, \quad \forall i \in \mathcal{I}_{\max}, \quad (12a)$$

$$-\xi_i(\lambda) > 0, \quad \forall \lambda \in \Delta_i, \quad \forall i \in \mathcal{I}_{\min}, \quad (12b)$$

then ξ as in (8) satisfies (6).

Proof. If (12a) holds for some $i \in \mathcal{I}_{\max}$, then $\xi_i(\lambda) > 0$ for all $\lambda \in \Delta_i$ implies $\xi(x) > 0$ for all $x \in \mathcal{R}_i$ by (11). By enforcing validity of (12a) for each $i \in \mathcal{I}_{\max}$ we have that $\xi(x) > 0$ for all $x \in \bar{\mathcal{R}}$. The argument behind (12b) is similar, just with an opposite sign. \square

Remark 4.4. The coefficients which multiply various powers of λ_k in $\xi_i(\lambda)$ will be different for each polytope since they depend on the vertices \mathcal{V}_i and the parameters α in (8). But since \mathcal{V}_i are known, the coefficients will only be a linear function of α .

Theorem 4.5. Let the collections of polytopes $\bar{\mathcal{R}}$ and $\underline{\mathcal{R}}$ be given. Obtain vertices \mathcal{V}_i of all polytopes in $\bar{\mathcal{R}}$ and $\underline{\mathcal{R}}$. Form polynomials $\xi_i(\lambda)$ by substituting for $x = \sum_k \lambda_k [\mathcal{V}_i]_k$ into (8). Homogenize each $\xi_i(\lambda)$ by multiplying its single monomials by $(\sum_{k=1}^{|\mathcal{V}_i|} \lambda_k)$ until all monomials have the same degree. Select a Pólya degree P and create extended polynomials $\tilde{\xi}_i$ per (10). Denote by $\text{coeffs}(\tilde{\xi}_i)$ the symbolic representation of coefficients of $\tilde{\xi}_i$, cf. Remark 4.4. If the linear program

$$\min \|\alpha\|_1 \quad (13a)$$

$$\text{s.t. } \text{coeffs}(\tilde{\xi}_i) > 0, \quad \forall i \in \mathcal{I}_{\max}, \quad (13b)$$

$$\text{coeffs}(-\tilde{\xi}_i) > 0, \quad \forall i \in \mathcal{I}_{\min}, \quad (13c)$$

is feasible, then ξ as in (8) solves Problem 4.1.

Proof. If (13) is feasible then (12) holds by Theorem 4.2. Then, (6) follows from (12) by Lemma 4.3. Therefore ξ as in (8) strictly separates $\bar{\mathcal{R}}$ and $\underline{\mathcal{R}}$. \square

By minimizing the ℓ_1 norm in (13a) we obtain a sparse α . Doing so is recommended to keep complexity of the separator (which is proportional to the number of non-zero coefficients) small. We remark that Theorem 4.5 can also be used to find a linear separator, i.e., with $\delta = 1$ in (8).

4.2. Implicit separation

Instead of searching for an explicit form of the separator ξ , in this section we present a procedure which implicitly defines the value of $\xi(x)$ for a given query point x . In particular, we show how to evaluate $\xi(x)$ using only the information provided by the unsaturated regions of κ . The advantage of the proposed procedure is that it does not require enumeration of vertices, which can be time consuming and/or numerically sensitive.

Consider a new PWA function $\hat{\kappa}$ which is defined only over the unsaturated regions $\mathcal{U} = \{\mathcal{R}_i\}_{i \in \mathcal{I}_{\text{unsat}}}$ of the original function κ :

$$\hat{\kappa}(x) = K_i x + L_i \quad \text{if } x \in \mathcal{R}_i, \quad i \in \mathcal{I}_{\text{unsat}}. \quad (14)$$

Theorem 4.6. Let κ be a continuous PWA function as in (5) with $\text{dom}(\kappa)$ convex and $\hat{\kappa}$ be as in (14). Let a query point $x \in \text{dom}(\kappa)$ be given. Denote by \hat{x} any point in \mathcal{U} closest to the query x , i.e.,

$$\hat{x} = \text{argmin} \{ \|z - x\| \mid z \in \mathcal{U} \}. \quad (15)$$

Then

$$\kappa(x) = \hat{\kappa}(\hat{x}), \quad \forall x \in \text{dom}(\kappa). \quad (16)$$

Proof. We first prove two intermediate statements:

$$x \in \overline{\mathcal{R}} \Rightarrow \mathcal{L}(\hat{x}, x) \subseteq \overline{\mathcal{R}}, \quad (17a)$$

$$x \in \underline{\mathcal{R}} \Rightarrow \mathcal{L}(\hat{x}, x) \subseteq \underline{\mathcal{R}}, \quad (17b)$$

where $\mathcal{L}(\hat{x}, x)$ denotes a line segment between points \hat{x} and x , i.e., $\mathcal{L}(\hat{x}, x) = \{\theta\hat{x} + (1 - \theta)x \mid 0 \leq \theta \leq 1\}$. To prove (17a), we shall show that

$$x \in \overline{\mathcal{R}} \Rightarrow \mathcal{L}(\hat{x}, x) \cap \mathcal{U} = \{\hat{x}\}, \quad (18a)$$

$$x \in \overline{\mathcal{R}} \Rightarrow \mathcal{L}(\hat{x}, x) \cap \underline{\mathcal{R}} = \emptyset, \quad (18b)$$

$$x \in \overline{\mathcal{R}} \Rightarrow \hat{x} \in \mathcal{U} \cap \overline{\mathcal{R}}. \quad (18c)$$

We show (18a) by contradiction. Assume there exists a point $z \neq \hat{x}$ such that $z \in \mathcal{L}(\hat{x}, x) \cap \mathcal{U}$. This would mean that z is closer to x than \hat{x} is, i.e., $\|z - x\| < \|\hat{x} - x\|$, a contradiction with (15). We remark that $\hat{x} \in \mathcal{L}(\hat{x}, x) \cap \mathcal{U}$ follows directly from (15) and from the definition of the line segment. Therefore (18a) holds. To prove (18b), assume by contradiction there exists a $z \in \mathcal{L}(\hat{x}, x) \cap \underline{\mathcal{R}}$. Without loss of generality take $z = \hat{x}$. Since $x \in \overline{\mathcal{R}}$ and $\hat{x} \in \underline{\mathcal{R}}$ is now assumed, we have $\kappa(x) = \bar{\kappa}$ and $\kappa(\hat{x}) = \underline{\kappa}$. Because κ is assumed to be continuous, there must be a point $y \neq \hat{x}$, $y \in \mathcal{L}(\hat{x}, x)$ which satisfies $\underline{\kappa} < \kappa(y) < \bar{\kappa}$. This can only happen if $y \in \mathcal{U}$. But due to (18a) we have that no such $y \neq \hat{x}$ exists, and therefore (18b) holds. Next, to show (18c) note that $\mathcal{L}(\hat{x}, x) \subseteq \text{dom}(\kappa)$ with $\text{dom}(\kappa) = \underline{\mathcal{R}} \cup \mathcal{U} \cup \overline{\mathcal{R}}$ since $\text{dom}(\kappa)$ is assumed convex. But due to (18b) we have $\mathcal{L}(\hat{x}, x) \cap \underline{\mathcal{R}} = \emptyset$ and therefore $\mathcal{L}(\hat{x}, x) \subseteq \mathcal{U} \cup \overline{\mathcal{R}}$. Since \mathcal{U} and $\overline{\mathcal{R}}$ are bounded and closed sets (though not necessarily convex), \hat{x} obtained from (15) will be on the boundary of \mathcal{U} . Since κ is assumed to be continuous, we have that \hat{x} is also on the boundary of $\overline{\mathcal{R}}$ and (18c) follows. Combining (18a)–(18c) with the fact that $\mathcal{L}(\hat{x}, x) \subseteq \text{dom}(\kappa)$ due to convexity of $\text{dom}(\kappa)$, we get (17a). The proof of (17b) follows the same lines and is therefore omitted.

Finally, we prove that (17) imply (16). If $x \in \mathcal{U}$, then (16) follows immediately from (14) since $\hat{x} = x$ minimizes (15) for any $x \in \mathcal{U}$. If $x \in \overline{\mathcal{R}}$ then we have $\kappa(\hat{x}) = \bar{\kappa}$ because $\hat{x} \in \overline{\mathcal{R}}$ by (18c). But simultaneously $\hat{x} \in \mathcal{U}$ by (18a). Since κ is assumed to be continuous, we therefore have $\kappa(\hat{x}) = \hat{\kappa}(\hat{x})$ by (14). Finally, since $\mathcal{L}(\hat{x}, x) \subseteq \overline{\mathcal{R}}$ by (17a), we have that $\kappa(y) = \bar{\kappa}$ for any $y \in \mathcal{L}(\hat{x}, x)$ and therefore $\kappa(x) = \hat{\kappa}(\hat{x})$. The proof for $x \in \underline{\mathcal{R}}$ implying (16) is identical. \square

Theorem 4.7. Let $\hat{\kappa}$ be as in (14) and denote by \hat{x} the projection of x onto \mathcal{U} per (15). Then

$$\xi(x) := \hat{\kappa}(\hat{x}) - \frac{1}{2}(\underline{\kappa} + \bar{\kappa}) \quad (19)$$

solves Problem 4.1.

Proof. For all $x \in \overline{\mathcal{R}}$ we have $\kappa(x) = \bar{\kappa}$ by definition of $\overline{\mathcal{R}}$, and $\hat{\kappa}(\hat{x}) = \bar{\kappa}$ by Theorem 4.6. Since $\bar{\kappa} > \underline{\kappa}$, the quantity $\bar{\kappa} - \frac{1}{2}(\underline{\kappa} + \bar{\kappa})$ is positive, which shows that (6a) holds. The proof of (6b) is similar and follows from the fact that $\underline{\kappa} - \frac{1}{2}(\underline{\kappa} + \bar{\kappa}) < 0$. \square

Theorem 4.7 says that we can obtain the value $\xi(x)$ by evaluating the simpler function $\hat{\kappa}$ as follows: (i) given x , compute \hat{x} from (15); (ii) evaluate $\hat{\kappa}(\hat{x})$ per (14); and (iii) obtain $\xi(x)$ from (19). The only technical difficulty is that the union $\mathcal{U} = \cup_i \mathcal{U}_i$ is non-convex, in general, therefore finding \hat{x} as a projection of x onto \mathcal{U} from (15) is not straightforward. In general, one can compute \hat{x}_i for each polytope \mathcal{U}_i by solving (15) as a quadratic program, i.e.,

$$\hat{x}_i = \text{argmin}\{\|z - x\| \mid z \in \mathcal{U}_i\}, \quad i = 1, \dots, |\mathcal{U}|, \quad (20)$$

followed by taking

$$\hat{x} = \text{argmin}\{\|\hat{x}_i - x\| \mid i = 1, \dots, |\mathcal{U}|\}. \quad (21)$$

However, determining \hat{x} by solving optimization problems on-line can be time consuming. Therefore we propose a simpler method for obtaining \hat{x} . We distinguish between two cases. If $x \in \mathcal{U}_i$ for some $i = 1, \dots, |\mathcal{U}|$, then $\hat{x} = x$ by (17). Consider therefore $x \notin \cup_i \mathcal{U}_i$ and note that Theorem 4.6 holds even when the search for \hat{x} from (20) to (21) is restricted to any line segment $\mathcal{L}(x, x_0)$ where x_0 is any point³ with $x_0 \in \text{int}(\mathcal{U})$, i.e.,

$$\hat{x}_i = \text{argmin}\{\|z - x\| \mid z \in \mathcal{U}_i, z \in \mathcal{L}(x, x_0)\}. \quad (22)$$

Then we can find the intersection between \mathcal{U}_i and $\mathcal{L}(x, x_0)$ as follows. Since each \mathcal{U}_i is a polytope, its half-space representation is $\mathcal{U}_i = \{x \mid F_i x \leq g_i\}$ where F_i contains the normal vectors of the defining half-spaces $f_{i,1}, \dots, f_{i,c_i}$, and c_i is the number of defining half-spaces of the i -th polytope. Then the intersection $\hat{x}_{i,j}$ between the hyperplane $\mathcal{H}_{i,j} = \{z \mid f_{i,j}^T z = g_{i,j}\}$ and the line $\{z \mid z = x + \theta(x_0 - x)\}$ passing through x_0 and x is given by

$$\hat{x}_{i,j} = x + \hat{\theta}_{i,j}(x_0 - x), \quad (23)$$

where

$$\hat{\theta}_{i,j} = \frac{g_{i,j} - f_{i,j}^T x}{f_{i,j}^T (x_0 - x)}. \quad (24)$$

If the denominator in (24) is zero, we adopt the notion that $\hat{\theta}_{i,j} = \infty$. The point $\hat{x}_{i,j}$ is a valid intersection between the line segment $\mathcal{L}(x, x_0)$ and $\mathcal{H}_{i,j}$ if $0 \leq \hat{\theta}_{i,j} \leq 1$. In such a case $\hat{\theta}_{i,j}$ also denotes the distance of $\hat{x}_{i,j}$ from x along $\mathcal{L}(x, x_0)$.

For a given polytope \mathcal{U}_i with c_i facets, we can find \hat{x}_i as in (22) by a simple procedure:

- (1) Compute intersection points $\hat{x}_{i,j}$ for $j = 1, \dots, c_i$ from (23) to (24).
- (2) Identify valid intersections, i.e., those satisfying $0 \leq \hat{\theta}_{i,j} \leq 1$ and $F_i \hat{x}_{i,j} \leq g_i$.
- (3) Among the valid intersections, select the one which is closest to x , i.e., the one with minimal value of $\hat{\theta}_{i,j}$. If there is no intersection between \mathcal{U}_i and $\mathcal{L}(x, x_0)$, return an empty set.

By repeating this procedure for all polytopes \mathcal{U}_i , $i = 1, \dots, |\mathcal{U}|$, we obtain a set of points $\{\hat{x}_i\}$ among which \hat{x} is chosen by (21). Note that the distances $\|\hat{x}_i - x\|$ in (21) are readily available in $\hat{\theta}_i$. Since x_0 is the interior point of \mathcal{U} , then there will always exist at least one polytope \mathcal{U}_i which intersects $\mathcal{L}(x, x_0)$ and therefore the minimum in (21) is always attained.

5. Complexity analysis

In terms of off-line pre-processing, searching for a polynomial separator from Theorem 4.2 requires two pre-processing steps. First, vertices of polytopes forming $\overline{\mathcal{R}}$ and $\underline{\mathcal{R}}$ need to be computed. Subsequently, coefficients of the separator are found by solving (13). The linear program in (13) has $\binom{n_x + \delta}{\delta}$ variables and $2 \sum_{i \in \mathcal{J}_{\max} \cup \mathcal{J}_{\min}} M_i$ constraints. Here, M_i is the number of coefficients of the Pólya polynomials $\tilde{\xi}$ in (10) and is given by $M_i = \binom{\delta_p + |\mathcal{V}_i| - 1}{\delta_p}$ with $\delta_p = \delta + P$. The implicit separator, described in Section 4.2, only requires one cheap pre-processing step in which the interior point $x_0 \in \text{int}(\mathcal{U})$, used in (22), is determined at the expense of a single LP with $n_x + 1$ variables.

In terms of on-line computation, obtaining the value of $\tilde{\kappa}(x)$ on-line for a given point x from (7) first requires to assess whether $x \in \mathcal{U}$. Searching through the unsaturated regions sequentially can answer this query in $\sum_{i \in \mathcal{J}_{\text{unsat}}} c_i (2n_x + 1)$ time (we recall that c_i is

³ If (2) is a regulation MPC problem, then x_0 can be chosen to be the origin.

the number of half-spaces defining the i -th polytope). If $x \in \mathcal{R}_r$ for some $r \in \mathcal{J}_{\text{unsat}}$, then $\kappa(x) = K_r x + L_r$. Otherwise, $\xi(x)$ is evaluated by, at most, $\delta \binom{n_x + \delta}{\delta}$ FLOPs, and based on its sign either $\kappa(x) = \bar{\kappa}$ or $\kappa(x) = \underline{\kappa}$ is returned. Obtaining the value $\xi(x)$ per the implicit description of Section 4.2 requires, at most, $2|\mathcal{J}_{\text{unsat}}|(n_x + 1) + \sum_{i \in \mathcal{J}_{\text{unsat}}} c_i(4n_x + 2)$ FLOPs. These figures are to be compared to $\sum_{i=1}^R c_i(2n_x + 1)$, the effort needed to evaluate the original function (3) by sequential search. Therefore the achievable reduction in on-line computation is proportional to the ratio between the number of unsaturated regions, $|\mathcal{J}_{\text{unsat}}|$, to the total number of regions, R , cf. Section 6.2.

Finally, we quantify the required on-line storage. Storing the original function κ would require $\sum_{i=1}^R c_i(n_x + 1)$ numbers. The simpler function $\tilde{\kappa}$ as in (7) only requires storing the separator and the unsaturated regions, i.e., only $\mathcal{S}(\xi) + \sum_{i \in \mathcal{J}_{\text{unsat}}} c_i(n_x + 1)$ numbers. Here, $\mathcal{S}(\xi)$ is the size of the separator with $\mathcal{S}(\xi) \leq \binom{n_x + \delta}{\delta}$ for coefficients of the polynomial separator and $\mathcal{S}(\xi) = n_x$ for the implicit separator, which only requires storing the single interior point x_0 .

5.1. Comparison to other approaches

Other methods can be used to derive the replacement function $\tilde{\kappa}$. The lattice representation (LR) of Wen et al. (2009) converts the original function κ into a series of min / max operations over linear functions, eliminating the need to store the underlying regions \mathcal{R}_i . Evaluation of such a lattice description requires $\mathcal{O}(R_{\text{unique}}^2)$ operations, where R_{unique} is the number of regions where the feedback law is unique. The memory storage is also proportional to $\mathcal{O}(R_{\text{unique}}^2)$. The clipping-based procedure (Kvasnica & Fikar, 2012) removes some of the saturated regions and replaces them by “extensions” of the unsaturated ones. In the best case, $\tilde{\kappa}$ is then defined over $|\mathcal{J}_{\text{unsat}}|$ regions. On average, $\tilde{\kappa}$ consists of $1.3|\mathcal{J}_{\text{unsat}}|$ regions. In the worst case, however, no reduction can be achieved even if $|\mathcal{J}_{\text{unsat}}| < R$.

6. Examples

6.1. Illustrative example

Consider a 2-state 1-input system of the form (1) with $A = \begin{bmatrix} 0.755 & 0.680 \\ 0.651 & -0.902 \end{bmatrix}$, $B = \begin{bmatrix} 0.825 \\ -0.139 \end{bmatrix}$, subject to constraints $|x_i| \leq 10$ for $i = 1, 2$ and $|u| \leq 1$. The MPC problem (2) was formulated with $N = 10$, $Q_x = \mathbb{1}$ and $Q_u = 1$ and solved as a pQP according to Theorem 2.1. Using the MPT Toolbox (Kvasnica, Grieder, & Baotić, 2004), the explicit RHMPC feedback κ was obtained in 4 s (on a 2.2 GHz Core i7 CPU with 8 GB of RAM using MATLAB 7.8 and MPT 2.6.3) as a PWA function defined over 225 regions, shown in Fig. 1. The partition of κ consists of 29 unsaturated regions \mathcal{U} , 98 regions $\bar{\mathcal{R}}$ where $\kappa(x) = 1$, and 98 regions $\underline{\mathcal{R}}$ with $\kappa(x) = -1$.

Solving for coefficients of (8) with $\delta = 3$ from (13) resulted in $\xi(x) := -x_1 - x_2 - 0.0011x_1^3 - 0.254x_2^3$. The total memory footprint of the original function κ with 225 regions is 27 kB. The equivalent replacement $\tilde{\kappa}$ in (7), on the other hand, only requires 3.5 kB (16 bytes of which are consumed by ξ). These figures correspond to a reduction of memory consumption by a factor of 7.7. The worst-case computational effort needed to evaluate the original function κ is 4470 FLOPs. By employing (7) this figure can be reduced to 570 FLOPs (560 FLOPs to find out whether $x \in \mathcal{U}$ and 10 FLOPs to evaluate $\xi(x)$), a reduction by a factor of 7.8. The replacement $\tilde{\kappa}$ can also be evaluated using the implicit separator per Section 4.2, without the need to construct ξ in its explicit form. In this case evaluating $\xi(x)$ per (21)–(24) requires 1792 FLOPs, in the worst case.

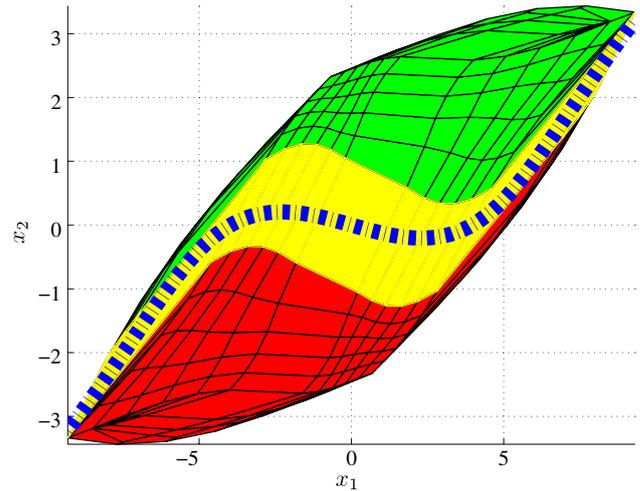


Fig. 1. Polytopic sets \mathcal{U} (yellow), $\bar{\mathcal{R}}$ (red), $\underline{\mathcal{R}}$ (green), and the zero-level set of polynomial separator (8) of degree 3. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 1

Likelihood of existence of a separator (8) of degree δ_{\min} .				
n_x/n_u	2/1	2/2	3/1	3/2
$\delta_{\min} = 1$	94	90	83	63
$\delta_{\min} = 3$	6	10	15	37
$\delta_{\min} = 5$	–	–	1	–
Σ	100	100	99	100

6.2. Random systems

Next, we have analyzed a large number of random RHMPC feedback laws κ generated by solving problem (2) for randomly selected LTI systems with 2–3 states, and 1–2 inputs. Magnitudes of the variables were constrained by $|x_i| \leq 10$ and $|u_i| \leq 5$, respectively. 100 random cases were considered for each n_x/n_u category. The largest investigated scenario had 12 651 regions. For each function κ we have constructed the equivalent replacement $\tilde{\kappa}$ as in (7). Based on the 400 random scenarios, Table 1 shows for how many cases a separator ξ of a given minimal degree δ_{\min} was found by solving (13). Only degrees $\delta \leq 5$ were considered to keep the complexity of (13) on a tractable level.

Table 2 reports the minimal, maximal, and average values of the complexity reduction ratio defined as $\Delta = \frac{|\mathcal{J}_{\text{unsat}}| + |\mathcal{J}_{\text{max}}| + |\mathcal{J}_{\text{min}}|}{|\mathcal{J}_{\text{unsat}}|}$. This figure represents a reduction in on-line memory and computation achievable by employing procedures of this paper compared to using the original explicit MPC feedback, cf. Section 5. The last row of Table 2 shows performance of the method of Kvasnica and Fikar (2012) on the same data set. In all cases where a linear separator was found, performance of the current scheme was the same as with (Kvasnica & Fikar, 2012). In the remaining cases the procedures of this paper achieved up to 3.5-times higher complexity reduction. Also important, from a practical point of view, is the fact that, compared to (Kvasnica & Fikar, 2012), the proposed approaches are up to 1000 times faster in terms of off-line pre-processing. In particular, in all successful cases the vertex enumeration time did not exceed 10 s and runtime of LP (13) was below 10 s as well.

7. Conclusions

Given an explicit RHMPC feedback function κ , we have shown how to construct its simpler replacement $\tilde{\kappa}$ which maintains the equivalence $\kappa(x) = \tilde{\kappa}(x)$ for all $x \in \text{dom } \kappa$. The approach is based on devising a separating function ξ which separates the regions

Table 2
Minimal, maximal, and average values of the complexity reduction ratio.

n_x/n_u	2/1	2/2	3/1	3/2
Δ_{\min}	2.3	1.8	2.1	1.9
Δ_{\max}	31.0	14.5	21.0	10.2
Δ_{avg}	13.4	5.9	7.1	3.6
Δ_{avg} (Kvasnica & Fikar, 2012)	13.0	5.3	6.6	2.9

over which κ attains a saturated value. We have shown how to construct the explicit form of such a separator by solving a single linear program. Polynomial (and linear, as a special case) separators feature a low memory footprint and allow for fast evaluation of ξ at x . This approach is recommended for lower-dimensional problems for which vertex enumeration can be performed in a numerically reliable manner. For higher-dimensional cases, where off-line pre-processing would be prohibitive and/or unreliable, we have shown a procedure which implicitly defines a value of the separator at a given point. Such an implicit separator avoids off-line pre-processing steps at the expense of more involved on-line computation. Regardless of the type of separator, the replacement function $\tilde{\kappa}$ requires only the storage of the unsaturated regions of κ , along with representation of ξ . By means of a large case study we have demonstrated that a significant reduction of complexity can be achieved in general.

References

- Bemporad, A., Morari, M., Dua, V., & Pistikopoulos, E. N. (2002). The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1), 3–20.
- Bemporad, A., Oliveri, A., Poggi, T., & Storace, M. (2011). Ultra-fast stabilizing model predictive control via canonical piecewise affine approximations. *IEEE Transactions on Automatic Control*, 56(12), 2883–2897.
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297.
- Geyer, T., Torrisi, F. D., & Morari, M. (2008). Optimal complexity reduction of polyhedral piecewise affine systems. *Automatica*, 44(7), 1728–1740.
- Grieder, P., & Morari, M. (2003). Complexity reduction of receding horizon control. In *IEEE conference on decision and control* (pp. 3179–3184). Maui, Hawaii, December.
- Hardy, G. H., Littlewood, J. E., & Pólya, G. (1952). *Inequalities* (2nd ed.). Cambridge University Press.
- Johansen, T. A., & Grancharova, A. (2003). Approximate explicit constrained linear model predictive control via orthogonal search tree. *IEEE Transactions on Automatic Control*, 48, 810–815.
- Jones, C. N., & Morari, M. (2010). Polytopic approximation of explicit model predictive controllers. *IEEE Transactions on Automatic Control*, 55(11), 2542–2553.
- Kvasnica, M., & Fikar, M. (2012). Clipping-based complexity reduction in explicit MPC. *IEEE Transactions on Automatic Control*, 57(7), 1878–1883.
- Kvasnica, M., Grieder, P., & Baotić, M. (2004). Multi-parametric toolbox (MPT). Available from: <http://control.ee.ethz.ch/~mpt/>.
- Kvasnica, M., Rauová, I., & Fikar, M. (2011). Simplification of explicit MPC feedback laws via separation functions. In *Preprints of the 18th IFAC world congress* (pp. 5383–5388). Milano, Italy.
- Scibilia, F., Oлару, S., & Hovd, M. (2009). Approximate explicit linear MPC via delaunay tessellation. In *Proceedings of the 10th european control conference*. Budapest, Hungary.
- Tøndel, P., Johansen, T. A., & Bemporad, A. (2003). Evaluation of piecewise affine control via binary search tree. *Automatica*, 39(5), 945–950.
- Wen, Ch., Ma, X., & Ydstie, B. E. (2009). Analytical expression of explicit MPC solution via lattice piecewise-affine function. *Automatica*, 45(4), 910–917.



Michal Kvasnica was born in 1977. He received his diploma in chemical engineering from the Slovak University of Technology in Bratislava, Slovakia and the Ph.D. in electrical engineering from the Swiss Federal Institute of Technology in Zurich, Switzerland. Since 2011 he is an Associate Professor at the Slovak University of Technology in Bratislava. His research interests are in model predictive control, modeling of hybrid systems, and development of software tools for control. He is the co-author and developer of the MPT Toolbox for explicit model predictive control.



Juraj Hledík was born in 1988. He obtained his bachelor and master degrees in financial mathematics from the Comenius University in Bratislava, Slovakia in 2011. He is currently a Ph.D. student at the Vienna Graduate School of Finance. His research interests are focused on optimal hedging in incomplete markets and contagion spread in financial networks.



Ivana Rauová was born in 1986. She studied at the Slovak University of Technology in Bratislava, Slovakia, in the field of automation and informatization in the chemistry and food industry. Her diploma thesis was conducted at the Automatic Control Laboratory of the Swiss Federal Institute of Technology in Zurich, Switzerland. The focus of her diploma work was model predictive control and she graduated in 2010 with honors. Since 2011 she works as a control engineer in Slovnaft, the biggest petrochemical company in Slovakia. Her current responsibilities involve application of advanced process control of petrochemical

processes.



Miroslav Fikar was born in Bratislava, Slovakia, in 1966 and received his ME and Ph.D. degrees in chemical engineering from the Slovak University of Technology in Bratislava in 1989 and 1994, respectively. He has stayed with the Faculty of Chemical and Food Technology STU where he is currently a professor and institute director. He was Postdoc Fellow in Nancy, France (Elf Aquitaine, Ministère de la Recherche), Alexander von Humboldt Fellow in Bochum, Germany and has spent several stays in Denmark, Germany, France, and Switzerland. His current research interests include optimal control, MPC, and

chemical process control.

Stabilizing low complexity feedback control of constrained piecewise affine systems[☆]

Pascal Grieder*, Michal Kvasnica, Mato Baotić, Manfred Morari

Automatic Control Laboratory, Swiss Federal Institute of Technology, ETLI 22, 8092 Zürich, Switzerland

Received 7 July 2004; received in revised form 4 March 2005; accepted 14 April 2005

Available online 20 June 2005

Abstract

Piecewise affine (PWA) systems are powerful models for describing both non-linear and hybrid systems. One of the key problems in controlling these systems is the inherent computational complexity of controller synthesis and analysis, especially if constraints on states and inputs are present. In addition, few results are available which address the issue of computing stabilizing controllers for PWA systems without placing constraints on the location of the origin.

This paper first introduces a method to obtain stability guarantees for receding horizon control of discrete-time PWA systems. Based on this result, two algorithms which provide low complexity state feedback controllers are introduced. Specifically, we demonstrate how multi-parametric programming can be used to obtain minimum-time controllers, i.e., controllers which drive the state into a pre-specified target set in minimum time. In a second segment, we show how controllers of even lower complexity can be obtained by separately dealing with constraint satisfaction and stability properties. To this end, we introduce a method to compute PWA Lyapunov functions for discrete-time PWA systems via linear programming. Finally, we report results of an extensive case study which justify our claims of complexity reduction.

© 2005 Elsevier Ltd. All rights reserved.

Keywords: Piecewise affine system; Optimal control; Model predictive control; Multi-parametric programming

1. Introduction

Optimal control of piecewise affine (PWA) systems has garnered increasing interest in the research community, since this system type represents a powerful tool for approximating non-linear systems and because of their equivalence to many classes of hybrid systems (Heemels, de Schutter, & Bemporad, 2001; Sontag, 1981). The optimal control inputs for discrete-time PWA systems may be obtained by solving mixed-integer optimization problems on-line (Bemporad & Morari, 1999; Mayne & Raković, 2003), or as was shown in Baotić, Christophersen, and Morari (2003a),

Borrelli, Baotić, Bemporad, and Morari (2003), Kerrigan and Mayne (2002) and Borrelli (2003), by solving off-line a number of multi-parametric programs. By multi-parametric programming, a linear (mpLP) or quadratic (mpQP) optimization problem is solved off-line for a range of parameters.

In their pioneering work (Bemporad, Morari, Dua, & Pistikopoulos, 2002), the authors show how to formulate an optimal control problem for constrained linear discrete-time systems as a multi-parametric program (by treating the state vector as a parameter) and how to solve such a program. Basic ideas from Bemporad et al. (2002) for linear systems were extended to PWA systems in Baotić et al. (2003a), Borrelli et al. (2003), Kerrigan and Mayne (2002) and Borrelli (2003). The associated solution (optimal control inputs) takes the form of a PWA state feedback law. In particular, the state space is partitioned into polyhedral sets and for each of these sets the optimal control law is given as

[☆] This paper was not presented at any IFAC meeting. This paper was recommended for publication in revised form by Associate Editor Denis Dochain under the direction of Editor Prof. F. Allgower.

* Corresponding author.

E-mail address: grieder@control.ee.ethz.ch (P. Grieder).

an affine function of the state. In the on-line implementation of such controllers, input computation reduces to a simple set-membership test. Even though the approaches in Baotić et al. (2003a), Borrelli et al. (2003), Kerrigan and Mayne (2002), and Borrelli (2003) rely on off-line computation of a feedback law, the computation quickly becomes prohibitive for larger problems. This is not only due to the high complexity of the multi-parametric programs involved (Grieder & Morari, 2003), but mainly because of the large number of multi-parametric programs which need to be solved when a controller is computed in a dynamic programming fashion (Borrelli et al., 2003; Kerrigan & Mayne, 2002).

In addition, there are few results in the literature which explicitly address the issue of computing feedback controllers for PWA systems which provide stability guarantees. The few publications that address this issue (e.g., Mayne & Raković, 2003) assume that the origin is contained in the interior of one system dynamic. The only exception is the infinite horizon solution proposed in Baotić, Christophersen, and Morari (2003b), which is computationally intractable for larger problems.

This paper addresses the clear need for low complexity controllers for PWA systems that provide stability guarantees even if the origin is located on the boundary of several different system dynamics. Two algorithms are presented in this paper which achieve this goal.

First, a general scheme for obtaining stability guarantees for generic PWA systems subject to receding horizon control will be presented. This scheme can be used in connection with other controller computation methods (e.g., Mayne & Raković, 2003; Baotić et al., 2003a; Borrelli et al., 2003; Kerrigan & Mayne, 2002) to obtain stability guarantees.

Subsequently, the computation of a minimum-time feedback controller is presented. As the final section will show, the resulting controller is of such low complexity compared to what one can obtain with traditional methods (Baotić et al., 2003a; Borrelli et al., 2003) that a whole new class of problems becomes tractable.

In a second segment, we show how controllers of even lower complexity can be obtained by separately dealing with the issue of constraint satisfaction and asymptotic stability. To this end, we introduce a method to compute a PWA Lyapunov function for discrete-time PWA systems via linear programming. The computation is guaranteed to find a PWA Lyapunov function for a given partition, if it exists.

2. Problem description and properties

This section covers some of the fundamentals of multi-parametric programming for linear systems before restating recent results for PWA systems. Consider a discrete-time linear time-invariant system

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

with $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$. Let $x(k)$ denote the measured state at time k and $x_k(u_k)$ the predicted state (input) at time k , given $x(0)$. Assume now that the states and the inputs of the system in (1) are subject to the following constraints:

$$x(k) \in \mathbb{X} \subseteq \mathbb{R}^n, \quad u(k) \in \mathbb{U} \subseteq \mathbb{R}^m \quad \forall k \geq 0, \quad (2)$$

where \mathbb{X} and \mathbb{U} are polytopic sets containing the origin in their interior.

Remark 1. For ease of notation, we restrict ourselves to separate constraints on state and input in (2). It is straightforward to modify all algorithms in this paper to deal with systems subject to mixed state-input-constraints, i.e., $C^x x(k) + C^u u(k) \leq C^c$, $\forall k \geq 0$.

Consider the constrained finite-time optimal control problem with a linear objective

$$J_N^*(x(0)) = \min_{u_0, \dots, u_{N-1}} \sum_{k=0}^{N-1} (\|Ru_k\|_{1,\infty} + \|Qx_k\|_{1,\infty}) + \|Q_f x_N\|_{1,\infty}, \quad (3a)$$

$$\text{subj. to } x_k \in \mathbb{X}, u_{k-1} \in \mathbb{U} \quad \forall k \in \{1, \dots, N\}, \quad (3b)$$

$$x_N \in \mathcal{T}_{\text{set}}, \quad (3c)$$

$$x_{k+1} = f(x_k, u_k), \quad x_0 = x(0), \quad (3d)$$

where (3c) is a user defined set-constraint on the final state and $\|\cdot\|_{1,\infty}$ denotes the 1- or ∞ -norm of a vector, respectively.

Definition 2.1. We define the N -step feasible set $\mathcal{F}_N \subseteq \mathbb{R}^n$ as the set of initial states $x(0)$ for which the optimal control problem (3) is feasible, i.e.,

$$\mathcal{F}_N = \{x(0) \in \mathbb{R}^n \mid \exists U_N \in \mathbb{R}^{Nm}, u_{k-1} \in \mathbb{U}, x_k \in \mathbb{X}, x_N \in \mathcal{T}_{\text{set}}, \forall k \in \{1, \dots, N\}\},$$

where $U_N = [u'_0, \dots, u'_{N-1}]'$ is the optimization vector. Assume now that the state update function is linear, i.e., $f(x_k, u_k) = Ax_k + Bu_k$ in (3d). By considering $x(0)$ as a parameter, problem (3) can then be stated as an mpLP (Bemporad, Borrelli, & Morari, 2000) which can be solved to obtain a feedback solution with the following properties (derived from Borrelli, 2003; Gal, 1995):

Theorem 2.1. Consider the finite time constrained regulation problem (3), with a linear objective in (3a) and a linear state update function $f(x_k, u_k) = Ax_k + Bu_k$ in (3d). Then, the set of feasible parameters \mathcal{F}_N is convex, there exists an optimizer $U_N^*: \mathcal{F}_N \rightarrow \mathbb{R}^{Nm}$ which is continuous and PWA over polyhedra, i.e.,

$$U_N^*(x(0)) = F_r x(0) + G_r \quad \text{if } x(0) \in \mathcal{P}_r \\ \mathcal{P}_r = \{x \in \mathbb{R}^n \mid H_r x \leq K_r\}, \quad r = 1, \dots, R$$

and the value function $J_N^*: \mathcal{F}_N \rightarrow \mathbb{R}$ is continuous, convex and PWA.

According to Theorem 2.1, the feasible state space \mathcal{F}_N is partitioned into R polytopic regions, i.e., $\mathcal{F}_N = \bigcup_{r=1,\dots,R} \mathcal{P}_r$. The results in Bemporad et al. (2000) were extended in Borrelli et al. (2003) to compute the optimal explicit feedback controller for PWA systems of the form

$$x(k+1) = A_i x(k) + B_i u(k) + f_i \tag{4a}$$

$$\text{if } x(k) \in \mathcal{D}_i, i \in \mathcal{I} \tag{4b}$$

subject to the constraints (2). Here, the dynamic set \mathcal{D}_i is polyhedral and the set \mathcal{I} is defined as $\mathcal{I} \triangleq \{1, 2, \dots, D\}$, where D denotes the number of different dynamics. We will henceforth assume that the different dynamic regions \mathcal{D}_i are non-overlapping and abbreviate (4a) and (4b) with $x(k+1) = f_{\text{PWA}}(x(k), u(k))$. The optimization problem considered in this paper is thus given by assuming a PWA state update equation, i.e., $f(x_k, u_k) = f_{\text{PWA}}(x_k, u_k)$ in (3d). In Baotić et al. (2003a), multi-parametric linear programs (mpLP) were solved in a dynamic programming fashion to obtain the feedback solution to (3) for PWA systems. In Borrelli et al. (2003), the feedback solution to (3) for PWA systems and quadratic objective in (3a) was computed by solving a sequence of multi-parametric quadratic programs (mpQP) in a dynamic programming fashion. Methods to obtain feedback solutions to linear or quadratic optimization problems for PWA systems are also given in Borrelli (2003), Mayne and Raković (2003), Baotić et al. (2003b) and Kerrigan and Mayne (2002). Note that we do not require $f_{\text{PWA}} : \mathbb{R}^m \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ to be continuous. However, if f_{PWA} is discontinuous, computing the solution to (3), if one exists, becomes rather cumbersome since special care has to be taken of the open and closed boundaries of \mathcal{D}_i .

3. Computation of stabilizing controllers for PWA systems

A large part of the literature has focussed on end-point constraints to guarantee asymptotic stability of the closed-loop PWA system (e.g., Bemporad & Morari, 1999; Borrelli, 2003). This type of constraint generally requires the use of large prediction horizons for the controller to cover the entire controllable state space, such that the computational complexity quickly becomes prohibitive. Other methods (e.g., Mayne & Raković, 2003) only provide stability guarantees if the origin is contained in the interior of one of the dynamics \mathcal{D}_i .

In this section, a method is presented for obtaining stabilizing controllers for generic PWA systems.¹ For any dynamical system, stability is guaranteed if an invariant set is imposed as a terminal state constraint (see (3c)) and the terminal cost in (3) corresponds to a Lyapunov function for that

set Mayne, Rawlings, Rao, and Sokaert (2000). In addition, the decay rate of the ‘terminal Lyapunov function’ must be greater than the stage cost. We here show how to obtain the invariant maximum admissible set $\mathcal{C}_\infty^{\text{PWA}}$ with the associated feedback law and Lyapunov function. In a first step, we select all dynamics $i \in \mathcal{I}_0$, which contain the origin, i.e.,

$$\mathcal{I}_0 \triangleq \{i \in \mathcal{I} \mid 0 \in \mathcal{D}_i\}.$$

We are assuming that the origin is an equilibrium state of the PWA system and hence the closed-loop dynamics $f_i = 0, \forall i \in \mathcal{I}_0$ (see (4)). If this assumption is not satisfied, the approach proposed here will fail since the system is unstable.

The search for stabilizing piecewise linear feedback controllers F_i and an associated Lyapunov function $V(x) = x'Px$ can now be posed as

$$x'Px \geq 0 \quad \forall x \in \mathbb{X},$$

$$\begin{aligned} &x'(A_i + B_i F_i)'P(A_i + B_i F_i)x - x'Px \\ &\leq -x'Qx - x'F_i'RF_i x \quad \forall x \in \mathcal{D}_i \quad \forall i \in \mathcal{I}_0. \end{aligned}$$

If we make this condition less restrictive by setting $\mathcal{D}_i = \mathbb{R}^n, \forall i \in \mathcal{I}_0$, the problem can be rewritten as an SDP by using Schur complements and introducing the new variables $Y_i = F_i Z$ and $Z = (1/\gamma)P^{-1}$ (see Kothare, Balakrishnan, & Morari, 1996; Mignone, Ferrari-Trecate, & Morari, 2000 for details),

$$\min_{Y_i, Z, \gamma} \gamma, \text{ subj. to} \tag{5a}$$

$$Z \succ 0, \tag{5b}$$

$$\begin{bmatrix} Z & * & * & * \\ (A_i Z + B_i Y_i)' & Z & * & * \\ (Q^{0.5} Z) & 0 & \gamma I & * \\ (R^{0.5} Y_i) & 0 & 0 & \gamma I \end{bmatrix} \geq 0 \quad \forall i \in \mathcal{I}_0, \tag{5c}$$

where the scalar γ is introduced to optimize for the worst case performance, whereby the ‘worst case’ corresponds to an arbitrary switching sequence. Note that it may not be possible for the worst case switching sequence considered in (5) to occur in practice, since not all dynamics i are defined over the entire state space.

Remark 2. If (5) is posed for an LTI system (i.e., $\mathcal{I}_0 = \{1\}$), the optimal LQR state feedback solution K and the solution to the Algebraic Riccati Equation P are recovered.

Alternatively, one can solve a max-det problem to obtain the largest invariant ellipsoidal target set (Boyd, El Ghaoui, Feron, & Balakrishnan, 1994). Large target sets generally make the subsequent controller computations (see Section 4) simpler. Note, however, that the feedback laws associated to the maximal volume invariant ellipsoidal set may not yield the maximal volume invariant polytopic set.

In a second step, the maximal admissible set $\mathcal{C}_\infty^{\text{PWA}}$ of the PWA system subject to the feedback controllers $F_i = Y_i Z^{-1}$ can be computed with the algorithm in

¹ Note that results virtually identical to what is presented in this section were simultaneously obtained by others (Lazar, Heemels, Weiland, & Bemporad, 2004).

Raković, Grieder, Kvasnica, Mayne, and Morari (2004), which is guaranteed to terminate in finite time for the problem at hand, since the closed-loop system is asymptotically stable. The proposed computation scheme is summarized in the following algorithm:

Algorithm 3.1. *Computation of maximal admissible set $\mathcal{O}_\infty^{\text{PWA}}$*

- (1) Identify all dynamics i which contain the origin, i.e., $\mathcal{I}_0 \triangleq \{i \in \mathcal{I} \mid 0 \in \mathcal{D}_i\}$.
- (2) Solve (5) for all $i \in \mathcal{I}_0$, to obtain F_i and P . If (5) is infeasible, abort the algorithm.
- (3) Compute the maximal output admissible set $\mathcal{O}_\infty^{\text{PWA}}$ corresponding to the closed-loop system $x_{k+1} = (A_i + B_i F_i)x_k$, if $x_k \in \mathcal{D}_i$ and constraints (2) with the method in Raković et al. (2004).
- (4) Return the target set $\mathcal{O}_\infty^{\text{PWA}}$, the feedback laws F_i and the associated matrix P .

Theorem 3.1. *Assume the optimization problem (3) is given with a quadratic objective (3a),*

$$J_N^*(x(0)) = \min_{u_0, \dots, u_{N-1}} \sum_{k=0}^{N-1} (u_k' R u_k + x_k' Q x_k) + x_N' Q_f x_N,$$

a terminal set $\mathcal{T}_{\text{set}} = \mathcal{O}_\infty^{\text{PWA}}$ and a terminal cost $Q_f = P$ (obtained with Algorithm 3.1). If this problem is solved at each time step for the PWA system (4) and only the first input is applied (receding horizon control), then the closed-loop system is asymptotically stable.

Proof. The result of Algorithm 3.1 trivially satisfies the conditions for asymptotic stability in Mayne et al. (2000, Section 3.3). \square

Note that we only need to consider a single convex terminal set for linear systems (Gilbert & Tan, 1991), whereas for PWA systems, the terminal set $\mathcal{O}_\infty^{\text{PWA}}$ is given as a union of several convex sets $\mathcal{O}_\infty^{\text{PWA}} = \bigcup \mathcal{O}_i$. If the union $\bigcup \mathcal{O}_i$ is convex, the regions can be merged with the method in Bemporad, Fukuda, and Torrisi (2001). This is a desirable property since simpler target sets \mathcal{T}_{set} generally lead to reduced algorithm run-time and solution complexity for the type of optimization problem given in (3).

Remark 3. The procedure described in this section is merely sufficient for asymptotic stability. We cannot guarantee that the Lyapunov function and the associated state feedback laws will be found in the suggested manner. However, we have observed in an extensive case study that the approach works very well in practice. Short of the computationally very demanding construction of the infinite horizon solution proposed in Baotić et al. (2003b), there is currently no alternative method for guaranteeing closed-loop stabil-

ity for control of generic PWA systems. Furthermore, the method we propose here can easily be combined with most other controller computation (e.g., Borrelli et al., 2003; Mayne & Raković, 2003; Kerrigan & Mayne, 2002; Baotić et al., 2003a).

4. Computation of low complexity controllers for PWA systems

The goal in this section is the design of explicit state feedback controllers, which ensure that the system constraints (2) are satisfied for all time and provide asymptotic stability guarantees. Without loss of generality, we restrict ourselves to the regulation problem, i.e., how the state $x(k)$ can be steered to the origin without violating any of the system constraints along the closed-loop trajectory. General tracking problems can easily be formulated as regulation problems by augmenting the state space appropriately (Pannocchia & Kerrigan, 2003).

One of the key problems in the control of PWA systems is the lack of convexity of the controlled sets, which produces a significant computational overhead. Furthermore, the complexity of the cost-to-go function in the dynamic programming approach in Borrelli et al. (2003); Kerrigan and Mayne (2002) makes it necessary to explore an exponentially growing number of possible target sets during the iterations. The algorithms presented here avoid these issues to some extent by considering ‘simpler’ control objectives (e.g., minimum time control). Note that all controllers presented here guarantee constraint satisfaction for all time as well as asymptotic stability.

4.1. Computation of a minimum time controller

The minimum time controller considered here aims at driving the system state $x(k)$ into a pre-specified target set $\mathcal{O}_\infty^{\text{PWA}}$ in minimum time. Unlike the approaches in Borrelli et al. (2003) and Kerrigan and Mayne (2002), the cost-to-go for the minimum-time controller assumes only integer values. Because, of the ‘simple’ cost-to-go, the target sets which need to be considered at each iteration step are larger and fewer in number than those which would be obtained if an optimal controller with a different cost objective were to be computed (Borrelli et al., 2003; Kerrigan & Mayne, 2002; Baotić et al., 2003b). Thus, both the complexity of the feedback law as well as the computation time are greatly reduced, in general.

When the proposed algorithm terminates, the associated feedback controller will cover the N -step stabilizable set $\mathcal{K}_N^{\text{PWA}}(\mathcal{O}_\infty^{\text{PWA}})$.

Definition 4.1. The set $\mathcal{K}_N^{\text{PWA}}(\mathcal{O}_\infty^{\text{PWA}})$ denotes the N -step stabilizable set for a PWA system (4), i.e., it contains all

states which can be steered into $\mathcal{C}_\infty^{\text{PWA}}$ in N steps. Specifically,

$$\begin{aligned} \mathcal{K}_N^{\text{PWA}}(\mathcal{C}_\infty^{\text{PWA}}) &= \{x(0) \in \mathbb{R}^n \mid \exists u(k) \in \mathbb{U}, \text{ s.t.} \\ x(N) &\in \mathcal{C}_\infty^{\text{PWA}}, x(k) \in \mathbb{X}, \\ x(k+1) &= f_{\text{PWA}}(x(k), u(k)), \forall k \in \{0, \dots, N\}\}. \end{aligned}$$

Accordingly, the set $\mathcal{K}_\infty^{\text{PWA}}(\mathcal{C}_\infty^{\text{PWA}})$ denotes the maximal stabilizable set for $N \rightarrow \infty$.

Note that the N -step stabilizable set $\mathcal{K}_N^{\text{PWA}}(\mathcal{C}_\infty^{\text{PWA}})$ is a control invariant set.

4.1.1. Minimum-time controller: off-line computation

Before presenting the algorithm, some preliminaries will be introduced. Assume a possibly non-convex union \mathcal{X}^0 of L^0 polytopes \mathcal{X}_l^0 , i.e., $\mathcal{X}^0 = \bigcup_{l \in \mathcal{L}^0} \mathcal{X}_l^0$, where $\mathcal{L}^0 \triangleq \{1, 2, \dots, L^0\}$. In the following, the set \mathcal{X} without subscript will be used to denote unions of polytopes while the subscript is used to denote polytopes. All states which can be driven into the set \mathcal{X}^0 for the PWA system (4) are defined by

$$\begin{aligned} \text{Pre}(\mathcal{X}^0) &= \{x \in \mathbb{X} \mid \exists u \in \mathbb{U}, f_{\text{PWA}}(x, u) \in \mathcal{X}^0\} \\ &= \bigcup_{i \in \mathcal{I}} \bigcup_{l \in \mathcal{L}^0} \{x \in \mathcal{D}_i \mid \exists u \in \mathbb{U}, \\ &\quad A_i x + B_i u + f_i \in \mathcal{X}_l^0\}. \\ &= \bigcup_{j \in \mathcal{J}^0} \mathcal{F}_{1,j} \end{aligned}$$

For a fixed i and l , the target set \mathcal{X}_l^0 is convex and the dynamics affine, such that it is possible to apply standard multi-parametric programming techniques to compute the set of states which can be driven into \mathcal{X}_l^0 (Bemporad et al., 2002). Therefore, the set $\text{Pre}(\mathcal{X}^0)$ is a union of polytopes and can be computed by solving $J^0 = D \cdot L^0$ multi-parametric programs, where D denotes the number of dynamics and L^0 is the number of polytopes which define \mathcal{X}^0 . Each of these multi-parametric programs will yield a controller partition $\{\mathcal{P}_{j,r}^0\}_{r=1}^R$ consisting of R controller regions whose union covers the feasible set $\mathcal{F}_{1,j} = \bigcup_{r=1, \dots, R} \mathcal{P}_{j,r}^0$ (see Definition 2.1). Since the set $\text{Pre}(\mathcal{X}^0)$ is computed via multi-parametric programming, we also obtain an associated feedback law $u(x)$ which provides feasible inputs as a function of the state (see Theorem 2.1). Note that the various controller partitions may overlap, but that each controller will drive the state into \mathcal{X}^0 in one time step, i.e., $f_{\text{PWA}}(x, u(x)) \in \mathcal{X}^0$. Henceforth, we will use the notation $\mathcal{X}^{iter+1} = \text{Pre}(\mathcal{X}^{iter}) = \bigcup_{j \in \mathcal{J}^{iter+1}} \mathcal{X}_j^{iter+1}$.

In the following, the algorithm for computing the minimum-time controller for PWA systems will be introduced. In principle, the algorithm is a discrete version of the viability kernel algorithm (Aubin, Lygeros, Quincampoix, Sastry, & Seube, 2002). However, since multi-parametric

programming techniques are applied, the algorithm proposed here will yield a control law and the associated controllable set.

Algorithm 4.1. Minimum-time controller computation

- (1) Compute the invariant set $\mathcal{C}_\infty^{\text{PWA}}$ around the origin (see Fig. 1(a)) as well as the associated Lyapunov function $V(x) = x'Px$ and feedback laws F_i as described by Algorithm 3.1.
- (2) Initialize the set list $\mathcal{X}^0 = \mathcal{C}_\infty^{\text{PWA}}$ and initialize the iteration counter $iter = 0$.
- (3) Compute $\mathcal{X}^{iter+1} = \text{Pre}(\mathcal{X}^{iter}) = \bigcup_{j \in \mathcal{J}^{iter+1}} \mathcal{X}_j^{iter+1}$, by solving a sequence of multi-parametric programs (see Fig. 1(b)). Thus, a feedback controller partition $\{\mathcal{P}_{j,r}^{iter+1}\}_{r=1}^R$ is associated with each obtained set \mathcal{X}_j^{iter+1} . Obviously, the number of regions R of each partition is a function of $iter$ and j .
- (4) For all $j^* \in \mathcal{J}^{iter+1}$: If $\mathcal{X}_{j^*}^{iter+1} \subseteq \{\bigcup_{j \in \mathcal{J}^{iter+1} \setminus \{j^*\}} \mathcal{X}_j^{iter+1}\} \cup \{\bigcup_{i \in \{1, \dots, iter\}} \mathcal{X}_i\}$, then discard $\mathcal{X}_{j^*}^{iter+1}$ from \mathcal{X}^{iter+1} and set $\mathcal{J}^{iter+1} = \mathcal{J}^{iter+1} \setminus \{j^*\}$ (see Figs. 1(c) and (d)).
- (5) If $\mathcal{X}^{iter+1} \neq \emptyset$, set $iter = iter + 1$ and go to step 3.
- (6) For all $k \in \{1, \dots, iter - 1\}$ and $r \in \mathbb{N}^+$ discard all controller regions $\mathcal{P}_{j,r}^{k+1}$ for which $\mathcal{P}_{j,r}^{k+1} \subseteq \bigcup_{i \in \{1, \dots, k\}} \mathcal{X}_i$ since the associated control laws are not time-optimal and will never be applied.

The index $iter$ corresponds to the number of steps in which a state trajectory will enter the terminal set $\mathcal{C}_\infty^{\text{PWA}}$ if a receding horizon control policy is applied. If the algorithm terminates in finite time, then the union of all controlled sets \mathcal{X}^{iter} is the maximum controllable set $\mathcal{K}_\infty^{\text{PWA}}(\mathcal{C}_\infty^{\text{PWA}})$ as given in Definition 4.1.

Remark 4. Note that Algorithm 4.1 may not terminate in finite time, even if the feasible state space is bounded. This is a problem inherent property and not a result of the computation scheme (see Kerrigan, 2000 for further details). It is therefore advisable to specify a maximum step distance N which can be used as a termination criterion in step 5 of Algorithm 4.1. The resulting controller computation will then terminate in finite time and the feedback controller will cover $\mathcal{K}_N^{\text{PWA}}(\mathcal{C}_\infty^{\text{PWA}})$.

Remark 5. The implementation of Algorithm 4.1 requires a function that can detect if a convex polyhedron \mathcal{P}_0 is covered by a finite set of non-empty convex polyhedra $\{\mathcal{P}_r\}_{r=1}^R$, i.e., if $\mathcal{P}_0 \subseteq \bigcup_{r \in \{1, \dots, R\}} \mathcal{P}_r$. For instance, this operation is needed to check if two unions of polyhedra cover the same non-convex set (Raković et al., 2004) (e.g., Step 5 of Algorithm 4.1). Due to space constraints, we refer the reader to Baotić and Torrisi (2003), where an efficient algorithm is given to perform this task.

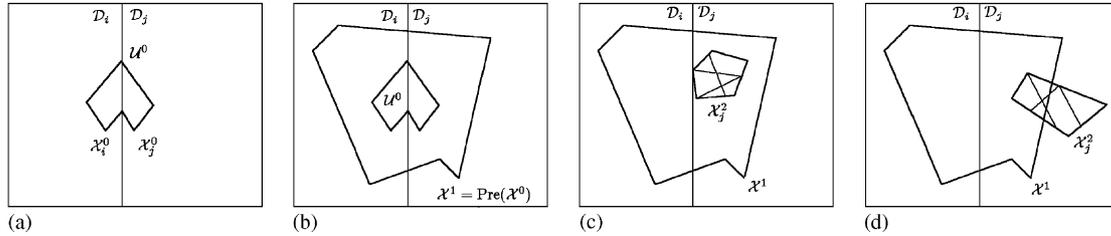


Fig. 1. Description of Algorithm 4.1. (a) Invariant target set $\mathcal{O}_\infty^{\text{PWA}}$. (b) Set of states \mathcal{X}^1 which enter \mathcal{X}^0 in one time step. (c) The transition partition does not expand the controlled set of states. (d) The transition controller expands the controllable set of states.

4.1.2. Minimum-time controller: on-line application

In the minimum-time algorithm presented in this paper, we can take advantage of some of the algorithm features to speed up the on-line region identification procedure. We propose a three-tiered search tree structure which serves to significantly speed up the region identification. Unlike the search tree proposed in Tøndel, Johansen, and Bemporad (2003), the tree structure proposed here is computed automatically by Algorithm 4.1, i.e., no post-processing is necessary. The three levels of the search tree are as follows:

Algorithm 4.2. On-line application of minimum-time controller

- (1) Identify the active dynamics i , such that $x \in \mathcal{D}_i$, $i \in \mathcal{I}$ (see Fig. 2(a)).
- (2) Identify controller set $\mathcal{X}_j^{\text{iter}}$ associated with dynamic i which is ‘closest’ to the target set \mathcal{X}^0 , i.e., $\min_{\text{iter}, j} \text{iter}$, s.t. $x \in \mathcal{X}_j^{\text{iter}}$, $j \in \mathcal{J}^{\text{iter}}$ (see Fig. 2(b)).
- (3) Extract the controller partition $\{\mathcal{P}_{j,r}^{\text{iter}}\}_{r=1}^R$ with the corresponding feedback laws F_r , G_r and identify the region r which contains the state $x \in \mathcal{P}_{j,r}^{\text{iter}}$ (see Fig. 2(c)).
- (4) Apply the control input $u = F_r x + G_r$. Go to 1.

Note that the association of controller partitions $\mathcal{X}_j^{\text{iter}}$ to active dynamics in step 2 is trivially implemented by building an appropriate lookup-table during the off-line computation in Algorithm 4.1.

Theorem 4.1. *The controller obtained with Algorithm 4.1 and applied to a PWA system (4) in a receding horizon control fashion according to Algorithm 4.2, guarantees asymptotic stability and feasibility of the closed-loop system, provided $x(0) \in \mathcal{K}_N^{\text{PWA}}(\mathcal{O}_\infty^{\text{PWA}})$.*

Proof. Assume the initial state $x(0)$ is contained in the set $\mathcal{X}^{\text{iter}}$ with a step distance to $\mathcal{O}_\infty^{\text{PWA}}$ of iter . The control law at step 4 of Algorithm 4.2 will drive the state into a set $\mathcal{X}^{\text{iter}-1}$ in one time step (see step 3 of Algorithm 4.1). Therefore, the state will enter $\mathcal{O}_\infty^{\text{PWA}}$ in iter steps. Once the state enters $\mathcal{O}_\infty^{\text{PWA}}$ the feedback controllers associated with the common quadratic Lyapunov ensure stability. \square

4.2. One-step controller

In the previous section, stability was guaranteed by imposing an appropriate terminal set constraint. In order to cover large parts of the state space, this type of constraint generally entails the use of large prediction horizons which results in controllers with a large number of regions.

In this section, instead of enforcing asymptotic stability with an appropriate terminal set (and the associated cost), we propose to enforce constraint satisfaction only. This can be easily achieved by imposing a set constraint on the first predicted state in the MPC formulation. Hence, the terminal-set constraint $x_N \in \mathcal{T}_{\text{set}}$ becomes superfluous and we do not need to rely on large prediction horizons. Asymptotic stability is analyzed in a second step. This scheme is inspired by promising complexity reduction results for LTI systems in Grieder, Parillo, and Morari (2003) and Grieder and Morari (2003).

4.2.1. Constraint satisfaction

If (3) is solved via multi-parametric programming for any prediction horizon N' with set constraints $x_1 \in \mathcal{K}_N^{\text{PWA}}(\mathcal{O}_\infty^{\text{PWA}})$ and $x_{N'} \in \mathcal{T}_{\text{set}} = \mathbb{R}^n$, the resulting MPC controller will guarantee that the state will remain within $\mathcal{K}_N^{\text{PWA}}(\mathcal{O}_\infty^{\text{PWA}})$ for all time. The set $\mathcal{O}_\infty^{\text{PWA}}$ is computed as described by Algorithm 3.1 and $\mathcal{K}_N^{\text{PWA}}(\mathcal{O}_\infty^{\text{PWA}})$ is obtained by applying Algorithm 4.1. The set constraint on the first step guarantees that the resulting controller partition will be positive invariant, which directly implies feasibility for all time (Blanchini, 1999; Kerrigan, 2000). Note that this allows us to control large volume sets $\mathcal{K}_N^{\text{PWA}}(\mathcal{O}_\infty^{\text{PWA}})$ with short prediction horizons N' , i.e., $N' \ll N$. We will henceforth assume $N' = 1$, $N \rightarrow \infty$ and refer to the proposed controller as one-step controller. Note that in the examples provided in Section 5.1, the set $\mathcal{K}_\infty^{\text{PWA}}(\mathcal{O}_\infty^{\text{PWA}})$ was always finitely determined. This is not always the case such that in practice it is advisable to limit N to be a large but finite value.

Since the target set $\mathcal{K}_\infty^{\text{PWA}}(\mathcal{O}_\infty^{\text{PWA}}) = \bigcup_{c \in \{1, \dots, C\}} \mathcal{K}_\infty^c$ is non-convex in general (i.e., a union of C polytopes \mathcal{K}_∞^c) a controller partition can be obtained by solving a sequence of $C \cdot D$ multi-parametric programs (3), where D corresponds to the total number of different dynamics. Specifically, the one-

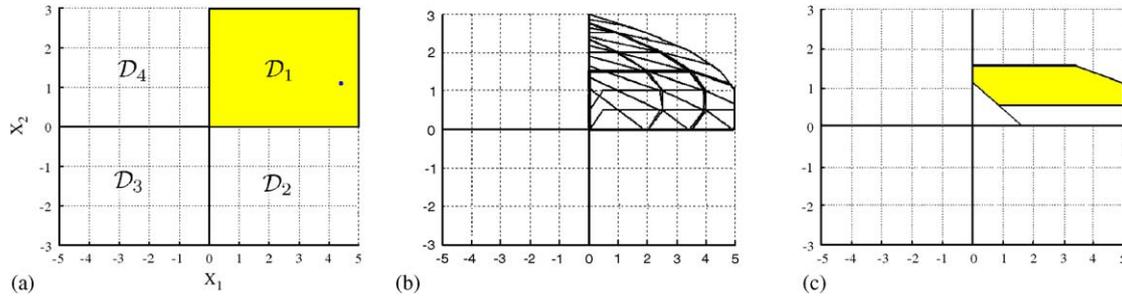


Fig. 2. Illustration of Algorithm 4.2. (a) Identify dynamics \mathcal{D}_i containing the state. (b) Identify feasible controller set \mathcal{X}_j^{iter} containing the state that has the smallest value for $iter$. (c) Extract controller partition $\{\mathcal{P}_{j,r}^{iter}\}_{r=1}^R$ associated to feasible set \mathcal{X}_j^{iter} and identify region $\mathcal{D}_{j,r}^{iter}$ containing the state.

step controller can be obtained by solving $C \cdot D$ problems (3) for $N' = 1$ with $x_1 \in \mathcal{T}_{set} = \mathcal{H}_{\infty}^c$ in (3c) (C different sets) and for D different dynamics in (3d).

4.2.2. Stability analysis

The controller partition obtained in Section 4.2.1 will generally contain overlaps such that the closed-loop dynamics associated with a given state $x(0)$ may not be unique. It is therefore not possible to perform a non-conservative stability analysis of the closed-loop system. However, by using the PWA value function $J_N^*(x)$ in (3a) as a selection criterion, it is possible to obtain a non-overlapping partition (Grieder, Kvasnica, Baotić, & Morari, 2003 or Borrelli, 2003, pp. 158–160) by solving a number of LPs, i.e., only the cost optimal controller is stored.

The resulting controller partition is invariant and a unique controller region r ($x \in \mathcal{P}_r, u = F_r x + G_r$) and unique dynamics l ($x \in \mathcal{D}_l$) is associated with each state x , i.e., the closed-loop system corresponds to an autonomous PWA system

$$x_{k+1} = (A_l + B_l F_r)x_k + B_l G_r + f_l, \quad x_k \in \mathcal{P}_r \cap \mathcal{D}_l \quad (6a)$$

$$= \tilde{A}_r x_k + \tilde{f}_r, \quad x_k \in \mathcal{P}_r. \quad (6b)$$

Since every controller region \mathcal{P}_r is only contained in one unique dynamic \mathcal{D}_l , the update matrix \tilde{A}_r and vector \tilde{f}_r are uniquely defined. In the following, it can now be shown how to formulate the search for a PWA Lyapunov function for autonomous PWA systems as a linear program (LP).

It was shown how to use semi-definite programming (SDP) to compute piecewise quadratic (PWQ) Lyapunov functions for continuous-time PWA systems in Johansson and Rantzer (1998) and for discrete-time PWA systems in Ferrari-Trecate, Cuzzola, Mignone, and Morari (2002) and Grieder, Lüthi, Parillo, and Morari (2003). The search for a PWQ Lyapunov function is conservative, since the SDP formulation is based on the S -procedure, which is not lossless for the cases considered (Boyd et al., 1994). Therefore, instead of searching for a PWQ Lyapunov function via SDP, we here show how to compute a PWA Lyapunov function via LP. The proposed scheme is based on results for con-

tinuous time systems which were published in Johansson (2001).

The computation scheme for the PWA Lyapunov function is non-conservative (i.e., if a PWA Lyapunov function exists for the given partition, it will be found) such that it may succeed when no PWQ Lyapunov function can be found with the schemes in Ferrari-Trecate et al. (2002) and Grieder et al. (2003). However, the converse is also true (see survey Biswas, Grieder, Löfberg, & Morari, 2005 for details). Furthermore, the value function associated with a mpLP controller partition is PWA, such that this function type is a natural candidate in the search for a Lyapunov function. The following theorem is based on Vidyasagar (1993, p. 267) and Ferrari-Trecate et al. (2002):

Theorem 4.2 (Asymptotic stability). *The origin $x = 0$ is asymptotically stable on the set \mathcal{X} if there exists a function $V(x)$ and scalar coefficients $\alpha > 0, \beta > 0, \rho > 0$ such that: $\beta \|x_k\| \geq V(x_k) \geq \alpha \|x_k\|$ and $V(x_{k+1}) - V(x_k) \leq -\rho \|x_k\|, \forall x_k \in \mathcal{X}$ and $V(x) = \infty, \forall x \notin \mathcal{X}$. The successor state x_{k+1} is defined in (6b), $\|\cdot\|$ denotes a vector norm and \mathcal{X} denotes the state space of interest.*

In order to pose the problem of finding a PWA Lyapunov function without introducing conservative relaxations, a region transition map is created. Specifically, a transition map \mathcal{S} is created according to

$$\mathcal{S}(i, j) = \begin{cases} 1 & \text{if } \exists x_k \in \text{int}(\mathcal{P}_i), \text{ s.t. } x_{k+1} \in \mathcal{P}_j, \\ 0 & \text{otherwise,} \end{cases}$$

where x_{k+1} is defined by (4) and Theorem 2.1 and $\text{int}(\cdot)$ denotes the strict interior of a set.

Remark 6. In principle, one LP needs to be solved for each element of the transition map \mathcal{S} , i.e., a total of R^2 LPs, where R denotes the total number of system dynamics. However, instead of solving LPs directly, it is advisable to first compute bounding boxes (hyper-rectangles) for each region \mathcal{P}_r ($r \in \mathcal{R}$). In addition, a bounding box of the affine map of the region $\mathcal{P}_r^+ = \{\tilde{A}_r x + \tilde{f}_r \in \mathbb{R}^n | x \in \mathcal{P}_r\}$ needs to be computed. The number of LPs which need to be solved in order to compute the bounding boxes is linear in the number

of regions R and state space dimension n . This computation is tractable even for very complex partitions. The bounding boxes can be efficiently checked for intersections, such that certain transitions $i \rightarrow j$ can be ruled out. In our experience, the bounding box implementation is the most effective way to compute \mathcal{F} for complex region partitions.

In a second step, the polytopic transition sets \mathcal{P}_{ij} for system (6b) are explicitly computed $\forall i, j \in \{1, \dots, R\} | \mathcal{S}(i, j) = 1$ according to:

$$\mathcal{P}_{ij} = \{x_k \in \mathbb{R}^n | x_k \in \mathcal{P}_i, x_{k+1} \in \mathcal{P}_j\}.$$

If $\mathcal{S}(i, j) = 0$, we can directly set $\mathcal{P}_{ij} = \emptyset$. Subsequently, the vertices of the transition sets ($\text{vert}(\mathcal{P}_{ij})$) and the controller sets ($\text{vert}(\mathcal{P}_i)$) are computed. The problem of finding a PWA Lyapunov function,

$$\text{PWA}_i(x) = L_i x + C_i \quad \text{if } x \in \mathcal{P}_i, \quad i = 1, \dots, R,$$

for the autonomous PWA system (6b) such that the conditions in Theorem 4.2 are satisfied can now be stated as

$$\beta \|x\|_1 \geq \text{PWA}_i(x) \geq \alpha \|x\|_1, \quad \alpha, \beta > 0, \quad (7a)$$

$$\text{PWA}_j(x_{k+1}) - \text{PWA}_i(x_k) \leq \rho \|x_k\|_1, \quad \rho < 0, \quad (7b)$$

$$\forall x \in \text{vert}(\mathcal{P}_i) \quad \forall x_k \in \text{vert}(\mathcal{P}_{ij}) \quad \forall i, j \in \{1, \dots, R\}. \quad (7c)$$

Since the vertices of all sets \mathcal{P}_i and \mathcal{P}_{ij} are known, the resulting problem is linear in $L_i, C_i, \alpha, \beta, \rho$ and can therefore be solved as an LP.

Theorem 4.3. *If the LP (7) associated with the autonomous PWA system (6b) is feasible, then this system is asymptotically stable.*

Proof. Since the function $\text{PWA}_i(x)$ is PWA, it follows that satisfaction of (7a) for all vertices of \mathcal{P}_i implies that the inequalities in (7a) will also hold $\forall x \in \mathcal{P}_i$. Furthermore, if (7b) holds for all vertices of \mathcal{P}_{ij} , it follows from linearity of the system dynamics (6b) that the inequality will hold for all states $x \in \mathcal{P}_{ij}$. Since the partition \mathcal{F}_N is invariant, it follows that $\mathcal{F}_N = \bigcup_{i \in \{1, \dots, R\}} \mathcal{P}_i = \bigcup_{i, j \in \{1, \dots, R\}} \mathcal{P}_{ij}$. Therefore, the inequalities in (7a) and (7b) hold $\forall x \in \mathcal{F}_N$ such that the conditions in Theorem 4.2 are satisfied, i.e., feasibility of (7) implies asymptotic stability of the autonomous PWA system (6b). \square

It should be noted that the required computation time may become large because of the extensive reachability analysis, vertex enumeration and size of the final LP. Specifically, the LP (7) introduces one constraint for each vertex of each region $\mathcal{P}_r, \forall r \in \{1, \dots, R\}$ (see (7a)) and one constraint for each vertex of each $\mathcal{P}_{ij}, \forall i, j \in \{1, \dots, R\}$ (see (7b)). The number of variables is $(n+1)R$, where R denotes the number of regions and n the state space dimension.

However, in the authors experience (Biswas et al., 2005), stability analysis problems for a couple of hundred regions

in a state space dimension of less than five are tractable and the necessary computation effort is comparable to the approaches in Ferrari-Trecate et al. (2002) and Grieder et al. (2003).

4.2.3. One-step controller computation

The one-step control scheme utilizes tools from invariant set computation and stability analysis in order to compute controllers with small prediction horizons which guarantee constraint satisfaction as well as asymptotic stability. The basic procedure consists of two main stages. In the first stage, a one-step optimal controller is computed which guarantees constraint satisfaction for all time (Section 4.2.1). Since constraint satisfaction does not imply asymptotic stability, it is necessary to analyze the stability properties of the closed-loop system in a second stage (Section 4.2.2). Specifically, the algorithm works as follows.

Algorithm 4.3. Computation: one-step controller

- (1) Compute the invariant set $\mathcal{O}_\infty^{\text{PWA}}$ around the origin and an associated Lyapunov function as described by Algorithm 3.1.
- (2) Compute the set $\mathcal{K}_N^{\text{PWA}}(\mathcal{O}_\infty^{\text{PWA}}) = \bigcup_{c \in \{1, \dots, C\}} \mathcal{K}_N^c$ ($N \rightarrow \infty$) by applying Algorithm 4.1.
- (3) Solve a sequence of $C \cdot D$ mpLPs (3) for prediction horizon $N' = 1$ with $\mathcal{T}_{\text{set}} = \mathcal{K}_N^c, \forall c \in \{1, \dots, C\}$ in (3c) and affine dynamics $d \in \{1, \dots, D\}$ in (3d).
- (4) Remove the region overlaps by using the PWA value function as a selection criterion (see Baotić & Torrisi, 2003; Borrelli, 2003 for details).
- (5) Attempt to find a PWA Lyapunov function by solving the LP (7) or attempt to find a PWQ Lyapunov function as described in Grieder et al. (2003).

There is no guarantee that step 2 of Algorithm 4.3 will terminate in finite time or that a Lyapunov function can be found in step 5. The finite time termination conditions are discussed in Remark 4. If no Lyapunov function is found, the resulting controller is still guaranteed to satisfy the system constraints for all time, but no proof of asymptotic stability can be given. Note that this does not imply that the closed-loop system is unstable, it merely shows that no PWA Lyapunov function exists for the given partition.

Theorem 4.4. *If the stability analysis in Step 5 of Algorithm 4.3 is successful and the feedback law obtained in Step 4 is applied to system (4) in a RHC fashion, then the closed-loop system is exponentially stable on $\mathcal{K}_N^{\text{PWA}}(\mathcal{O}_\infty^{\text{PWA}})$ and the system constraints are satisfied for all time.*

Proof. The partition computed in Step 4 is invariant by construction, hence constraint satisfaction is guaranteed. Exponential stability follows trivially from the successful stability analysis in Step 5. \square

Table 1
Off-line CPU-time t and number of controller regions $\#R$ for different algorithms

	Algorithm 4.1		Algorithm 4.3		Algorithm [3]	
	t	$\#R$	t	$\#R$	t	$\#R$
Example 1	1153 s	1519	286 s	522	*	*

The * denotes that the computations were not completed after 7 days. The computation was run on a 2.8GHz Pentium IV CPU running the Windows version of MATLAB 6.5 along with the NAG foundation LP solver.

Remark 7. If the stability analysis in Step 5 of Algorithm 4.3 fails, it is advisable to recompute the controller in Step 3 using different weights R , Q , Q_f and/or a different prediction horizon N' in (3). Slight modifications in these parameters may make the subsequent stability analysis in Step 5 feasible. We have observed that large weights on the states (i.e., Q , Q_f large) and a larger prediction horizon N' have a positive effect on the likelihood of success.

5. Numerical examples

5.1. Controller computation

As was shown in Grieder and Morari (2003) and Grieder et al. (2003) and will also be illustrated in this section, algorithms of type 4.1 and 4.3 generally yield controllers of significantly lower complexity than those which are obtained if a linear norm-objective is minimized as in (3) (Baotić et al., 2003a,b).

Example 1. Consider the three-dimensional PWA system introduced in Mayne and Raković (2003), with $x(k+1) = f^1(x(k))$ if $x_2(k) \leq 1$ and $x(k+1) = f^2(x(k))$ otherwise:

$$f^1(x) = \begin{bmatrix} 1 & 0.5 & 0.3 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(k),$$

$$f^2(x) = \begin{bmatrix} 1 & 0.2 & 0.3 \\ 0 & 0.5 & 1 \\ 0 & 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(k) + \begin{bmatrix} 0.3 \\ 0.5 \\ 0 \end{bmatrix}.$$

Subject to constraints $-10 \leq x_1(k) \leq 10$, $-5 \leq x_2(k) \leq 5$, $-10 \leq x_3(k) \leq 10$, and $-1 \leq u(k) \leq 1$. The weights in the cost function are $Q = I$, $R = 0.1$.

Once the set $\mathcal{C}_\infty^{\text{PWA}}$ is computed, Algorithms 4.1 and 4.3 are applied to Example 1. A runtime comparison of the computation procedures as well as complexity of the resulting solutions are reported in Table 1. Even though the proposed computation schemes are significantly more efficient than existing approaches, it is easy to come up with examples where the associated computation time is prohibitive.

In order to compare low complexity control strategies discussed in this paper with the cost-optimal approach of Baotić et al. (2003b), we generated 10 random PWA sys-

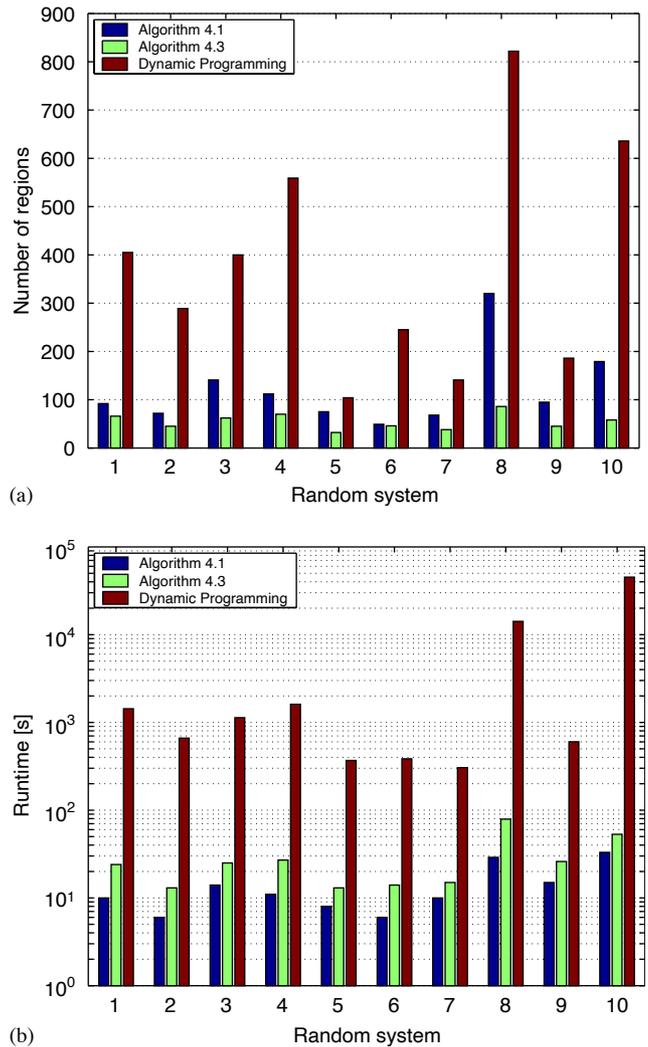


Fig. 3. Complexity and runtime for 10 random PWA systems. (a) Number of regions generated by different algorithms. (b) Runtime for different algorithms.

tems with two states, one input and four PWA dynamics. All elements in the state space matrices were assigned random values between $[-2, 2]$ (i.e., stable and unstable systems were considered). Each of the random PWA systems consists of four different affine dynamics which are defined over non-overlapping random sets whose union covers the square $\mathbb{X} = [-5, 5] \times [-5, 5]$. The origin was chosen to be on the boundary of multiple dynamics. All simulation runs as

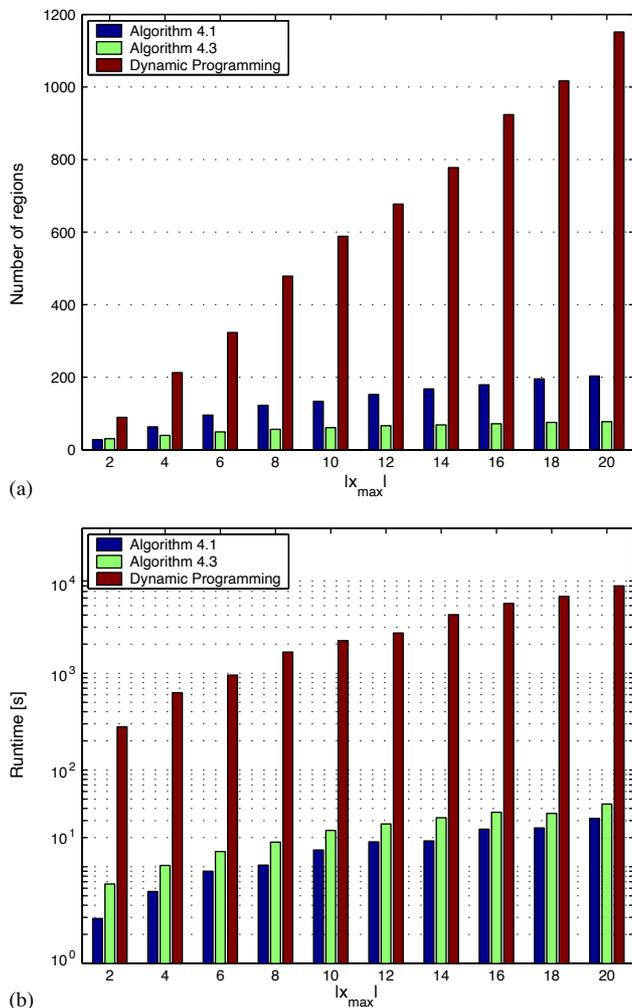


Fig. 4. Complexity and runtime versus size of exploration space (average over 10 random PWA systems). (a) Number of regions vs. size of exploration space. (b) Runtime vs. size of exploration space.

well as the random system generation was performed with the MPT toolbox (Kvasnica, Grieder, & Baotić, 2004).

Algorithms 4.1 and 4.3, as well as the cost-optimal strategy of Baotić et al. (2003b) were applied to these systems. Complexity of the resulting solution and run time of each algorithm are depicted graphically in Figs. 3(a) and (b).

To further investigate the behavior of different control strategies, another test on a set of 10 random PWA systems was performed to show how the complexity of Algorithms 4.1 and 4.3 scales with increasing volume of the exploration space. A comparison with the approach in Baotić et al. (2003b) is depicted in Figs. 4(a) and (b). For the random systems considered here, the necessary runtime is reduced by two orders of magnitude and the solution complexity is reduced by one order of magnitude, on average. In addition, these differences become larger with increasing size of the state constraints. Although we have not come across any examples where the proposed schemes are inferior to the ap-

proaches in Borrelli et al. (2003) and Kerrigan and Mayne (2002), we are not able to prove that no such cases exist.

Due to space constraints, we refer the reader to Grieder (2004) for a more comprehensive comparison between the algorithms.

6. Conclusion

A scheme to compute terminal sets (along with the associated ‘cost-to-go’) for generic PWA systems was presented, which may be used in the context of receding horizon control to obtain asymptotic stability guarantees for the closed-loop system. These sets are subsequently used to derive two novel algorithms to compute low complexity feedback controllers for constrained PWA systems. Both controllers guarantee constraint satisfaction for all time as well as asymptotic stability. The computation scheme iteratively solves a series of multi-parametric programs such that a feedback controller is obtained which drives the state into a target set in minimum time. As a side product, a search tree for efficient on-line identification of the optimal feedback law is automatically constructed. A second computation scheme (referred to as one-step control) is also presented, which separately deals with the requirement of constraint satisfaction and asymptotic stability. In the one-step scheme, stability is not enforced but merely verified a posteriori. While the resulting controller is of even lower complexity than the minimum-time controller, there is no a priori guarantee that the closed-loop system will be asymptotically stable. In order to analyze stability, the paper introduces a method of computing PWA Lyapunov functions for a given autonomous PWA system. The proposed method is based on linear programming and is guaranteed to find a PWA Lyapunov function for a given partition, if it exists.

In an extensive case study, it is observed that both algorithms reduce complexity versus optimal controllers (Borrelli et al., 2003; Kerrigan & Mayne, 2002) by several orders of magnitude. The proposed procedures make problems tractable that were previously too complex to be tackled by the methods in Borrelli et al. (2003) and Kerrigan and Mayne (2002). Although we have not come across any examples where the proposed schemes are inferior to the approaches in Borrelli et al. (2003) and Kerrigan and Mayne (2002), we are not able to prove that no such cases exist.

The presented algorithms are contained in the MPT toolbox (Kvasnica et al., 2004) <http://control.ee.ethz.ch/~mpt>.

References

- Aubin, J.-P., Lygeros, J., Quincampoix, M., Sastry, S., & Seube, N. (2002). Impulse differential inclusions: A viability approach to hybrid systems. *IEEE Transactions on Automatic Control*, 47(1), 2–20.
- Baotić, M., Christophersen, F. J., & Morari, M. (2003a). A new algorithm for constrained finite time optimal control of hybrid systems with a linear performance index. In *European control conference*, Cambridge, UK, September.

- Baotić, M., Christophersen, F. J., & Morari, M. (2003b). Infinite time optimal control of hybrid systems with a linear performance index. In *Proceedings of the conference on decision and control*, Maui, Hawaii, USA, December.
- Baotić, M., & Torrisi, F. (2003). Polycover. Technical Report AUT03-11, Automatic Control Lab, ETHZ, Switzerland. <http://control.ee.ethz.ch>.
- Bemporad, A., Borrelli, F., & Morari, M. (2000). Explicit solution of LP-based model predictive control. In *Proceedings 39th IEEE conference on decision and control*, Sydney, Australia, December.
- Bemporad, A., Fukuda, K., & Torrisi, F. (2001). Convexity recognition of the union of polyhedra. *Computational Geometry*, 18, 141–154.
- Bemporad, A., & Morari, M. (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3), 407–427.
- Bemporad, A., Morari, M., Dua, V., & Pistikopoulos, E. (2002). The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1), 3–20.
- Biswas, P., Grieder, P., Löfberg, J., & Morari, M. (2005). A survey on stability analysis of discrete-time piecewise affine systems. In *Proceedings of 16th IFAC World congress*, June.
- Blanchini, F. (1999). Set invariance in control—a survey. *Automatica*, 35(11), 1747–1767.
- Borrelli, F. (2003). Constrained optimal control of linear and hybrid systems. Lecture notes in control and information sciences, vol. 290. Berlin: Springer.
- Borrelli, F., Baotić, M., Bemporad, A., & Morari, M. (2003). An efficient algorithm for computing the state feedback optimal control law for discrete time hybrid systems. In *Proceedings of the American control conference*, Denver, Colorado, USA, June.
- Boyd, S., El Ghaoui, L., Feron, E., & Balakrishnan, V. (1994). *Linear matrix inequalities in system and control theory studies in applied mathematics*. Philadelphia, PA: SIAM.
- Ferrari-Trecate, G., Cuzzola, F. A., Mignone, D., & Morari, M. (2002). Analysis of discrete-time piecewise affine and hybrid systems. *Automatica*, 38(12), 2139–2146.
- Gal, T. (1995). *Postoptimal analyses, parametric programming, and related topics* (2nd ed.). Berlin: de Gruyter.
- Gilbert, E. G., & Tan, K. T. (1991). Linear systems with state and control constraints: the theory and applications of maximal output admissible sets. *IEEE Transactions on Automatic Control*, 36(9), 1008–1020.
- Grieder, P. (2004). *Efficient computation of feedback controllers for constrained systems*. Ph.D. thesis, ETH, Zurich, Switzerland, October.
- Grieder, P., Kvasnica, M., Baotić, M., & Morari, M. (2003). *Low complexity control of piecewise affine systems with stability guarantee*. Technical Report AUT03-13, Automatic Control Lab, ETHZ, Switzerland. <http://control.ee.ethz.ch>.
- Grieder, P., Lüthi, M., Parillo, P., & Morari, M. (2003). Stability & feasibility of receding horizon control. In *European control conference*, Cambridge, UK, September.
- Grieder, P., & Morari, M. (2003). Complexity reduction of receding horizon control. In *Proceedings of the 42th IEEE conference on decision and control*, Maui, Hawaii, USA, December.
- Grieder, P., Parillo, P., & Morari, M. (2003). Robust receding horizon control—analysis and synthesis. In *Proceedings of the 42th IEEE conference on decision and control*, Maui, Hawaii, USA, December.
- Heemels, W. P. M. H., de Schutter, B., & Bemporad, A. (2001). Equivalence of hybrid dynamical models. *Automatica*, 37(7), 1085–1091.
- Johansson, M. (2002). Piecewise linear control systems—a computational approach. In Lecture notes in control and information sciences, vol. 284. Berlin: Springer.
- Johansson, M., & Rantzer, A. (1998). Computation of piecewise quadratic Lyapunov functions for hybrid systems. *IEEE Transactions on Automatic Control*, 43(4), 555–559.
- Kerrigan, E. C. (2000). *Robust constraints satisfaction: invariant sets and predictive control*. Ph.D. thesis, Department of Engineering, The University of Cambridge, Cambridge, England.
- Kerrigan, E. C., & Mayne, D. Q. (2002). Optimal control of constrained, piecewise affine systems with bounded disturbances. In *Proceedings of 41st IEEE conference on decision and control*, Las Vegas, Nevada, USA, December.
- Kothare, M. V., Balakrishnan, V., & Morari, M. (1996). Robust constrained model predictive control using linear matrix inequalities. *Automatica*, 32(10), 1361–1379.
- Kvasnica, M., Grieder, P., & Baotić, M. (2004). *Multi parametric toolbox (MPT)*. <http://control.ee.ethz.ch/~mpt/>.
- Lazar, M., Heemels, W., Weiland, S., Bemporad, A. (2004). Stabilization conditions for model predictive control of constrained PWA systems. In *Proceedings of the 43th IEEE conference on decision and control*. Bahamas, December.
- Mayne, D. Q., & Raković, S. (2003). Model predictive control of constrained piecewise affine discrete-time systems. *International Journal of Robust and Nonlinear Control*, 13(3), 261–279.
- Mayne, D. Q., Rawlings, J., Rao, C., & Scokaert, P. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6), 789–814.
- Mignone, D., Ferrari-Trecate, G., & Morari, M. (2000). Stability and stabilization of piecewise affine and hybrid systems: An LMI approach. In *Proceedings of the 39th IEEE conference on decision and control*, December.
- Pannocchia, G., & Kerrigan, E. (2003). Offset-free control of constrained linear discrete-time systems subject to persistent unmeasured disturbances. In *42nd conference on decision and control*, Maui, Hawaii, USA, December.
- Raković, S. V., Grieder, P., Kvasnica, M., Mayne, D. Q., & Morari, M. (2004). Computation of invariant sets for piecewise affine discrete time systems subject to bounded disturbances. In *IEEE conference on decision and control*, December.
- Sontag, E. (1981). Nonlinear regulation: The piecewise linear approach. *IEEE Transactions on Automatic Control*, 26(2), 346–358.
- Tøndel, P., Johansen, T., & Bemporad, A. (2003). Evaluation of piecewise affine control via binary search tree. *Automatica*, 39(5), 945–950.
- Vidyasagar, M. (1993). *Nonlinear systems analysis*. 2nd ed., Englewood Cliffs, NJ: Prentice-Hall.



Pascal Grieder received a diploma in electrical engineering in 2002 from the Swiss Federal Institute of Technology (ETH), Zurich, Switzerland. He completed his diploma thesis research at Stanford University before returning to Zurich as a Dr.sc.tech. student at the Automatic Control Laboratory of ETH, where he got his Ph.D. in 2005. His research interests include receding horizon control, hybrid systems and computational geometry.



Michal Kvasnica was born in 1977. He obtained a diploma in chemical engineering in 2000 from the Slovak Technical University, Bratislava, Slovakia. He is currently a Ph.D. candidate at the Swiss Federal Institute of Technology (ETH), Zurich, Switzerland. His research interests are in efficient software tools for optimal control and analysis of hybrid systems and computational geometry.



Mato Baotić received the B.Sc. and M.Sc. degrees, both in Electrical Engineering, from the Faculty of Electrical Engineering and Computing (FEEC), University of Zagreb, Croatia in 1997 and 2000, respectively. He received a scholarship of the Swiss Government and spent a school year 2000/01 as a visiting researcher at the Automatic Control Lab, Swiss Federal Institute of Technology (ETH) Zurich. In 2005, he received the Ph.D. from the ETH Zurich, Switzerland.

Currently, he is a post-doc at the Department of Control and Computer Engineering in Automation, FEEC, University of Zagreb, Croatia. His research interests include mathematical programming, hybrid systems, optimal control and model predictive control.



Manfred Morari was appointed head of the Automatic Control Laboratory in 1994 at the Swiss Federal Institute of Technology (ETH) in Zurich. Earlier 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 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, and was elected to the National Academy of Engineering (US). Professor Morari has held appointments with Exxon and ICI plc and serves on the technical advisory board of several major corporations.



Brief paper

Enumeration-based approach to solving parametric linear complementarity problems[☆]



Martin Herceg^a, Colin N. Jones^b, Michal Kvasnica^c, Manfred Morari^a

^a Automatic Control Laboratory, ETH Zurich, Physikstrasse 3, 8092 Zurich, Switzerland

^b Automatic Control Laboratory, EPFL Lausanne, Station 9, 1015 Lausanne, Switzerland

^c Institute of Inf. Eng., Automation, and Mathematics, STU Bratislava, Radlinského 9, 81237 Bratislava, Slovakia

ARTICLE INFO

Article history:

Received 4 November 2014

Received in revised form

3 August 2015

Accepted 2 September 2015

Keywords:

Model predictive control
Multiparametric programming
Parametric linear-complementarity
problem

ABSTRACT

This paper presents a solution method for parametric linear complementarity problems (PLCP) that relies on an enumeration technique to discover all feasible bases. The enumeration procedure is based on evaluating all possible combinations of active constraints and testing for feasibility. Although the enumeration approach is known to grow exponentially in the number of constraints, the formulation of the PLCP allows incorporation of cheap rank tests to quickly prune the infeasible directions in the exploration. The motivation for the development of the enumeration based PLCP solver is that it represents a direct method to solve parametric linear and quadratic optimization problems as well as their mixed-integer counterparts. These types of problems often arise in the field of model predictive control for linear and hybrid systems. The enumeration based PLCP solver offers another alternative to compute explicit solutions in the field of hybrid model predictive control that can be extremely effective in some important cases.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

Hybrid model predictive control (MPC) represents a successful approach to tackle constrained control problems that involve logical decisions (Bemporad & Morari, 1999). The leading idea here is to employ optimization tools that provide effective solutions to control problems involving binary variables. For special classes of hybrid control problems it has been shown in Borrelli (2003) that the resulting optimization problems can be solved explicitly by employing parametric programming techniques. In particular, these problems are referred to as parametric mixed-integer linear programs (PMILP) and parametric mixed-integer quadratic programs (PMIQP).

In the literature Borrelli, Bemporad, and Morari (2014, Ch. 18), there are two main approaches that solve mixed-integer parametric problems. In the batch approach, the idea is to employ a branch and bound strategy to decompose the problem into sequences of simpler parametric linear problems (Acevedo & Pistikopoulos, 1997; Dua, Bozinis, & Pistikopoulos, 2002; Dua & Pistikopoulos, 2000). In the recursive approach, the optimization problem is solved recursively in a dynamic programming fashion. For a detailed overview of methods to parametric programming problems, the reader is referred to the survey (Pistikopoulos, Dominguez, Panos, Kouramas, & Chinchuluun, 2012).

The majority of developed strategies for solving parametric optimization problems directly avoids the complete enumeration approach because in the worst case it may require exploring all possible combinations of active constraints. Despite this fact, there have been research efforts that explore enumerative approaches in recent years. In particular, Gupta, Bhartiya, and Nataraj (2011) present the enumerative approach to solve parametric quadratic problems (PQP), which features pruning of combinations in infeasible directions. The enumerative PQP algorithm has been improved in Feller, Johansen, and Olaru (2013) by employing symmetry properties of MPC formulations. Furthermore, in Feller and Johansen (2013) it has been shown on a case study with 82 parameters that this approach can be a viable method to solve simple input–output constrained MPC problems explicitly.

[☆] M. Kvasnica gratefully acknowledges the contribution of the Scientific Grant Agency of the Slovak Republic under the grant 1/0403/15, and the financial support of the Slovak Research and Development Agency under the project APVV 0551-11. The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Martin Guay under the direction of Editor Ian R. Petersen.

E-mail addresses: herceg@control.ee.ethz.ch (M. Herceg), colin.jones@epfl.ch (C.N. Jones), michal.kvasnica@stuba.sk (M. Kvasnica), morari@control.ee.ethz.ch (M. Morari).

The motivation in this work is to extend the enumerative algorithm of Gupta et al. (2011) to the linear complementarity problem (LCP) framework because it represents a superclass for parametric linear and quadratic problems (Murty, 1997) and includes their mixed-integer forms. While in the literature there can be found methods to tackle general LCPs without restrictive assumptions on the structure (Al-Khayyal, 1987; De Schutter & De Moor, 1995; Júdice, 2012; Moort, Vandenberghe, & Vandewalle, 1992; Pardalos & Rosen, 1988; Sherali, Krishnamurthy, & Al-Khayyal, 1998) there is a lack of approaches to solve general PLCPs. The lexicographic extension of Lemke's method for convex PLCPs has been studied by Columbano, Fukuda, and Jones (2009) and Jones and Morari (2006) and the latest implementation of the algorithm is available in the Multi-Parametric Toolbox (Herceg, Kvasnica, Jones, & Morari, 2013). The extension to general PLCPs has been proposed in Li and Ierapetritou (2010) where the problem is transformed into a parametric mixed-integer optimization problem and solved using multiparametric techniques.

The interest here is to present an enumeration based PLCP solver that can handle general PLCPs and test its applicability to a hybrid MPC problem. In particular, it will be shown that the hybrid MPC problem can be formulated as PLCP and solved explicitly. The algorithm has been implemented in MPT3 and the readers are welcome to download the toolbox and to experiment with some of the hybrid MPC examples.

The first part of the paper presents the PLCP problem and its relationship with PMILP/PMIQP is highlighted. The second part presents the main algorithm and analyzes its properties. Subsequently, the example section shows the performance of the algorithm on two MPC examples. Based on these results, concluding remarks are given in the final section of the paper.

2. Notation and definitions

For a finite set I , $|I|$ denotes its cardinality. For a real matrix $C \in \mathbb{R}^{m \times n}$ and the index set $I \subseteq \{1, \dots, m\}$, we denote by $C_{I,*} \in \mathbb{R}^{|I| \times n}$ the matrix formed by rows of C indexed by I . If $J \subseteq \{1, \dots, n\}$, then $C_{I,J} \in \mathbb{R}^{|I| \times |J|}$ denotes the submatrix of C formed by rows indexed by I and columns indexed by J . Similarly, the matrix $C_{*,J} \in \mathbb{R}^{m \times |J|}$ is formed from C taking all rows and columns indexed by the set J . If $c \in \mathbb{R}^n$ is a vector, then c_J is the vector formed by the elements of c in J . The identity matrix with dimension n is denoted as I_n .

A polyhedron $\mathcal{P} = \{\theta \in \mathbb{R}^d \mid L\theta \leq l\}$ is a convex set given as the intersection of a finite number of closed halfspaces and a polytope is a bounded polyhedron. The polyhedron is called full-dimensional if there exists a d -dimensional ball with a positive radius contained in \mathcal{P} and no $(d+1)$ -dimensional ball with positive radius contained in \mathcal{P} .

3. Problem formulation

3.1. Parametric linear complementarity problem

Consider the following PLCP

$$\forall \theta \in \Theta \quad \text{find } w, z \quad (1a)$$

$$\text{s.t.: } w - Mz = q + Q\theta, \quad (1b)$$

$$w^T z = 0, \quad (1c)$$

$$w, z \geq 0, \quad (1d)$$

where the problem data is given by a real matrix $M \in \mathbb{R}^{n \times n}$, real vector $q \in \mathbb{R}^n$, and real matrix $Q \in \mathbb{R}^{n \times d}$. The $w \in \mathbb{R}^n$ and $z \in \mathbb{R}^n$ are the decision variables and $\theta \in \mathbb{R}^d$ is the vector of parameters that is restricted to lie in a full-dimensional polytope Θ (1a). The relation (1c) corresponds to a linear complementarity constraint

where for each pair of variables w_i and z_i , one must be zero for the condition to hold.

Using the substitution $A = [I_n, -M]$, $x = [w^T, z^T]^T$, the problem (1) can be rewritten in a more compact form

$$\forall \theta \in \Theta \quad \text{find } x \quad (2a)$$

$$\text{s.t.: } A_{*,B} x_B = q + Q\theta, \quad (2b)$$

$$x_N = 0, \quad (2c)$$

$$x_B \geq 0 \quad (2d)$$

where the index set $B \subset \{1, \dots, 2n\}$ is given such that $|B| = n$ and $\text{rank}(A_{*,B}) = n$. The set B is referred to as a basis, and its complement is $N = \{1, \dots, 2n\} \setminus B$. The variables x_B corresponding to the basis B are denoted as basic, and to x_N as non-basic. The basis B is called complementary if the complementarity condition (1c) holds, that is for any index $i \in \{1, \dots, 2n\}$, i is in the basis B if and only if its complement $\bar{i} = (i + n) \bmod 2n$ is not. The importance of the formulation (2) is that the complementarity condition (1c) is no longer present in (2) but it has been encoded with respect to the complementary basis B .

If the solution of linear equations (2b) in the basic variables, i.e.

$$x_B = A_{*,B}^{-1}(q + Q\theta), \quad (3)$$

is nonnegative for some θ , then the complementary basis B is called feasible. The set of parameters for which $x_B \geq 0$ then forms the critical region \mathcal{P}_B

$$\mathcal{P}_B = \{\theta \in \Theta \mid -A_{*,B}^{-1}Q\theta \leq A_{*,B}^{-1}q\}, \quad (4)$$

in which the affine solution map $\theta \in \mathbb{R}^d \mapsto x \in \mathbb{R}^{2n}$ is given by

$$\begin{pmatrix} x_B \\ x_N \end{pmatrix} = \begin{pmatrix} A_{*,B}^{-1}Q \\ 0 \end{pmatrix} \theta + \begin{pmatrix} A_{*,B}^{-1}q \\ 0 \end{pmatrix} \quad \text{if } \theta \in \mathcal{P}_B. \quad (5)$$

Every feasible basis leads to a critical region (4) in the parameter space Θ . The union of all critical regions forms the explicit solution to the problem (2). For practical reasons, only the non-empty and full-dimensional regions are considered as a part of the solution.

In the next section we provide an algorithm to solve (1) for the following four types of optimization problems that often arise in the formulation of linear and hybrid model predictive control:

- Parametric linear program (PLP)
- Parametric quadratic program (PQP)
- Parametric mixed-integer linear program (PMILP)
- Parametric mixed-integer quadratic program (PMIQP)

In this work it is assumed that the matrix M in (1) encodes above optimization problems and the recent methods developed in Columbano et al. (2009), Herceg et al. (2013) and Jones and Morari (2006) do not apply in this general setting. The objective of the next section is to derive the PLCP representation (1) for all four aforementioned optimization programs.

3.2. Relation to mixed-integer linear/quadratic program

The common representation of PLP/PQP/PMILP/PMIQP problems can be put in the form

$$\min_u \frac{1}{2} u^T H u + (F\theta + f)^T u \quad (6a)$$

$$\text{s.t.: } G u \leq g + D\theta, \quad (6b)$$

$$u \geq 0, \quad (6c)$$

$$u_{l_b} \in \{0, 1\}^{n_b} \quad (6d)$$

where the problem data is given by $H \succeq 0$, $H \in \mathbb{R}^{n_u \times n_u}$, $F \in \mathbb{R}^{n_u \times d}$, $f \in \mathbb{R}^{n_u}$ in the objective function (6a) and $G \in \mathbb{R}^{m \times n_u}$, $g \in \mathbb{R}^m$, $D \in \mathbb{R}^{m \times d}$ in the constraints (6b). The decision variables $u \in \mathbb{R}^{n_u}$

are restricted to be nonnegative by (6c) and some of the variables, indexed by the set I_b , can belong to a binary set (6d) with the cardinality n_b . The problem is parameterized in the parameters $\theta \in \mathbb{R}^d$ which are restricted to a bounded polytope Θ .

The formulation (6) corresponds to a PMIQP. If there is no binary set I_b present, the problem represents a convex QQP because $H \succeq 0$. One can easily check that if the matrix $H = 0$, the formulation reduces to a PMILP. In case there are no binary variables and $H = 0$, the problem formulation (6) results in a PLP.

Define by I_r the index set of variables that correspond to continuous elements of the vector u with the cardinality $n_r = |I_r|$. Furthermore, by $v \in \mathbb{R}^{n_u}$ and $s \in \mathbb{R}^m$ we denote the additional slack variables that help to formulate the PLCP.

It has been shown in Audet, Hansen, Jaumard, and Savard (1997), De Schutter, Heemels, and Bemporad (2002), Judice and Mitra (1988) and Pardalos (1994) that the binary variables (6d) can be encoded as the linear complementarity constraint (1c) with the help of slack variables v_{I_b} as follows:

$$v_{I_b} + u_{I_b} = 1, \tag{7a}$$

$$v_{I_b}^T u_{I_b} = 0, \tag{7b}$$

$$v_{I_b}, u_{I_b} \geq 0. \tag{7c}$$

Practically, the constraint set (7) describes an LCP that captures all possible combinations of binary variables u_{I_b} and therefore is of combinatorial nature. For example, in the case of five elements in the binary set $|I_b| = 5$, the feasible set comprises of $2^5 = 32$ different combinations where the values of u_{I_b} are fixed.

The specificity of the problem (6) is that for a fixed value of binary variables u_{I_b} , the structure results in a convex program for which one can derive the optimality conditions

$$s = -G_{*,I_b} u_{I_b} - G_{*,I_r} u_{I_r} + g + D\theta \geq 0 \tag{8a}$$

$$v_{I_r} = \frac{1}{2}(H_{I_b,I_r}^T + H_{I_r,I_b}^T)u_{I_b} + H_{I_r,I_r} u_{I_r} + \dots + (F_{I_r,*}\theta + f_{I_r}) + G_{I_r,*}^T \lambda \geq 0 \tag{8b}$$

$$0 = v_{I_r}^T u_{I_r} + s^T \lambda \tag{8c}$$

where $\lambda \in \mathbb{R}^m$ denotes Lagrange multipliers for the inequality constraints. Although the conditions (8) characterize the solution of the convex subproblem, the inclusion of the complementarity constraints (7) defines the feasible set of the original, non-convex problem (6) for all possible combinations of the binary variables. Putting all relations (8) and (7) to a matrix form gives

$$\underbrace{\begin{pmatrix} v_{I_b} \\ v_{I_r} \\ s \end{pmatrix}}_w - \underbrace{\begin{pmatrix} -I_{n_b} & 0 & 0 \\ \frac{1}{2}(H_{I_b,I_r}^T + H_{I_r,I_b}^T) & H_{I_r,I_r} & G_{I_r,*}^T \\ -G_{*,I_b} & -G_{*,I_r} & 0 \end{pmatrix}}_M \underbrace{\begin{pmatrix} u_{I_b} \\ u_{I_r} \\ \lambda \end{pmatrix}}_z = \underbrace{\begin{pmatrix} 1 \\ f_{I_r} \\ g \end{pmatrix}}_q + \underbrace{\begin{pmatrix} 0 \\ F_{I_r,*} \\ D \end{pmatrix}}_Q \theta, \tag{9a}$$

$$v_{I_b}^T u_{I_b} + v_{I_r}^T u_{I_r} + s^T \lambda = 0, \tag{9b}$$

$$v_{I_b}, v_{I_r}, u_{I_b}, u_{I_r}, s, \lambda \geq 0, \tag{9c}$$

$$\theta \in \Theta, \tag{9d}$$

which corresponds to the PLCP formulation (1) with new variables $w \in \mathbb{R}^{n_u+m}$ and $z \in \mathbb{R}^{n_u+n}$. Note that the non-convexity of the original problem (6) has been transformed to the indefinite matrix M in (9a) with n_b negative eigenvalues because of negative identity matrix that resides on the diagonal of M . Hence, the single PLCP formulation (9) encompasses all four PLP/QQP/PMILP/PMIQP

optimization problems (6). In the case of PMILP/PMIQP, the feasible set is built from all admissible combinations of binary variables that give the objective function (6a) different values. The optimal solution to the problem (6) corresponds to a particular combination of decision variables (9) with the minimum cost function (6a).

Remark 1. Problem formulations with equality constraints can be transformed to (6) with the help of auxiliary variables, see for instance Boyd and Vandenberghe (2004) for reference.

4. Enumeration-based solution approach

In this section we show how to synthesize the explicit solution to (1), i.e., to find all full-dimensional critical regions (4) along with the associated affine solution maps (5). The proposed approach is based on enumerating feasible complementary bases, coupled with pruning of infeasible combinations. The first part of this section presents sequential generation of all possible combinations of bases. In the second part the main algorithm will be introduced that identifies the feasible bases. Finally, the third part investigates the complexity properties of the algorithm.

4.1. Generation of bases

Every complementary basis B in the PLCP problem (2) is characterized by a pair of basic variables x_B and nonbasic variables $x_{N'}$ such that the complementarity constraint (1c) is satisfied. While the basic variables are restricted to be nonnegative, the nonbasic variables are fixed to zero, i.e., $x_{N'} = 0$. The formulation (2) represents the PLP and it is essentially a subproblem of (1) for a fixed basis B . Hence, the implicit enumeration strategy of Gupta et al. (2011) can be applied here to generate sequences of complementary bases. The main idea in the enumeration strategy is to consecutively test each row of (2b) constraint to check which column index forms a feasible (3) or infeasible basis.

Denote by $i \in \{1, \dots, n\}$ the i th constraint row of (2b). From a complementarity condition (1c) it follows that for a single constraint either the column index i belongs to the basis, or its complement $\bar{i} = i + n$. In this manner one can generate partial complementary bases B' starting from the index 1 as follows

i	$S' = \{B', \dots\}$
{1}	{{1}, {n+1}}
{1, 2}	{{1, 2}, {1, n+2}, {n+1, 2}, {n+1, n+2}}
...	...

Define the logical function that operates over the partial complementary bases B'

$$\mathcal{F}(B') = \begin{cases} 1 & \text{if } \exists x \geq 0, \theta \in \Theta \text{ s.t.} \\ & Ax = q + Q\theta \\ & x_{B'} \geq 0, x_{N'} = 0 \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

where N' denotes the set of complementary indices to B' . The function (10) evaluates to $\mathcal{F}(B') = 1$ if and only if the partial complementary basis B' is feasible. The evaluation of (10) requires solving a feasibility LP with x and θ as decision variables.

The objective of the enumeration strategy is to organize the feasibility checks in a tree such that not all partial complementary bases B' will be tested, and therefore avoiding the exploration of 2^n combinations. The pruning part is covered by the following lemma which is introduced without the proof because it is an obvious consequence of Theorem by Gupta et al. (2011).

Lemma 2. If $\mathcal{F}(B') = 0$, then $\mathcal{F}(B' \cup i) = 0$ and $\mathcal{F}(B' \cup i + n) = 0 \forall i \in (\{1, \dots, n\} \setminus B')$.

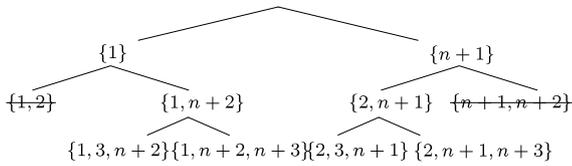


Fig. 1. Generation of basis candidates starting from the constraint index $i = 1$. If the set B' results in an infeasible basis, then all its children are discarded from the tree.

Generation of partial complementary bases and applying the Lemma 2 lead to a tree structure. An example of such a tree with the starting index $i = 1$ to n is depicted in Fig. 1. The details of the proposed enumeration algorithm are revealed next.

4.2. The enumerative algorithm

The main contribution of this paper is summarized by the following theorem:

Theorem 3. Consider the PLCP (1) with the problem data M , q , Q , and a full-dimensional polytope Θ . The Algorithm 1 finds all feasible bases that define the solution given by (5).

Proof. The algorithm proceeds consecutively from the first row of (2) to the last row and generates all possible combinations of partial complementary bases B' as shown graphically in Fig. 1. If the partial complementary basis B' evaluates (10) to 0, i.e. $\mathcal{F}(B') = 0$, then a superset of partial complementary bases containing B' remains infeasible and according to Lemma 2 can be pruned. Thus, only the set of partial feasible bases $T = \{B', \dots\}$ is then carried over to a next iteration and therefore only feasible bases are returned by the algorithm. \square

Algorithm 1 Enumeration based algorithm to solve PLCP (1).

Input: PLCP input data M , q , Q , and Θ
Output: Feasible bases $S = \{B_1, B_2, \dots\}$

```

1:  $S = \emptyset$ 
2: for  $i = \{1, \dots, n\}$  do
3:    $T = \emptyset$ 
4:   for each  $B \in S$  do
5:      $S' = \{B, i, \{B, i+n\}\}$ 
6:     for each  $B' \in S'$  do
7:       if  $\text{rank}(A_{*,B'}) = i$  then
8:         if  $\mathcal{F}(B') = 1$  then
9:            $T \leftarrow \{T, B'\}$ 
10:        end if
11:       end if
12:     end for
13:   end for
14:    $S = T$ 
15: end for
  
```

The enumeration-based PLCP algorithm is shown in Algorithm 1. The inputs are given by the problem data M , q , Q , and Θ of PLCP (1). The algorithm loops sequentially through each equality constraint in (2b) which is evident at the line 2. The partial complementary bases B' are generated at the line 5 to form the set S' . After the bases generation, each of the new partial complementary basis B' is added to a list $T = \{T, B'\}$ only if $\mathcal{F}(B') = 1$ is feasible by (10), which is the pruning part located at lines 6–12.

An important part of the Algorithm 1 is the pruning condition located at the line 7. It is derived from (10) and it corresponds to an invertibility requirement for the related linear system (3). This

condition is crucial in the enumeration approach because the rank test is computationally simpler and can be executed faster than solving the feasibility problem in (10).

The output from the Algorithm 1 represents a set S of feasible bases for which there exist a critical region (4) and solution (5). For simplicity, the construction of the critical regions (4) and the associated solution maps (5) is not the part of Algorithm 1. The explicit solution but can be efficiently constructed, analyzed, and evaluated given the problem data M , q , Q , Θ , and the set of feasible bases S inside, for example, the Multi-Parametric Toolbox (Herceg et al., 2013).

4.3. Properties of the enumerative PLCP algorithm

The finite termination of Algorithm 1 follows from the finite number of combinations to explore. Given the dimension n , there exist 2^n possible combinations to explore which gives worst-case exponential complexity. Even if some of the possible combinations will be pruned, it still does not change the fact that the algorithm has a worst-case exponential complexity. It should be noted that the worst case complexity relates to the number of equality constraints n in (2b) and not the dimension of the parameters d . Therefore, the approach suits problems with high number of parameters but rather small number of constraints.

Moreover, Algorithm 1 guarantees discovery of all feasible bases because it evaluates all possible combinations and leaves no space for unexplored regions. However, in practice it is desirable to only keep critical regions which exhibit certain numerical robustness, such as full dimensionality.

Algorithm 1 could be potentially improved in efficiency when implementing the main pruning parts at the lines 7 and 8. It is recommended to perform rank tests at the line 7 with high precision and to employ an active set method to obtain reliable infeasibility certificates at the line 8 for (10). The feasibility test can be formulated in a way to determine if the constraints are strictly or weakly active, as suggested by Gupta et al. (2011). An active set method is also recommended in other enumerative approach of Feller et al. (2013).

The innermost part of the algorithm (at the line 9) that carries the partial feasible bases over iterations is critical for memory footprint. It could be therefore of interest from an implementation point of view to encode approaches with efficient basis storage.

5. Examples

We demonstrate the functionality of Algorithm 1 on two sets of examples.

5.1. Hybrid MPC

The first example of Necoara et al. (2004) considers a regulation problem of a PWA system that is comprised of four local models. The MPC problem is given by

$$\min_{u_0, \dots, u_{N-1}} \sum_{k=0}^{N-1} (\|Qx_k\|_2^2 + \|Ru_k\|_2^2) \quad (11a)$$

$$\text{s.t.: } x_{k+1} = \begin{cases} A_1x_k + Bu_k & \text{if } E_1x_k \geq 0 \\ A_2x_k + Bu_k & \text{if } E_2x_k \geq 0 \\ A_3x_k + Bu_k & \text{if } E_3x_k \geq 0 \\ A_4x_k + Bu_k & \text{if } E_4x_k \geq 0 \end{cases} \quad (11b)$$

$$x_k \in [-5, 5] \times [-5, 5], \quad (11c)$$

$$u_k \in [-2, 2], \quad (11d)$$

Table 1
Explicit solution to the regulation problem (11) with varying horizon N .

N	# binaries	dim. of PLCP	# LPs	# of regions
1	4	42	1335	20
2	8	80	67575	222
3	12	118	4893984	1136

where $Q = 10^{-4}I_2$, $R = 10^{-3}$ and the matrices of the system dynamics are

$$A_1 = \begin{pmatrix} 0.5 & 0.61 \\ 0.9 & 1.345 \end{pmatrix}, \quad A_2 = \begin{pmatrix} -0.92 & 0.644 \\ 0.758 & -0.71 \end{pmatrix},$$

$$A_3 = A_1, \quad A_4 = A_2,$$

$$E_1 = \begin{pmatrix} -1 & 1 \\ -1 & -1 \end{pmatrix}, \quad E_2 = \begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix},$$

$$E_3 = -E_1, \quad E_4 = -E_2.$$

The prediction horizon $N \in \{1, 2, 3, 4\}$ is varying in order to construct PMIQPs of different size. The explicit solution for each horizon has been computed using the enumerative PLCP scheme of Section 4.2 and the results are reported in Table 1. The combinatorial nature of the approach can be seen in the increasing number of LPs to be checked, as indicated by the data in Table 1.

5.2. Comparison with alternative approaches

In this section we show how Algorithm 1 compares to other methods for computing explicit solutions. We are specifically interested in those methods that employ enumeration techniques, such as the enumeration PQP approach of Gupta et al. (2011), and geometric techniques, i.e. Baotić (2002). By this comparison we want to show a trend how this approach scales with the increasing dimension of parameters for the two research directions in multiparametric solvers. We remark that both alternative techniques can only be applied to solve PLPs/PQPs that arise as subproblems of PMILPs/PMIQPs for a fixed value of binary variables, whereas Algorithm 1 is applicable to PLCP/PMILP/PMIQP directly. From this reason only convex PQPs are compared in the sequel.

The tests were performed by considering a PQP problem constructed from the typical MPC setup of the form

$$\min_{u_0, \dots, u_{N-1}} x_N^T P x_N + \sum_{k=0}^{N-1} x_k^T Q x_k + u_k^T R u_k \quad (12a)$$

$$\text{s.t.} : x_{k+1} = A x_k + B u_k \quad (12b)$$

$$x_k \in \mathcal{X}, u_k \in \mathcal{U}, \quad (12c)$$

with $x \in \mathbb{R}^{n_x}$, $u \in \mathbb{R}$, $P = Q = I_{n_x}$, $R = 1$, $\mathcal{X} = \{x \mid -10 \leq x_i \leq 10, i = 1, \dots, n_x\}$, and $\mathcal{U} = \{u \mid -1 \leq u \leq 1\}$. The prediction model in (12b) was obtained as a discrete-time version (with sampling time of 1 s) of

$$G(s) = \frac{1}{(s+1)^{n_x}}, \quad (13)$$

with $n_x \in \{4, 6, 8, 10\}$ and $N \in \{2, 3\}$. The rationale behind such a choice is to assess the performance of various parametric solvers for varying dimensions of the parametric space and of the decision space in (6). The runtime comparison was performed on an Intel Core i7 1.7 GHz CPU running Matlab R2013a. All algorithms were implemented using the Multi-Parametric Toolbox (Herceg et al., 2013).

The obtained results are summarized in Table 2. Two main conclusions can be drawn from the reported results. First, the

Table 2
Runtime in seconds of individual parametric solvers for a varying dimensionality of the PQP problem (6).

N	n_x	Runtime (Baotić, 2002)	Runtime (Gupta et al., 2011)	Runtime Algorithm 1
2	4	0.8	0.7	0.6
	6	2.1	1.7	1.0
	8	14.3 ^a	11.0	5.0
	10	24.6 ^a	18.3	8.0
3	4	1.1	1.2	1.1
	6	4.9 ^a	5.4	5.4
	8	64.2 ^a	81.4	40.1
	10	181.8 ^a	126.5	91.7

^a Denotes an incomplete solution being generated by the corresponding solver due to numerical problems.

parametric solver of Baotić (2002), which uses a geometric-based approach to exploration of active constraints, performs poorly in cases where the dimension of the parametric space is large. Specifically, once $n_x > 6$, the solver always returned an incomplete solution. This was due to the fact that high-dimensional geometric problems (such as computing the centers of lower-dimensional facets) could not be solved reliably even with state-of-the-art solvers (GUROBI in this case). Moreover, the discussed parametric solver was also the slowest one. The second main conclusion is that the presented enumeration-based PLCP algorithm is, on average, twice as fast as the enumeration method of Gupta et al. (2011) for $N = 2$ and about 1.5 times faster for $N = 3$. The reduced computational time can be attributed to a structure of the LP in (10) that contains reduced system of inequalities due to a transformation step from MPQP to PLCP. This also emphasizes the fact that PLCP formulation can be more efficient than tackling directly MPQP, specially if the transformation procedure returns problem defined over reduced number of constraints. Both Algorithm 1 as well as the method of Gupta et al. (2011) scale rather equally with increasing dimensionality of the parametric space.

6. Conclusions

In this paper an enumerative PLCP approach has been presented to tackle optimization problems arising in hybrid MPC. The advantage of the underlying PLCP formulation is that a single algorithm can be used to tackle four types of optimization problems: PLP/PQP/PMILP/PMIQP. The enumeration-based algorithm is general as it does not assume any specific properties of the structure of the problem. Despite the combinatorial worst-case behavior in the number of equality constraints of PLCP, the approach scales well with increasing number of parameters in practice, as was documented by the case studies. The enumerative algorithm has been implemented in the Multi-Parametric Toolbox and is thus available to the community. Any future extensions of the method could focus on efficient strategies for pruning and basis generation that involve large number of parameters.

References

Acevedo, J., & Pistikopoulos, E. N. (1997). A multiparametric programming approach for linear process engineering problems under uncertainty. *Industrial and Engineering Chemistry Research*, 36, 717–728.

Al-Khayyal, F. A. (1987). An implicit enumeration procedure for the general linear complementarity problem. In K. L. Hoffman, R. H. F. Jackson, & J. Telgen (Eds.), *Mathematical programming studies: Vol. 31. Computation mathematical programming* (pp. 1–20). Berlin, Heidelberg: Springer.

Audet, C., Hansen, P., Jaumard, B., & Savard, G. (1997). Links between linear bilevel and mixed 0–1 programming problems. *Journal of Optimization Theory and Applications*, 93(2), 273–300.

Baotić, M. (2002). *An efficient algorithm for multi-parametric quadratic programming*. Technical report AUT02-05. ETH Zurich, Switzerland: Automatic Control Laboratory.

- Bemporad, A., & Morari, M. (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3), 407–427.
- Borrelli, F. (2003). *Lect. notes in control and inf. sciences: Vol. 290. Constrained optimal control of linear and hybrid systems*. Springer.
- Borrelli, F., Bemporad, A., & Morari, M. (2014). Predictive control for linear and hybrid systems. Preprint. <http://www.mpc.berkeley.edu/mpc-course-material>.
- Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.
- Columbano, S., Fukuda, K., & Jones, C. N. (2009). An output-sensitive algorithm for multi-parametric LCPs with sufficient matrices. In *CRM Proceedings and Lecture Notes: Vol. 48. Polyhedral Computation* (p. 73).
- De Schutter, B., & De Moor, B. (1995). The extended linear complementarity problem. *Mathematical Programming*, 71(3), 289–325.
- De Schutter, B., Heemels, W. P. M. H., & Bemporad, A. (2002). On the equivalence of linear complementarity problems. *Operations Research Letters*, 30(4), 211–222.
- Dua, V., Bozinis, N. A., & Pistikopoulos, E. N. (2002). A multiparametric programming approach for mixed-integer quadratic engineering problems. *Computers & Chemical Engineering*, 26, 715–733.
- Dua, V., & Pistikopoulos, E. N. (2000). An algorithm for the solution of multiparametric mixed integer linear programming problems. *Annals of Operations Research*, 99, 123–139.
- Feller, C., & Johansen, T. A. (2013). Explicit MPC of higher-order linear processes via combinatorial multi-parametric quadratic programming. In *Proc. of the eur. control conf.* July (pp. 536–541).
- Feller, C., Johansen, T. A., & Oлару, S. (2013). An improved algorithm for combinatorial multi-parametric quadratic programming. *Automatica*, 49(5), 1370–1376.
- Gupta, A., Bhartiya, S., & Nataraj, P. S. V. (2011). A novel approach to multiparametric quadratic programming. *Automatica*, 47(9), 2112–2117.
- Herceg, M., Kvasnica, M., Jones, C. N., & Morari, M. (2013). Multi-parametric toolbox 3.0. In *Proc. of the eur. control conf.*, Zürich, Switzerland, July 17–19 (pp. 502–510) <http://control.ee.ethz.ch/~mpt>.
- Jones, C. N., & Morari, M. (2006). Multiparametric linear complementarity problems. In *IEEE conf. on decision and control*. December.
- Júdice, J. J. (2012). Algorithms for linear programming with linear complementarity constraints. *TOP*, 20(1), 4–25.
- Judice, J. J., & Mitra, G. (1988). Reformulation of mathematical programming problems as linear complementarity problems and investigation of their solution methods. *Journal of Optimization Theory and Applications*, 57(1), 123–149.
- Li, Z., & Ierapetritou, M. G. (2010). A method for solving the general parametric linear complementarity problem. *Annals of Operations Research*, 181(1), 485–501.
- Moort, B., Vandenberghe, L., & Vandewalle, J. (1992). The generalized linear complementarity problem and an algorithm to find all its solutions. *Mathematical Programming*, 57(1), 415–426.
- Murty, K. G. (1997). *Linear complementarity, linear and nonlinear programming*. Ann Arbor, Michigan, USA: University of Michigan, http://ioe.engin.umich.edu/people/fac/books/murty/linear_complementarity_webbook/.
- Necoara, I., De Schutter, B., Heemels, W. P. M. H., Weiland, S., Lazar, M., & van den Boom, T. J. J. (2004). *Control of PWA systems using a stable receding horizon method: Extended report. Technical Report 04-019a*. Netherlands: Delft University of Technology.
- Pardalos, P. M. (1994). The linear complementarity problem. In Susana Gomez, & Jean-Pierre Hennart (Eds.), *Mathematics and its applications: Vol. 275. Advances in optimization and numerical analysis* (pp. 39–49). Netherlands: Springer.
- Pardalos, P. M., & Rosen, J. B. (1988). Global optimization approach to the linear complementarity problem. *SIAM Journal on Scientific and Statistical Computing*, 9(2), 341–354.
- Pistikopoulos, E. N., Dominguez, L., Panos, C., Kouramas, K., & Chinchuluun, A. (2012). Theoretical and algorithmic advances in multi-parametric programming and control. *Computer and Management Sciences*, 9(2), 183–203.
- Sherali, H. D., Krishnamurthy, R. S., & Al-Khayyal, F. A. (1998). Enumeration approach for linear complementarity problems based on a reformulation-linearization technique. *Journal of Optimization Theory and Applications*, 99(2), 481–507.



Martin Herceg is currently a Senior Research Engineer at Advanced Technology Laboratory in Honeywell Prague, Czech Republic, with focus on applications of model predictive control in automotive industry. Prior to that he was a Postdoctoral Researcher at the Automatic Control Lab at ETH Zurich in Switzerland from 2010 to 2015. He obtained Ph.D. degree in real-time explicit model predictive control and a master degree in chemical engineering from the Slovak University of Technology in Bratislava, Slovakia. Martin's experiences are in research and development of multi-parametric methods for model predictive control and he is one of the main developers of Multi-Parametric Toolbox 3.0. His interests cover optimization methods for fast implementation of advanced control, multi-parametric programming, and numerical algorithms.



Colin N. Jones is an Assistant Professor in the Automatic Control Laboratory at the Ecole Polytechnique Federale de Lausanne (EPFL) in Switzerland. He was a Senior Researcher at the Automatic Control Lab at ETH Zurich until 2011 and obtained a Ph.D. 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 commercial building control to the development of custom optimization tools focusing on retail human resource scheduling. His current research interests are in the theory and computation of predictive control and optimization, and their application to green energy generation, distribution and management.



Michal Kvasnica was born in 1977. He received his diploma in chemical engineering from the Slovak University of Technology in Bratislava, Slovakia and the Ph.D. in electrical engineering from the Swiss Federal Institute of Technology in Zurich, Switzerland. Since 2011 he is an Associate Professor at the Slovak University of Technology in Bratislava. His research interests are in model predictive control, modeling of hybrid systems, and development of software tools for control. He is the co-author and developer of the MPT Toolbox for explicit model predictive control.



Manfred Morari was head of the Department of Information Technology and Electrical Engineering at ETH Zurich from 2009 to 2012 and 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. From 1977 to 1983 he was on the faculty of the University of Wisconsin. 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. Morari's research is internationally recognized. The analysis techniques and software developed in his group are used in universities and industry throughout the world. He has received numerous awards, including the Eckman Award, Ragazzini Award and Bellman Control Heritage Award from the American Automatic Control Council; the Colburn Award, Professional Progress Award and CAST Division Award from the American Institute of Chemical Engineers; the Control Systems Technical Field Award and the Bode Lecture Prize from IEEE. He is a Fellow of IEEE, AIChE and IFAC. In 1993 he was elected to the US National Academy of Engineering, in 2015 to the UK Royal Academy of Engineering.

Manfred Morari served on the technical advisory boards of several major corporations.



Brief paper

Convex liftings-based robust control design[☆]Ngoc Anh Nguyen^{a,b,1}, Sorin Olaru^b, Pedro Rodríguez-Ayerbe^b, Michal Kvasnica^c^a Institute for Design and Control of Mechatronical Systems, Johannes Kepler University Linz, Austria^b Laboratory of Signals and Systems, CentraleSupélec-CNRS-UPS, Université Paris Saclay, Gif-sur-Yvette, France^c Department of Information Engineering and Process Control, Slovak University of Technology in Bratislava, Slovakia

ARTICLE INFO

Article history:

Received 17 November 2015

Received in revised form

17 August 2016

Accepted 24 October 2016

Keywords:

Convex liftings

Robust control

Bounded additive disturbances

Polytopic uncertainties

PWA Lyapunov function

ABSTRACT

This paper presents a new approach for control design of constrained linear systems affected by bounded additive disturbances and polytopic uncertainties. This method hinges on so-called *convex liftings* which emulate control Lyapunov function by providing a constructive framework for optimization based control implementation. It will be shown that this method can guarantee the recursive feasibility and robust stability. Finally, a numerical example will be presented to illustrate this method.

© 2016 Elsevier Ltd. All rights reserved.

1. Introduction

Originated in the seminal work (Lyapunov, 1907), Lyapunov stability stands as a fundamental concept in control theory (Loría & Panteley, 2006). In stability analysis, a Lyapunov function is usually of use to prove closed-loop stability, see Kalman and Bertram (1960), Brayton and Tong (1979) and Molchanov and Pyatnitskiy (1989). On the other hand, in control design, control Lyapunov functions are usually employed to design stabilizing/robust controllers, see among others (Khalil, 2002; Zubov & Boron, 1964). Accordingly, whenever such control Lyapunov functions are used in optimization based strategies, these should be chosen such that the recursive feasibility and closed-loop stability are all fulfilled. Different classes of control Lyapunov functions have been proposed in control theory (Michel, Nam, & Vittal, 1984; Polanski, 1995). In the context of linear quadratic control, infinite/finite quadratic cost functions usually serve as control Lyapunov functions, as shown in Anderson and Moore (2007), Chmielewski

and Manousiouthakis (1996) and Sznaier and Damborg (1987). In particular, in linear model predictive control (MPC), such a control Lyapunov function has been used to design robust controllers to cope with polytopic uncertainties, leading to a linear matrix inequality problem, see Kothare, Balakrishnan, and Morari (1996). Polyhedral control Lyapunov functions have also been exploited in several studies, e.g., Bitsoris (1988b), Bitsoris and Vassilaki (1995), Blanchini (1994, 1995), Gutman and Cwikel (1987), Lazar (2010) and Vassilaki, Hennes, and Bitsoris (1988), since they lead to simple design procedures, i.e., composed of linear constraints. Convex piecewise affine control Lyapunov function for piecewise affine systems has also been considered in Baotic, Christophersen, and Morari (2006) and solved using dynamic programming, which may be impractical if disturbances and uncertainties are considered.

It is worth emphasizing that the robust control design proposed in Kothare et al. (1996) requires at each sampling time solving a linear matrix inequality (LMI) problem, the online evaluation thus becomes computationally demanding. Some improvements of this method are presented in Cuzzola, Geromel, and Morari (2002) and Wan and Kothare (2003). An effort to simplify this complexity has been proposed in Kouvaritakis, Rossiter, and Schuurmans (2000). However, this method can only guarantee the positive invariance of the initially ellipsoidal feasible set instead of asymptotic stability of the origin. Also, although the number of LMIs is decreased, however, solving online an LMI problem is still expensive in comparison to strict real-time requirements. Some extensions of the latter method have been proposed to reduce complexity, e.g., Khan and Rossiter (2012).

[☆] The material in this paper was partially presented at the 8th IFAC Symposium on Robust Control Design, July 8–11, 2015, Bratislava, Slovakia. This paper was recommended for publication in revised form by Associate Editor Akira Kojima under the direction of Editor Ian R. Petersen.

E-mail addresses: Ngocanh.Nguyen.rs@gmail.com (N.A. Nguyen), Sorin.Olaru@centralesupelec.fr (S. Olaru), Pedro.Rodriguez@centralesupelec.fr (P. Rodríguez-Ayerbe), michal.kvasnica@stuba.sk (M. Kvasnica).

¹ Fax: +43 732 2468 6213.

Note however that making use of *degree of freedom* n_c is nothing other than solving a finite horizon MPC problem. Also, in the context of MPC, the optimal cost function usually serves as a Lyapunov function, therefore minimizing a nominal cost function as in this reference is meaningless, and robust stability is thus guaranteed by the constraint set. Further, the pre-imposition on the structure of controllers leads to conservativeness and possible loss of recursive feasibility. An alternative robust MPC scheme has been presented in [Mayne, Seron, and Raković \(2005\)](#) to take bounded additive disturbances into account. However, polytopic uncertainties considerably increase its computational complexity with respect to the prediction horizon. As an extension of this method, parameterized tube MPC has recently been proposed in [Rakovic, Kouvaritakis, Cannon, Panos, and Findeisen \(2012\)](#) to cope with bounded additive disturbances. Although implicit controller is computed based on its decomposed elements, the number of decision variables is of order $\mathcal{O}(q^N)$, with q to be the number of vertices of the given disturbance set and N to be the prediction horizon. As a consequence, accounting for polytopic uncertainty makes the online computation much more demanding, as the number of decision is of order $\mathcal{O}(q^N p^N)$, with p to be the number of vertices of the given polytopic uncertainty set. Further, dealing with tube cost function in this case becomes more complicated.

This paper proposes a method which only requires resolution of a linear programming problem at each sampling instant. Moreover, unlike the method in [Blanchini \(1994\)](#), which guarantees robust stability in the sense of Lyapunov (input-to-state stability), this paper proves a more flexible result by guaranteeing that the state converges to a given robust positively invariant set (minimal/maximal robust positively invariant set) as time tends to infinity. Note that such a constructed convex lifting is not a control Lyapunov function, which represents a relaxation and a supplementary degree of freedom with respect to the method in [Blanchini \(1994\)](#). Finally, to our best knowledge, convex liftings have never been used in control design and can be a valuable tool, offering additional flexibility for the existing constrained control methods.

2. Notation and definitions

Throughout this paper, $\mathbb{N}, \mathbb{N}_{>0}, \mathbb{R}, \mathbb{R}_+$ denote the set of nonnegative integers, the set of positive integers, the set of real numbers and the set of nonnegative numbers, respectively. For ease of presentation, with a given $N \in \mathbb{N}_{>0}$, by \mathcal{I}_N , we denote the index set: $\mathcal{I}_N := \{i \in \mathbb{N}_{>0} : i \leq N\}$. Also, we use \mathcal{I}_N^2 to denote the set defined as: $\mathcal{I}_N^2 = \mathcal{I}_N \times \mathcal{I}_N$.

A polyhedron is the intersection of finitely many closed halfspaces. A polytope is a bounded polyhedron. If P is an arbitrary polytope, then by $\mathcal{V}(P)$, we denote the set of its vertices. If \mathcal{S} is an arbitrary set, then $\text{conv}(\mathcal{S})$ denotes the convex hull of \mathcal{S} . Also, we use $\dim(\mathcal{S})$ to denote the dimension of its affine hull. Moreover, if \mathcal{S} is a full-dimensional set, then we use $\text{int}(\mathcal{S})$ to denote the interior of \mathcal{S} . Given a set $\mathcal{S} \subset \mathbb{R}^d$ and a matrix $A \in \mathbb{R}^{d \times d}$, then $A\mathcal{S}$ is defined as follows: $A\mathcal{S} := \{As : s \in \mathcal{S}\}$. Also, for any vector $x \in \mathbb{R}^d$, $\rho_{\mathcal{S}}(x)$ is defined as follows: $\rho_{\mathcal{S}}(x) := \min_{y \in \mathcal{S}} \sqrt{(y-x)^T(y-x)}$. Given two sets $\mathcal{S}_1, \mathcal{S}_2 \subset \mathbb{R}^d$, their Minkowski sum is denoted by $\mathcal{S}_1 \oplus \mathcal{S}_2$ and is defined by: $\mathcal{S}_1 \oplus \mathcal{S}_2 := \{y_1 + y_2 \in \mathbb{R}^d : y_1 \in \mathcal{S}_1, y_2 \in \mathcal{S}_2\}$. Also, $\mathcal{S}_1 \setminus \mathcal{S}_2$ is defined as follows: $\mathcal{S}_1 \setminus \mathcal{S}_2 := \{x \in \mathbb{R}^d : x \in \mathcal{S}_1, x \notin \mathcal{S}_2\}$.

3. Problem settings

In this paper, we consider a discrete-time linear system:

$$x_{k+1} = A(k)x_k + B(k)u_k + w_k, \quad (1)$$

where x_k, u_k, w_k denote the state, control variable and additive disturbance at time k . The state-space matrices $[A(k) B(k)]$ are

time-varying and assumed to belong to an *uncertainty matrix polytope* denoted by Ψ and defined below:

$$[A(k) B(k)] \in \Psi := \text{conv} \{[A_1 B_1], \dots, [A_L B_L]\}. \quad (2)$$

The state, control variables and disturbances are subject to constraints:

$$x_k \in \mathbb{X} \subset \mathbb{R}^{d_x}, \quad u_k \in \mathbb{U} \subset \mathbb{R}^{d_u}, \quad w_k \in \mathbb{W} \subset \mathbb{R}^{d_w}, \quad (3)$$

where $d_x, d_u \in \mathbb{N}_{>0}$, and $\mathbb{X}, \mathbb{U}, \mathbb{W}$ are polytopes containing the origin in their interior.

The objective is to find robust control laws which can cope with bounded additive disturbances and polytopic model uncertainties such that the closed loop is robustly stable. It is clear that if w_k is unknown, one cannot expect to guarantee asymptotic stability of the origin. In this case, asymptotic stability is replaced with an ultimate boundedness concept ([Khalil, 2002; Kofman, Haimovich, & Seron, 2007](#)) or input to state stability ([Jiang & Wang, 2001](#)).

4. Robust control design based on convex liftings

4.1. Robust positively invariant sets

Positively invariant sets have been studied over several decades. Due to their relevance in control theory, they turn out to be useful in many control related studies, e.g., [Bitsoris \(1988a,b\)](#), [Bitsoris and Vassilaki \(1995\)](#), [Blanchini and Miani \(2007\)](#) and [Kerrigan \(2001\)](#). The definition of a robust positively invariant set for system (1) is recalled below.

Definition 4.1. Given an admissible control law $u_k = Kx_k \in \mathbb{U}$, a set $\Omega \subseteq \mathbb{X}$ is called *robust positively invariant* with respect to (1) if

$$(A(k) + B(k)K)\Omega \oplus \mathbb{W} \subseteq \Omega, \quad \forall [A(k) B(k)] \in \Psi,$$

where Ψ is defined in (2).

To compute such a robust positively invariant set Ω , it is important to choose an appropriate unconstrained control law to cope with given bounded additive disturbances and polytopic uncertainties. More clearly, this control law should satisfy that there exists a Lyapunov function $V(x) : \mathbb{R}^{d_x} \rightarrow \mathbb{R}_+$ such that

$$V((A(k) + B(k)K)x_k) - V(x_k) < 0, \quad \forall [A(k) B(k)] \in \Psi.$$

The computation of such a gain K was studied in, e.g., [Daafouz and Bernussou \(2001\)](#) and [Kothare et al. \(1996\)](#). A simpler formulation is presented below:

$$\begin{aligned} & \min_{Z, Y} -\log \det(Z) \\ & \text{subject to} \\ & Z = Z^T > 0 \\ & \begin{bmatrix} Z & (A_i Z + B_i Y)^T \\ A_i Z + B_i Y & Z \end{bmatrix} > 0, \quad \forall i \in \mathcal{I}_L. \end{aligned}$$

Then, gain K is determined by $K = YZ^{-1}$. It is already known that the above formulation is an LMI problem and is solvable by using semidefinite programming. The interested reader can find details in [Boyd, El Ghaoui, Feron, and Balakrishnan \(1994\)](#).

With respect to the state feedback $u_k = Kx_k$, the computation of a robust positively invariant set Ω for system (1) has been put forward in [Nguyen \(2014\)](#), as a simple extension of the idea presented in [Gilbert and Tan \(1991\)](#). Note also that prominent studies on the computation of the maximal and minimal positively invariant sets for a linear, discrete-time invariant system affected by bounded additive disturbances can be found in [Kolmanovsky and Gilbert \(1998\)](#) and [Rakovic, Kerrigan, Kouramas, and Mayne \(2005\)](#). Still, in the case system (1) is not affected by additive disturbances, then the minimal robust positively invariant set coincides with the origin due to its asymptotic stability, i.e., $\Omega = \{0\}$. Without loss of generality, we are hereafter interested in the case $\Omega \subseteq \mathbb{X} \subset \mathbb{R}^{d_x}$ represents a full-dimensional set.

4.2. Domain of attraction

Given a robust positively invariant set Ω associated with an admissible state feedback $u = Kx \in \mathbb{U}$ for all $x \in \Omega$, the domain of attraction is defined as the set of all points in \mathbb{X} which can be driven to Ω , see Khalil (2002). More precisely, the domain of attraction contains all points $x_0 \in \mathbb{X}$ such that there always exists control law satisfying constraints (3) which is able to steer the state to Ω as $k \rightarrow \infty$, i.e., $\lim_{k \rightarrow \infty} \rho_\Omega(x_k) = 0$. Computing exactly the domain of attraction is difficult. Instead, approximation of the domain of attraction is usually of use. For simplicity, in this paper, we restrict our attention to a contractive set. The definition of a contractive set for system (1) is recalled in the sequel.

Definition 4.2. Consider system (1) subject to model uncertainty (2) and constraints (3). A set $\mathcal{X} \subseteq \mathbb{X}$ is called λ -contractive for a given $0 \leq \lambda < 1$ if there exists a control law $u_k = \kappa(x_k) \in \mathbb{U}$ such that

$$(A(k)x_k + B(k)\kappa(x_k)) \oplus \mathbb{W} \subseteq \lambda\mathcal{X}, \\ \forall x_k \in \mathcal{X}, \forall [A(k) \ B(k)] \in \Psi.$$

The maximal λ -contractive set, denoted as P_λ , is defined as the set containing all the λ -contractive sets in \mathbb{X} . An algorithm for the computation of the maximal λ -contractive set is put forward in Blanchini (1994). For completeness, this algorithm is recalled in Appendix A. Hereafter, we will use the maximal λ -contractive set as an estimation of the domain of attraction for a given $0 \leq \lambda < 1$, i.e., $\mathcal{X} = P_\lambda \subseteq \mathbb{X}$. Without loss of generality, we assume that $\Omega \subset P_\lambda$.

4.3. Convex liftings construction

In control theory, convex liftings have been used to facilitate implementation of piecewise affine control laws (Gulan, Nguyen, Oлару, Rodríguez-Ayerbe, & Rohal'-Ilkiv, 2015; Nguyen, 2015; Nguyen, Gulan, Oлару, & Rodríguez-Ayerbe, 2016). Recently, they have been exploited to solve inverse parametric linear/quadratic programming problem (Nguyen, Oлару, & Rodríguez-Ayerbe, 2015a,b; Nguyen, Oлару, Rodríguez-Ayerbe, Hovd, & Necoara, 2014, 2016). In this paper, we will show that such convex liftings are also useful in robust control design, based on preliminary results in Nguyen, Oлару, and Rodríguez-Ayerbe (2015c). Before recalling the definition of a convex lifting, additional definitions need to be recalled.

Definition 4.3. A collection of N full-dimensional polyhedra $\mathcal{X}_i \subset \mathbb{R}^d$, denoted by $\{\mathcal{X}_i\}_{i \in \mathcal{I}_N}$, is called a *polyhedral partition of a polyhedron* $\mathcal{X} \subseteq \mathbb{R}^d$ if the following conditions hold:

- $\bigcup_{i \in \mathcal{I}_N} \mathcal{X}_i = \mathcal{X}$,
- $\text{int}(\mathcal{X}_i) \cap \text{int}(\mathcal{X}_j) = \emptyset, \forall (i, j) \in \mathcal{I}_N^2, i \neq j$.

Two regions $\mathcal{X}_i, \mathcal{X}_j$ are called *neighboring* or *adjacent* if $i \neq j, (i, j) \in \mathcal{I}_N^2, \dim(\mathcal{X}_i \cap \mathcal{X}_j) = d - 1$. Further, if \mathcal{X} is a polytope, then $\{\mathcal{X}_i\}_{i \in \mathcal{I}_N}$ is called a *polytopical partition*.

Definition 4.4. Given a polyhedral partition $\{\mathcal{X}_i\}_{i \in \mathcal{I}_N}$ of a polyhedron $\mathcal{X} \subseteq \mathbb{R}^d$, a *piecewise affine lifting* is described by a function $z : \mathcal{X} \rightarrow \mathbb{R}$ with:

$$z(x) := a_i^T x + b_i \quad \text{for any } x \in \mathcal{X}_i, \quad (4)$$

and $a_i \in \mathbb{R}^d, b_i \in \mathbb{R}, \forall i \in \mathcal{I}_N$.

Definition 4.5. Given a polyhedral partition $\{\mathcal{X}_i\}_{i \in \mathcal{I}_N}$ of a polyhedron $\mathcal{X} \subseteq \mathbb{R}^d$, a piecewise affine lifting $z(x) = a_i^T x + b_i$ for $x \in \mathcal{X}_i$, is called a *convex piecewise affine lifting* if the following conditions hold true:

- $z(x)$ is continuous over \mathcal{X} ,
- for each $i \in \mathcal{I}_N, z(x) > a_i^T x + b_i$ for all $x \in \mathcal{X}_i \setminus \mathcal{X}_j$ and all $j \neq i, j \in \mathcal{I}_N$.

Note that the second condition in Definition 4.5 implies that any pair of neighboring regions are lifted onto two distinct hyperplanes. Also, it implies the convexity of this piecewise affine lifting. For ease of presentation, a slight abuse of notation is hereafter used: a *convex lifting* is understood as a convex piecewise affine lifting.

We now present an algorithm to construct a class of convex liftings which will be of use later in the proposed robust control design. Let $\ell(x)$ denote this convex lifting defined over an estimation of the domain of attraction \mathcal{X} . As discussed in Section 4.2, we restrict our attention to the maximal λ -contractive set P_λ for a given $0 \leq \lambda < 1$, i.e., $\mathcal{X} = P_\lambda$.

Algorithm 1 Construct a suitable convex lifting

Input: A given robust positively invariant set $\Omega \subset \mathbb{R}^{d_x}$, an estimation of the domain of attraction $\mathcal{X} = P_\lambda \subset \mathbb{R}^{d_x}$ with a given $0 \leq \lambda < 1$ and a scalar constant $c > 0$.

Output: A convex lifting $\ell(x)$ such that $\ell(x) = 0$ for all $x \in \Omega$.

- 1: $V_1 := \mathcal{V}(\Omega), \widehat{V}_1 := \left\{ \begin{bmatrix} x^T & 0 \end{bmatrix}^T : x \in V_1 \right\} \subset \mathbb{R}^{d_x+1}$.
- 2: $V_2 := \mathcal{V}(\mathcal{X}), \widehat{V}_2 := \left\{ \begin{bmatrix} x^T & c \end{bmatrix}^T : x \in V_2 \right\} \subset \mathbb{R}^{d_x+1}$.
- 3: $\Pi := \text{conv}(\widehat{V}_1 \cup \widehat{V}_2)$.
- 4: Solve the parametric linear programming problem:

$$\ell(x) := \min_z z \quad \text{s.t.} \quad \begin{bmatrix} x^T & z \end{bmatrix}^T \in \Pi. \quad (5)$$

Steps 1-2 in Algorithm 1 aim to lift the vertices of Ω and \mathcal{X} to \mathbb{R}^{d_x+1} with appropriate heights. Namely, the vertices of Ω are lifted with heights equal to 0, whereas the vertices of \mathcal{X} are lifted with heights equal to the given constant $c > 0$. Note that (5) is a parametric linear programming problem, its optimal solution is thus a piecewise affine function defined over a polytopical partition denoted as follows: $\ell(x) = a_i^T x + b_i$ for $x \in \mathcal{X}_i$. Note also that by construction, there exists a region in the partition associated with $\ell(x)$ which coincides with Ω , since the vertices of Ω are lifted onto a lower facet of Π . The following observation describes the properties of such an $\ell(x)$, generated from Algorithm 1.

Lemma 4.6. The function $\ell(x)$ over \mathcal{X} , generated from Algorithm 1, is continuous, convex, piecewise affine function.

Lemma 4.7. The function $\ell(x)$ over \mathcal{X} , generated from Algorithm 1, is a convex lifting over the associated partition $\{\mathcal{X}_i\}_{i \in \mathcal{I}_N}$.

Lemma 4.8. The function $\ell(x)$ over \mathcal{X} , generated from Algorithm 1, satisfies $\ell(x) = 0$ for any $x \in \Omega$ and $\ell(x) > 0$ for all $x \in \mathcal{X} \setminus \Omega$.

Lemma 4.9. For any $x \in \mathcal{X}$ and $0 \leq \beta \leq 1, \ell(\beta x) \leq \beta \ell(x)$.

For reading ease, the proof of the above lemmas is provided in the Appendix.

Algorithm 2 Robust control design procedure based on convex liftings

Input: A robust positively invariant set Ω associated with a stabilizing control law $u = Kx$ over Ω and a convex lifting $\ell(x) = a_i^T x + b_i$ for $x \in \mathcal{X}_i$, $i \in \mathcal{I}_N$ as in Algorithm 1.

Output: Control law $u^*(x_k)$ at each sampling time.

- 1: Compute $\ell(x_k)$.
- 2: **If** $x_k \in \Omega$ then $u^*(x_k) := Kx_k$, jump to Step 6.
- 3: **Else** Solve the following linear programming problem:

$$\begin{aligned} & [\alpha^* (u_k^*)^T]^T := \arg \min_{\alpha, u_k} \alpha \\ \text{s.t. } & a_i^T (A_j x_k + B_j u_k + w) + b_i \leq \alpha \ell(x_k) \\ & \alpha \geq 0, u_k \in \mathbb{U}, \forall i \in \mathcal{I}_N, \forall w \in \mathcal{V}(\mathbb{W}), \\ & \forall [A_j B_j] \in \mathcal{V}(\Psi). \end{aligned} \quad (6)$$

- 4: Apply $u^*(x_k) := u_k^*$
- 5: **End**
- 6: $k \leftarrow k + 1$, return to Step 1.

4.4. Robust control design procedure

This subsection introduces the procedure for designing robust control laws based on convex liftings. This procedure can guarantee robust stability of the closed loop by showing that $\lim_{k \rightarrow \infty} \rho_\Omega(x_k) = 0$. Our design procedure based on a convex lifting, computed from Algorithm 1, is summarized in Algorithm 2.

Natural questions arise here whether or not the linear programming problem (6) is feasible and whether closed-loop stability is guaranteed by the proposed procedure. These questions are answered via the following theorem. Accordingly, it will be shown that convex lifting constructed in Algorithm 1 is strictly decreasing to 0 along the state evolution outside Ω .

Theorem 4.10. *Given a robust positively invariant set Ω associated with a robust control law gain K and an estimation of the domain of attraction $\mathcal{X} = P_\lambda$ for a given $0 \leq \lambda < 1$, if the initial condition $x_k \in \mathcal{X}$, then the linear programming problem (6) is recursively feasible. Furthermore, the closed loop is robustly stable.*

Proof. For the feasibility of (6), one can easily see that $0 \leq \ell(x) \leq c$ by the construction in Algorithm 1. Therefore, due to the contractivity of \mathcal{X} , for any $x_k \in \mathcal{X}$ there always exists $u(x_k) \in \mathbb{U}$ such that:

$$A(k)x_k + B(k)u(x_k) + w_k \in \lambda\mathcal{X} \subset \mathcal{X}$$

for all $w_k \in \mathbb{W}$ and for all $[A(k) B(k)] \in \Psi$. Therefore, if $u^*(x_k)$ denotes an optimal solution to (6), then one has:

$$\begin{aligned} 0 & \leq \ell(A(k)x_k + B(k)u^*(x_k) + w_k) \\ & \leq \ell(A(k)x_k + B(k)u(x_k) + w_k) \\ & \leq c, \quad \forall w_k \in \mathbb{W}, \quad \forall [A(k) B(k)] \in \Psi. \end{aligned}$$

Due to this boundedness, the recursive feasibility of the linear programming problem (6) is ensured for a finite, large enough scalar α at each sampling time.

As for robust stability, we prove that for any $x_k \in \mathcal{X} \setminus \Omega$:

$$\begin{aligned} \ell(A(k)x_k + B(k)u^*(x_k) + w_k) & < \ell(x_k), \\ \forall w_k \in \mathbb{W}, \forall [A(k) B(k)] \in \Psi. \end{aligned}$$

Indeed, due to the contractivity of \mathcal{X} , for any $v \in \mathcal{V}(\mathcal{X}) \setminus \Omega$, there exists a control law, denoted by $u(v) \in \mathbb{U}$ such that $A(k)v + B(k)u(v) + w_k \in \lambda\mathcal{X}$ despite any disturbances $w_k \in \mathbb{W}$ and for all $[A(k) B(k)] \in \Psi$. For each $w_k \in \mathbb{W}$ and each $[A(k) B(k)] \in \Psi$, there

exists $y(k, w_k) \in \mathcal{X}$ such that $A(k)v + B(k)u(v) + w_k = \lambda y(k, w_k)$. Due to Lemma 4.9, this inclusion leads to

$$\begin{aligned} \ell(A(k)v + B(k)u(v) + w_k) & = \ell(\lambda y(k, w_k)) \\ & \leq \lambda \ell(y(k, w_k)). \end{aligned} \quad (7)$$

By the construction of $\ell(x)$ in Algorithm 1, we obtain:

$$\ell(y(k, w_k)) \leq c. \quad (8)$$

Also, according to Algorithm 1 and $v \in \mathcal{V}(\mathcal{X}) \setminus \Omega$,

$$\ell(v) = c. \quad (9)$$

From (7), (8), (9), one can deduce that

$$\ell(A(k)v + B(k)u(v) + w_k) \leq \lambda \ell(v). \quad (10)$$

Recall that (10) holds for all $w_k \in \mathbb{W}$ and for all $[A(k) B(k)] \in \Psi$.

Now, consider a point $x_k \in \mathcal{X}_i$ in the polytopic partition $\{\mathcal{X}_i\}_{i \in \mathcal{I}_N}$ of \mathcal{X} over which $\ell(x)$ is defined. Without loss of generality, suppose $\mathcal{X}_i \neq \Omega$, then x_k can be described via a convex combination of the vertices of \mathcal{X}_i , i.e.,

$$x_k = \sum_{v \in \mathcal{V}(\mathcal{X}_i)} \alpha(v)v, \quad \text{where } \alpha(v) \geq 0, \quad \sum_{v \in \mathcal{V}(\mathcal{X}_i)} \alpha(v) = 1.$$

Recall that due to the definition of convex lifting, $\ell(x)$ over \mathcal{X}_i is an affine function, then $\ell(x_k)$ can be written in the following form:

$$\ell(x_k) = \sum_{v \in \mathcal{V}(\mathcal{X}_i)} \alpha(v)\ell(v). \quad (11)$$

If $v \in \mathcal{V}(\mathcal{X}_i)$ belongs to Ω , then due to the robust positive invariance of Ω with respect to a linear feedback $u(x) = Kx$, it satisfies

$$\begin{aligned} \ell(v) = 0 & = \ell((A(k) + B(k)K)v + w_k), \\ \forall w_k \in \mathbb{W}, \forall [A(k) B(k)] \in \Psi. \end{aligned} \quad (12)$$

Otherwise, if $v \in \mathcal{V}(\mathcal{X}_i)$ is a vertex of \mathcal{X} and $v \notin \Omega$, then it satisfies (10). Therefore, due to the convexity of $\ell(x)$ proved in Lemma 4.6 and (10), (11), (12), the following is obtained:

$$\begin{aligned} \lambda \ell(x_k) & = \sum_{v \in \mathcal{V}(\mathcal{X}_i)} \alpha(v)(\lambda \ell(v)) \\ & \geq \sum_{v \in \mathcal{V}(\mathcal{X}_i)} \alpha(v)\ell(A(k)v + B(k)u(v) + w_k) \\ & \geq \ell(A(k) \sum_{v \in \mathcal{V}(\mathcal{X}_i)} \alpha(v)v + B(k) \sum_{v \in \mathcal{V}(\mathcal{X}_i)} \alpha(v)u(v) + w_k) \\ & = \ell(A(k)x_k + B(k) \sum_{v \in \mathcal{V}(\mathcal{X}_i)} \alpha(v)u(v) + w_k). \end{aligned} \quad (13)$$

Recall that $u(v) \in \mathbb{U}$, $\forall v \in \mathcal{V}(\mathcal{X}_i) \setminus \Omega$ and $u(v) = Kv \in \mathbb{U}$, $\forall v \in \mathcal{V}(\mathcal{X}_i) \cap \Omega$, then it follows that

$$\sum_{v \in \mathcal{V}(\mathcal{X}_i)} \alpha(v)u(v) \in \mathbb{U}. \quad (14)$$

Therefore, (14) leads to:

$$\begin{aligned} \ell(A(k)x_k + B(k) \sum_{v \in \mathcal{V}(\mathcal{X}_i)} \alpha(v)u(v) + w_k) \\ \geq \ell(A(k)x_k + B(k)u^*(x_k) + w_k). \end{aligned} \quad (15)$$

From (13) and (15), the following inclusion is obtained:

$$\begin{aligned} \lambda \ell(x_k) & \geq \ell(A(k)x_k + B(k)u^*(x_k) + w_k), \\ \forall w_k \in \mathbb{W}, \forall [A(k) B(k)] \in \Psi. \end{aligned} \quad (16)$$

Recall that $0 \leq \lambda < 1$, therefore any $x_k \in \mathcal{X} \setminus \Omega$ satisfies

$$\begin{aligned} \ell(x_k) & > \ell(A(k)x_k + B(k)u^*(x_k) + w_k), \\ \forall w_k \in \mathbb{W}, \forall [A(k) B(k)] \in \Psi, \end{aligned} \quad (17)$$

meaning $\{\ell(x_k)\}_{k=0}^{\infty}$ is a strictly decreasing sequence outside Ω and bounded in the interval $[0, c]$. Thus, this sequence is convergent to 0. In other words, x_k tends to Ω as time tends to infinity. \square

Remark 4.11. Note that by construction, the partition associated with a convex lifting in Algorithm 1, may not be a Delaunay decomposition as in Scibilia, Oлару, and Hovd (2009). This method does not rely on such a decomposition, but relies on a convex lifting defined over this partition. This approach is simple and only requires solving a linear programming problem at each sampling instant. However, the associated control law is not continuous at the moment the state switches into Ω (see step 2 of Algorithm 2; this idea is similar to the one presented in Nguyen, Oлару, & Hovd, 2012). Note also that checking whether the current state belongs to Ω can be relaxed. Accordingly, one can continue solving problem (6) while trajectories still stay inside Ω . Indeed, if $x_k \in \Omega$, then due to the construction $\ell(x_k) = 0$. Consider the next state, one can see that $Kx_k \in \mathbb{U}$, then it leads to:

$$\begin{aligned} 0 &\leq \ell(A(k)x_k + B(k)u^*(x_k) + w_k) \\ &\leq \ell(A(k)x_k + B(k)Kx_k + w_k) = 0 = \ell(x_k). \end{aligned}$$

This inclusion implies that optimal control law $u^*(x_k) \in \mathbb{U}$ to problem (6) also keeps the trajectories inside Ω , if $x_k \in \Omega$.

Remark 4.12. We also remark that if the set $\mathcal{E} := \{x : \ell(x) \leq c\}$ and P_λ are not identical, then the constraints $A_j x_k + B_j u_k + w \in P_\lambda$ for all $[A_j \ B_j] \in \mathcal{V}(\Psi)$, $w \in \mathcal{V}(\mathbb{W})$, should be included in problem (6) to guarantee that $\ell(x)$ is exclusively restricted to P_λ .

Remark 4.13. An open problem is to guarantee robust stability of the proposed method for another estimation of the domain of attraction as the N -step robust controllable set denoted by $\mathcal{K}_N(\Omega)$, c.f., Kerrigan (2001). Note that in this case, proving the strict decrease of $\ell(x)$ becomes more difficult. Also, this strict decrease may not be successive, since $\mathcal{K}_N(\Omega)$ is not usually contractive.

Remark 4.14. Note that the explicit robust controller of (6) can be obtained by replacing $\alpha \ell(x_k)$ with a variable, denoted by, e.g., z . Accordingly, the optimization problem (6) becomes a parametric linear programming problem with the decision argument $[z \ u_k^T]^T$ and the parameter as the current state x_k .

Remark 4.15. Recall that the constructed convex lifting is equal to 0 over the given robust positively invariant set Ω , whereas optimal quadratic cost function usually deployed in MPC is strictly positive except at the origin. On the other hand, as shown in Algorithm 1, a scalar $c > 0$ is freely chosen, therefore one can always choose a small enough value such that $\ell(x)$ is smaller than the aforementioned quadratic cost function, leading to a smaller control performance of the proposed method relative to a given quadratic one in MPC.

5. Numerical example

To illustrate the proposed procedure, consider Example 1 in Kothare et al. (1996) where an angular antenna positioning system is modeled by the following equation:

$$x_{k+1} = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 - 0.1\alpha_k \end{bmatrix} x_k + \begin{bmatrix} 0 \\ 0.1\kappa \end{bmatrix} u_k,$$

where $\kappa = 0.787$ and the uncertain parameter α_k ranges in interval $[0.1 \ 10]$. The state and control variables are subject to the following constraints: $\|x_k\|_\infty \leq 1$, $\|u_k\|_\infty \leq 2$. Unconstrained controller is chosen as follows: $u = [-3.9922 \ -6.5135]x$. Accordingly, the maximal robust positively invariant set associated

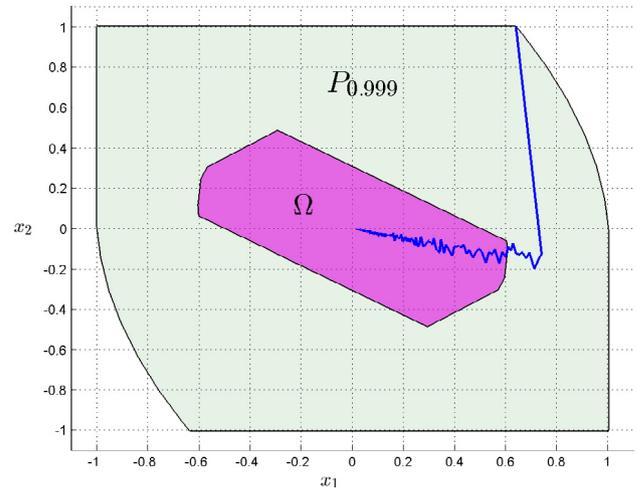


Fig. 1. The maximal robust positively invariant set Ω , an estimation of the domain of attraction $\mathcal{X} = P_{0.999}$ and the closed-loop trajectories.

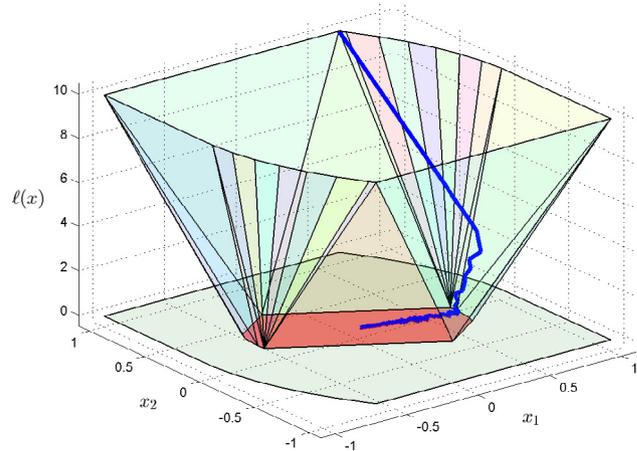


Fig. 2. A convex lifting $\ell(x)$ constructed by Algorithm 1 with $c = 10$ and its strict decrease over $\mathcal{X} \setminus \Omega$ along the state.

with the above controller, i.e., Ω is shown in Fig. 1. Also, the maximal 0.999-contractive set $P_{0.999}$ is presented therein. This set is computed from procedure (18). A convex lifting $\ell(x)$ is visualized in Fig. 2 according to Algorithm 1 with $c = 10$. The closed-loop trajectories are shown in Fig. 1 to be convergent to the origin, since the unconstrained control law can cope with the given set of polytopic uncertainties over Ω . Finally, the strict decrease of $\ell(x_k)$ over $\mathcal{X} \setminus \Omega$, is illustrated in Fig. 2.

To clarify the benefit of the proposed method in comparison to the methods in Kothare et al. (1996) and Kouvaritakis et al. (2000), we will consider their online computational time. To this end, we consider again the above angular antenna positioning system. As aforementioned, these two methods require resolutions of LMI problems at each sampling time, therefore their online computation is expensive. Whereas, the proposed method only needs to solve a linear programming problem. To clarify this aspect, we present in Fig. 3 the online computational time along the simulation of these three approaches at the same initial condition. Clearly, the online computation of the proposed approach is much cheaper than the other ones. We also compare the feasible region of the proposed method and the one by Kothare et al. (1996). This end is visualized in Fig. 4, the feasible region by the proposed method is clearly bigger than the one by Kothare et al. (1996).

Recall that robust MPC schemes, e.g., Mayne et al. (2005) and Pluymer, Rossiter, Suykens, and De Moor (2005), minimize cost functions of finite prediction horizons; thus terminal constraints

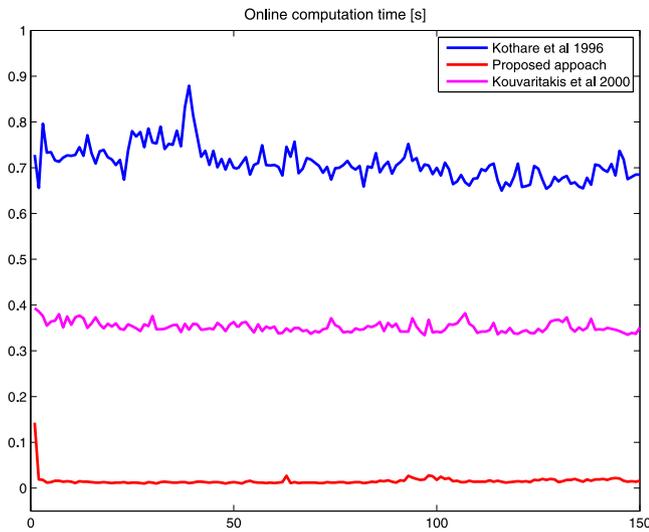


Fig. 3. The online computational time of the methods in Kothare et al. (1996) and Kouvaritakis et al. (2000) and the proposed one.

are usually imposed to guarantee closed-loop stability. Note also that the bigger the prediction horizon is, the larger the feasible region becomes. However, these methods suffer from considering polytopic uncertainties, since they lead to complicated formulations. Moreover, solving a linear programming problem is in general cheaper than solving a quadratic one; accordingly, the methods in Mayne et al. (2005) and Pluymers et al. (2005) are more computationally demanding than the proposed method.

To compare the proposed method with the robust MPC design in Bemporad, Borrelli, and Morari (2003), it is reasonable to compare their number of constraints in their design formulations, even if the latter method synthesizes explicit robust controllers. To illustrate this point, we again consider the angular antenna positioning system. Accordingly, an ∞ -norm cost function is chosen with weighting matrices $P, Q \in \mathbb{R}^{2 \times 2}, R \in \mathbb{R}$. Also, we choose the terminal constraints as Ω which is described by 10 halfspaces. The numbers of constraints by the method in Bemporad et al. (2003) via different prediction horizons N and of the proposed approach are summarized in Table 1. It can be seen that the number of constraints via the approach in Bemporad et al. (2003) increases exponentially with the prediction horizon, thus this method becomes much more expensive than the proposed one as the prediction horizon increases.

Finally, to compare the proposed method with the MPC design in Khan and Rossiter (2012), we both compare the feasible region and online computation. Their feasible regions are also included in Fig. 4, where the generalized parameterization MPC (RGMPC) in Khan and Rossiter (2012) is configured with $a = 0.65, b = 0.67, c = 0.64$. Note that the brown and blue regions represent the feasible regions obtained from the method in Khan and Rossiter (2012) with respectively $n_c = 3$ and $n_c = 2$. In these cases, the feasible regions obtained by this method are smaller than the one of the proposed method. One can argue that increasing n_c can enlarge the feasible region, however, also lead to more demanding online evaluation since the number of constraints scales exponentially with n_c . The numbers of constraints corresponding to this example are listed in Table 1. Recall that this method requires solving a quadratic cost function at each sampling time, thus more expensive online computation in comparison to the proposed one.

The numerical examples of this paper have been performed by means of MPT in Herceg, Kvasnica, Jones, and Morari (2013) and YALMIP in Löfberg (2004).

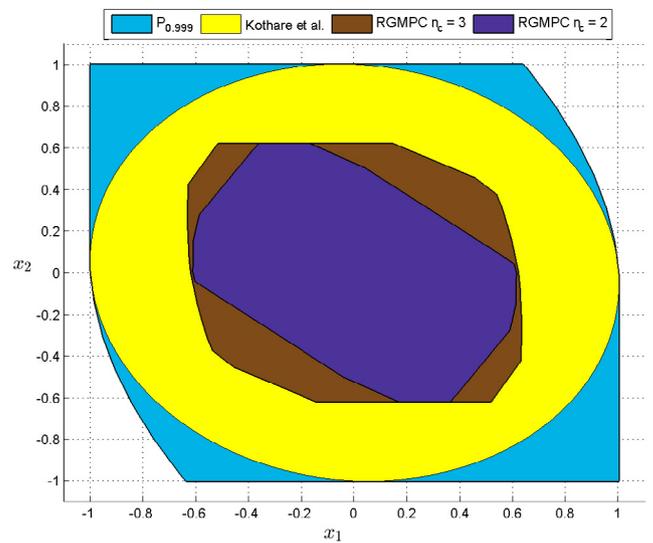


Fig. 4. Comparison of feasible regions via different approaches.

Table 1

Comparison of number of constraints via different approaches.

N/n_c	# constraints		
	1	2	3
Bemporad et al. (2003)	32	81	167
Khan and Rossiter (2012)		66	138
The proposed approach		61	

6. Conclusions

This paper presented a new method to design robust control law for constrained linear systems affected by bounded additive disturbances and polytopic uncertainties. This method was based on convex liftings. It was shown to guarantee the recursive feasibility and robust stability as well. The benefit of the proposed method was also shown via a numerical example relative to several MPC methods.

Acknowledgments

The authors would like to thank the Associate Editor and the anonymous reviewers for constructive comments to improve this manuscript. This work has partially been supported by the Linz Center of Mechatronics (LCM) in the framework of the Austrian COMET-K2 program. The research by the second author has benefited from the financial support of the European Unions 7th Framework Programme under EC-GA No. 607957 TEMPO—Training in Embedded Model Predictive Control and Optimization. Michal Kvasnica gratefully acknowledges the contribution of the Scientific Grant Agency of the Slovak Republic under the grant 1/0403/15 and the contribution of the Slovak Research and Development Agency under the project APVV 15-0007.

Appendix

A.1. Algorithm to compute the maximal λ -contractive set

$$\begin{aligned}
 S_1 &:= \mathbb{X}, \\
 S_{i+1} &:= \left\{ x \in S_i : \exists u(x) \in \mathbb{U} \text{ s.t. } (A_i x + B_i u(x)) \oplus \mathbb{W} \subseteq \lambda S_i, \right. \\
 &\quad \left. \forall j \in \mathcal{L}_L \right\}, \\
 P_\lambda &:= S_\infty.
 \end{aligned} \tag{18}$$

A.2. Proof of Lemma 4.6

$\ell(x)$ is a piecewise affine function since it is induced from a parametric linear programming problem. The continuity and convexity of $\ell(x)$ can easily be derived from Theorems IV-3 and IV-4 in Gal (1995). \square

A.3. Proof of Lemma 4.7

To prove that $\ell(x)$ is a convex lifting for $\{\mathcal{X}_i\}_{i \in \mathcal{I}_N}$, we need to prove that for any pair of $(\mathcal{X}_i, \mathcal{X}_j)$ the associated optimal solutions are different, i.e., $(a_i, b_i) \neq (a_j, b_j)$. Suppose the converse situation happens, more precisely, there exist two regions $(\mathcal{X}_i, \mathcal{X}_j)$ such that $(a_i, b_i) = (a_j, b_j)$.

First, it can easily be seen that the optimal solution to the parametric linear programming problem (5) is unique. In fact, suppose there exist two different optimal solutions to (5), i.e., $z_1^*(x)$ and $z_2^*(x)$. Consider a region \mathcal{X}_i in the associated partition over which $z_1^*(x)$, $z_2^*(x)$ are defined, i.e., $z_1^*(x) := (a_i^{(1)})^T x + b_i^{(1)}$, $z_2^*(x) := (a_i^{(2)})^T x + b_i^{(2)}$. Since z is the cost function of (5), therefore, we obtain:

$$(a_i^{(1)})^T x + b_i^{(1)} = (a_i^{(2)})^T x + b_i^{(2)} \quad \text{for all } x \in \mathcal{X}_i. \quad (19)$$

Note that the set of all x satisfying (19) describes a set of dimension lower than d_x , whereas (19) also holds true for all $x \in \mathcal{X}_i$ as a full-dimensional polyhedron. This case only holds if $(a_i^{(1)}, b_i^{(1)}) = (a_i^{(2)}, b_i^{(2)})$. This leads to the uniqueness of the optimal solution to (5).

Consider now two regions $(\mathcal{X}_i, \mathcal{X}_j)$ such that $(a_i, b_i) = (a_j, b_j)$. Let the optimization problem (5) be written in the following form:

$$\min_z z \quad \text{s.t. } Gz \leq W + Ex. \quad (20)$$

Without loss of generality, the constraint set of (20) is assumed to be in minimal representation. Also, suppose the constraints active at $[x^T \ a_j^T x + b_j]^T$ are as follows:

$$G^{(j)}z = W^{(j)} + E^{(j)}x.$$

Due to the uniqueness of the optimal solution to (20), $G^{(j)} \in \mathbb{R} \setminus \{0\}$. Consider $x \in \mathcal{X}_i$, it can be seen that $[x^T \ a_i^T x + b_i]^T$ satisfies the set of constraints in (20). However, since $\mathcal{X}_i \neq \mathcal{X}_j$, $[x^T \ a_i^T x + b_i]^T$ does not make constraint $G^{(j)}z \leq W^{(j)} + E^{(j)}x$ active; more precisely

$$G^{(j)}(a_i^T x + b_i) < W^{(j)} + E^{(j)}x. \quad (21)$$

As assumed $(a_i, b_i) = (a_j, b_j)$, $G^{(j)}z \leq W^{(j)} + E^{(j)}x$ thus becomes active at $[x^T \ a_i^T x + b_i]^T$ for $x \in \mathcal{X}_j$, namely,

$$G^{(j)}(a_i^T x + b_i) = W^{(j)} + E^{(j)}x. \quad (22)$$

Since \mathcal{X}_j is a full-dimensional polyhedron, then inclusion (22) yields

$$G^{(j)}a_i^T = E^{(j)}, \quad G^{(j)}b_i = W^{(j)}. \quad (23)$$

Inclusions (21) and (23) are clearly contradictory. In other words, for any pair of different regions $(\mathcal{X}_i, \mathcal{X}_j)$, the optimal solution to (5), i.e., $\ell(x)$ satisfies $(a_i, b_i) \neq (a_j, b_j)$.

In addition, Lemma 4.6 shows that $\ell(x)$ is a continuous, convex, piecewise affine function. Therefore, $\ell(x)$ is a convex lifting for $\{\mathcal{X}_i\}_{i \in \mathcal{I}_N}$ according to Definition 4.5. \square

A.4. Proof of Lemma 4.8

Indeed, consider $x \in \Omega$, then x can be written as a convex combination of the vertices of Ω , i.e., $x = \sum_{v \in \mathcal{V}(\Omega)} \alpha(v)v$ with $\alpha(v) \geq 0$ and $\sum_{v \in \mathcal{V}(\Omega)} \alpha(v) = 1$. It is known that $\ell(x)$ over Ω is an affine function, then $\ell(x) = a_i^T x + b_i$ leads to $\ell(x) = 0$ for every $x \in \Omega$.

To complete the proof, it is necessary to show that $\ell(x) > 0$ for $x \in \mathcal{X} \setminus \Omega$. Indeed, as shown above, $\ell(x) = a_i^T x + b_i = 0$ for every $x \in \Omega$, then since Ω is of full dimension, it follows $a_i = 0$, $b_i = 0$. Consider a region $\mathcal{X}_j \neq \Omega = \mathcal{X}_i$ of the polytopical partition $\{\mathcal{X}_i\}_{i \in \mathcal{I}_N}$ associated with $\ell(x)$, $\ell(x) = a_j^T x + b_j$ for $x \in \mathcal{X}_j$. According to Lemma 4.7, $\ell(x)$ satisfies the convexity and continuity conditions of a convex lifting:

$$a_j^T x + b_j > a_i^T x + b_i = 0, \quad \text{for all } x \in \mathcal{X}_j \setminus \mathcal{X}_i,$$

$$a_j^T x + b_j = a_i^T x + b_i = 0, \quad \text{for all } x \in \mathcal{X}_j \cap \mathcal{X}_i.$$

The same inclusion for the other affine functions of $\ell(x)$, leads to the nonnegativity of $\ell(x)$. Moreover, $\ell(x) > 0$ for any $x \in \mathcal{X} \setminus \Omega$. The proof is complete. \square

A.5. Proof of Lemma 4.9

Due to the convexity of $\ell(x)$ over \mathcal{X} as proved in Lemma 4.6, it leads to

$$\ell(\beta x + (1 - \beta)0) \leq \beta \ell(x) + (1 - \beta)\ell(0).$$

Due to the assumption that $0 \in \text{int}(\mathbb{W})$, then $0 \in \text{int}(\Omega)$, meaning that $\ell(0) = 0$ according to Lemma 4.8. This inclusion and the above one imply that $\ell(\beta x) \leq \beta \ell(x)$. \square

References

- Anderson, B. D., & Moore, J. B. (2007). *Optimal control: linear quadratic methods*. Courier Corporation.
- Baotic, M., Christophersen, F. J., & Morari, M. (2006). Constrained optimal control of hybrid systems with a linear performance index. *IEEE Transactions on Automatic Control*, 51, 1903–1919.
- Bemporad, A., Borrelli, F., & Morari, M. (2003). Min-max control of constrained uncertain discrete-time linear systems. *IEEE Transactions on Automatic Control*, 48, 1600–1606.
- Bitsoris, G. (1988a). On the positive invariance of polyhedral sets for discrete-time systems. *Systems & Control Letters*, 11, 243–248.
- Bitsoris, G. (1988b). Positively invariant polyhedral sets of discrete-time linear systems. *International Journal of Control*, 47, 1713–1726.
- Bitsoris, G., & Vassilaki, M. (1995). Constrained regulation of linear systems. *Automatica*, 31, 223–227.
- Blanchini, F. (1994). Ultimate boundedness control for uncertain discrete-time systems via set-induced lyapunov functions. *IEEE Transactions on Automatic Control*, 39, 428–433.
- Blanchini, F. (1995). Nonquadratic lyapunov functions for robust control. *Automatica*, 31, 451–461.
- Blanchini, F., & Miani, S. (2007). *Set-theoretic methods in control*. Springer.
- Boyd, S. P., El Ghaoui, L., Feron, E., & Balakrishnan, (1994). *Linear matrix inequalities in system and control theory*. Vol. 15. SIAM.
- Brayton, R. K., & Tong, C. H. (1979). Stability of dynamical systems: A constructive approach. *IEEE Transactions on Circuits and Systems*, 26, 224–234.
- Chmielewski, D., & Manousiouthakis, V. (1996). On constrained infinite-time linear quadratic optimal control. *Systems & Control Letters*, 29, 121–129.
- Cuzzola, F. A., Geromel, J. C., & Morari, M. (2002). An improved approach for constrained robust model predictive control. *Automatica*, 38, 1183–1189.
- Daafouz, J., & Bernussou, J. (2001). Parameter dependent lyapunov functions for discrete time systems with time varying parametric uncertainties. *Systems & Control Letters*, 43, 355–359.
- Gal, T. (1995). *Postoptimal analyses, parametric programming and related topics*. Walter de Gruyter.
- Gilbert, E. G., & Tan, K. T. (1991). Linear systems with state and control constraints: The theory and application of maximal output admissible sets. *IEEE Transactions on Automatic Control*, 36, 1008–1020.
- Gulan, M., Nguyen, N. A., Oлару, S., Rodríguez-Ayerbe, P., & Rohal-Ilkiv, B. (2015). Implications of inverse parametric optimization in model predictive control. In S. Oлару, A. Grancharova, & F. L. Pereira (Eds.), *Developments in model-based optimization and control* (pp. 49–70). Springer.

- Gutman, P. O., & Cwikel, M. (1987). An algorithm to find maximal state constraint sets for discrete-time linear dynamical systems with bounded controls and states. *IEEE Transactions on Automatic Control*, 32, 251–254.
- Herceg, M., Kvasnica, M., Jones, C., & Morari, M. (2013). Multi-parametric toolbox 3.0. In *Control conference (ECC), 2013 European*, (pp. 502–510).
- Jiang, Z. P., & Wang, Y. (2001). Input-to-state stability for discrete-time nonlinear systems. *Automatica*, 37, 857–869.
- Kalman, R. E., & Bertram, J. E. (1960). Control system analysis and design via the second method of lyapunov: ldiscrete-time systems. *Journal of Fluids Engineering*, 82, 394–400.
- Kerrigan, E. C. (2001). *Robust constraint satisfaction: Invariant sets and predictive control*. (Ph.D. thesis), University of Cambridge.
- Khalil, H. K. (2002). *Nonlinear systems*. (p. 323). Prentice Hall, Inc..
- Khan, B., & Rossiter, J. A. (2012). Robust mpc algorithms using alternative parameterisations. In *2012 UKACC international conference on control* (pp. 882–887). IEEE.
- Kofman, E., Haimovich, H., & Seron, M. M. (2007). A systematic method to obtain ultimate bounds for perturbed systems. *International Journal of Control*, 80, 167–178.
- Kolmanovskiy, I., & Gilbert, E. G. (1998). Theory and computation of disturbance invariant sets for discrete-time linear systems. *Mathematical Problems in Engineering*, 4, 317–367.
- Kothare, M. V., Balakrishnan, V., & Morari, M. (1996). Robust constrained model predictive control using linear matrix inequalities. *Automatica*, 32, 1361–1379.
- Kouvaritakis, B., Rossiter, J. A., & Schuurmans, J. (2000). Efficient robust predictive control. *IEEE Transactions on Automatic Control*, 45, 1545–1549.
- Lazar, M. (2010). On infinity norms as lyapunov functions: Alternative necessary and sufficient conditions. In *2010 49th IEEE conference on decision and control, (CDC)*. (pp. 5936–5942). IEEE.
- Löfberg, J. (2004). Yalmip: A toolbox for modeling and optimization in matlab. In *2004 IEEE international symposium on computer aided control systems design* (pp. 284–289). IEEE.
- Loria, A., & Panteley, E. (2006). Stability, told by its developers. In *Advanced topics in control systems theory* (pp. 199–258). Springer.
- Lyapunov, A. M. (1907). Problème général de la stabilité du mouvement. In *Annales de la Faculté des Sciences de Toulouse* (pp. 203–474). Université Paul Sabatier.
- Mayne, D. Q., Seron, M. M., & Raković, S. V. (2005). Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41, 219–224.
- Michel, A. N., Nam, B. H., & Vittal, V. (1984). Computer generated lyapunov functions for interconnected systems: Improved results with applications to power systems. *IEEE Transactions on Circuits and Systems*, 31, 189–198.
- Molchanov, A. P., & Pyatnitskiy, Y. S. (1989). Criteria of asymptotic stability of differential and difference inclusions encountered in control theory. *Systems & Control Letters*, 13, 59–64.
- Nguyen, H. N. (2014). *Lecture notes in control and information sciences, Constrained control of uncertain, time-varying, discrete-time systems: an interpolation-based approach*.
- Nguyen, N. A. (2015). *Explicit robust constrained control for linear systems: analysis, implementation and design based on optimization*. (Ph.D. thesis), France: CentraleSupélec, Université Paris-Saclay.
- Nguyen, N.A., Gulan, M., Oлару, S., & Rodríguez-Ayerbe, P. (2016). Convex liftings: Theory and control applications. URL: <https://hal-centralesupelec.archives-ouvertes.fr/hal-01326804/document>.
- Nguyen, H. N., Oлару, S., & Hovd, M. (2012). A patchy approximation of explicit model predictive control. *International Journal of Control*, 85, 1929–1941.
- Nguyen, N. A., Oлару, S., & Rodríguez-Ayerbe, P. (2015a). Any discontinuous PWA function is optimal solution to a parametric linear programming problem. In *54th conference on decision and control* (pp. 5926–5931). Osaka, Japan: IEEE.
- Nguyen, N. A., Oлару, S., & Rodríguez-Ayerbe, P. (2015b). Inverse parametric linear/quadratic programming problem for continuous PWA functions defined on polyhedral partitions of polyhedra. In *54th conference on decision and control* (pp. 5920–5925). Osaka, Japan: IEEE.
- Nguyen, N. A., Oлару, S., & Rodríguez-Ayerbe, P. (2015c). Robust control design based on convex liftings. In *8th IFAC symposium on robust control design* (pp. 308–313). Elsevier.
- Nguyen, N.A., Oлару, S., Rodríguez-Ayerbe, P., Hovd, M., & Necoara, I. (2014). Inverse parametric convex programming problems via convex liftings. In *Proc. of the 19th IFAC world congress*, Cape Town, SA.
- Nguyen, N. A., Oлару, S., Rodríguez-Ayerbe, P., Hovd, M., & Necoara, I. (2016). Constructive solution of inverse parametric linear/quadratic programming problems. *Journal of Optimization Theory and Applications*, 1–26. <http://dx.doi.org/10.1007/s10957-016-0968-0>.
- Pluymers, B., Rossiter, J.A., Suykens, J.A.K., & De Moor, B. (2005). A simple algorithm for robust mpc. In *Proceedings of the IFAC world congress*.
- Polanski, A. (1995). On infinity norms as lyapunov functions for linear systems. *IEEE Transactions on Automatic Control*, 40, 1270–1274.
- Rakovic, S. V., Kerrigan, E. C., Kouramas, K. I., & Mayne, D. Q. (2005). Invariant approximations of the minimal robust positively invariant set. *IEEE Transactions on Automatic Control*, 50, 406–410.
- Rakovic, S. V., Kouvaritakis, B., Cannon, M., Panos, C., & Findeisen, R. (2012). Parameterized tube model predictive control. *IEEE Transactions on Automatic Control*, 57, 2746–2761.
- Scibilia, F., Oлару, S., & Hovd, M. (2009). Approximate explicit linear MPC via Delaunay tessellation. In *European control conference* (pp. 2833–2838). Budapest, Hungary: IEEE.
- Sznaier, M., & Damborg, M.J. (1987). Suboptimal control of linear systems with state and control inequality constraints. In *IEEE, conference on decision and control*.
- Vassilaki, M., Hennes, J. C., & Bitsoris, G. (1988). Feedback control of linear discrete-time systems under state and control constraints. *International Journal of Control*, 47, 1727–1735.
- Wan, Z., & Kothare, M. V. (2003). An efficient off-line formulation of robust model predictive control using linear matrix inequalities. *Automatica*, 39, 837–846.
- Zubov, V. I., & Boron, L. F. (1964). *Methods of AM Lyapunov and their application*. Groningen: Noordhoff.



Ngoc Anh Nguyen received a double Dip.Ing. degree in Electrical Engineering at Hanoi University of Science and Technology, Vietnam and Grenoble Institute of Technology, France, both in 2012. He is currently doing a postdoc at the Johannes Kepler University Linz, Austria. Prior to that, he obtained his Ph.D. degree at Laboratory of Signals and Systems, CentraleSupélec, University Paris Saclay, France in December 2015. His main research interests lie in computational geometry, optimization based control and set theoretic methods.



Sorin Oлару is a Professor in CentraleSupélec, member of the CNRS Laboratory of Signals and Systems and of the INRIA team DISCO, all these institutions being part of the Paris-Saclay University in France. His research interests are encompassing the optimization-based control design, set-theoretic characterization of constrained dynamical systems as well as the numerical methods in control. He is currently involved in research projects related to embedded predictive control, fault tolerant control and time-delay systems.



Pedro Rodríguez-Ayerbe received the technical engineering Diploma in electronics from Mondragon University, Arrasate, Spain, in 1993, and the Engineering degree in electrical engineering from SUPELEC, Gif sur Yvette, France, in 1996. In 2003, he received the Ph.D. degree in automatic control from SUPELEC and the Université Paris Sud, Orsay, France. He is currently an Associate Professor in CentraleSupélec, member of the CNRS Laboratory of Signals and Systems being part of the Paris-Saclay University in France. His research interests include constrained predictive control and robust control theory.



Michal Kvasnica was born in 1977. He received his diploma in chemical engineering from the Slovak University of Technology in Bratislava, Slovakia and the Ph.D. in electrical engineering from the Swiss Federal Institute of Technology in Zurich, Switzerland. Since 2011 he is an Associate Professor at the Slovak University of Technology in Bratislava. His research interests are in model predictive control, modeling of hybrid systems, and development of software tools for control. He is the co-author and developer of the MPT Toolbox for explicit model predictive control.

Clipping-Based Complexity Reduction in Explicit MPC

Michal Kvasnica and Miroslav Fikar

Abstract—The idea of explicit model predictive control (MPC) is to characterize optimal control inputs as an explicit piecewise affine (PWA) function of the initial conditions. The function, however, is often too complex and either requires too much processing power to evaluate on-line, or consumes a prohibitive amount of memory. The paper focuses on the memory issue and proposes a novel method of replacing a generic continuous PWA function by a different function of significantly lower complexity in such a way that the simple function guarantees the same properties as the original. The idea is based on eliminating regions of the PWA function over which the function attains a saturated value. An extensive case study is presented which confirms that a significant reduction of complexity is achieved in general.

Index Terms—Computational complexity, piecewise linear techniques, predictive control.

I. INTRODUCTION

As shown in [5], the effort of implementing model predictive control (MPC) in the Receding Horizon fashion (RHMP) can be substantially reduced by pre-computing the feedback law $u^*(x)$, for all feasible initial conditions, as a piecewise affine (PWA) function $\kappa(x)$. Such a function is defined over R polyhedral regions with associated affine feedback laws. The problem being that on-line evaluation speed, as well as the associated memory storage, are proportional to R . Therefore, it is of imminent importance to keep complexity of $\kappa(x)$ as low as possible.

The problem is usually attacked by approximating the optimal feedback or the optimal value function in such a way that a less complex, albeit suboptimal, feedback function $\tilde{\kappa}(x)$ is obtained, see e.g., [4], [16]. Suboptimal approximations $\tilde{\kappa}(x)$ can be also obtained by replacing the regions of $\kappa(x)$ by simpler objects, e.g., by hypercubes [7] or by simplices [14]. Another line is focused on deriving a simpler representation of the PWA function $\kappa(x)$ with no implications on optimality of the feedback. Here, one can either characterize $\kappa(x)$ in terms of its lattice representation [17], or exploit convexity of the optimal value function [3]. In [9] the optimal region merging (ORM) method was proposed. It is based on merging together regions of $\kappa(x)$ whose union is convex and which share the same affine expression for $u^*(x)$. The result is a new PWA function $\tilde{\kappa}(x)$, which is an *equivalent* replacement of $\kappa(x)$ in the sense that $\tilde{\kappa}(x) = \kappa(x)$ for all feasible initial conditions x . The downside being that merging regions optimally is an NP-hard problem.

In this paper, we present a different way of computing an equivalent replacement function $\tilde{\kappa}(x)$. The main benefit, compared to the ORM method [9], is that the construction of the replacement scales significantly better with growing complexity of the original function $\kappa(x)$. The approach is based on the premise that the RHMP controller operates at the limits of the admissible control freedom for some x . Simply

put, the idea is to remove the regions in which $\kappa(x)$ attains a saturated value and subsequently cover the “holes” by expanding the regions in which the value of $\kappa(x)$ is unsaturated. The replacement function $\tilde{\kappa}(x)$ is then passed through a so-called *clipping filter* $\phi(\tilde{\kappa}(x))$ such that the equivalence $\phi(\tilde{\kappa}(x)) = \kappa(x)$ is established for all feasible initial conditions x . A similar approach was proposed in [13] for cases with linear feedbacks of the form $\kappa(x) := Kx$.

We illustrate that complexity of $\tilde{\kappa}(x)$, measured by the number \tilde{R} of regions over which $\tilde{\kappa}(x)$ is defined, fulfills $\tilde{R} \leq R$. Moreover, we provide an extensive case study which shows that, typically, \tilde{R} is equal to the number of unsaturated regions R_{unsat} of the original function $\kappa(x)$. In addition, we show that $\tilde{R} \ll R$ for a vast majority of practical MPC setups. By decreasing the number of regions, two goals are achieved simultaneously: the memory footprint of explicit RHMP is decreased by a factor of R/\tilde{R} and the on-line implementation speed is increased accordingly. Moreover, the proposed method can be further combined either with ORM, or with more advanced evaluation strategies such as extrapolating the solution from closest regions [6], or omitting rarely visited regions [1]. In addition to our previous work [11], in this paper we propose a modified algorithm with improved properties, show how to process feedback laws for systems with multiple inputs, analyze complexity of the approach and provide a large case study which justifies that it scales well with problem size.

II. NOTATION AND DEFINITIONS

For a matrix or a vector A , $[A]_{\setminus \mathcal{I}}$ represents all rows of A except of those belonging to some index set \mathcal{I} . A finite set of n elements $\mathcal{R} := \{\mathcal{R}_1, \dots, \mathcal{R}_n\}$ will be denoted as $\{\mathcal{R}_i\}_{i=1}^n$ and its cardinality by $|\mathcal{R}|$. A polyhedron is the convex intersection of c closed affine half-spaces, i.e., $\mathcal{R} := \{x \in \mathbb{R}^{n_x} \mid R^x x \leq R^0\}$. We call the collection of polyhedra $\{\mathcal{R}_i\}_{i=1}^R$ the *partition* of polyhedron \mathcal{R} if $\mathcal{R} = \bigcup_{i=1}^R \mathcal{R}_i$, and $\text{int}(\mathcal{R}_i) \cap \text{int}(\mathcal{R}_j) = \emptyset$ for all $i \neq j$. Each polyhedron \mathcal{R}_i will be referred to as a *region* of the partition. Regions \mathcal{R}_i and \mathcal{R}_j of a partition \mathcal{R} are *adjacent* if $\mathcal{R}_i \cap \mathcal{R}_j$ is an $(n_x - 1)$ -dimensional facet of both \mathcal{R}_i and \mathcal{R}_j , $i \neq j$. For each facet j of region \mathcal{R}_i of the partition \mathcal{R} we denote by $\mathcal{A}_{i,j}(\mathcal{R})$ the index set of regions adjacent to \mathcal{R}_i along the j -th facet. Vector-valued function $\kappa(x) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_u}$ is called *Piecewise Affine (PWA) over polyhedra* $\{\mathcal{R}_i\}_{i=1}^R$ if $\kappa(x) := K_i x + L_i \forall x \in \mathcal{R}_i$, $i = 1, \dots, R$. PWA function $\kappa(x)$ is *continuous* if $K_i x + L_i = K_j x + L_j$ holds $\forall x \in \mathcal{R}_i \cap \mathcal{R}_j$, $i \neq j$.

III. EXPLICIT MODEL PREDICTIVE CONTROL

We consider the class of discrete-time, stabilizable linear time-invariant systems $x_{k+1} = Ax_k + Bu_k$, which are subject to polytopic constraints $x \in \mathcal{X} \subseteq \mathbb{R}^{n_x}$ and $u \in \mathcal{U} \subseteq \mathbb{R}^{n_u}$. Assume the following constrained finite-time optimal control problem:

$$\min_{U_N} \sum_{k=0}^{N-1} x_{k+1}^T Q_x x_{k+1} + u_k^T Q_u u_k \quad (1a)$$

$$\text{s.t.} \quad x_{k+1} = Ax_k + Bu_k, \quad x_k \in \mathcal{X}, \quad u_k \in \mathcal{U} \quad (1b)$$

where x_k and u_k denote, respectively, state and input predictions over a finite horizon $k = 0, \dots, N - 1$, given the initial condition x_0 . It is assumed that $Q_x = Q_x^T \succeq 0$, $Q_u = Q_u^T \succ 0$ in (1a), i.e., that (1) is a strictly convex QP. The receding horizon MPC feedback then becomes $u^*(x_0) = [\mathbf{1} \ \mathbf{0} \ \dots \ \mathbf{0}] U_N^*$, where the optimal vector $U_N^* := [u_0^T, \dots, u_{N-1}^T]^T$ can be found by solving (1) as a QP for a given value of the initial condition x_0 . For problems of modest size, it is also possible to characterize the optimal feedback $u^*(x_0)$ explicitly

Manuscript received September 21, 2010; revised April 04, 2011; accepted November 22, 2011. Date of publication December 09, 2011; date of current version June 22, 2012. This work was supported by the Scientific Grant Agency of the Slovak Republic under Grants 1/0071/09 and 1/0095/11 and of the Slovak Research and Development Agency under Contract LPP-0092-07. Recommended by Associate Editor P. Tsiftas.

The authors are with the Slovak University of Technology, 81237 Bratislava, Slovakia (e-mail: michal.kvasnica@stuba.sk; miroslav.fikar@stuba.sk).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TAC.2011.2179428

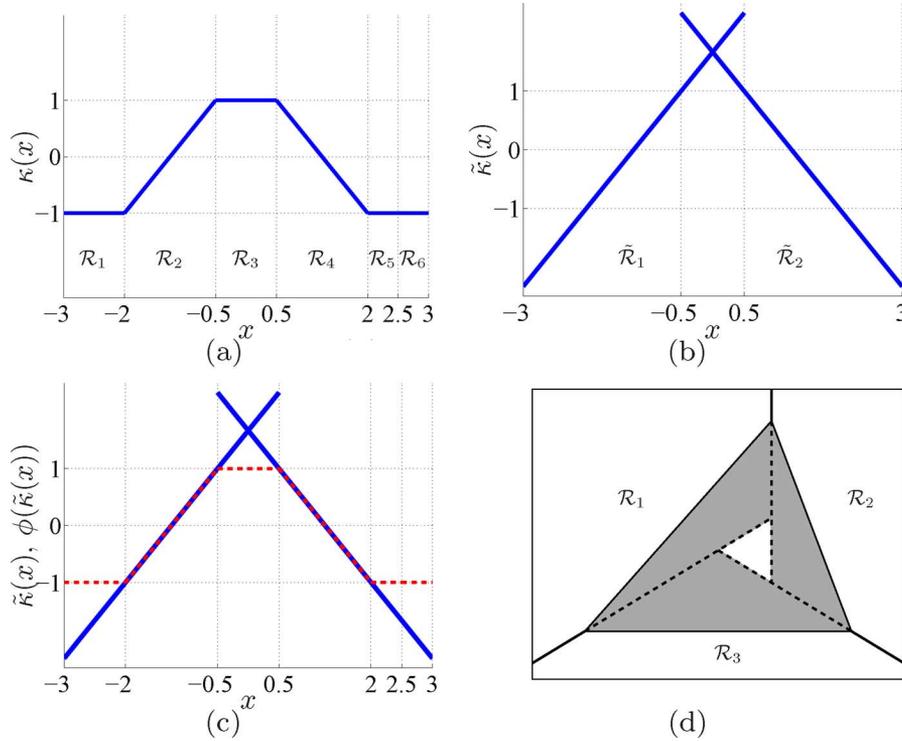


Fig. 1. Illustration of a suitable augmentation. (a) 1-D PWA function $\kappa(x)$ with 4 saturated regions ($\mathcal{R}_1, \mathcal{R}_3, \mathcal{R}_5, \mathcal{R}_6$) and two unsaturated regions \mathcal{R}_2 and \mathcal{R}_4 . (b) Extensions of unsaturated regions over their weakly adjacent saturated neighbors. (c) Suitable augmentation $\tilde{\kappa}(x)$ (solid line) and the result of clipping $\phi(\tilde{\kappa}(x))$ at $\bar{\kappa} = 1$ and $\underline{\kappa} = -1$ (dashed line). (d) Unsaturated regions $\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3$ in \mathbb{R}^2 whose extension does not fully cover a saturated region (the large shaded triangle).

as a PWA function of x_0 [5] by solving (1) as a *parametric quadratic program* (pQP).

Theorem 3.1 ([5]): The RHMPC feedback $u^*(x_0)$ for problem (1) is given by $u^*(x_0) = \kappa(x_0)$ where: (i) the set of feasible initial conditions $\Omega := \{x_0 \mid \exists u_0, \dots, u_{N-1} \text{ s.t. (1b) hold}\}$ is a convex polyhedron; (ii) $\kappa(x_0) : \Omega \rightarrow \mathcal{U}$ is a continuous PWA function defined over R regions $\mathcal{R}_i, i = 1, \dots, R$; (iii) \mathcal{R}_i are polyhedra with a closure $\bar{\mathcal{R}}_i = \{x \mid R_i^x \leq R_i^0\}$; and (iv) $\{\mathcal{R}_i\}_{i=1}^R$ is a partition of Ω .

The advantage of such an explicit representation is obvious: obtaining the optimal control action for a given x_0 reduces to a mere evaluation of the function $\kappa(x_0)$, which is henceforth denoted as the *explicit RHMPC feedback law*. The crucial limitation, however, is that the number of regions tends to be large, often above the limits of typical hardware implementation platforms both in respect to on-line computation as well as to memory storage.

IV. MAIN RESULTS

Theorem 3.1 gives the explicit RHMPC feedback $u^*(x) = \kappa(x)$ as a continuous PWA function defined over R polyhedral regions, which maps a vector of initial conditions onto the optimal control inputs. The complexity reduction problem to be solved can be formally stated as follows:

Problem 4.1: Given a continuous PWA function $\kappa(x)$ with R regions, find a simpler function $\tilde{\kappa}(x)$ with $\tilde{R} \leq R$ regions, and a suitable filter $\phi(\cdot)$, such that $\phi(\tilde{\kappa}(x)) = \kappa(x)$ for all $x \in \text{dom}(\kappa(x))$.

In the sequel, we show how to find the simpler function $\tilde{\kappa}(x)$ by using basic tools of computational geometry. The simplification is based on removing the regions of $\kappa(x)$ over which the function attains a saturated value. Subsequently, the “holes” are covered by expanding the unsaturated regions (cf. Definition 4.2). Therefore, in the best case, the number of regions of $\tilde{\kappa}(x)$ is equal to R_{unsat} , the number of unsaturated regions of $\kappa(x)$. We remark that, typically, $R_{\text{unsat}} \ll R$.

The procedure is first explained assuming that $\kappa(x) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ is scalar-valued, i.e., that the number of control inputs is $n_u = 1$. The multi-dimensional case is then discussed in Section IV-C.

A. Single-Input Case

Definition 4.2 (Saturated Region): Let $\bar{\kappa}$ and $\underline{\kappa}$ denote, respectively, the maximum and minimum values which the PWA function $\kappa(x) := \tilde{K}_i x + \tilde{L}_i$ attains over $\text{dom}(\kappa(x))$. Denote by \mathcal{I}_{max} (\mathcal{I}_{min}) the index set of regions where $\kappa(x)$ is saturated at the maximum (minimum), and let $\mathcal{I}_{\text{sat}} = \mathcal{I}_{\text{max}} \cup \mathcal{I}_{\text{min}}$. We call a region \mathcal{R}_i the *saturated region* if it is either saturated at the minimum or at the maximum, i.e., if $i \in \mathcal{I}_{\text{sat}}$. Otherwise the region is called *unsaturated*. The index set of unsaturated regions is denoted by $\mathcal{I}_{\text{unsat}}$.

Definition 4.3: Saturated region \mathcal{R}_j is *weakly adjacent* to an unsaturated region \mathcal{R}_i either if they are directly adjacent, or if \mathcal{R}_j is adjacent to some other saturated region(s) weakly adjacent to \mathcal{R}_i .

Definition 4.4 (Suitable Augmentation): Given is a continuous PWA function $\kappa(x)$, defined over the partition $\{\mathcal{R}_i\}_{i=1}^R$. We call the PWA function $\tilde{\kappa}(x) := \tilde{K}_i x + \tilde{L}_i$ a *suitable augmentation* of $\kappa(x)$ if the following properties hold:

- P1: $\tilde{\kappa}(x)$ is defined over regions $\{\tilde{\mathcal{R}}_j\}_{j=1}^{\tilde{R}}$ such that $\bigcup_j \tilde{\mathcal{R}}_j = \bigcup_i \mathcal{R}_i$, i.e., $\text{dom}(\tilde{\kappa}(x)) = \text{dom}(\kappa(x))$.
- P2: $\tilde{\kappa}(x) = \kappa(x)$ for all $x \in \mathcal{R}_{\mathcal{I}_{\text{unsat}}}$.
- P3: $\tilde{\kappa}(x) \geq \bar{\kappa}$ for all $x \in \mathcal{R}_{\mathcal{I}_{\text{max}}}$.
- P4: $\tilde{\kappa}(x) \leq \underline{\kappa}$ for all $x \in \mathcal{R}_{\mathcal{I}_{\text{min}}}$.

Here, $\bar{\kappa}$, $\underline{\kappa}$, $\mathcal{I}_{\text{unsat}}$, \mathcal{I}_{max} , and \mathcal{I}_{min} are as in Definition 4.2, and $\mathcal{R}_{\mathcal{I}}$ denotes the subset of regions $\{\mathcal{R}_i\}_{i \in \mathcal{I}}$ for some index set $\mathcal{I} \subseteq \{1, \dots, R\}$.

Fig. 1(a) shows an illustrative 1-D PWA function $\kappa(x)$, while Fig. 1(b) depicts its suitable augmentation. Notice that a suitable augmentation $\tilde{\kappa}(x)$ is not, by Definition 4.4, required to be continuous, nor does it require that $\tilde{\kappa}(x) = \kappa(x)$ for all $x \in \text{dom}(\kappa(x))$. It merely

suggests that one can replace the affine expression $\kappa(x) = K_i x + L_i$ in the saturated regions by an arbitrary $\tilde{K}_i x + \tilde{L}_i$ which satisfies P3–P4. As will be shown in the sequel, this freedom allows to construct a simpler function $\tilde{\kappa}(x)$ by enlarging the unsaturated regions such that they completely cover the saturated ones. Once such a function is obtained, we recover $\kappa(x)$ by applying a simple clipping filter $\phi(\cdot)$ such that $\phi(\tilde{\kappa}(x)) = \kappa(x) \forall x \in \text{dom}(\kappa(x))$. A procedure for computing $\tilde{\kappa}(x)$ is reported as Algorithm 1, which is the first main result of the paper. We will explain the algorithm on the following example.

Algorithm 1 Construction of a suitable augmentation

INPUT: Saturated continuous PWA function $\kappa(x)$ defined over the polyhedral partition $\mathcal{R} = \{\mathcal{R}_i\}_{i=1}^R$ with $\mathcal{R}_i = \{x \mid R_i^x x \leq R_i^0\}$ and $\Omega = \cup_i \mathcal{R}_i$ being a convex polyhedron.

OUTPUT: Suitable augmentation $\tilde{\kappa}(x) = \tilde{K}_j x + \tilde{L}_j$ if $x \in \tilde{\mathcal{R}}_j$, $j = 1, \dots, \tilde{R}$.

1: Obtain the adjacency list $\mathcal{A}_{i,j}(\mathcal{R})$ and index sets $\mathcal{I}_{\text{unsat}}$ and \mathcal{I}_{sat} representing indices of unsaturated and saturated regions, respectively.

2: **for each** unsaturated region $\mathcal{R}_i \in \mathcal{R}_{\mathcal{I}_{\text{unsat}}}$ **do**

3: Using the adjacency list $\mathcal{A}_{i,j}$ identify the subset of half-space indices \mathcal{J} over which the neighbor of \mathcal{R}_i is a saturated region.

4: Form a new polyhedron $\tilde{\mathcal{R}}_r = \{x \mid \tilde{R}_r^x x \leq \tilde{R}_r^0\}$ by removing from \mathcal{R}_i the half-spaces indexed by \mathcal{J} , i.e., $\tilde{R}_r^x = [R_i^x]_{\setminus \mathcal{J}}$ and $\tilde{R}_r^0 = [R_i^0]_{\setminus \mathcal{J}}$. Let $\tilde{\mathcal{R}}_r = \tilde{\mathcal{R}}_r \cap \Omega$.

5: Let $\mathcal{I}_{\text{cand}} = (\mathcal{I}_{\text{unsat}} \setminus \{i\}) \cup (\mathcal{I}_{\text{sat}} \setminus \mathcal{I}_{\text{adj},i})$ where $\mathcal{I}_{\text{adj},i}$ is the index set of saturated regions weakly adjacent to \mathcal{R}_i . Denote by \mathcal{I}_{int} indices of regions $\mathcal{R}_{\mathcal{I}_{\text{cand}}}$ intersecting with $\tilde{\mathcal{R}}_r$.

6: **if** $\mathcal{I}_{\text{int}} \neq \emptyset$ **then** $\tilde{\mathcal{R}}_r = \tilde{\mathcal{R}}_r \setminus \mathcal{R}_{\mathcal{I}_{\text{int}}}$.

7: Store region(s) $\tilde{\mathcal{R}}_r$ and matrices $\tilde{K}_r = K_i$, $\tilde{L}_r = L_i$.

8: **end for**

9: Determine the index set of saturated regions not fully covered by $\tilde{\mathcal{R}}$, i.e., find $\mathcal{I}_{\text{nocover}} = \{i \mid \mathcal{R}_i \not\subseteq \tilde{\mathcal{R}}, i \in \mathcal{I}_{\text{sat}}\}$.

10: **if** $\mathcal{I}_{\text{nocover}} \neq \emptyset$ **then** $\tilde{\mathcal{R}} = \tilde{\mathcal{R}} \cup \mathcal{R}_{\mathcal{I}_{\text{nocover}}}$ and update \tilde{K} and \tilde{L} accordingly.

11: **if** $|\tilde{\mathcal{R}}| > |\mathcal{R}|$ **then** $\tilde{\kappa}(x) := \kappa(x)$.

Example 4.5: Consider a 1-D PWA function $\kappa(x) : \mathbb{R} \rightarrow \mathbb{R}$ shown in Fig. 1(a) with $R = 6$, $\mathcal{R}_1 = \{x \mid -3 \leq x \leq -2\}$, $\mathcal{R}_2 = \{x \mid -2 \leq x \leq -0.5\}$, $\mathcal{R}_3 = \{x \mid -0.5 \leq x \leq 0.5\}$, $\mathcal{R}_4 = \{x \mid 0.5 \leq x \leq 2\}$, $\mathcal{R}_5 = \{x \mid 2 \leq x \leq 2.5\}$, $\mathcal{R}_6 = \{x \mid 2.5 \leq x \leq 3\}$, $\Omega = \{x \mid -3 \leq x \leq 3\}$, $\bar{\kappa} = 1$, $\underline{\kappa} = -1$, $\mathcal{I}_{\text{sat}} = \{1, 3, 5, 6\}$ and $\mathcal{I}_{\text{unsat}} = \{2, 4\}$. The algorithm iterates through all unsaturated regions in an arbitrary order. Take $i = 2$. Then in Step 3, unsaturated region \mathcal{R}_2 has \mathcal{R}_1 and \mathcal{R}_3 as saturated neighbors over half-spaces $x \geq -2$ and $x \leq -0.5$. Therefore, on Step 4, region $\tilde{\mathcal{R}}_1$ is formed by removing these two half-spaces and intersecting with Ω , leading to $\tilde{\mathcal{R}}_1 = \{x \mid -3 \leq x \leq 3\}$. However, $\tilde{\mathcal{R}}_1$ then intersects with an another unsaturated region \mathcal{R}_4 and with saturated regions \mathcal{R}_5 and \mathcal{R}_6 not weakly adjacent to \mathcal{R}_2 (i.e., $\mathcal{I}_{\text{adj},2} = \{1, 3\}$ and $\mathcal{I}_{\text{sat}} \setminus \mathcal{I}_{\text{adj},2} = \{5, 6\}$), violating P2–P4 of Def. 4.4 in these regions. Therefore $\tilde{\mathcal{R}}_1 = \tilde{\mathcal{R}}_1 \setminus (\mathcal{R}_4 \cup \mathcal{R}_5 \cup \mathcal{R}_6) = \{x \mid -3 \leq x \leq 0.5\}$ per Step 6 to recover the augmentation properties. Then the second unsaturated region \mathcal{R}_4 is explored in a similar fashion, i.e., $\tilde{\mathcal{R}}_2 = \{x \mid -3 \leq x \leq 3\}$ after Step 4 and $\tilde{\mathcal{R}}_2 = \tilde{\mathcal{R}}_2 \setminus (\mathcal{R}_2 \cup \mathcal{R}_1) = \{x \mid -0.5 \leq x \leq 3\}$

after Steps 5 and 6 where $\mathcal{I}_{\text{adj},4} = \{3, 5, 6\}$ and $\mathcal{I}_{\text{sat}} \setminus \mathcal{I}_{\text{adj},4} = \{1\}$ (note that \mathcal{R}_6 is weakly adjacent to \mathcal{R}_4 via \mathcal{R}_5). Finally, coverage of all saturated regions is checked on Step 10, revealing that $\mathcal{I}_{\text{nocover}} = \emptyset$, i.e., extended regions $\tilde{\mathcal{R}}_1$ and $\tilde{\mathcal{R}}_2$ fully cover all saturated regions and hence P1 of Def. 4.4 is fulfilled. $\tilde{\kappa}(x)$ is then represented just by two regions $\tilde{\mathcal{R}}_1$ and $\tilde{\mathcal{R}}_2$, shown in Fig. 1(b). One case where Step 10 would need to be executed is depicted on a 2–D scenario in Fig. 1(d). Here, the not fully covered saturated region needs to be included into $\tilde{\mathcal{R}}$ in order to achieve full coverage of Ω .

Remark 4.6: The adjacency list in Step 1 is usually automatically generated as a by-product of most pQP solvers, see e.g., [5]. Should it not be available at hand, it can be computed a-posteriori by the MPT Toolbox [12]. The index set $\mathcal{I}_{\text{adj},i}$ of saturated regions weakly adjacent to the i -th unsaturated region can be easily obtained by running a breath first search [8] on the graph representation of the adjacency list.

Remark 4.7: In theory [2], the set difference operation in Step 6 can produce exponentially many regions. Therefore Step 11 is formally needed to ensure that $\tilde{\kappa}(x)$ is no more complex than the original function $\kappa(x)$. We remark that we have never observed such a case, though.

Next, we provide a formal proof of correctness of Algorithm 1.

Theorem 4.8: Given a continuous PWA function $\kappa(x)$, Algorithm 1 constructs its suitable augmentation $\tilde{\kappa}(x)$ which fulfills all prerequisites of Definition 4.4.

Proof: First we show that $\cup_j \tilde{\mathcal{R}}_j = \cup_i \mathcal{R}_i = \Omega$, i.e., that P1 of Def. 4.4 holds. If \mathcal{R}_i has no saturated neighbors, then $\tilde{\mathcal{R}}_j = \mathcal{R}_i$. Otherwise by removing at least one half-space from \mathcal{R}_i in Step 4 we have $\tilde{\mathcal{R}}_j \supseteq \mathcal{R}_i$ and therefore $\cup_j \tilde{\mathcal{R}}_j \supseteq \mathcal{R}_{\mathcal{I}_{\text{unsat}}}$. Moreover, $\cup_j \tilde{\mathcal{R}}_j \supseteq \mathcal{R}_{\mathcal{I}_{\text{sat}}}$ follows directly from Step 10 where any uncovered part of $\mathcal{R}_{\mathcal{I}_{\text{sat}}}$ is added to $\tilde{\mathcal{R}}$. Therefore $\cup_j \tilde{\mathcal{R}}_j \supseteq \cup_i \mathcal{R}_i$. Since each extended region $\tilde{\mathcal{R}}_j$ is intersected with Ω in Step 4, we have that $\cup_j \tilde{\mathcal{R}}_j = \cup_j (\tilde{\mathcal{R}}_j \cap \Omega) = \Omega \cap (\cup_j \tilde{\mathcal{R}}_j) = \Omega$ and therefore $\cup_j \tilde{\mathcal{R}}_j = \cup_i \mathcal{R}_i = \Omega$. To prove that $\tilde{\kappa}(x) = \kappa(x)$ for all $x \in \mathcal{R}_{\mathcal{I}_{\text{unsat}}}$ it is enough to show that $\tilde{\mathcal{R}} \cap \mathcal{R}_{\mathcal{I}_{\text{unsat}}} = \emptyset$, i.e., that the extended regions $\tilde{\mathcal{R}}$ do not overlap with unsaturated regions. Due to Step 6, we have $(\tilde{\mathcal{R}} \setminus \mathcal{R}_{\mathcal{I}_{\text{unsat}}}) \cap \mathcal{R}_{\mathcal{I}_{\text{unsat}}} = \tilde{\mathcal{R}} \cap (\mathcal{R}_{\mathcal{I}_{\text{unsat}}} \setminus \mathcal{R}_{\mathcal{I}_{\text{unsat}}}) = \tilde{\mathcal{R}} \cap \emptyset = \emptyset$. Therefore, $\tilde{\kappa}(x)$ meets P2 of Def. 4.4. Finally, P3 and P4 follow directly from Step 6 since $\kappa(x)$ is assumed to be continuous. ■

Theorem 4.9: The number of regions \tilde{R} of the augmented function $\tilde{\kappa}(x)$ generated by Algorithm 1 is bounded by $R_{\text{unsat}} \leq \tilde{R} \leq R$.

Proof: The lower bound comes from two facts: (i) Algorithm 1 does not modify the number of unsaturated regions; and (ii) the saturated regions are replaced by “expansion” of unsaturated regions, therefore $\tilde{R} = R_{\text{unsat}}$ in the best case. However, additional regions might be added in Steps 6 and 10 and therefore $\tilde{R} \geq R_{\text{unsat}}$, in general. The upper bound follows directly from Step 11, cf. Remark 4.7. ■

Corollary 4.10: If Steps 6 and 10 are never invoked during the run of Algorithm 1, then $\tilde{\kappa}(x)$ is defined over $\tilde{R} = R_{\text{unsat}}$ regions.

Remark 4.11: Efficiency of the presented procedure, expressed as the ratio \tilde{R}/R , depends on the number of unsaturated regions. If $\kappa(x)$ does not contain any saturated regions, then no simplification can be achieved. As observed e.g., in [10], the number of unsaturated regions depends mainly on two factors: tightness of input constraints \mathcal{U} and selection of the input penalty Q_u in (1). The tighter the constraints and/or the lower Q_u is, the more regions will become saturated, hence enabling our approach to be more efficient.

Theorems 4.8 and 4.9 say that $\kappa(x)$ can be replaced by its suitable augmentation $\tilde{\kappa}(x)$ of (possibly) lower complexity in terms of number of regions. As will be documented in Section V, usually $\tilde{R} \ll R$ for the case of problems considered in this paper. The augmented function $\tilde{\kappa}(x)$, however, cannot be readily applied as an RHMP feedback since, in general, $\tilde{\kappa}(x) \neq \kappa(x)$ for $x \in \mathcal{R}_{\mathcal{I}_{\text{sat}}}$. The equivalence can be achieved by passing $\tilde{\kappa}(x)$ through a very simple clipping filter $\phi(\cdot)$, as noted by Theorem 4.12, which is the second main result.

Theorem 4.12: Consider a saturated continuous PWA function $\kappa(x)$ and its suitable augmentation $\tilde{\kappa}(x)$. Let

$$\phi(\tilde{\kappa}(x)) := \max\{\underline{\kappa}, \min\{\tilde{\kappa}(x), \bar{\kappa}\}\}. \quad (2)$$

Then the equivalence $\phi(\tilde{\kappa}(x)) = \kappa(x)$ is established for all $x \in \text{dom}(\kappa(x))$, and therefore $\phi(\tilde{\kappa}(x))$ solves Problem 4.1.

Proof: Notice that (2) is a compact encoding of three IF-THEN rules

$$\phi(\tilde{\kappa}(x)) = \begin{cases} \bar{\kappa} & \text{if } \tilde{\kappa}(x) \geq \bar{\kappa}, \\ \underline{\kappa} & \text{if } \tilde{\kappa}(x) \leq \underline{\kappa}, \\ \tilde{\kappa}(x) & \text{otherwise.} \end{cases} \quad (3)$$

Then we get $\phi(\tilde{\kappa}(x)) = \bar{\kappa}$ for all $x \in \mathcal{R}_{\mathcal{I}_{\max}}$ by P3 of Definition 4.4, $\phi(\tilde{\kappa}(x)) = \underline{\kappa}$ for all $x \in \mathcal{R}_{\mathcal{I}_{\min}}$ by P4, and $\phi(\tilde{\kappa}(x)) = \tilde{\kappa}(x) = \kappa(x)$ for all $x \in \mathcal{R}_{\mathcal{I}_{\text{unsat}}}$ by P2. Since $\text{dom}(\kappa(x)) = \mathcal{R}_{\mathcal{I}_{\max}} \cup \mathcal{R}_{\mathcal{I}_{\min}} \cup \mathcal{R}_{\mathcal{I}_{\text{unsat}}}$ by Theorem 3.1, it follows that $\phi(\tilde{\kappa}(x)) = \kappa(x)$ for all $x \in \text{dom}(\kappa(x))$. ■

B. Multi-Input Case

If $\kappa(x)$ is a vector-valued function (i.e., when $n_u > 1$), it can be decomposed to individual functions $\kappa_j(x) : \Omega \rightarrow \mathbb{R}, j = 1, \dots, n_u$, each defined over the original partition $\Omega = \{\mathcal{R}_i\}_{i=1}^R$. Therefore one obvious way of approaching this case is to process each subfunction $\kappa_j(x)$ individually by Algorithm 1. This will give rise to n_u sets of regions $\{\tilde{\mathcal{R}}_{j,i}\}_{i=1}^{\tilde{R}_j}, j = 1, \dots, n_u$, and n_u clipping filters $\phi_j(\tilde{\kappa}_j(x)) := \max\{\min\{\tilde{\kappa}_j(x), \bar{\kappa}_j\}, \underline{\kappa}_j\}$. Even though the total number of regions is then $\sum_{j=1}^{n_u} \tilde{R}_j$, significant reduction of complexity can still be achieved if $\tilde{R}_j \ll R$ for all $j \in [1, \dots, n_u]$.

Another option is to perform a direct simplification of $\kappa(x)$ as a vector-valued function. If $\kappa(x)$ contains regions in which the values of all subfunctions $\kappa_j(x)$ are *jointly* saturated at the same extrema (i.e., either $\kappa_j(x) = \bar{\kappa}_j(x)$ holds $\forall j$, or $\kappa_j(x) = \underline{\kappa}_j(x), \forall j$), then Algorithm 1 can be readily applied as is. The only difference being that $\underline{\kappa} := [\underline{\kappa}_1, \dots, \underline{\kappa}_{n_u}]^T$ and $\bar{\kappa} := [\bar{\kappa}_1, \dots, \bar{\kappa}_{n_u}]^T$ in the clipping filter $\phi(\cdot)$ are now vectors, and therefore the max and min operators in (2) have to be applied element-wise. It is also possible to extend Algorithm 1 to tackle cases where the subfunctions $\kappa_j(x)$ are saturated for all j , but not necessarily jointly at one type of extrema (e.g., $\kappa_1(x) = \bar{\kappa}_1, \kappa_2(x) = \underline{\kappa}_2$ for $n_u = 2$). Naturally, the joint-saturation scenario occurs more frequently if the input constraints \mathcal{U} are hyper-rectangular.

C. Implementation and Complexity

The original function $\kappa(x)$, composed of R polyhedral regions, requires $\mathcal{O}(R)$ memory for its storage. Evaluating $\kappa(x)$ for a given value of x is a two-stage process. First, the index a of the region which contains x is identified by a suitable *region traversal* procedure. Subsequently, the a -th elements K_a and L_a are extracted from memory and $\kappa(x) = K_a x + L_a$ is evaluated. Traditionally, regions are traversed sequentially, which requires $\mathcal{O}(R)$ floating point operations (FLOPS). This figure can be decreased by employing more advanced traversal approaches. The binary search tree (BST) approach [15] can traverse the regions in $\mathcal{O}(\log_2 R)$ time by building a suitable search tree. The downside being that the off-line construction of such a tree requires solving $\mathcal{O}(R^2)$ linear programs, which easily becomes prohibitive if R is large (say, $R > 2000$). Other alternatives include employing the lattice representation [17] with runtime complexity of $\mathcal{O}(R_{\text{unique}}^2)$ (where R_{unique} is the number of unique control laws), or suboptimal procedures like extrapolating the solution from closest regions [6], or omitting rarely visited regions [1].

With the proposed simplification technique, the memory footprint of $\tilde{\kappa}(x)$ is proportional to $\mathcal{O}(\tilde{R})$, a direct reduction by a factor of R/\tilde{R} .

TABLE I
RESULTS FOR THE F14 EXAMPLE

u_{\max}	Original feedback			Equivalent replacement			R/\tilde{R}
	R	FLOPS	Memory	\tilde{R}	FLOPS	Memory	
2	1167	49998	47500	170	8226	7705	6.9
4	1290	56292	53360	221	11442	10640	5.8
6	1391	61086	57860	289	14874	13840	4.8
8	1438	63096	59770	317	16188	15075	4.5
10	1507	66030	62560	364	18192	16980	4.1

We remark that $\tilde{R} = R_{\text{unsat}}$ in the best case and $\tilde{R} = R$ in the worst case. The extra memory required to store the filter $\phi(\cdot)$ is $2n_u$ floating point numbers (vectors $\bar{\kappa}$ and $\underline{\kappa}$), negligible compared to the memory footprint of polyhedral regions. On-line evaluation of $\phi(\tilde{\kappa}(x))$ for a given value of x can be done as described above with an additional step of passing the function value through the filter $\phi(\cdot)$, which always performs only $2n_u$ comparisons, insignificant compared to the complexity of region traversal. If sequential region traversal is used to evaluate $\tilde{\kappa}(x)$, then our procedure decreases the on-line evaluation effort in terms of FLOPS by a factor of R/\tilde{R} . If the BST approach is used, reduction is proportional to $\log_2(R/\tilde{R})$. Our approach can also be combined with the suboptimal traversal techniques described in the previous paragraph to further speed up the on-line implementation. Since $R \ll \tilde{R}$ in practice (as demonstrated in the next section), the proposed simplification technique can decrease both the evaluation time as well as required memory storage.

V. EXAMPLES

A. F14 Fighter Jet

We consider the open-loop unstable model of an F14 fighter jet in the lateral axis whose states represent the pitch and attack angles and the respective angular velocities, with the flap angle as control input

$$\dot{x} = \begin{bmatrix} -0.015 & -60.57 & 0 & -31.170 \\ 0.0001 & -1.341 & 0.993 & 0 \\ 0.0002 & 43.25 & -0.869 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x + \begin{bmatrix} 13.14 \\ -0.251 \\ -1.577 \\ 0 \end{bmatrix} u. \quad (4)$$

States are constrained by $|x_i| \leq 10, i = 1, \dots, 4$, and the control command is bounded by $|u| \leq u_{\max}$. Model (4) was discretized using sampling time 0.01 seconds. MPC problem (1) was formulated with $Q_x = \mathbf{I}$ and $Q_u = 1$ and prediction horizon $N = 15$. We have investigated how tightness of input constraints (represented by u_{\max}) impacts the number of unsaturated regions, which is the main factor that determines efficiency of the proposed scheme, cf. Remark 4.11. Therefore, we have computed explicit RHMPC feedbacks for $u_{\max} = \{2, 4, 6, 8, 10\}$. Each feedback $\kappa(x)$ was then processed by Algorithm 1 to obtain the replacement function $\tilde{\kappa}(x)$. Results are summarized in Table I. Columns of the table report, respectively, maximal control amplitude u_{\max} , complexity of the original feedback $\kappa(x)$ (in terms of number of regions, worst-case evaluation effort¹ in FLOPS and memory footprint² in floating point numbers), complexity of the replacement function $\tilde{\kappa}(x)$, and the complexity reduction ratio. In all cases we got $\tilde{R} = R_{\text{unsat}}$, cf. Corollary 4.10. As expected, the on-line evaluation speed and required memory storage both improve proportionally to R/\tilde{R} . This fraction decreases when the input constraints become less strict (cf. Remark 4.11).

¹Number of floating point operations needed to evaluate the PWA function using sequential search.

²Amount of floating point numbers needed to represent polyhedral regions $\mathcal{R}_i = \{x \mid R_i^x \leq R_i^0\}$.

TABLE II
RESULTS FOR RANDOM SYSTEMS WITH $n_u = 1$

n_x	R	R_{unsat}	# of regions		Runtime [sec]	
			Alg. 1	ORM [9]	Alg. 1	ORM [9]
2	319	59	59	89	1.4	112.4
	401	19	19	37	0.3	208.0
	495	35	51	67	1.1	294.1
3	527	85	91	205	8.0	2838.4
	773	113	113	†	13.3	†
	1241	303	303	†	146.3	†
4	523	47	47	125	2.3	4900.3
	2513	387	387	†	197.6	†
	5643	657	657	†	462.2	†

TABLE III
RESULTS FOR RANDOM SYSTEMS WITH $n_u = 2$

n_x	R	R_{unsat}	# of regions		Runtime [sec]	
			Alg. 1	ORM [9]	Alg. 1	ORM [9]
2	117	42	42	55	0.9	69.9
	365	30	30	91	2.6	346.5
	787	118	118	248	9.3	561.6
3	309	106	106	79	9.3	1600.0
	961	134	134	†	23.1	†
	1759	402	420	†	213.7	†
4	259	174	174	154	7.4	1259.4
	8785	1838	1924	†	2160.5	†
	12651	2338	2338	†	3478.2	†

TABLE IV
AGGREGATED RESULTS FOR 600 RANDOM PROBLEMS

n_x/n_u	Complexity reduction ratio R/\tilde{R}		% of cases with $\tilde{R} = R_{\text{unsat}}$
	Algorithm 1	ORM [9]	
2/1	13.1	5.2	97.5 %
3/1	8.5	3.8	83.9 %
4/1	7.0	1.8	81.0 %
2/2	6.9	2.9	95.2 %
3/2	4.5	1.9	71.7 %
4/2	3.9	1.5	59.8 %

B. Random Systems

To assess how Algorithm 1 scales with increasing problem size and how it compares to the optimal region merging (ORM) method of [9], we have analyzed random LTI systems with two to four states, and one to two inputs, subject to constraints $\mathcal{X} = \{x | -10 \leq x \leq 10\}$ and $\mathcal{U} = \{u | -1 \leq u \leq 1\}$. Both open-loop stable and unstable systems were considered. For each random system we have then solved the MPC problem (1) parametrically with $Q_x = \mathbf{1}$, $Q_u = \mathbf{1}$, $N = 10$ using the MPT Toolbox [12]. Each resulting PWA solution $u^*(x) = \kappa(x)$ was subsequently post-processed independently by Algorithm 1 and by ORM. A total of 600 random systems was considered, with 100 for each n_x/n_u combination. A representative selection of obtained results is shown in Tables II–III. Entries marked with † denote cases where the ORM approach failed to converge within of 12 h of computation. We remark that in all 600 investigated cases a perfect coverage of all saturated regions was achieved, i.e., Step 10 of Alg. 1 never had to be executed. Runtime of the coverage check in Step 9 attributes to around 75% of the total runtime reported in the tables.

The average complexity reduction ratios R/\tilde{R} over the complete set of random problems are reported in Table IV. We remind that the reduction ratio has two direct implications: it shows how much memory can be saved by employing the replacement $\tilde{\kappa}(x)$ instead of the original feedback $\kappa(x)$, and how much faster can $\phi(\tilde{\kappa}(x))$ be evaluated on-line. The table also reports likelihood of Algorithm 1 achieving the theoretical lower bound of complexity with $\tilde{R} = R_{\text{unsat}}$, cf. Corollary 4.10.

The presented results clearly show that the clipping-based procedure scales significantly better with increasing problem size than the ORM procedure. Algorithm 1 was able to devise a simpler representation of the original PWA function even when $\kappa(x)$ was very complex and defined over several thousands of regions. The ORM procedure, on the other hand, is limited to situations with few hundreds of regions. In addition, the complexity reduction ratio is significantly higher for Algorithm 1, which follows from the fact that saturated regions are removed completely from the function description. The ORM approach, on the other hand, keeps the saturated regions and merely tries to merge them into larger convex objects.

Results for the multi-input case with $n_u = 2$ shown in Table III were obtained by the scalarization-based procedure discussed in Section IV-C. This approach turned out to be more efficient compared to treating $\kappa(x)$ as a single vector-valued function. Specifically, in 80% of the 600 investigated problems scalarization was significantly more efficient than the vector case, leading to, on average, 3.5 times less regions compared to the vector approach. In the remaining 20%, scalarization performed only slightly worse than the vector method. In particular, the only case from Table III where scalarization has done worse than the vector approach was the first instance, where scalarization gives 42 regions, whereas vectorization leads to 39 regions. Although the achievable complexity reduction naturally decreases with increasing dimension of the input space, the presented procedure still provides higher complexity reduction ratio compared to the ORM method at significantly lower computational cost.

VI. CONCLUSIONS

In this paper we have shown how to reduce complexity of explicit RHMPC feedback laws which contain saturated regions. Given a RHMPC function $\kappa(x)$, the procedure constructs its simpler replacement $\tilde{\kappa}(x)$ using basic polyhedral operations. Regions where the control action attains a saturated value are removed and replaced by extensions of unsaturated regions. As a consequence, the implementation complexity of the replacement is substantially reduced compared to employing the original explicit feedback $\kappa(x)$. Specifically, the memory footprint of $\tilde{\kappa}(x)$ and its evaluation time are improved by a factor of R/\tilde{R} , which often exceeds one order of magnitude. Evaluation speed can be further increased by devising advanced region traversal strategies, as discussed in Section IV-D. In addition, it was illustrated that construction of the replacement scales well with increasing problem size and that it is significantly faster compared to using the ORM procedure.

REFERENCES

- [1] A. Alessandro and A. Bemporad, "A survey on explicit model predictive control," in *Nonlinear Model Predictive Control*. Berlin/Heidelberg, Germany: Springer, 2009, vol. 384, Lecture Notes in Control and Information Sciences, pp. 345–369.
- [2] M. Baotic, "Polytopic computations in constrained optimal control," *Automatika, J. Control, Measur., Electron., Comput. and Commun.*, vol. 50, pp. 119–134, 2009.
- [3] M. Baotic, F. Borrelli, A. Bemporad, and M. Morari, "Efficient on-line computation of constrained optimal control," *SIAM J. Control and Optimiz.*, vol. 47, no. 5, pp. 2470–2489, Sep. 2008.
- [4] A. Bemporad and C. Filippi, "Suboptimal explicit RHC via approximate multiparametric quadratic programming," *J. Optimiz. Theory and Applic.*, vol. 117, no. 1, pp. 9–38, Apr. 2003.
- [5] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, Jan. 2002.
- [6] F. J. Christophersen, M. N. Zeilinger, C. N. Jones, and M. Morari, "Controller complexity reduction for piecewise affine systems through safe region elimination," in *Proc. Conf. Decision & Control*, New Orleans, LA, Dec. 2007.

- [7] M. Cychowski and T. O'Mahony, "Efficient off-line solutions to robust model predictive control using orthogonal partitioning," in *Proc. 16th IFAC World Congr.*, Prague, Czech Republic, Jul. 2005.
- [8] S. Dasgupta, C. Papadimitriou, and U. Vazirani, *Algorithms*, 1 ed. New York: McGraw-Hill Science/Engineering/Math, 2006.
- [9] T. Geyer, F. D. Torrisi, and M. Morari, "Optimal complexity reduction of polyhedral piecewise affine systems," *Automatica*, vol. 44, no. 7, pp. 1728–1740, Jul. 2008.
- [10] P. Grieder and M. Morari, "Complexity reduction of receding horizon control," in *Proc. IEEE Conf. Decision and Control*, Maui, HI, Dec. 2003, pp. 3179–3184.
- [11] M. Kvasnica and M. Fikar, "Performance-lossless complexity reduction in explicit MPC," in *Proc. 49th IEEE Conf.*, 2010, pp. 5270–5275.
- [12] M. Kvasnica, P. Grieder, and M. Baotić, Multi-Parametric Toolbox (MPT) 2004 [Online]. Available: <http://control.ee.ethz.ch/~mpt/>
- [13] D. Limon, J. Gomes da Silva, T. Alamo, and E. Camacho, "Improved MPC design based on saturating control laws," *Eur. J. Control*, vol. 11, no. 2, pp. 112–122, 2005.
- [14] F. Scibilia, S. Oлару, and M. Hovd, "Approximate explicit linear MPC via delaunay tessellation," in *Proc. 10th European Control Conf.*, Budapest, Hungary, 2009.
- [15] P. Tøndel, T. A. Johansen, and A. Bemporad, "Evaluation of piecewise affine control via binary search tree," *Automatica*, vol. 39, no. 5, pp. 945–950, May 2003.
- [16] A. Ulbig, S. Oлару, D. Dumur, and P. Boucher, "Explicit solutions for nonlinear model predictive control: A linear mapping approach," in *Proc. European Control Conf. 2007*, 2007, pp. 3295–3302.
- [17] C. Wen, X. Ma, and B. E. Ydstie, "Analytical expression of explicit MPC solution via lattice piecewise-affine function," *Automatica*, vol. 45, no. 4, pp. 910–917, 2009.

Control Design for Quantized Linear Systems With Saturations

Sophie Tarbouriech and Frédéric Gouaisbaut

Abstract—This paper deals with systems involving input saturation and quantized control laws, which can be of two types: input quantization case and state quantization case. In both cases, the state feedback control design problem is addressed. Therefore, based on some modified sector conditions and appropriate variable changes, regional (local) uniform ultimate boundedness stabilization problem is tackled. Computational oriented solutions are derived to solve suboptimal convex optimization problems able to give a constructive solution to the design problem of the state feedback gain.

Index Terms—Quantized control law, regional uniform ultimate boundedness stability, saturation.

I. INTRODUCTION

Many physical control systems are subject to magnitude limitation in the input. This type of nonlinearity may reduce the performance of the closed-loop system or even lead to instability. Therefore the stability analysis or stabilization problems of control systems with saturation in the input attracted research efforts for several decades (see, e.g., [14],

[17], [25]). Furthermore, another important feature resides in the fact that the output of the plant can be injected to the controller through a quantizer, which may lead to limit cycles and chaotic behavior even if the controller is a stabilizing one [3], [12], [15]. It is one of the reasons why quantization in control systems has recently become an active research topic. Actually, quantization can arise when digital networks or control with limited information are part of the feedback loop or when poor capabilities sensor and actuators are involved [6], [28]. Emerging control theory framework has been developed pointing out how limited information (as due to quantizer) can affect the performance of the resulting closed-loop system and therefore what type of solution can be investigated [1], [20]. Thus, some tools issued from robust control theory can be adapted to deal with stability and performance purposes, as H_∞ performance or L_2 stability, [9], [10], [16]. In this robust framework, discrete-time systems with input quantization but without saturation have been considered in [23], [24]. The case of logarithmic quantizer has been addressed in a global context by using absolute stability in [29], or in a regional context in [9] by using quadratic Lyapunov arguments. Results addressing systems with uncertainties and/or with delays have been published: see, for example, [5], [8], [27] and references therein. In [2], planar systems of a particular structure with single input and single output are considered. Differently from the current paper, the quantization appears on the output and a globally practically stabilizing output feedback control is designed.

The current paper deals with control design problem for systems involving both input saturation and quantized control law. In this setup, the quantizer is chosen to be uniform [9], [22]. Two quantization cases are considered: input quantization case and state quantization case. In both cases, the state feedback control design problem is addressed by using some modified sector conditions and appropriate variable changes. The regional (local) uniform ultimate boundedness stabilization [18] of the system is then carried out. Hence, the proposed approach allows to characterize both an inner and an outer set such that the closed-loop trajectories initiated in the outer set converge toward the inner set. It is important to emphasize that the technique proposed does not require the open-loop system to be stable. Synthesis conditions in a quasi-linear matrix inequality (quasi-LMI) form are stated in a regional (local) stability context. The objective of the related LMI-based optimization problems is then to maximize a measure of the size of the outer set (stability domain), whereas a measure of the inner set is minimized. Depending on the open-loop stability, the global stability context is also carried out. In this case, the outer set corresponds to the whole state space and the inner one reduces to the origin.

Our work is related to the work of Liberzon [20], where such an approach is proposed in a nonlinear context. Except the nonlinear nominal system, the major difference is the way to deal with the different nonlinearities involved. In [20], the saturation and quantizer blocks are in fact a one and only block, the saturation being a particular effect of the quantizer. In our setup, the two nonlinearities are disjointed, allowing to characterize them precisely. Furthermore, we can also deal with state quantization case where the control gain is in sandwich between the saturation and the quantizer. Moreover, the contribution of the paper can be viewed also as complementary to the results developed in [8], even if in our case the way chosen to deal with the nonlinearities issued from saturation and quantizer is based on the use of modified sector conditions and not on LDI (Linear Differential Inclusion) and associated polytopic representation tools as in [14]. This tool was chosen preferably to LDI tools [25], [26] since the numerical complexity of the conditions increases more slowly than that one associated to LDI tools with respect to the dimension of the conditions. Finally, the technique developed in the paper provides constructive conditions, in the

Manuscript received October 05, 2010; revised May 24, 2011; accepted November 24, 2011. Date of publication December 14, 2011; date of current version June 22, 2012. Recommended by Associate Editor H. Ishii.

The authors are with CNRS, LAAS, F-31077 Toulouse, France and are also with the Université de Toulouse, LAAS, F-31077 Toulouse, France.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TAC.2011.2179845

Parallel MPC for Linear Systems with Input Constraints

Yuning Jiang, Juraj Oravec*, Boris Houska, and Michal Kvasnica

Abstract—This paper is about a real-time model predictive control (MPC) algorithm for large-scale, structured linear systems with polytopic control constraints. The proposed controller receives the current state measurement as an input and computes a sub-optimal control reaction by evaluating a finite number of piecewise affine functions that correspond to the explicit solution maps of small-scale parametric quadratic programming (QP) problems. We provide asymptotic stability guarantees, which can be verified offline. The feedback controller is computing approximations of the optimal input, because we are enforcing real-time requirements assuming that it is not possible to solve the given large-scale QP in the given amount of time. Here, a key contribution of this paper is that we provide a bound on the sub-optimality of the controller. The approach is illustrated by benchmark case studies.

Index Terms—Model Predictive Control, Parametric Optimization.

I. INTRODUCTION

The advances of numerical optimization methods over the last decades [1], in particular, the development of efficient quadratic programming problem (QP) solvers [2], have enabled numerous industrial applications of MPC [3]. Modern real-time optimization and control software packages [4], [5] achieve run-times in the milli- and microsecond range by generating efficient and reliable C-code [6], [7]. However, as much as these algorithms perform well on desktop computers or other devices with comparable computation power, the number of successful implementations of MPC on embedded industrial hardware, such as programmable logic controllers (PLC) and field-programmable gate arrays (FPGA), remains limited [8]. Here, the main question is what can be done if an embedded device has not enough computational power or storage space to solve the exact MPC problem in real-time.

Many researchers have attempted to address this question. For example, the development of Explicit MPC [9] aims at reducing both the online run-time and the memory footprint of MPC by optimizing pre-computed solution maps of multi-parametric optimization problems. However, Explicit MPC has the disadvantage that the number of polytopic regions over which the piecewise affine solution map of a parametric quadratic program is defined, grows, in the worst case, exponentially with the number of constraints. Some authors [10] have suggested addressing this issue by simplifying the MPC problem formulation by using move-blocking [11], but the associated control reactions can be sub-optimal by a large

margin. Other authors [12] have worked on reducing the memory footprint of Explicit MPC—certainly making considerable progress yet failing to meet the requirement of many practical systems with more than just a few states. In fact, despite all these developments in Explicit MPC, these methods are often applicable to problems of modest size only. As soon as one attempts to scale up to larger systems, Explicit MPC is often outperformed by iterative online solvers such as active set [2] or interior-point methods [5].

A recent trend in optimization-based control is to solve large MPC problems by breaking them into smaller ones. This trend has been initiated by research on distributed optimization [13]. For example, dual decomposition [14], ADMM [13], and ALADIN [15] have been applied to MPC in various contexts and by many authors [16], [17], [18], [19], [20]. Additionally, applications of accelerated variants of ADMM to MPC can be found in [21], [22]. However, modern distributed optimization methods, such as ADMM or ALADIN, typically converge to an optimal solution in the limit, if the number of iterations tends to infinity. Thus, if real-time constraints are present, one could at most implement a finite number of such ADMM or ALADIN iterations returning a control input that may be infeasible or sub-optimal by a large margin.

Therefore, this paper asks the question whether it is possible to approximate MPC feedback laws by evaluating a constant, finite number of pre-computed, explicit solution maps that are associated to MPC problems of a smaller scale. Here, a key requirement is that uniform asymptotic stability and performance guarantees of the implemented closed-loop controller have to be verifiable offline. The contribution of this paper is the development of a controller, which meets this requirement under the restricting assumption that the original MPC problem is a strongly convex QP, as introduced in Section II. The control scheme itself is presented in the form of Algorithm 1 in Section III. This algorithm alternates between solving explicit solution maps that are associated with small-scale decoupled QPs and solving a linear equation system of a larger scale. However, in contrast to ALADIN, ADMM or other existing distributed optimization algorithms, Algorithm 1 performs a constant number of iterations per sampling time.

The stability and performance properties of Algorithm 1, which represent the main added value compared to our preliminary work [23], are summarized in Sections III-C, III-D, and III-E, respectively. Instead of relying on existing analysis concepts from the field of distributed optimization, the mathematical developments in this paper rely on results that find their origin in Explicit MPC theory [24]. In particular, the technical developments around Theorem 1 make use of the solution properties of multi-parametric QPs in order to derive convergence rate estimates for Algorithm 1. Moreover, Theorem 2 establishes an asymptotic stability guarantee of

*Corresponding author.

Yuning Jiang and Boris Houska are with School of Information Science and Technology, ShanghaiTech University, China. (e-mail: [jiangyn, borish]@shanghaitech.edu.cn).

Juraj Oravec and Michal Kvasnica are with Institute of Information Engineering, Automation, and Mathematics, Faculty of Chemical and Food Technology, Slovak University of Technology in Bratislava, Slovakia. (e-mail: [juraj.oravec, michal.kvasnica]@stuba.sk).

the presented real-time closed-loop scheme. This result is complemented by Corollary 1, which provides bounds on the sub-optimality of the presented control scheme. Finally, Section IV-A discusses implementation details with a particular emphasis on computational and storage complexity exploiting the fact that the presented scheme can be realized by using static memory only while ensuring a constant run-time, as illustrated by numerical case studies.

II. LINEAR-QUADRATIC MPC

This paper concerns discrete-time MPC problems,

$$J(x_0) = \min_{x,u} \mathcal{M}(x_N) + \sum_{k=0}^{N-1} \ell(x_k, u_k) \quad (1)$$

$$\text{s.t.} \quad \begin{cases} \forall k \in \{0, \dots, N-1\}, \\ x_{k+1} = Ax_k + Bu_k, \\ u_k \in \mathbb{U}, \end{cases}$$

with strictly convex quadratic stage and terminal cost,

$$\ell(x, u) = x^\top Qx + u^\top Ru \quad \text{and} \quad \mathcal{M}(x) = x^\top Px.$$

Here, $x_k \in \mathbb{R}^{n_x}$ denotes the state at time k and $u_k \in \mathbb{R}^{n_u}$ the associated control input assuming that the current time of the MPC controller is set to 0. The matrices $A, P, Q \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$, $R \in \mathbb{R}^{n_u \times n_u}$ are given and constant. Notice that (1) is a parametric optimization problem with respect to the current state measurement x_0 . The optimization variable $x = [x_1^\top, x_2^\top, \dots, x_N^\top]^\top$ includes all but the first element of the state sequence and the control sequence $u = [u_0^\top, u_1^\top, \dots, u_{N-1}^\top]^\top$ is defined accordingly.

Assumption 1 *We assume that*

- the control constraint set $\mathbb{U} \subseteq \mathbb{R}^{n_u}$ is a closed and convex polyhedron satisfying $0 \in \mathbb{U}$;*
- the matrices $Q, R,$ and P are all symmetric and positive definite.*

Assumptions 1a) and 1b) imply strong convexity such that the primal solution of (1) is unique whenever it exists.

A. Asymptotic stability

Notice that the stability properties of MPC controllers have been analyzed exhaustively [25]. In this context, a standard assumption can be formulated as follows.

Assumption 2 *The terminal cost \mathcal{M} in (1) admits a control law $\mu : \mathbb{R}^{n_x} \rightarrow \mathbb{U}$ such that for all $x \in \mathbb{R}^{n_x}$*

$$\ell(x, \mu(x)) + \mathcal{M}(Ax + B\mu(x)) \leq \mathcal{M}(x).$$

The MPC controller (1) is asymptotically stable if Assumptions 1 and 2 hold [25].

III. SUBOPTIMAL REAL-TIME MPC

In this section we propose and analyze a real-time algorithm for finding approximate solutions of (1).

A. Preliminaries

Let us introduce the vectors $y_0 = u_0$, $y_k = [x_k^\top \quad u_k^\top]^\top$, $y_N = x_N$, and their associated constraint sets

$$\mathbb{Y}_0 = \mathbb{U} \quad \text{and} \quad \mathbb{Y}_k = \{y \in \mathbb{R}^{n_u+n_x} \mid [0 \ I]y \in \mathbb{U}\} \quad (2)$$

for all $k \in \{1, \dots, N-1\}$. Moreover, we introduce

$$F_k(y_k) = \ell(x_k, u_k), \quad F_N(y_N) = \mathcal{M}(x_N), \quad (3)$$

for $k \in \{1, \dots, N-1\}$ and matrices

$$H_0 = B, \quad H_k = [A \ B], \quad G_k = [I \ 0], \quad G_N = I,$$

as well as $h_0 = Ax_0$, $h_k = 0$ for all $k \in \{1, \dots, N-1\}$. Now, (1) can be written in the form

$$J(x_0) = \min_y \sum_{k=0}^N F_k(y_k) \quad (4)$$

$$\text{s.t.} \quad \begin{cases} \forall k \in \{0, \dots, N-1\}, \\ G_{k+1}y_{k+1} = H_k y_k + h_k \quad | \quad \lambda_k, \\ y_k \in \mathbb{Y}_k. \end{cases}$$

The notation “ $| \lambda_k$ ” behind the affine constraints in the above optimization problems indicates that λ_k denotes their associated multipliers. It is helpful to keep in mind that both the function F_0 and the vector h_0 depend on x_0 . In addition, we introduce a shorthand for the objective in (4) and its convex conjugate function,

$$F(y) = \sum_{k=0}^N F_k(y_k), \quad F^*(\lambda) = \max_y \{-F(y) + \langle \lambda, y \rangle\},$$

where the shorthand notation

$$\langle \lambda, y \rangle = -(H_0^\top \lambda_0)^\top y_0 + \sum_{k=1}^N (G_k^\top \lambda_{k-1} - H_k^\top \lambda_k)^\top y_k + \lambda_{N-1}^\top G_N^\top y_N$$

is used to denote a weighted (non-symmetric) scalar product of primal and dual variables. Notice that the functions F and F^* are strongly convex quadratic forms with $F(0) = 0$ and $F^*(0) = 0$ as long as Assumption 1 is satisfied. The optimal primal and dual solutions of (4) are denoted by x^* and λ^* , respectively. It is well-known that x^* and λ^* are continuous and piecewise affine functions of x_0 , see [26].

B. Algorithm

The main idea for solving (4) approximately and in real time is to consider the auxiliary optimization problem

$$J(x_0) = \min_y \sum_{k=0}^N F_k(y_k - y_k^{\text{ref}}) \quad (5)$$

$$\text{s.t.} \quad \begin{cases} \forall k \in \{0, \dots, N-1\}, \\ G_{k+1}y_{k+1} = H_k y_k + h_k \quad | \quad \lambda_k, \end{cases}$$

with reference trajectory y^{ref} . If $y^{\text{ref}} = y^*$ is equal to the minimizer of (4), then y^* is a minimizer of (5). Notice that the main motivation for introducing the coupled QP (5) is that this problem approximates (4) without needing inequality

constraints. Thus, this problem can be solved by using a sparse linear algebra solver.

Let us assume that y^m and λ^m are the current approximations of the primal and dual solution of (4). Algorithm 1 constructs the next iterate y^{m+1} and λ^{m+1} by performing two main operations. First, we solve augmented Lagrangian optimization problems of the form

$$\min_{\xi^m \in \mathbb{Y}} F(\xi^m) + \langle \lambda, y \rangle + F(\xi^m - y^m). \quad (6)$$

with $\mathbb{Y} = \mathbb{Y}_0 \times \dots \times \mathbb{Y}_{N-1} \times \mathbb{R}^{n_x}$. Problem (6) can be solved in parallel, see Step 2.a) of Algorithm 1. In the following, we set $Q = \frac{1}{2} \nabla^2 F(0)$ such that $\|\xi^m - y^m\|_Q^2 = F(\xi^m - y^m)$ recalling that F is a centered positive-definite quadratic form. And second, we solve QP (5) for the reference point $y^{\text{ref}} = 2\xi^m - y^m$ without considering the input constraints. These two main steps correspond to Step 2a) and Step 2b) in Algorithm 1.

Algorithm 1 Parallel real-time MPC

Initialization:

- Choose $y^1 = [y_0^1, \dots, y_N^1]^\top$, $\lambda^1 = [\lambda_0^1, \dots, \lambda_{N-1}^1]^\top$, a constant $\gamma > 0$, and a maximum number \bar{m} of iterations per sampling time.

Online:

- Wait for the state measurement x_0 and compute the constant

$$f^1 = F(y^1) + F^*(\lambda^1).$$

If $f^1 \geq \gamma^2 x_0^\top Q x_0$, rescale

$$y^1 \leftarrow y^1 \sqrt{\frac{\gamma^2 \|x_0\|_Q^2}{f^1}} \quad \text{and} \quad \lambda^1 \leftarrow \lambda^1 \sqrt{\frac{\gamma^2 \|x_0\|_Q^2}{f^1}},$$

where $\|x_0\|_Q^2 \triangleq x_0^\top Q x_0$.

- For $m = 1 \rightarrow \bar{m}$

a) solve the small-scale decoupled QPs in parallel

$$\begin{aligned} \min_{\xi_0^m \in \mathbb{Y}_0} & F_0(\xi_0^m) - (H_0^\top \lambda_0^m)^\top \xi_0^m + F_0(\xi_0^m - y_0^m) \\ \min_{\xi_k^m \in \mathbb{Y}_k} & F_k(\xi_k^m) + (G_k^\top \lambda_{k-1}^m - H_k^\top \lambda_k^m)^\top \xi_k^m + F_k(\xi_k^m - y_k^m) \\ \min_{\xi_N^m} & F_N(\xi_N^m) + (G_N^\top \lambda_{N-1}^m)^\top \xi_N^m + F_N(\xi_N^m - y_N^m) \end{aligned}$$

for all $k \in \{1, \dots, N-1\}$ and denote solutions by $\xi^m = [\xi_0^m, \xi_1^m, \dots, \xi_N^m]$.

b) Solve the coupled QP

$$\begin{aligned} \min_{y^{m+1}} & \sum_{k=0}^N F_k(y_k^{m+1} - 2\xi_k^m + y_k^m) \\ \text{s.t.} & \begin{cases} \forall k \in \{0, \dots, N-1\}, \\ G_{k+1} y_{k+1}^{m+1} = H_k y_k^{m+1} + h_k & | \delta_k^m, \end{cases} \end{aligned} \quad (7)$$

and set $\lambda^{m+1} = \lambda^m + \delta^m$.

End

- Send $u_0 = \xi_0^{\bar{m}}$ to the real process.
 - Set $y^1 = [y_1^{\bar{m}}, \dots, y_N^{\bar{m}}, 0]^\top$, $\lambda^1 = [\lambda_1^{\bar{m}}, \dots, \lambda_{N-1}^{\bar{m}}, 0]^\top$, go to Step 1.
-

Algorithm 1 is initialized with guesses,

$$y^1 = [y_0^1, \dots, y_N^1]^\top \quad \text{and} \quad \lambda^1 = [\lambda_0^1, \dots, \lambda_N^1]^\top,$$

for the primal and dual solution of (4) offline. Notice that Algorithm 1 receives a state measurement x_0 in every iteration (Step 1) and returns a control input to the real process (Step 3). Similar to the classical real-time MPC scheme [6], or related warm-start techniques [27], Step 4) shifts primal and dual variables y^1 and λ^1 , which are, however, rescaled in Step 1), based on a tuning parameter $\gamma > 0$.

Assumption 3 The constant γ in Algorithm 1 is such that

$$F(y^*) + F^*(\lambda^*) \leq \gamma^2 x_0^\top Q x_0.$$

Notice that such a bound γ exists and can be computed offline, because y^* and λ^* are Lipschitz continuous and piecewise affine functions of x_0 [26]. Notice that the choice $\gamma = \infty$ would mean that the variables are never rescaled. In this case, Algorithm 1 is unstable in general. In order to see this, consider the scenario that a user initializes the algorithm with an arbitrary $(y^1, \lambda^1) \neq 0$. Now, if the first measurement happens to be at $x_0 = 0$, the optimal control input is at $u^* = 0$. But, if we run Algorithm 1 with $\bar{m} < \infty$, it returns an approximation $u_0 \approx u^* = 0$, which will introduce an excitation as we have $u_0 \neq 0$ in general. Thus, if we would not rescale the initialization in Step 1), it would be impossible to establish stability.

C. Convergence properties of Algorithm 1

This section provides a concise overview of the theoretical convergence properties of Algorithm 1. Here, we initially focus on establishing conditions for convergence of the iterates of this algorithm (Lemma 1), which are then, in a second step, used to establish a linear convergence rate estimate (Theorem 1).

Lemma 1 Let Assumption 1 be satisfied and let (4) be feasible, such that a unique minimizer y^* and an associated dual solution λ^* exist. Then the iterates of Algorithm 1 satisfy

$$\sum_{m=\hat{m}}^{\bar{m}} F(\xi^m - y^*) \leq \frac{F(y^{\hat{m}} - y^*) + F^*(\lambda^{\hat{m}} - \lambda^*)}{4}$$

for all $\bar{m} \geq \hat{m}$ and all $\hat{m} \geq 2$.

Notice that the statement of Lemma 1 is useful in the sense that an immediate consequence of this statement is that the iterates of Algorithm 1 would converge to the exact solution of (4), if we would set $\bar{m} = \infty$, i.e.,

$$\lim_{m \rightarrow \infty} \xi^m = y^* \quad \text{and} \quad \lim_{m \rightarrow \infty} \lambda^m = \lambda^*.$$

The proof of the above lemma is technical but important for the developments in this paper:

Proof. Let us introduce the auxiliary functions

$$\mathcal{F}_0(\phi_0) = F_0(\phi_0) - (H_0^\top \lambda_0^m)^\top \phi_0^m + \nabla F_0(\xi_0^m - y_0^m)^\top \phi_0,$$

$$\begin{aligned} \mathcal{F}_k(\phi_k) &= F_k(\phi_k) + (G_k^\top \lambda_{k-1}^m - H_k^\top \lambda_k^m)^\top \phi_k^m \\ &\quad + \nabla F_k(\xi_k^m - y_k^m)^\top \phi_k, \end{aligned}$$

$$\mathcal{F}_N(\phi_N) = F_N(\phi_N) + G_N^\top \lambda_{N-1}^m \phi_N^m + \nabla F_N(\xi_N^m - y_N^m)^\top \phi_N.$$

Because ξ_k^m is a minimizer of the k -th decoupled QP in Step 2a) of Algorithm 1, it must also be a minimizer of \mathcal{F}_k on \mathbb{Y}_k . Thus, because \mathcal{F}_k is strongly convex with Hessian $\nabla^2 \mathcal{F}_k$, we must have

$$\sum_{k=0}^N \mathcal{F}_k(\xi_k^m) + \sum_{k=0}^N F_k(\xi_k^m - y_k^*) \leq \sum_{k=0}^N \mathcal{F}_k(y_k^*).$$

On the other hand, due to duality, we have

$$\begin{aligned} & \sum_{k=0}^N F_k(y_k^*) + \langle \lambda^*, y^* \rangle + \sum_{k=0}^N F_k(\xi_k^m - y_k^*) \\ & \leq \sum_{k=0}^N F_k(\xi_k^m) + \langle \lambda^*, \xi^m \rangle. \end{aligned}$$

Adding both inequalities and collecting terms yields

$$\begin{aligned} 0 & \geq \sum_{k=0}^N \nabla F_k(\xi_k^m - y_k^m)^\top (\xi_k^m - y_k^m) + 2 \sum_{k=0}^N F_k(\xi_k^m - y_k^m) \\ & \quad + \langle \lambda^m - \lambda^*, \xi^m - y^* \rangle \\ & = (\xi^m - y^m)^\top \mathcal{Q}(\xi^m - y^*) + 2 \sum_{k=0}^N F_k(\xi_k^m - y_k^m) \\ & \quad + (\lambda^m - \lambda^*)^\top \mathcal{A}(\xi^m - y^*) \end{aligned} \quad (8)$$

with $\mathcal{A} = \nabla_{\lambda, x} \langle \lambda, x \rangle$. Similarly, the stationarity condition QP (7) can be written as

$$\mathcal{Q}(y^{m+1} - 2\xi^m + y^m) + \mathcal{A}^\top \delta^m = 0.$$

Because \mathcal{Q} is positive definite, we solve this equation with respect to ξ^m finding

$$\xi^m = \frac{1}{2} \mathcal{Q}^{-1} \mathcal{A}^\top (\lambda^{m+1} - \lambda^m) + \frac{y^m + y^{m+1}}{2}. \quad (9)$$

Here, we have additionally substituted the relation

$$\delta^m = \lambda^{m+1} - \lambda^m.$$

Notice that we have $\mathcal{A}y^m = \mathcal{A}y^{m+1} = \mathcal{A}y^*$ for all $m \geq 2$, because the solutions of the QP (7) must satisfy the equality constraints in (4). If we substitute this equation and the expression for ξ^m in (8), we find that

$$\begin{aligned} & -2F(\xi^m - y^*) \\ & \geq (\xi^m - y^m)^\top \mathcal{Q}(\xi^m - y^*) + (\lambda^m - \lambda^*)^\top \mathcal{A}(\xi^m - y^*) \\ & = \frac{1}{4} (\lambda^{m+1} - \lambda^m)^\top \mathcal{A} \mathcal{Q}^{-1} \mathcal{A}^\top (\lambda^{m+1} - \lambda^m) \\ & \quad + \frac{1}{4} (y^{m+1} - y^m) \mathcal{Q} (y^m - 2y^* + y^{m+1}) \\ & \quad + \frac{1}{2} (\lambda^m - \lambda^*)^\top \mathcal{A} \mathcal{Q}^{-1} \mathcal{A}^\top (\lambda^{m+1} - \lambda^m) \\ & = \frac{1}{2} (F(y^{m+1} - y^*) - F(y^m - y^*)) \\ & \quad + \frac{1}{2} (F^*(\lambda^{m+1} - \lambda^*) - F^*(\lambda^m - \lambda^*)) \end{aligned} \quad (10)$$

for all $m \geq 2$. Now, the statement of Lemma 1 follows by summing up the above inequalities for $m = \hat{m}$ to $m = \bar{m}$ and using that the last element in the telescoping sum on the right hand,

$$\frac{F(y^{\bar{m}+1} - y^*) + F^*(\lambda^{\bar{m}+1} - \lambda^*)}{2} \geq 0$$

is non-negative. ■

The following theorem uses the above result in order to derive a convergence rate estimate of Algorithm 1.

Theorem 1 *Let Assumption 1 be satisfied and let (4) be feasible, such that a unique minimizer y^* and an associated dual solution λ^* exist. Then there exists a positive constant $\kappa < 1$ such that*

$$\begin{aligned} & F(y^{m+1} - y^*) + F^*(\lambda^{m+1} - \lambda^*) \\ & \leq \kappa (F(y^m - y^*) + F^*(\lambda^m - \lambda^*)) \end{aligned} \quad (11)$$

for all $m \geq 2$.

Proof. Let $\hat{\mathbb{Y}}_k$ denote the intersection of all active supporting hyperplanes at the solutions of the small scale QPs of Step 2a) in Algorithm 1 for $k \in \{0, \dots, N-1\}$. We construct the auxiliary optimization problem

$$\begin{aligned} & \min_{\hat{y}} \sum_{k=0}^N F_k(\hat{y}_k) \\ & \text{s.t.} \begin{cases} \forall k \in \{0, \dots, N-1\}, \\ G_{k+1} \hat{y}_{k+1} = H_k \hat{y}_k + h_k & | \hat{\lambda}_k, \\ 0 = H_N \hat{y}_N & | \hat{\lambda}_N, \\ \hat{y}_k \in \hat{\mathbb{Y}}_k \end{cases} \end{aligned} \quad (12)$$

and denote optimal primal and dual solutions of this problem by \hat{y}^* and $\hat{\lambda}^*$. Next, we also construct the auxiliary QPs

$$\begin{aligned} & \min_{\xi_0^m \in \hat{\mathbb{Y}}_0} F_0(\xi_0^m) - (H_0^\top \lambda_0^m)^\top \xi_0^m + F_0(\xi_0^m - y_0^m), \\ & \min_{\xi_k^m \in \hat{\mathbb{Y}}_k} F_k(\xi_k^m) + (G_k^\top \lambda_{k-1}^m - H_k^\top \lambda_k^m)^\top \xi_k^m + F_k(\xi_k^m - y_k^m), \\ & \min_{\xi_N^m} F_N(\xi_N^m) + (G_N^\top \lambda_{N-1}^m)^\top \xi_N^m + F_N(\xi_N^m - y_N^m). \end{aligned}$$

Because these QPs have equality constraints only, their parametric solutions must be affine. Thus, there exists a matrix T_1 such that

$$\xi^m - \hat{y}^* = T_1 \begin{pmatrix} y^m - \hat{y}^* \\ \lambda^m - \hat{\lambda}^* \end{pmatrix}.$$

Similarly, the coupled QP (7) has equality constraints only; that is, there exists a matrix T_2 such that

$$\begin{pmatrix} y^{m+1} - \hat{y}^* \\ \delta^m \end{pmatrix} = T_2 \begin{pmatrix} \xi^m - \hat{y}^* \\ y^m - \hat{y}^* \end{pmatrix}.$$

Now, we use the equation $\lambda^{m+1} - \lambda^* = \lambda^m - \lambda^* + \delta$ and substitute the above equations finding that

$$\begin{pmatrix} y^{m+1} - \hat{y}^* \\ \lambda^{m+1} - \hat{\lambda}^* \end{pmatrix} = T \begin{pmatrix} y^m - \hat{y}^* \\ \lambda^m - \hat{\lambda}^* \end{pmatrix} \quad (13)$$

with

$$T = \left(T_2 \begin{pmatrix} T_1 \\ I & 0 \end{pmatrix} + \begin{pmatrix} 0 & I \end{pmatrix} \right).$$

Next, we know from Lemma 1 that if we would apply Algorithm 1 to the auxiliary problem (12), the corresponding primal and dual iterates would converge to \hat{y}^* and $\hat{\lambda}^*$. In particular, inequality (10) yields

$$\begin{aligned} & (y^{m+1} - \hat{y}^*)^\top \mathcal{Q} (y^{m+1} - \hat{y}^*) \\ & \quad + (\lambda^{m+1} - \hat{\lambda}^*)^\top \mathcal{Q}^{-1} \mathcal{A}^\top (\lambda^{m+1} - \hat{\lambda}^*) \\ & < (y^m - \hat{y}^*)^\top \mathcal{Q} (y^m - \hat{y}^*) \\ & \quad + (\lambda^m - \hat{\lambda}^*)^\top \mathcal{A} \mathcal{Q}^{-1} \mathcal{A}^\top (\lambda^m - \hat{\lambda}^*), \end{aligned} \quad (14)$$

whenever $\begin{pmatrix} y^m - \hat{y}^* \\ \lambda^m - \hat{\lambda}^* \end{pmatrix} \neq 0$. By substituting the linear equation (13), we find that this is only possible if

$$T^\top \begin{pmatrix} \mathcal{Q} & 0 \\ 0 & \mathcal{A} \mathcal{Q}^{-1} \mathcal{A}^\top \end{pmatrix} T \preceq \kappa_A I \quad (15)$$

for a constant $\kappa_A < 1$. Now, one remaining difficulty is that the constant κ_A (as well as the matrix T) depends on the particular set \mathbb{A} of active supporting hyperplanes in the small-scale QPs. Nevertheless, because there exists only a finite number of possible active sets, the maximum $\kappa = \max_{\mathbb{A}} \kappa_A$ must exist and satisfy $\kappa < 1$. Now, the equation

$$\begin{pmatrix} y^{m+1} - y^* \\ \lambda^{m+1} - \lambda^* \end{pmatrix} = T \begin{pmatrix} y^m - y^* \\ \lambda^m - \lambda^* \end{pmatrix} \quad (16)$$

holds only for our fixed m and the associated matrix T for a particular active set, but the associated decent condition

$$\begin{aligned} & (y^{m+1} - y^*)^\top \mathcal{Q} (y^{m+1} - y^*) \\ & + (\lambda^{m+1} - \lambda^*)^\top \mathcal{A} \mathcal{Q}^{-1} \mathcal{A}^\top (\lambda^{m+1} - \lambda^*) \\ & \leq \kappa \left[(y^m - y^*)^\top \mathcal{Q} (y^m - y^*) \right. \\ & \quad \left. + (\lambda^m - \lambda^*)^\top \mathcal{A} \mathcal{Q}^{-1} \mathcal{A}^\top (\lambda^m - \lambda^*) \right], \end{aligned} \quad (17)$$

holds independently of the active set of the QPs in the m -th iteration and is indeed valid for all m . A resubstitution of F and F^* yields the statement of the theorem. ■

D. Asymptotic stability of Algorithm 1

The goal of this section is to establish asymptotic stability of Algorithm 1. Because we send the control input $u_0 = \xi_0^{\bar{m}}$ to the real process, the next measurement will be at $x_0^+ = Ax_0 + B\xi_0^{\bar{m}}$. Notice that, in general, we may have $x_0^+ \neq x_1^* = Ax_0 + By_0^*$, since we run Algorithm 1 with a finite $\bar{m} < \infty$.

Theorem 2 *Let Assumptions 1, 2 and 3 be satisfied. Let the constant $\sigma > 0$ be such that the semi-definite inequality $B^\top \mathcal{Q} B \preceq \sigma R$ holds and let the constants $\eta, \tau > 0$ be such that*

$$|J(x_0^+) - J(x_1^*)| \leq \eta \|x_0^+ - x_1^*\|_Q + \frac{\tau}{2} \|x_0^+ - x_1^*\|_Q^2 \quad (18)$$

If the constant $\bar{m} \in \mathbb{N}$ satisfies

$$\bar{m} > \frac{2 \log \left(2\eta\gamma \sqrt{\frac{\sigma(1+\kappa)}{\kappa}} + 2\tau\sigma\gamma^2 \frac{1+\kappa}{\kappa} \right)}{\log(1/\kappa)}, \quad (19)$$

then the controller in Algorithm 1 is asymptotically stable.

Proof. Because we have $x_0^+ - x_1^* = B(\xi_0^{\bar{m}} - y_0^*) = \mathcal{P}(\xi^{\bar{m}} - y^*)$ with $\mathcal{P} = [B, 0, \dots, 0]$, we can substitute (9) to find

$$\begin{aligned} & x_0^+ - x_1^* \\ & = \mathcal{P} \left[\frac{\mathcal{Q}^{-1} \mathcal{A}^\top (\lambda^{\bar{m}+1} - \lambda^{\bar{m}})}{2} + \frac{y^{\bar{m}+1} + y^{\bar{m}}}{2} - y^* \right] \\ & = \frac{1}{2} \mathcal{P} \mathcal{Q}^{-1} \mathcal{A}^\top (\lambda^{\bar{m}+1} - \lambda^*) + \frac{1}{2} \mathcal{P} \mathcal{Q}^{-1} \mathcal{A}^\top (\lambda^* - \lambda^{\bar{m}}) \\ & \quad + \frac{1}{2} \mathcal{P} (y^{\bar{m}+1} - y^*) + \frac{1}{2} \mathcal{P} (y^{\bar{m}} - y^*). \end{aligned}$$

The particular definition of σ implies $\mathcal{P}^\top \mathcal{Q} \mathcal{P} \preceq \sigma \mathcal{Q}$ and

$$\begin{aligned} & 4(x_0^+ - x_1^*)^\top \mathcal{Q} (x_0^+ - x_1^*) \\ & \leq 4(\lambda^{\bar{m}+1} - \lambda^*)^\top \mathcal{A} \mathcal{Q}^{-1} \mathcal{P}^\top \mathcal{Q} \mathcal{P} \mathcal{Q}^{-1} \mathcal{A}^\top (\lambda^{\bar{m}+1} - \lambda^*) \\ & \quad + 4(\lambda^{\bar{m}} - \lambda^*)^\top \mathcal{A} \mathcal{Q}^{-1} \mathcal{P}^\top \mathcal{Q} \mathcal{P} \mathcal{Q}^{-1} \mathcal{A}^\top (\lambda^{\bar{m}} - \lambda^*) \\ & \quad + 4(y^{\bar{m}+1} - y^*)^\top \mathcal{P}^\top \mathcal{Q} \mathcal{P} (y^{\bar{m}+1} - y^*) \\ & \quad + 4(y^{\bar{m}} - y^*)^\top \mathcal{P}^\top \mathcal{Q} \mathcal{P} (y^{\bar{m}} - y^*) \\ & \leq 4\sigma (F(y^{\bar{m}+1} - y^*) + F^*(\lambda^{\bar{m}+1} - \lambda^*)) \\ & \quad + 4\sigma (F(y^{\bar{m}} - y^*) + F^*(\lambda^{\bar{m}} - \lambda^*)) \\ & \leq 4\sigma(1 + \kappa) \kappa^{\bar{m}-1} (F(y^1 - y^*) + F^*(\lambda^1 - \lambda^*)) \\ & \leq 16\sigma(1 + \kappa) \kappa^{\bar{m}-1} \gamma^2 F_0(y_0^*). \end{aligned}$$

The last inequality holds based on the inequalities

$$\begin{aligned} F(y^*) + F^*(\lambda^*) & \leq \gamma^2 x_0^\top \mathcal{Q} x_0 \leq \gamma^2 F_0(y_0^*), \\ F(y^1) + F^*(\lambda^1) & \leq \gamma^2 x_0^\top \mathcal{Q} x_0 \leq \gamma^2 F_0(y_0^*), \end{aligned}$$

which hold due to Assumption 3 and the particular construction in Step 1 of Algorithm 1. Now, a division by 4 yields

$$\|x_0^+ - x_1^*\|_Q^2 \leq 4\sigma\gamma^2 \left(\frac{1 + \kappa}{\kappa} \right) \kappa^{\bar{m}} F_0(y_0^*). \quad (20)$$

By combining this inequality with (18) we find

$$\begin{aligned} & |J(x_0^+) - J(x_1^*)| \\ & \leq 2 \left[\eta\gamma \sqrt{\frac{\sigma(1+\kappa)}{\kappa}} + \tau\sigma\gamma^2 \frac{1+\kappa}{\kappa} \right] \kappa^{\frac{\bar{m}}{2}} F_0(y_0^*). \end{aligned} \quad (21)$$

Thus, if we set

$$\alpha = 1 - 2 \left[\eta\gamma \sqrt{\frac{\sigma(1+\kappa)}{\kappa}} + \tau\sigma\gamma^2 \frac{1+\kappa}{\kappa} \right] \kappa^{\frac{\bar{m}}{2}} > 0,$$

we have

$$\begin{aligned} J(x_0^+) & \leq J(x_0) - (F_0(y_0^*) - J(x_0^+) + J(x_1^*)) \\ & \leq J(x_0) - \alpha F_0(y_0^*), \end{aligned} \quad (22)$$

which is sufficient to establish asymptotic stability [28]. ■

E. Performance of Algorithm 1

The result of Theorem 2 can be extended in order to derive an upper bound on the sub-optimality of Algorithm 1.

Corollary 1 *Let the assumption of Theorem 2 hold with*

$$\alpha = 1 - 2 \left[\eta\gamma \sqrt{\frac{\sigma(1+\kappa)}{\kappa}} + \tau\sigma\gamma^2 \frac{1+\kappa}{\kappa} \right] \kappa^{\frac{\bar{m}}{2}}.$$

If $y_i^{\text{cl}} = (x_i^{\text{cl}}, u_i^{\text{cl}})$ denotes the sequence of closed-loop states and controls that are generated by the controller in Algorithm 1, an a-priori bound on the associated infinite-horizon closed-loop performance is given by

$$\sum_{i=0}^{\infty} \ell(x_i^{\text{cl}}, u_i^{\text{cl}}) \leq \frac{J(x_0)}{\alpha}.$$

Proof. Because (22) holds, we have

$$J(x_{i+1}^{\text{cl}}) \leq J(x_i^{\text{cl}}) - \alpha F_0(y_i^{\text{cl}}),$$

which yields the inequality

$$\sum_{i=0}^{\infty} F_0(y_i^{\text{cl}}) \leq \frac{1}{\alpha} \sum_{i=0}^{\infty} (J(x_i^{\text{cl}}) - J(x_{i+1}^{\text{cl}})).$$

The statement of the corollary follows after simplifying the telescoping sum on the right and substituting the equation $F_0(y_i^{\text{cl}}) = \ell(x_i^{\text{cl}}, u_i^{\text{cl}})$. ■

Remark 1 (MPC with state constraints) *Notice that (1) admits control constraints only. A complete discussion of how to extend the presented algorithm and analysis for MPC problems with state constraint would go beyond the scope of this paper. However, one method for taking such state*

constraints into account can be obtained by adding L_1 -penalty functions to the stage cost ℓ . Our stability and convergence proofs can be extended for this case because adding L_1 -penalties does not change the fact that the cost-to-go function J is piecewise quadratic.

IV. IMPLEMENTATION DETAILS AND CASE STUDIES

This section applies Algorithm 1 to benchmark case studies.

A. Implementation on hardware with limited memory

Algorithm 1 has two main steps, Step 2a) and Step 2b). In Step 2a) decoupled QPs have to be solved on-line. We solve these QPs off-line using multi-parametric programming by pre-computing the solution maps

$$\begin{aligned} \xi_0^*(\theta_0, x_0) &= \arg \min_{\xi_0 \in \mathbb{Y}_0} 2F_0(\xi_0) + \theta_0^\top \xi_0, \\ \xi_1^*(\theta_1) &= \arg \min_{\xi_1 \in \mathbb{Y}_1} 2F_1(\xi_1) + \theta_1^\top \xi_1, \\ \xi_N^*(\theta_N) &= \arg \min_{\xi_N} 2F_N(\xi_N) + \theta_N^\top \xi_N \end{aligned} \quad (23)$$

with parameters $\theta_0 \in \mathbb{R}^{n_u+n_z}$, $\theta_1 \in \mathbb{R}^{n_x+n_u}$, and $\theta_N \in \mathbb{R}^{n_x}$. Here, ξ_0^* depends on x_0 recalling that this dependency had been hidden in our definition of F_0 and \mathbb{Y}_0 . We use MPT [29] to pre-compute and store the maps ξ_0^* , ξ_1^* and ξ_N^* . Consequently, Step 2a) in Algorithm 1 can be replaced by:

- **Step 2a')** Compute the parameters

$$\theta_0^m = -H_0^\top \lambda_0^m - 2\Sigma_0 y_0^m, \quad (24a)$$

$$\theta_k^m = G_k^\top \lambda_{k-1}^m - H_k^\top \lambda_k^m - 2\Sigma_k y_k^m, \quad (24b)$$

$$\theta_N^m = G_N^\top \lambda_{N-1}^m - 2\Sigma_N y_N^m \quad (24c)$$

with $\Sigma_0 = R$, $\Sigma_k = \text{blkdiag}\{Q, R\}$, $k \in \{1, \dots, N-1\}$, $\Sigma_N = P$ and set

$$\xi_0^m = \xi_0^*(\theta_0^m, x_0), \quad \xi_k^m = \xi_1^*(\theta_k^m)$$

for all $k \in \{1, \dots, N\}$ by evaluating the respective explicit solution maps (23). In this paper, we use the enumeration-based multi-parametric QP algorithm from [30] for generating these maps.

Notice that the complexity of pre-processing the small-scale QPs (23) depends on the number $N_R = \max\{N_{R,0}, N_{R,1}\}$ of critical regions over which the PWA optimizers ξ_0^* , ξ_1^* and ξ_N^* are defined [31], but N_R is independent of the prediction horizon N as summarized in the first row in Table I. For a derivation of the associated run-time and memory complexity results we refer to [23], [32], [33].

TABLE I
COMPLEXITY OF STEPS 2A') AND 2B) OF ALGORITHM 1.

Step	Offline CPU time	Online CPU time	Memory Requirement
2a')	$\mathcal{O}(N_R^2)$	$\mathcal{O}(N \log_2(N_R))$	$\mathcal{O}(N_R)$
2b)	$\mathcal{O}(Nn_x^3)$	$\mathcal{O}(Nn_x^2)$	$\mathcal{O}(Nn_x^2)$

In Step 2b) coupled QP (7) must be solved. Because this QP has equality constraints only, (7) is equivalent to a large but sparse system of equations. Moreover, all matrices in (7) are given and constant during the online iterations. This means

that all linear algebra decompositions can be pre-computed offline. If one uses standard Riccati recursions for exploiting the band-structure of (7), the computational complexity for all offline computations is at most of order $\mathcal{O}(Nn_x^3)$, while the online implementation has complexity $\mathcal{O}(Nn_x^2)$ [34] as summarized in the second row in Table I.

B. Parallel MPC with Long Horizons

The first benchmark considers a linear dynamic system with

$$A = \begin{bmatrix} 0.9993 & -3.0083 & -0.1131 & -1.6081 \\ 0 & 0.9862 & 0.0478 & 0 \\ 0 & 2.0833 & 1.0089 & 0 \\ 0 & 0.0526 & 0.0498 & 1 \end{bmatrix},$$

$$\text{and } B = \begin{bmatrix} -0.0804 & -0.0291 & -0.8679 & -0.0216 \\ -0.6347 & -0.0143 & -0.0917 & -0.0022 \end{bmatrix}^\top.$$

The states of this system can be interpreted as the yaw, pitch, roll and the attack angles of an aircraft while the controls are given by the elevator and the flaperon angles [35]. The state constraint and control constraint are given by

$$\mathbb{X} = \left\{ x \in \mathbb{R}^4 \mid \begin{bmatrix} -0.5 \\ -100 \end{bmatrix} \leq \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x \leq \begin{bmatrix} 0.5 \\ 100 \end{bmatrix} \right\},$$

$$\mathbb{U} = [-25, 25] \times [-25, 25],$$

the stage cost weights are set to

$$Q = \text{diag}(0.1, 100, 0.1, 100), \quad R = \text{diag}(10, 10),$$

and the initial state is given by $x_0 = [20 \ 0 \ 20 \ 20]$. The matrix P is computed by solving an algebraic Riccati equation such that the terminal cost is locally equal to the unconstrained infinite horizon cost [25]. Moreover, the parameter $\gamma = 10$ is fixed in our implementations.

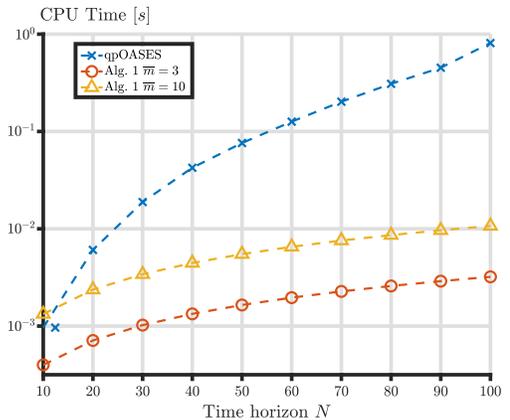


Fig. 1. CPU time comparison: Algorithm 1 vs Traditional MPC (Condensing + qpOASES) both run in Matlab R2018a interfacing C/C++ code.

Figure 1 shows a CPU time comparison of Algorithm 1 (with $\bar{m} = 3$ and $\bar{m} = 10$) and traditional MPC in dependence on the prediction horizon. The implementations of Algorithm 1 uses Matlab R2018a with YALMIP [36] and MPT 3.1.5 [29] but the comparison is based on qpOASES [2]. Algorithm 1 is faster for large N , but this speed-up comes along with a loss of control performance (see Figure 2). For $\bar{m} = 10$ the

sub-optimal closed-loop state trajectory is, however, almost indistinguishably close to the optimal trajectory.

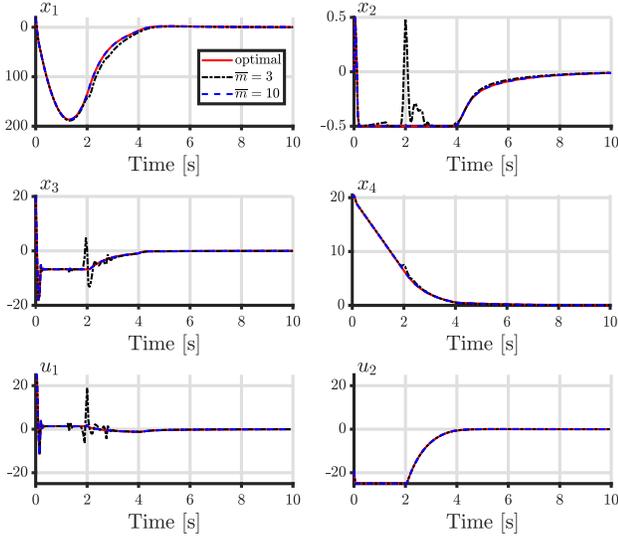


Fig. 2. Closed-loop state and control trajectories: $\bar{m} \in \{3, 10, \infty\}$, $N = 40$.

Our implementation of Algorithm 1 requires 81 kB memory corresponding to 92 regions (independent of N). These numbers can be compared with the following results for a standard Explicit MPC implementation using the geometric parametric LCP solver of MPT 3.1.5 [29]:

N	# of regions	memory [kB]	CPU time [μ s]
3	427	233	23
5	3 649	2 566	89
10	64 556	70 609	304

For $N > 10$ our implementation of Explicit MPC ran out of memory.

C. Spring-Vehicle-Damper System

Our second case study considers a spring-vehicle-damper system with \bar{I} vehicles with mass $m_v = 1$ kg, as visualized below.

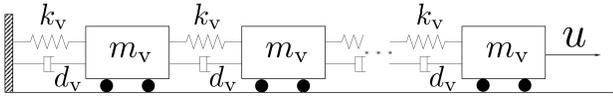


Fig. 3. Sketch of a spring-vehicle-damper system.

The non-zero blocks of the system matrices are given by

$$\begin{aligned}
 A_{i,i} &= \mathbb{I} + T_s \begin{pmatrix} 0 & 1 \\ -2 \frac{k_v}{m_v} & -2 \frac{d_v}{m_v} \end{pmatrix}, & B_i &= \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \\
 A_{\bar{I},\bar{I}} &= \mathbb{I} + T_s \begin{pmatrix} 0 & 1 \\ -\frac{k_v}{m_v} & -\frac{d_v}{m_v} \end{pmatrix}, & B_{\bar{I}} &= \begin{pmatrix} 0 \\ \frac{T_s}{m_v} \end{pmatrix}, \\
 A_{i-1,i} &= A_{i,i+1} = T_s \begin{pmatrix} 0 & 0 \\ \frac{k_v}{m_v} & \frac{d_v}{m_v} \end{pmatrix},
 \end{aligned}$$

for $i \in \{1, \dots, \bar{I} - 1\}$. Here, $k_v = 3$ N/m denotes the spring constant, $d_v = 3$ Ns/m a damping coefficient, and $T_s = 0.1$ s

the step-size of an Euler discretization. The state and control constraints are set to

$$\mathbb{X} = \mathbb{X}_1 \times \dots \times \mathbb{X}_{\bar{I}}, \mathbb{U} = [-2, 0.5],$$

$$\text{where } \mathbb{X}_1 = \dots = \mathbb{X}_{\bar{I}} = [-0.5, 1.5] \times [-0.5, 1].$$

The weighting matrices of the stage cost are set to $Q = 10 I$ and $R = I$.

In this example, an implementation of Algorithm 1 requires 287 kB corresponding to 432 critical regions. This memory requirement is independent of the number of vehicles \bar{I} and the prediction horizon N . In contrast to this, the number of regions for standard Explicit MPC depends on both \bar{I} and N :

(\bar{I}, N)	# of regions	memory [kB]
(1, 10)	58	14
(1, 50)	144	169
(2, 10)	2 244	877
(3, 10)	4 247	2 324

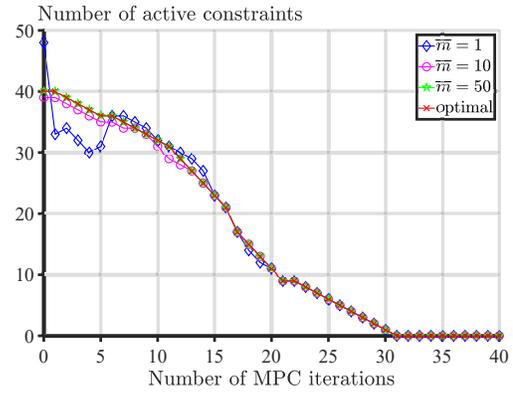


Fig. 4. The total number of active constraints of all distributed QP solvers during the MPC iterations for different choices of \bar{m} .

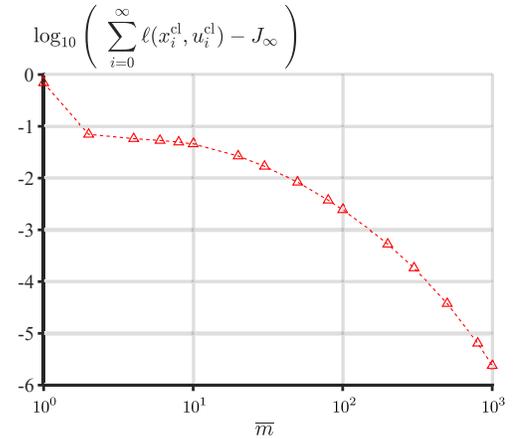


Fig. 5. Closed-loop performance degradation (log scale) with respect to the optimal objective function J_∞ as a function of \bar{m} .

Figure 4 shows the total number of active constraints of all distributed QP solvers for different choices of \bar{m} . Here, the number of active constraints of optimal MPC (corresponding to $\bar{m} = \infty$) are shown in the form of red crosses in Figure 4. If we compare these optimal red crosses with the blue diamonds ($\bar{m} = 1$), we can see that the choice $\bar{m} = 1$ still leads to many wrongly chosen active sets. However, for

$\bar{m} \geq 10$ a reasonably accurate approximation of the optimal number of active constraints is maintained during all iterations. Finally, Figure 5 shows the sub-optimality of Algorithm 1 in dependence on \bar{m} for a representative case study with $\bar{I} = 3$ and $N = 30$.

V. CONCLUSIONS

This paper has introduced a parallelizable and real-time verifiable MPC scheme, presented in the form of Algorithm 1. This control algorithm evaluates at every sampling time a finite number of pre-computed, explicit piecewise affine solution maps that are associated with parametric small-scale QPs. Theorem 2 and Corollary 1 provide both asymptotic stability guarantees as well as bounds on sub-optimality. The presented explicit MPC approach can be used to reduce the storage and run-time of explicit MPC by orders of magnitude.

ACKNOWLEDGMENT

All authors were supported via a bilateral grant agreement between China and Slovakia (SK-CN-2015-PROJECT-6558, APVV SK-CN-2015-0016). The work of Yuning Jiang and Boris Houska was supported by the National Science Foundation China (NSFC), Nr. 61473185, as well as ShanghaiTech University, Grant-Nr. F-0203-14-012. Juraj Oravec and Michal Kvasnica acknowledge the contribution of the Scientific Grant Agency of the Slovak Republic under the grants 1/0585/19, 1/0545/20, and the Slovak Research and Development Agency under the project APVV-15-0007.

REFERENCES

- [1] M. Diehl, H. Ferreau, and N. Haverbeke, "Efficient numerical methods for nonlinear MPC and moving horizon estimation," in *Nonlinear model predictive control*, ser. Lecture Notes in Control and Information Sciences, L. Magni, M. Raimondo, and F. Allgöwer, Eds. Springer, 2009, vol. 384, pp. 391–417.
- [2] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "qpOases: A parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014.
- [3] S. Qin and T. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 93, no. 316, pp. 733–764, 2003.
- [4] B. Houska, H. Ferreau, and M. Diehl, "An auto-generated real-time iteration algorithm for nonlinear MPC in the microsecond range," *Automatica*, vol. 47, p. 2279–2285, 2011.
- [5] J. Mattingley and S. Boyd, "Automatic code generation for real-time convex optimization," *Convex optimization in signal processing and communications*, pp. 1–41, 2009.
- [6] M. Diehl, H. Bock, J. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer, "Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations," *Journal of Process Control*, vol. 12, no. 4, pp. 577–585, 2002.
- [7] V. Zavala and L. Biegler, "The advanced-step NMPC controller: Optimality, stability and robustness," *Automatica*, vol. 45, no. 1, pp. 86–93, 2009.
- [8] D. Ingole and M. Kvasnica, "FPGA implementation of explicit model predictive control for closed loop control of depth of anesthesia," in *5th IFAC Conference on Nonlinear Model Predictive Control*, 2015, pp. 484–489.
- [9] R. Oberdieck, N. Diangelakis, I. Nascu, M. Papathanasiou, M. Sun, S. Avraamidou, and E. Pistikopoulos, "On multi-parametric programming and its applications in process systems engineering," *Chemical Engineering Research and Design*, vol. 116, pp. 61–82, 2016.
- [10] P. Grieder and M. Morari, "Complexity reduction of receding horizon control," in *42nd IEEE International Conference on Decision and Control (CDC)*, 2003, pp. 3179–3190.
- [11] R. Cagienard, P. Grieder, E. Kerrigan, and M. Morari, "Move blocking strategies in receding horizon control," *Journal of Process Control*, vol. 17, no. 6, pp. 563–570, 2007.
- [12] M. Kvasnica, B. Takács, J. Holaza, and S. Di Cairano, "On region-free explicit model predictive control," in *2015 54th IEEE Conference on Decision and Control (CDC)*, 2015, pp. 3669–3674.
- [13] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundation Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [14] H. Everett, "Generalized Lagrange multiplier method for solving problems of optimum allocation of resources," *Operations Research*, vol. 11, no. 3, pp. 399–417, 1963.
- [15] B. Houska, J. Frasch, and M. Diehl, "An augmented Lagrangian based algorithm for distributed non-convex optimization," *SIAM Journal on Optimization*, vol. 26, no. 2, pp. 1101–1127, 2016.
- [16] P. Giselsson, M. Dang Doan, T. Keviczky, B. De Schutter, and A. Rantzer, "Accelerated gradient methods and dual decomposition in distributed model predictive control," *Automatica*, vol. 49, no. 3, pp. 829–833, 2013.
- [17] A. Kozma, J. Frasch, and M. Diehl, "A distributed method for convex quadratic programming problems arising in optimal control of distributed systems," in *52nd IEEE Conference on Decision and Control*, 2013, pp. 1526–1531.
- [18] I. Necoara and J. Suykens, "Application of a smoothing technique to decomposition in convex optimization," *IEEE Transactions on Automatic Control*, vol. 53, no. 11, pp. 2674–2679, 2008.
- [19] B. O'Donoghue, G. Stathopoulos, and S. Boyd, "A splitting method for optimal control," *IEEE Transactions on Control Systems Technology*, vol. 21(6), pp. 2432–2442, 2013.
- [20] S. Richter, M. Morari, and C. Jones, "Towards computational complexity certification for constrained MPC based on Lagrange relaxation and the fast gradient method," in *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, 2011, pp. 5223–5229.
- [21] L. Ferranti, G. Stathopoulos, C. N. Jones, and T. Keviczky, "Constrained LQR using online decomposition techniques," in *IEEE Conference on Decision and Control*, Las Vegas, USA, 2016, pp. 2339–2344.
- [22] G. Stathopoulos, T. Keviczky, and Y. Wang, "A hierarchical time-splitting approach for solving finite-time optimal control problems," in *European Control Conference*, 2013, pp. 3089–3094.
- [23] J. Oravec, Y. Jiang, B. Houska, and M. Kvasnica, "Parallel explicit MPC for hardware with limited memory," in *In Proceedings of the 20th IFAC World Congress, Toulouse, France*, 2017, pp. 3356–3361.
- [24] F. Borrelli, *Constrained Optimal Control Of Linear And Hybrid Systems*, ser. Lecture Notes in Control and Information Sciences. Springer, 2003, vol. 290.
- [25] J. Rawlings, D. Mayne, and M. Diehl, *Model Predictive Control: Theory and Design*. Madison, WI: Nob Hill Publishing, 2018.
- [26] F. Borrelli, A. Bemporad, and M. Morari, "Geometric algorithm for multiparametric linear programming," *Journal of Optimization Theory and Applications*, vol. 118, no. 3, pp. 515–540, 2003.
- [27] P. Otta, O. Santin, and V. Havlena, "Measured-state driven warm-start strategy for linear MPC," in *2015 European Control Conference (ECC)*, 2015, pp. 3132–3136.
- [28] L. Grüne, "Analysis and design of unconstrained nonlinear mpc schemes for finite and infinite dimensional systems," *SIAM Journal on Control and Optimization*, vol. 48, no. 2, pp. 1206–1228, 2009.
- [29] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, "Multi-parametric toolbox 3.0," in *2013 European control conference (ECC)*. IEEE, 2013, pp. 502–510.
- [30] M. Herceg, C. N. Jones, M. Kvasnica, and M. Morari, "Enumeration-based approach to solving parametric linear complementarity problems," *Automatica*, no. 62, pp. 243–248, 2015.
- [31] 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.
- [32] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [33] F. Borrelli, A. Bemporad, and M. Morari, *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.
- [34] D. Bertsekas, *Dynamic Programming and Optimal Control*, 3rd ed. Belmont, Massachusetts: Athena Scientific Dynamic Programming and Optimal Control, 2012.
- [35] A. Bemporad, A. Casavola, and E. Mosca, "Nonlinear control of constrained linear systems via predictive reference management," *IEEE Transactions on Automatic Control*, vol. 42, no. 3, pp. 340–349, 1997.
- [36] J. Löfberg, "YALMIP," 2004, available from <http://users.isy.liu.se/johanl/yalmip/>.

Explicit MPC-Based RBF Neural Network Controller Design With Discrete-Time Actual Kalman Filter for Semiactive Suspension

Lehel Huba Csekő, Michal Kvasnica, and Béla Lantos

Abstract—Many applications require fast control action and efficient constraint handling, such as in aircraft or vehicle control, where instead of the slow online computation of the model predictive control (MPC) the explicit MPC can be an alternative solution. Explicit MPC controllers consist of several affine feedback gains, each of them valid over a polyhedral region of the state space. The exponential blow-up of the number of regions with increasing the prediction horizon increases the searching time among the regions extremely which together with the requirement of the full state measurement decreases its applicability for real systems. First, discrete-time actual Kalman filter is designed for the semiactive suspension and applied to explicit MPC controller that requires only measurement of the suspension deflection. Second, this paper presents a systematic way to design Gaussian radial basis function-based neural network (NN) approximation of the explicit MPC controller and shows that a well-tuned NN with some neurons can replace the explicit MPC controller. This nonlinear state-feedback controller can ensure the fast control action but price of the approximation is some deterioration of the performance value. The complete novel nonlinear control system with Kalman filter is analyzed in detail. The derived controllers are evaluated through simulations, where shock tests and white noise velocity disturbances are applied to a real quarter car vertical model.

Index Terms—Approximation methods, Kalman filters, limiting energy dissipation, optimal control, predictive control, radial basis function (RBF) networks, vehicle suspensions.

I. INTRODUCTION

THE automotive suspension supports the vehicle body on the axles and provides good ride quality against the road disturbances, while keeping good road traction. In the newest luxury cars, intelligent suspension is part of a vehicle dynamic control system and one may change the vehicle characteristic by pushing a button. The drive feeling can be set to a comfort mode as in a limousine, to a sporty mode, or to automatic.

Manuscript received April 6, 2014; revised September 21, 2014; accepted November 14, 2014. Date of publication January 28, 2015; date of current version August 7, 2015. Manuscript received in final form December 9, 2014. Recommended by Associate Editor E. Kerrigan.

L. H. Csekő and B. Lantos are with the Department of Control Engineering and Information Technology, Budapest University of Technology and Economics, Budapest 1111, Hungary (e-mail: csekol@yahoo.com; lantos@iit.bme.hu).

M. Kvasnica is with the Institution of Information Engineering, Automation and Mathematics, Slovak University of Technology in Bratislava, Bratislava 812 45, Slovakia (e-mail: michal.kvasnica@stuba.sk).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCST.2014.2382571

The system influences the characteristic of gear change, steering, engine, and suspension.

On the basis of the analogy between the electrical and mechanical circuits, a new mechanical circuit element, the inerter, has been developed and applied to vehicle suspension with success. Control design of suspension with inerter is based on the determination of the positive real admittance, which meets with the specified performance measures [1], [2]. The first deployment of the inerter-based suspension under the name J-damper happened in the McLaren Formula One Racing team, leading to significant performance gains in handling and grip [3].

The quarter car suspension model is adequate to analyze the car response to irregular road surface and design an approximately optimal suspension controller to increase the good drive feeling. The performance of the suspension in the time domain can be expressed by ℓ_2 norm. The suspension can be classified into three groups according to operation: 1) passive; 2) semiactive; and 3) active. Passive suspension consists only of spring, dampers, and inverters. The semiactive setup utilizes variable damper and in the active suspension hydraulic, air, or electric actuator forces are applied. The semiactive suspension has a simpler mechanical structure than the active one, requires power only to change the dissipative force characteristic, and it cannot become unstable because it is a passive system. Due to its many advantageous properties, the automotive industry builds the semiactive suspension often into top vehicles. The controller design challenge in semiactive suspension is due to its nonlinearity with dissipativity and saturation constraints. If these constraints are not considered by the controller and only the clipped strategy is applied instead, then performance may be lost. In addition to the automotive industry, the semiactive dampers can also be used in buildings to compensate for oscillations during earthquakes and anywhere where the vibration is undesirable. Examples mentioned previously show that the research area of the controlled dampers is very active and can take advantage from new damper technology and new control methods.

Although lots of modern control methods exist, only few can treat constraints in efficient way. The main objective of optimal control is to determine the solution of the infinite-horizon linear quadratic regulator problem with constraints (CLQR) that was studied by many researchers in [4]–[10]. The solution

can be approximated by repeatedly solving constrained finite-horizon optimal control problems in a receding horizon fashion, which is also called model predictive control (MPC) and accepted mainly in the process industry. Unfortunately, the time-consuming repetitive solution of quadratic program (QP) and linear program (LP) limits the application of MPC mainly to processes with slow dynamics.

To overcome this limitation, the method of multiparametric programming can be applied to precalculate the solution of the finite-horizon CLQR problem in the form of a piecewise affine (PWA) function. This technique enlarges the scope of applicability of MPC, allows insight into the controller structure, and enables detection of the reachable states and fault operations in advance. A serious drawback of explicit MPC solutions, however, lies in the exponential growth of the number of control regions when the prediction horizon is increasing [11]. A high number of regions causes large online implementation time required to find the control action and also impacts the storage requirements. New research directions study efficient searching algorithms to choose the feedback gains [12]–[14], and/or develop techniques to reduce the number of regions [15], [16]. An another approach to overcome the above mentioned limitations is to apply some kind of approximation of the explicit MPC controller using, for example, neural networks (NNs) or polynomials [17]–[19]. In this paper, we approximate the explicit MPC by a Gaussian radial basis function NN (RBF NN) that provides a nonlinear state-feedback controller. The RBF NN is a two-layer NN, where only the output layer is trained by the least squares method. This ensures a fast and simple learning, while providing similar approximation quality as multilayer perception (MLP) networks [20]. Important to notice is that the training happens offline. Once the NN is working, the online computation of control actions boils down to a fast evaluation of a nonlinear state-feedback controller. From a practical viewpoint, if the approximation error was below the conversion errors of the Digital-to-Analog conversion then, the approximation would yield the same result as the explicit MPC [19]. In this paper, we analyze goodness of our approximation from different aspects.

An another limitation of MPC implemented in the receding horizon manner is that persistent satisfaction of constraints is not guaranteed *a priori*. To mitigate this issue, one can use non-MPC type of controllers (a good choice can be, e.g., Linear-Quadratic (LQ) controller) in the part of the state space not covered by MPC. Alternatively, the constraints can be softened. Then, the feasible space will be completed with new regions. The price of the softening is that the state trajectory started from the earlier infeasible regions will violate the hard constraints. Moreover, application of the soft constraints will rapidly increase the number of regions [11]. The RBF NN approximation requires a training set and the idea is to use the previously mentioned completion methods of the regions to cover the whole available state space to create the training points easily. Hence, we do not need to analyze the more dimensional shape of the union of feasible regions for generating training set.

Finally, MPC is a state-feedback policy, and hence, requires

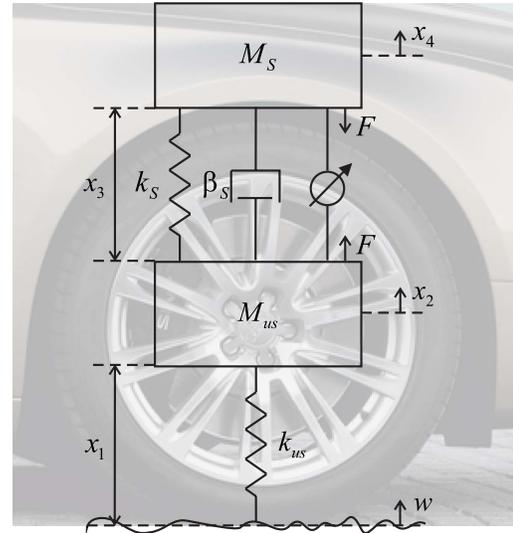


Fig. 1. Semiactive quarter car model.

full state measurement. These, however, are not always available in practice. Therefore, this paper suggests to employ a discrete-time actual Kalman filter connected to the explicit MPC controller and the NN approximation of the explicit MPC controller, which requires only measurement of the suspension deflection. It is well known from the Linear-Quadratic-Gaussian (LQG) theory that Kalman filter is the optimal filter for LQ control but we do not analyze if it is the optimal filter for explicit MPC or for NN controller. The results will show that the Kalman filter can be an appropriate estimator in case of presented controllers. The designed controllers are compared with each other and the results are presented through simulation of a real quarter car model. This paper builds upon results of the research article [11].

The remainder of this paper is organized as follows. Section II introduces the model of the semiactive suspension and the passivity constraints. Furthermore, it also presents the clipped optimal control, the discrete-time MPC, and the Kalman filter. Section III summarizes the theoretical background of PWA systems to deal with nonlinearities. Explicit MPC and multiparametric programming are discussed in Section IV. Section V is devoted to design of discrete-time actual Kalman filter. In Section VI, the explicit MPC-based RBF NN approximator is designed and the overall control system with Kalman filter is analyzed. The conclusion is drawn in Section VII.

II. QUARTER CAR MODEL OF THE SEMIACTIVE SUSPENSION AND THE OPTIMAL CONTROL PROBLEM

Motion equations of a 2-DOF quarter car in Fig. 1 can be described as in [21] by

$$\begin{aligned} \dot{x}_1 &= x_2 - w, & \dot{x}_2 &= \frac{1}{M_{us}}[k_s x_3 + \beta_s(x_4 - x_2) - k_{us} x_1 + F] \\ \dot{x}_3 &= x_4 - x_2, & \dot{x}_4 &= \frac{1}{M_s}[-k_s x_3 - \beta_s(x_4 - x_2) - F] \end{aligned} \quad (1)$$

where M_s and M_{us} are the sprung and unsprung mass, respectively, k_s and k_{us} [N/m] are the spring stiffness coefficients,

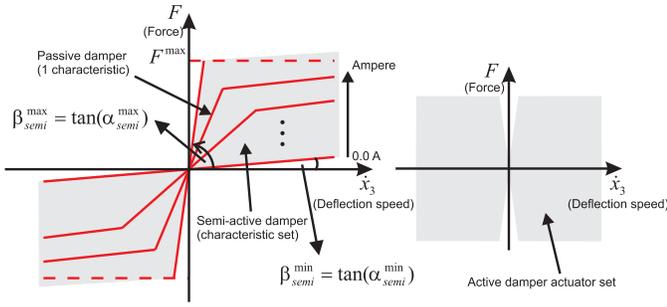


Fig. 2. Speed/effort rule of a passive semiactive (left) and active (right) suspension system.



Fig. 3. MR damper [22].

β_s [N/m/s] is the damping coefficient, x_1 [m] is the tire deflection, x_2 [m/s] is the unsprung mass velocity, x_3 [m] is the suspension deflection, x_4 [m/s] is sprung mass velocity, F [N] is the adjustable force, and w [m/s] is the road velocity disturbance.

The following normalized parameters will be introduced: sprung-to-unsprung mass ratio ρ , sprung mass and wheel-hop natural frequencies ω_s, ω_{us} [rad/s], and the normalized adjustable force u [N/kg] which imply the normalized damping coefficient $\zeta = \beta_s / (2(M_s k_s)^{1/2})$ to obtain numerically better conditioned state equations:

$$\begin{aligned} \dot{x}_2 &= -\underbrace{\frac{k_{us}}{M_{us}}}_{\omega_{us}^2} x_1 - \underbrace{\frac{\beta_s}{M_{us}}}_{2\rho\zeta\omega_s} x_2 + \underbrace{\frac{k_s}{M_{us}}}_{\rho\omega_s^2} x_3 + \underbrace{\frac{\beta_s}{M_{us}}}_{2\rho\zeta\omega_s} x_4 + \underbrace{\frac{M_s}{M_{us}}}_{\rho} \underbrace{\frac{F}{M_s}}_u \\ \dot{x}_4 &= \underbrace{\frac{\beta_s}{M_s}}_{2\zeta\omega_s} x_2 - \underbrace{\frac{k_s}{M_s}}_{\omega_s^2} x_3 - \underbrace{\frac{\beta_s}{M_s}}_{2\zeta\omega_s} x_4 - \underbrace{\frac{F}{M_s}}_u. \end{aligned} \quad (2)$$

According to Fig. 2, suspensions systems can be categorized into three groups. Passive suspension always dissipates energy through a fixed damping force characteristic. Semiactive suspension can also only dissipate energy but with varying damping force characteristic (left). Active suspension can both dissipate (1 and 3 quarters) or generate energy (2 and 4 quarters) using the almost total damping force plane (right), where the actuator model (e.g., electrohydraulic actuator) defines the exact characteristic on the plane.

Due to their simple mechanical structure, low-energy consumption, fast time response, and low cost, the semiactive suspensions are preferred over the active ones when increasing the vehicle performance is required. The magnetorheological (MR) (Fig. 3) damper is one of the most applied semiactive dampers, which uses MR fluid (e.g., oil and ferro particles) whose viscosity, i.e., damping value β_{semi} , can be varied by

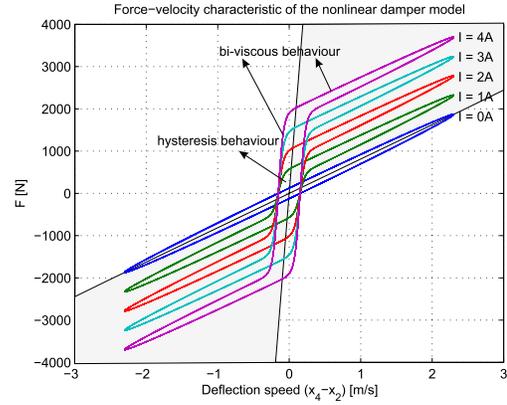


Fig. 4. Force-velocity characteristic of the damper model.

applying magnetic field controlled by current. The magnetic field orders the particles in such a direction as to increase the damping value. The damping characteristic can be controlled accurately by changing the magnetic field. An example of a nonlinear model of the MR-damper, which can describe the biviscous and hysteresis behaviors of the damper very well [23], [24], is given by

$$F = c_0 \dot{x}_3 + k_0 x_3 + y_{mr} I \cdot \tanh(c_1 \dot{x}_3 + k_1 x_3) \quad (3)$$

where the parameters $c_0, k_0, y_{mr}, c_1,$ and k_1 are defined in [24]. The physical meaning of these parameters is discussed in [23]. The input electric current applied to MR damper is denoted by I . If the following sinusoidal suspension deflection x_3 is considered:

$$\begin{aligned} t &= [0 \dots 2\pi], \quad x_3 = 0.2 \sin(2\pi \cdot 1.83 \cdot t) \\ \dot{x}_3 &= 0.2 \cdot 2\pi \cdot 1.83 \cos(2\pi \cdot 1.83 \cdot t) \end{aligned} \quad (4)$$

and the current I is changed from 0 up to 4 A, then Fig. 4 shows the nonlinear MR damper characteristic. The semiactive suspension systems are passive systems, since the power consumption is required only for purposes of changing dissipative force characteristic in real time. Consequently, they cannot become unstable. From another viewpoint, the semiactive suspension does not actively generate energy to the vibratory suspension system but only dissipates energy from it.

Some researchers study the semiactive suspension system as a bilinear system, where the control input β_{semi} is used [25]. In this formulation, the product of the states $(x_4 - x_2)$ and the control input β_{semi} appears in the model: $F = \beta_{semi}(x_4 - x_2)$ [see equations in (1)]. The variable damper β_{semi} is constrained by

$$\beta_{semi}^{\min} \leq \beta_{semi} \leq \beta_{semi}^{\max}. \quad (5)$$

According to a recently applied more practical approach, the semiactive damper is simply modeled as a static map of the deflection speed force, while the control input F has to satisfy the dissipativity and the saturation constraints [Fig. 2 (left)], see [21], [22]. The assumed semiactive quarter car model in (1) contains a virtual input F and measured output $y_{obs} = x_3$. Since the relation is linear, a linear Kalman filter can be used to estimate the states. Notice that the real physical input I (current of the MR-damper) can be expressed from (3) based

on the estimated state and the designed F . The introduction of F reduces the complexity of the problem in the design of F (dissipativity and saturation constraints should be satisfied), however, the estimated state remains available. As a consequence, the controller design problem can be considered as a dynamic optimal control problem for linear system under nonlinear state-dependent control constraints. This paper suggests to solve this problem by employing an explicit MPC as a (sub) optimal controller satisfying control constraints, approximated by a NN to reduce complexity. The NN approach also allows to cope with more complicated nonlinear MR-damper models, where the NN approximation of that model could be solved to deduce the current I for the actuator.

Since the semiactive damper ensures stability, our aim is to achieve performance requirements. In this paper, this is achieved by employing MPC. As its name suggests, one needs a model to predict the future behavior of the plant and the optimization is based on the predicted future of the plant. The semiactive suspension system can be modeled as

$$\begin{aligned} \dot{x} &= Ax + Bu + B_w w, & y_{\text{perf}} &= \dot{x}_4 = C_{\text{perf}} x + D_{\text{perf}} u \\ y_{\text{obs}} &= x_3 = C_{\text{obs}} x \end{aligned} \quad (6)$$

where the state-update and output matrices are

$$\begin{aligned} A &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\omega_{\text{us}}^2 & -2\rho\zeta\omega_s & \rho\omega_s^2 & 2\rho\zeta\omega_s \\ 0 & -1 & 0 & 1 \\ 0 & 2\zeta\omega_s & -\omega_s^2 & -2\zeta\omega_s \end{bmatrix} \\ B &= \begin{bmatrix} 0 \\ \rho \\ 0 \\ -1 \end{bmatrix}, & B_w &= \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, & D_{\text{perf}} &= [-1] \\ C_{\text{perf}} &= [0 & 2\zeta\omega_s & -\omega_s^2 & -2\zeta\omega_s] \\ C_{\text{obs}} &= [0 & 0 & 1 & 0]. \end{aligned} \quad (7)$$

The output y_{perf} (sprung mass acceleration) is used to design the MPC controller, while the suspension deflection y_{obs} is the only measured (observed) output. The quantity y_{perf} will be used as a performance measure later and it is equivalent to \dot{x}_4 . The semiactive damper is modeled as a static map, see Fig. 2. It determines the achievable forces and thus represents constraints. The following dissipating power constraints are considered:

$$\begin{aligned} \text{if } (\dot{x}_3 = x_4 - x_2) &\geq 0 \\ &\bar{\beta}_{\text{semi}}^{\min}(x_4 - x_2) \leq u \leq \bar{\beta}_{\text{semi}}^{\max}(x_4 - x_2) \\ \text{if } (\dot{x}_3 = x_4 - x_2) &\leq 0 \\ &\bar{\beta}_{\text{semi}}^{\max}(x_4 - x_2) \leq u \leq \bar{\beta}_{\text{semi}}^{\min}(x_4 - x_2) \end{aligned} \quad (8)$$

where $\bar{\beta}_{\text{semi}}^{\min}$ and $\bar{\beta}_{\text{semi}}^{\max}$ stand for the normalized damping lower and upper slopes with M_s . The saturation constraints are

$$u_{\min} \leq u \leq u_{\max}. \quad (9)$$

Note that the constraints in (8) are state dependent. Consequently, the current control affects not only the future states of the system but also impacts the future constraints

of the force u through $(x_4 - x_2)$. The range of the achievable control actions depends on the previous history of the control values. The performance index J includes the $y_{\text{perf}} = \dot{x}_4$ performance measure to reduce the vehicle body acceleration, x_1 to keep good road holding, and x_3 to hold the vehicle static weight [21]

$$J = \int_0^{\infty} (q_1 x_1^2 + q_3 x_3^2 + \dot{x}_4^2) dt = \int_0^{\infty} (x^T Q_0 x + y_{\text{perf}}^2) dt \quad (10)$$

with

$$Q_0 = \begin{bmatrix} q_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & q_3 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (11)$$

Substituting \dot{x}_4^2 from the state equations into (10), we obtain the performance function in the usual form

$$J = \int_0^{\infty} (x^T Q x + 2x^T N^T u + u^T R u) dt \quad (12)$$

where

$$\begin{aligned} Q &= \begin{bmatrix} q_1 & 0 & 0 & 0 \\ * & (2\zeta\omega_s)^2 & -2\zeta\omega_s^3 & -(2\zeta\omega_s)^2 \\ * & * & \omega_s^4 + q_3 & 2\zeta\omega_s^3 \\ * & * & * & (2\zeta\omega_s)^2 \end{bmatrix} \\ N^T &= \begin{bmatrix} 0 \\ -2\zeta\omega_s \\ \omega_s^2 \\ 2\zeta\omega_s \end{bmatrix} = B_4 A_{(4,:)}^T \triangleq S_0^T, \quad [R = 1] \end{aligned} \quad (13)$$

and the stars denote symmetric components.

The linearized real quarter-car semiactive suspension parameters are listed in Table I.

The following theorem, adopted from [26], describes the solution of the LQ optimal control problem with constraints:

Theorem 1: Assume the full state measurement is available. Then, the optimal control u for (6) and (7) with the passivity and saturation constraints (8) and (9), and the performance function defined in (12) can be obtained as

$$\dot{P} = -PA(x, P) - A^T(x, P)P + PR(x, P)P - Q(x, P) \quad (15)$$

$$u_{\text{opt}} = \text{sat}[-K_{\text{semi}}(P(t))x] = \text{sat}[-(B^T P(t) + S_0)x] \quad (16)$$

$$J = x_0^T P(0)x_0. \quad (17)$$

A. Clipped Optimal Control

It is important to note that the matrix Riccati differential equation in Theorem 1 cannot be simplified to an algebraic Riccati equation ($P(t) = P$) in spite of tending of the final time to infinity. This is due to the fact that the saturation causes switchings of matrices $A(x, P)$, $R(x, P)$, and $Q(x, P)$ along the trajectory. Therefore, by taking constant matrix $P(t) = P$ and consequently $\dot{P} = 0$ and solving an algebraic Riccati equation, only a *suboptimal* solution is obtained which is called the *clipped optimal* LQ solution in the literature. The name refers to the situation when the desired semiactive

TABLE I
LINEARIZED SEMIACTIVE SUSPENSION PARAMETERS [21], [22]

Parameter	Value	Description
T_s	10 ms	Sampling time
M_s	315 kg	Sprung mass
$M_{u.s}$	37.5 kg	Unsprung mass
k_s	29500 N/m	Suspension stiffness
$k_{u.s}$	208000 N/m	Tire stiffness
β_s	0 N/(m/s)	Suspension damping
β_{semi}^{min}	700 N/(m/s)	Susp. damping lower slope
β_{semi}^{max}	4000 N/(m/s)	Susp. damping upper slope
F^{max}	4000 N	Sat. constraint
x_1	$[-0.05, 0.05]$ m	Tire deflection
x_2	$[-5, 5]$ m/s	Unsprung mass velocity
x_3	$[-0.2, 0.2]$ m	Suspension deflection,
x_4	$[-2, 2]$ m/s	Sprung mass velocity
q_1	1100	Weight on tire deflection
q_2	100	Weight on susp. deflection
A_{road}	$4.9 \cdot 10^{-6}$	Road constant
v	88 km/h	Car velocity

force u is clipped according to (16), whenever it exceeds its passivity or actuator limitation constraints (8) and (9). Note that semiactive force in (16) consists of two parts: one part is the desirable total suspension force $-B^T P(t)x$ and the other part $u_p = -S_0x = -(\omega_s^2 x_3 + 2\zeta\omega_s(x_4 - x_2))$ cancels the passive spring and damper forces.

Without the passivity constraints in (8) for u , the active suspension is obtained. In this case, $P(t) = P$ and the matrix Riccati equation leads to the same algebraic Riccati equation as in the clipped optimal control. The analysis of the semiactive performance index relating to optimal active or passive control leads to the clipped LQ and to the steepest gradient method-based suboptimal control laws, but in this paper only the first one is presented and applied [11], [26]. The following theorem considers the relation between the performance of the optimal semiactive suspension and that of the optimal active suspension [26].

Theorem 2: The cost of the semiactive suspension is always greater than that of the optimal active suspension and the relation can be quantified such as

$$J_{semi} = \underbrace{x_0^T P_a x_0}_{J_{active,LQR}} + \int_0^\infty (u_a - u)^2 dt \quad (18)$$

subject to constraints (8) and (9), where $u_a = -(B^T P(t) + S_0)x$ and P_a is the solution of the following Riccati equation:

$$0 = -P_a(A - BS_0) - (A - BS_0)^T P_a + P_a B B^T P_a - Q + S_0^T S_0. \quad (19)$$

Since the first term in the integral is independent of the control signal, only the second-term (whole integral) minimization is needed, which is not trivial. An approximate solution can be derived by minimizing the integrand only. This approach leads to the clipped LQ suboptimal semiactive control law

$$\begin{aligned} \frac{d}{du} \{(u_a - u)^2\} &= -2(u_a - u) = 0 \quad (20) \\ \frac{d^2}{(du)^2} \{(u_a - u)^2\} &= 2 > 0 \rightarrow \text{minimum} \\ &\Downarrow \\ u &= \text{sat}[u_a]. \end{aligned}$$

B. Discrete-Time MPC and the Kalman Filter

In the real car suspension it is not possible to measure all states. Therefore, an output-feedback controller will be designed. Since MPC requires discrete-time modeling, and because the suspension is a stochastic system, the discrete-time actual Kalman filter, which requires only measurement of the suspension deflection, seems to be an appropriate choice to estimate the states. The state-space model with new discrete-time A , B , and B_w matrices is given by

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + B_w w(k) \\ y_{\text{perf}}(k) &= x_4(k+1) = C_{\text{perf}}x(k) + D_{\text{perf}}u(k) \\ y_{\text{obs}}(k) &= x_3(k) = C_{\text{obs}}x(k) + v(k) \end{aligned} \quad (21)$$

together with constraints (8) and (9) and $x_{\min} \leq x(k) \leq x_{\max}$. If the pair $(A, C_{\text{obs}}A)$ is observable, then a discrete-time actual state estimator can be designed. Since the disturbance (process noise) $w(k)$ from the road and the measurement noise $v(k)$ are stochastic signals, the Kalman filter estimator, which is the most optimal state estimator in case of stochastic signals, is more appropriate choice than the deterministic observer [11]. The actual Kalman filter applies the actual value of the measured output $y_{\text{obs}}(k)$ in the estimation of the state such that better control actions are generated [27]. The white noise road velocity disturbance w is modeled as a discrete-time Gaussian distribution with zero mean and the following standard deviation [21]:

$$\sigma = \sqrt{\frac{2 \cdot \pi \cdot v \cdot A_{\text{road}}}{T_s}}$$

where $A_{\text{road}} = 4.9 \cdot 10^{-6}$, $v = 88$ km/h, and $T_s = 10$ ms. Consequently, the autocovariance matrix is $E[w_k w_l^T] = \sigma^2 \delta_{kl}$. The measurement noise is assumed to be white, zero-mean, and with $E[v_k v_l^T] = 10^{-5} \delta_{kl}$, which corresponds to a very accurate sensor. Noise processes are uncorrelated, thus $E[w_k v_l^T] = 0$ and $E[v_k w_l^T] = 0$. The discrete-time actual Kalman filter is defined by the following dynamical system:

$$\hat{x}(0) = [0 \dots 0]_{n_x}^T \quad (22)$$

$$\hat{x}(k) = F\hat{x}(k-1) + Gy_{\text{obs}}(k) + Hu(k-1)$$

$$F = A - GC_{\text{obs}}A, \quad H = B - GC_{\text{obs}}B \quad (23)$$

where the innovation gain G is looked for such that it yields optimal unbiased and minimum-mean-square-error estimate

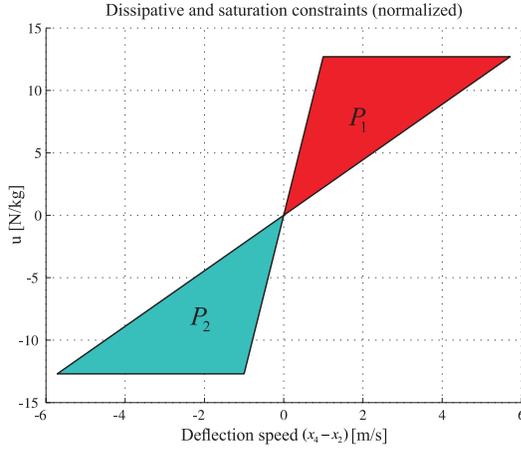


Fig. 5. Normalized dissipative and saturation constraints of the control signal.

\hat{x}_k of the state x_k

$$\begin{aligned} E[\hat{x}_k] &= x_k \Rightarrow E[x_e(k)] = E[x(k) - \hat{x}(k)] = 0 \\ E[(x(k) - \hat{x}(k))(x(k) - \hat{x}(k))^T] &\rightarrow \text{infimum } \forall k. \end{aligned} \quad (24)$$

In the above formulas, we used the same letters for the system matrices as earlier in the continuous-time case but from now on they represent discrete-time matrices. Design of the discrete-time actual Kalman filter in case of time-invariant system and constant covariance matrices can be executed simply based on design of a dual, time-invariant, and discrete-time LQ optimal control system [27]. Notice that the nonlinear constraints are not considered in the Kalman filter design.

Discrete-time implementation of the performance function can be obtained simply by applying the rectangular integration rule with sampling time T_s . Then, the following state-feedback MPC formulation can be defined for the semiactive suspension:

$$\min x_N^T Q_N x_N + \sum_{k=0}^{N-1} x_k^T Q x_k + y_{k,\text{perf}}^T y_{k,\text{perf}} \quad (25a)$$

$$\text{s.t. } x_{k+1} = Ax_k + Bu_k \quad (25b)$$

$$y_{k,\text{perf}} = C_{\text{perf}} x_k + D_{\text{perf}} u_k \quad (25c)$$

$$(x_k, u_k) \in (\mathcal{P}_1 \cup \mathcal{P}_2) \quad (25d)$$

$$x_{\min} \leq x_k \leq x_{\max} \quad (25e)$$

$$x_0 = x(0) \quad (25f)$$

where (25a) is the objective function, (25b) represents the prediction equation, (25c) defines predicted outputs, (25d) describes the passivity and saturation constraints, (25e) accounts for state constraints, and (25f) initializes the optimization problem with the current state measurements, or estimates thereof. Moreover, N is the (normalized) time horizon. In (25), we distinguish the current state $x(k)$ from the predicted state x_k . Moreover, we denote by U_N the open-loop input sequence over the horizon, and Q represents the weighting matrix.

The constraint in (25d) is shown in Fig. 5, which shows the normalized dissipative and saturation constraints for the semiactive suspension. Notice that although the constraint

is nonconvex, it can be described as the union of two polyhedral constraints.

Polyhedron \mathcal{P}_1 :

$$\underbrace{\begin{bmatrix} 0 & 2\zeta_{\max}\omega_s & 0 & -2\zeta_{\max}\omega_s & 1 \\ 0 & -2\zeta_{\min}\omega_s & 0 & 2\zeta_{\min}\omega_s & -1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{H_1} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ u \end{bmatrix} \leq \underbrace{\begin{bmatrix} 0 \\ 0 \\ u_{\max} \end{bmatrix}}_{K_1} \quad (26a)$$

Polyhedron \mathcal{P}_2 :

$$\underbrace{\begin{bmatrix} 0 & -2\zeta_{\max}\omega_s & 0 & 2\zeta_{\max}\omega_s & -1 \\ 0 & 2\zeta_{\min}\omega_s & 0 & -2\zeta_{\min}\omega_s & 1 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix}}_{H_2} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ u \end{bmatrix} \leq \underbrace{\begin{bmatrix} 0 \\ 0 \\ -u_{\min} \end{bmatrix}}_{K_2}. \quad (26b)$$

In the objective function (25a), we approximate the constrained discrete-time infinite-horizon LQ regulation problem (CLQR) as a finite-time optimal control problem (with a short horizon). Such a problem needs to be solved repeatedly in a receding horizon fashion. At each time instant an open-loop finite-time optimal control problem is solved and only the first optimal control command is applied to the process. At the next time step the finite-time optimal control is again solved over a shifted horizon based on new states. This type of the controller is called a receding horizon controller (RHC).

If the finite-time optimal control law is calculated by solving an online optimization at each time step, then the control method is also referred to as online MPC. The CLQR with quadratic or piecewise linear (1-norm, ∞ -norm) performance index implies QP or LP that can be solved online by efficient tools based on active-set or interior-point methods.

Several researchers recognized that the constrained finite-time optimal control (CFTOC) with the choice $Q_N = P_\infty$, where P_∞ is the solution of the unconstrained infinite-horizon LQ problem, sometimes also yields the solution of CLQR [4]–[10]. The set of initial conditions $x(0)$ for which the equivalence holds, depends on the length of the horizon N .

Several algorithms exist to compute the sufficiently long horizon N for any compact set of the initial states such that a CFTOC solves the infinite-time CLQR problem, assuming the constraints are inactive for $k \geq N$. In such a case the cost from N to ∞ can be calculated by $x(N)Q_N x(N)$, where Q_N equals the solution of the unconstrained infinite-horizon Riccati equation ($Q_N = P_\infty$). These algorithms usually yield large horizons N , which leads to complex optimization problems.

To model the nonconvex passivity constraint (25d) in a computationally tractable manner, we propose to employ the concept of PWA models, which are discussed in the subsequent section. Furthermore, the solution of the constrained MPC optimization problem including the discrete-time actual Kalman filter is very complicated. Therefore, we design the controller separately from the Kalman filter.

Specifically, we will derive an explicit MPC controller where we assume the measurement states are available, and we compute separately the state estimator (Kalman filter) for the controller implementation. The idea comes from LQG control area where the separation principle proves the optimal solution if this technique is applied. Whether or not such a separation principle holds for the assumed framework that employs an NN approximation of the MPC controller is an open problem that remains to be proved rigorously. In this paper, we assume the principle is valid based on LQG theory.

III. PIECEWISE AFFINE SYSTEMS

PWA dynamical systems belong to the class of hybrid systems [28]–[30], which combine continuous dynamics with discrete logic. Simply put, PWA systems allow the state-update equation (the continuous component) to take different expressions in different parts of the state–input space. The association of a particular state–input pair to a particular region is driven by logic conditions. In particular, consider a dynamical system with states $x \in \mathbb{R}^n$ and control inputs $u \in \mathbb{R}^m$. Then, the PWA representation of such a system is given by

$$x^+ = \begin{cases} A_1x + B_1u, & \text{if } (x, u) \in \mathcal{R}_1 \\ \vdots \\ A_px + B_pu, & \text{if } (x, u) \in \mathcal{R}_p \end{cases} \quad (27)$$

where x^+ denotes the successor state, p is the number of different realizations of the state-update equation, and \mathcal{R}_i is the region of the state–input space, where the i th state-update equation is valid. For the PWA system in (27) to be well posed, we must have that $\mathcal{R}_i \cap \mathcal{R}_j = \emptyset$ for all $i \neq j$. In other words, the corresponding regions of validity must not overlap.

It is important to note that the regions of validity, i.e., \mathcal{R}_i , can naturally include any state and/or input constraints which the PWA system is supposed to respect. To convert the IF-THEN rules of (27) into a computationally tractable form, one can proceed by defining binary selectors $\delta_i \in \{0, 1\}$ for $i = 1, \dots, p$ such that

$$(\delta_i = 1) \Leftrightarrow (x, u) \in \mathcal{R}_i. \quad (28)$$

If the regions \mathcal{R}_i are polytopes given by $\mathcal{R}_i = \{(x, u) \mid H_i(x^T, u^T)^T \leq h_i\}$, (28) can be rewritten, as suggested by [31], into

$$H_i \begin{pmatrix} x \\ u \end{pmatrix} - h_i \leq M(1 - \delta_i), \quad i = 1, \dots, p \quad (29a)$$

$$\sum_{i=1}^p \delta_i = 1 \quad (29b)$$

where M is a sufficiently large constant. Note that all constraints in (29) are linear in the corresponding variables. The exclusive-or condition in (29b) guarantees that only one binary selector is active for each state–input pair. With such a property the PWA state-update equation from (27) can be compactly written as

$$x^+ = \sum_{i=1}^p \delta_i (A_i x + B_i u) \quad (30)$$

which, however, is nonlinear due to a product between δ_i and the states/inputs. Such a nonlinearity can be avoided [31] by employing basic rules of propositional logic

$$x^+ - (A_i x + B_i u) \leq M(1 - \delta_i) \quad (31a)$$

$$x^+ - (A_i x + B_i u) \geq -M(1 - \delta_i). \quad (31b)$$

It is easy to verify that if $\delta_i = 1$, then (31) reduces to $x^+ = A_i x + B_i u$. On the other hand, if $\delta_i = 0$, the constraints in (31) are inactive since M is assumed to be sufficiently large.

To establish the relation between the MPC problem (25) and the PWA modeling framework, two important properties of the models in (27) are worth noting. First, evolution of the PWA system is only defined for state–input pairs, which reside in the union of corresponding regions of validity, i.e., in $\cup_i \mathcal{R}_i$. Second, even though each region \mathcal{R}_i is assumed to be a polytope (hence, a convex set), their union $\cup_i \mathcal{R}_i$ can be nonconvex. Therefore, the nonconvex passivity constraint in (25d) can be embedded into the PWA framework, together with state-update and output equations (25b) and (25c) as follows:

$$x_{k+1} = \begin{cases} Ax_k + Bu_k, & \text{if } (x_k, u_k) \in \mathcal{P}_1 \\ Ax_k + Bu_k, & \text{if } (x_k, u_k) \in \mathcal{P}_2 \end{cases} \quad (32a)$$

$$y_{k,\text{perf}} = \begin{cases} C_{\text{perf}}x_k + D_{\text{perf}}u_k, & \text{if } (x_k, u_k) \in \mathcal{P}_1 \\ C_{\text{perf}}x_k + D_{\text{perf}}u_k, & \text{if } (x_k, u_k) \in \mathcal{P}_2. \end{cases} \quad (32b)$$

Here, \mathcal{P}_1 and \mathcal{P}_2 are defined by (26). Even though identical state-update and output equations are used in (32), the domain of validity of such a model is restricted to the union of \mathcal{P}_1 and \mathcal{P}_2 , which, as is evident from Fig. 5, is nonconvex.

One can thus replace constraints (25b)–(25d) by (32)

$$\min x_N^T Q_N x_N + \sum_{k=0}^{N-1} x_k^T Q x_k + y_{k,\text{perf}}^T y_{k,\text{perf}} \quad (33a)$$

$$\text{s.t. } x_{k+1} = \begin{cases} Ax_k + Bu_k, & \text{if } (x_k, u_k) \in \mathcal{P}_1 \\ Ax_k + Bu_k, & \text{if } (x_k, u_k) \in \mathcal{P}_2 \end{cases} \quad (33b)$$

$$y_{k,\text{perf}} = \begin{cases} C_{\text{perf}}x_k + D_{\text{perf}}u_k, & \text{if } (x_k, u_k) \in \mathcal{P}_1 \\ C_{\text{perf}}x_k + D_{\text{perf}}u_k, & \text{if } (x_k, u_k) \in \mathcal{P}_2 \end{cases} \quad (33c)$$

$$x_{\min} \leq x_k \leq x_{\max} \quad (33d)$$

$$x_0 = x(0). \quad (33e)$$

Finally, constraints (33b) and (33c) can be translated into a set of mixed-integer inequalities by introducing binary selectors $\delta_{1,k}$ and $\delta_{2,k}$, for each step of the prediction horizon, per (29) and (31). Then, the optimization problem (33) becomes a mixed-integer QP (MIQP) in the decision variables u_0, \dots, u_{N-1} and $\delta_{1,k}, \delta_{2,k}, k = 0, \dots, N-1$.

However, due to presence of binary selectors, solving MIQP problems of the form of (33) is not trivial. In the worst case, the solution needs to explore all feasible combinations of binary variables. In our formulation, at each step of the prediction horizon we have two binary selectors. Thus, the total is $2N$, which gives the worst case runtime complexity of (33) as $\mathcal{O}(2^{2N})$. Such a complexity often prohibits practical implementation, since the MIQP optimization problem needs to be solved repetitively at each sampling instant. To reduce

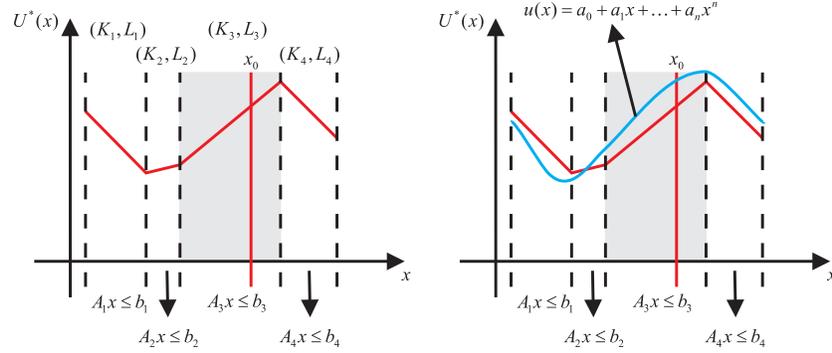


Fig. 6. Left: searching for the polyhedra containing x_0 . Right: polynomial approximation of the explicit control law.

the implementation effort, we therefore propose to obtain a so-called explicit representation of the optimal solution to (33) using multiparametric programming, which is discussed in the subsequent section.

IV. EXPLICIT MPC

The idea of explicit MPC is to obtain an explicit representation of the MPC feedback law $u_0^*(x_0)$ which maps states onto optimal control inputs. This explicit dependence can be obtained by multiparametric programming that constructs the analytic solution to a particular optimization problem for the whole range of feasible initial conditions. In particular, consider a general optimization problem of the form

$$\min f(z, \theta) \quad (34a)$$

$$\text{s.t. } g(z, \theta) \leq 0 \quad (34b)$$

$$h(z, \theta) = 0 \quad (34c)$$

where z is the decision variables, θ is the parameters (the initial conditions), $f(\cdot)$ is the objective function to be minimized, and $g(\cdot)$ and $h(\cdot)$ are specify constraints. The objective is to derive the explicit dependence of the optimizer z^* on the parameters θ , i.e., to obtain the analytic form of the function $z^*(\theta)$.

Deriving the analytic form of $z^*(\theta)$ is difficult in general. However, if the vector of decision variables z is composed of real numbers, $f(\cdot)$ is a linear or a quadratic function, and if $g(\cdot)$ and $h(\cdot)$ are linear functions, then such an explicit solution can be obtained offline by utilizing Karush–Kuhn–Tucker optimality conditions.

Theorem 3 [4]: Consider the parametric QP

$$z^* = \arg \min \{z^T H z \mid G z \leq w + E \theta\}. \quad (35)$$

Then, the optimizer $z^*(\theta)$ is a PWA function

$$z^*(\theta) = \begin{cases} a_1\theta + \beta_1, & \text{if } \theta \in \mathcal{R}_1 \\ \vdots \\ a_q\theta + \beta_q, & \text{if } \theta \in \mathcal{R}_q \end{cases} \quad (36)$$

where \mathcal{R}_i , $i = 1, \dots, q$ are critical regions, and q denotes the total number of critical regions.

The benefit of obtaining such an analytic solution for the MPC optimization problem in (33) stems from the subsequent

fast online implementation. Specifically, once the optimal control inputs are characterized as a PWA function in (36), the value of the optimal control input can be obtained easily just by evaluating the function. This can be done substantially faster than solving (33) as a MIQP.

However, Theorem 3 cannot be readily used to obtain an explicit representation of the receding horizon feedback law, since the MPC problem (33) features binary optimization variables. To circumvent this limitation, one can explicitly enumerate, completely offline, all feasible combinations of such binary variables. Then, once each binary combination is fixed, problem (33) translates to a QP in purely real decision variables u_0, \dots, u_{N-1} . For such a QP Theorem 3 can be applied to obtain the optimizer $u_0^*(x_0)$ as a function of the initial states x_0 , see (33e). By exploring each feasible combination of the binary selector one therefore obtains a whole set of optimizers, each defined over its set of critical regions. We remark that the critical regions, along with the associated affine expressions of the optimizer, can be computed, e.g., by the Multi-Parametric-Toolbox (MPT) toolbox [32].

To identify the value of the optimal control input associated to a particular state measurement, one then needs to identify the critical region which contains the initial state. This procedure is called the point-location problem. The simplest algorithms for the point-location problem are the sequential and binary tree [12] approaches, respectively. The first method traverses the regions in a predetermined order until the correct region is found. The second-method constructs and evaluates a binary tree, which allows for faster region identification [Fig. 6 (left)].

Unfortunately, the computation of explicit MPC controllers scales badly with increasing problem size. From a practical perspective, the procedure is applicable for systems with up to four state variables. Furthermore, we will see in the simulation that another large drawback of the explicit (offline) control law is that the number of polyhedral regions grows dramatically with the prediction horizon and the number of constraints which decreases the practical applicability in embedded systems.

For this reason, a lot of efficient searching and storage algorithms have been developed [12]–[14], [33]–[37]. In [17], the key idea is that the optimal explicit PWA controller is approximated by a single polynomial [Fig. 6 (right)], where

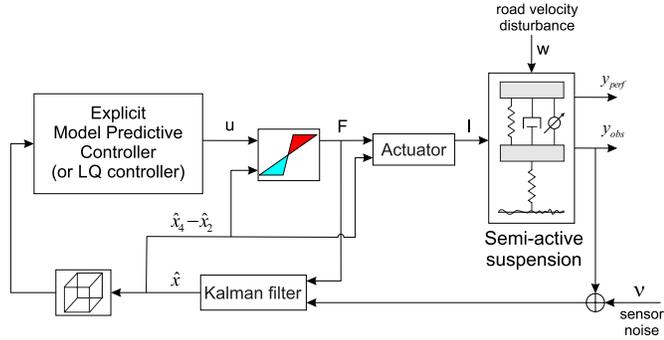


Fig. 7. Output-feedback control system with Kalman filter for semiactive suspension.

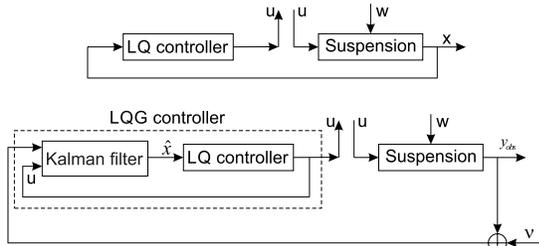


Fig. 8. Cutting point at the plant input.

number of the coefficients to be stored does not depend on the number of the regions. This type of controller does not require region storage and region identification. They prove the stability can be guaranteed and the constraints can also be satisfied.

The following sections will present an NN-based approximation of the explicit MPC controller and will analyze the whole semiactive suspension control system. To increase practical applicability of the control system first discrete-time actual Kalman filter is designed to the suspension model.

V. DISCRETE-TIME ACTUAL KALMAN FILTER DESIGN

The control system for the explicit MPC with Kalman filter is given in Fig. 7. The suspension deflection $y = x_3 = C_{obs}x$ is considered as the measured output of the semiactive suspension. Estimated states are bounded by the state constraints and, after calculation of the control signal, the dissipative and saturation constraints are applied. Finally, the actuator delivers the corresponding current to the MR damper in the suspension system based on the input control force. Modeling of the actuator and its inclusion into the control system is not the topic of this research article. The Kalman filter is designed for the linear suspension model without constraints [27]. The computed innovation gain equals to $G = [-0.1275 \ -31.3230 \ 0.6482 \ -0.0529]^T$. Fig. 9 shows the Bode diagram of the suspension plant and the Bode diagrams of the open-loop control systems in case of LQG controller and LQ state-feedback controller according to the cutting point in Fig. 8. Fig. 9 shows that if the same robustness property was required for LQG controller as for the LQ state-feedback controller, then the loop transfer recovery (LTR) method should remove effect of the zeros. It is important

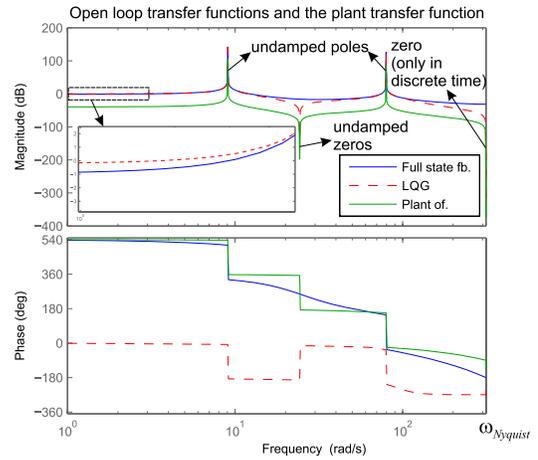


Fig. 9. Bode plot of the discrete transfer functions.

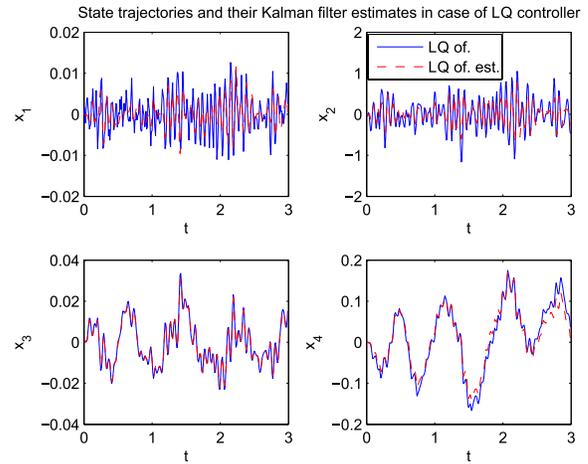


Fig. 10. State trajectories of the LQG control.

to note that robustness and the output disturbance rejection properties of a control system have tradeoff relationship with each other. That is, the improvement of the robustness will decrease output disturbance rejection properties of the control system. Furthermore, the state estimate feedback can destroy the phase margins (robustness property) but in contrary it can improve the measurement noise performance of the closed-loop system. Fig. 10 shows the estimated states of the LQG control when velocity disturbance is applied described in Section II-B. The simulation takes into consideration the constraints but activation of the constraints depends on road velocity disturbance. It can be observed that the estimated states track changes of states of the suspension well. However, in case of x_1 and x_2 the signals change so fast that the estimated states cannot reach the maximal amplitude of states of the suspension. Here, and in the sequel, of. means output feedback and of.est. means feedback of the estimated state. Fig. 10 also shows that the Kalman filter estimates the measured state x_3 most precisely. Fig. 11 compares estimated states of the Kalman filter and states of the full state-feedback LQ control. LTR technique can be used to recover the original properties (e.g., stability, states of the full state feedback) of the suspension. However, rejection of the measurement noise

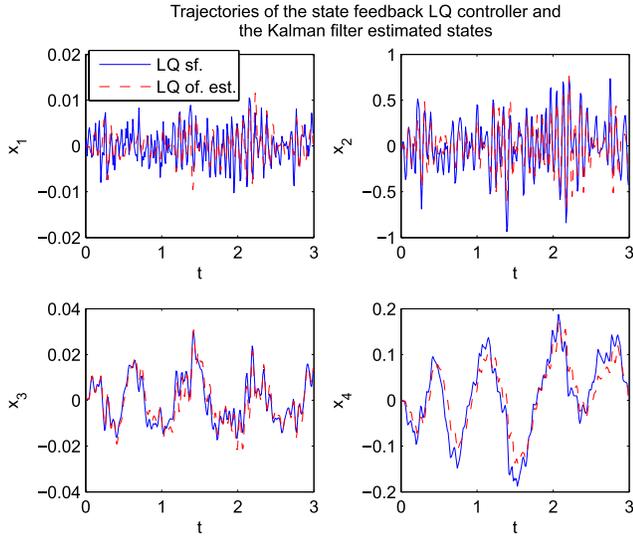


Fig. 11. Comparison of state trajectories of the LQG and the full state-feedback LQ control.

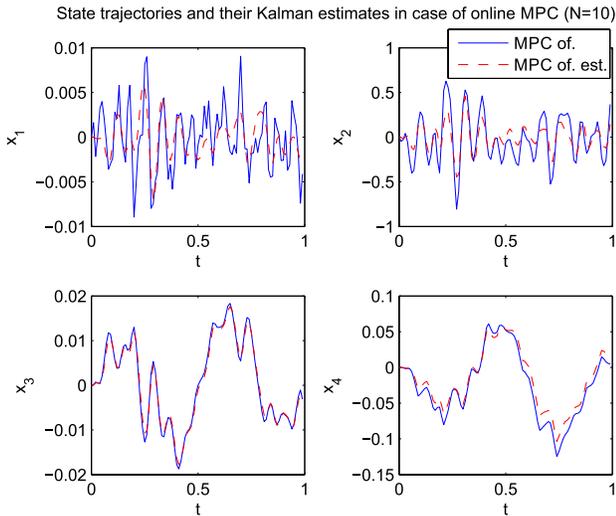


Fig. 12. State trajectories under MPC with Kalman filter.

is better if LTR are not used. Stability inherently follows from the physical setup. To check if the Kalman filter is a suitable state estimator for MPC, we depicted the state trajectories of the MPC ($N = 10$) with Kalman filter in Figs. 12 and 13. The results show that the Kalman filter is a good choice to estimate states of the suspension.

VI. EXPLICIT MPC-BASED NEURAL NETWORK CONTROLLER DESIGN

Csekő *et al.* [11] analyzed in detail, the optimal explicit MPC approach for the quarter car semiactive suspension model. The explicit MPC approach is a promising method to increase the practical applicability of MPC for systems where the time-consuming online optimization is prohibitive. Furthermore, the optimal MPC approach does have a linear state-feedback form. Two main disadvantages of the explicit MPC method are the exponential blow-up of the number

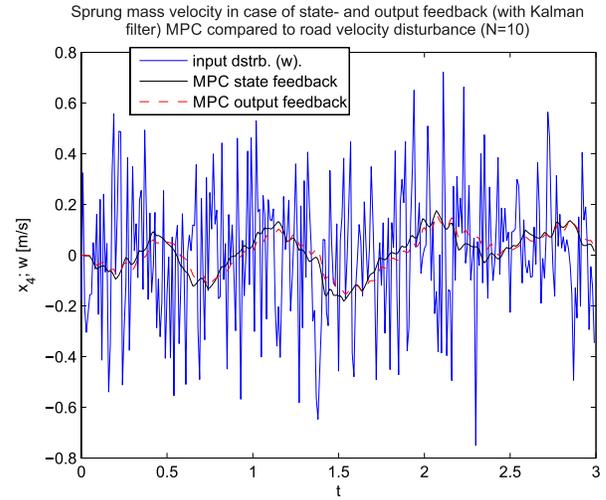


Fig. 13. Control of the sprung mass velocity with output-feedback MPC.

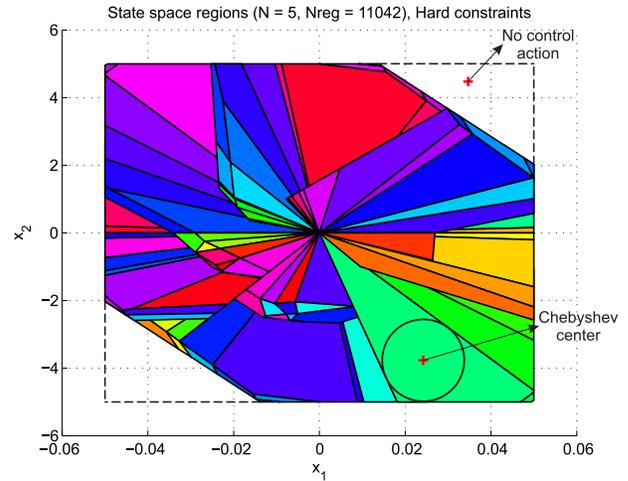


Fig. 14. Slices $x_1 - x_2$ of the optimal solution regions in case of hard constraints $N = 5$.

of regions, and the requirements of the having the full state measurement available. In this section, we create the RBFs-based NN approximation of the explicit MPC controller which gives a possibility to treat the number of regions problem. Moreover, we will also see that the Kalman filter design provides a solution to the second problem. In explicit MPC, the regions usually only cover a subset of state constraints, see Fig. 14. However, when the state observer is designed independently from the controller, or when disturbances or modeling uncertainties appear, the states might leave the controller regions. To treat this problem the paper [11] suggested the soft constraints approach and to combine MPC with clipped LQ controller. If soft constraints are applied then the whole space of state constraints will be filled out totally, as shown in Fig. 15. This solution also shows that the original regions remain the same and are completed by new regions at the corners. The price of the soft constraints is that the state trajectory started from the originally infeasible regions will violate the hard constraints. Moreover, the number of regions is increased as well.

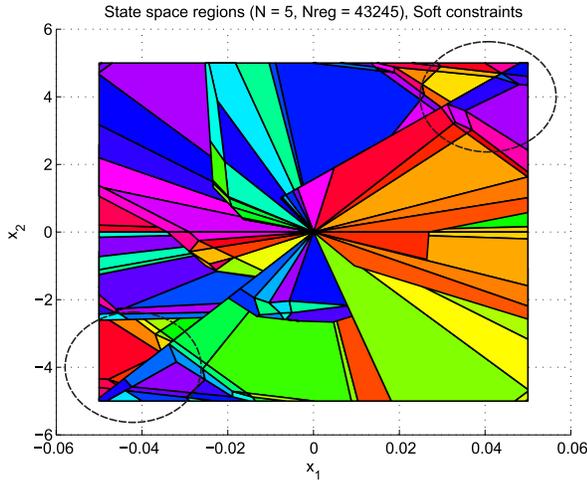


Fig. 15. Slices $x_1 - x_2$ of the optimal solution regions in case of soft constraints $N = 5$.

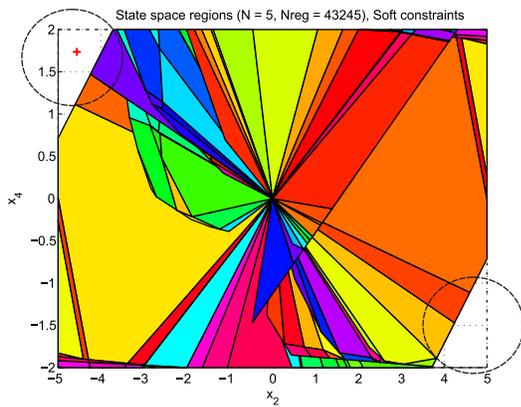


Fig. 16. Dissipative and saturation hard constraints cause cuts for slices $x_2 - x_4$ in case of soft constraints $N = 5$.

The learning set-based approximation methods, such as NNs, also require that the controller is defined for all feasible states to create the appropriate training set. On the basis of the soft-constraints method the generation of the training set would be possible and simple. However, the dissipative and saturation hard constraints (Fig. 5) can also complicate the creation of the training set (Fig. 16). The problem of missing control actions can be solved using combined controllers. The combined explicit MPC/LQ controller was suggested in [11] to handle this problem because the LQ controller fits into the explicit MPC framework very well. One can see that the number of the regions can be extremely large. This property is inherent to the whole approach of parametric programming where the very central idea of explicit MPC is to enumerate all possible combinations of active constraints. Since there can be exponentially many of them as a function of the prediction horizon, an exponential growth in the number of regions can be observed in the worst case. Specifically, the upper bound on the number of regions is $2^{(\text{number of binary variables in the MPC problem})}$, where the (number of binary variables in the MPC problem) = (prediction horizon) \times (number of binary variables in the PWA model).

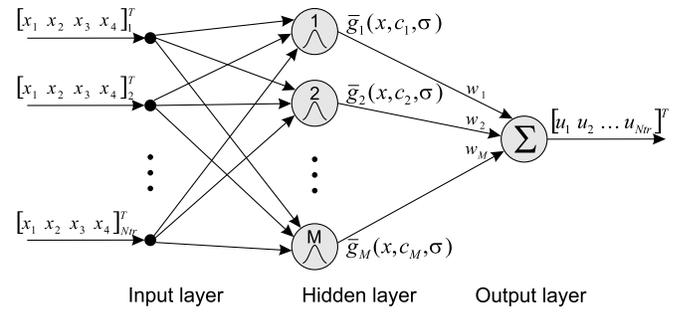


Fig. 17. Structure of the RBF NN.

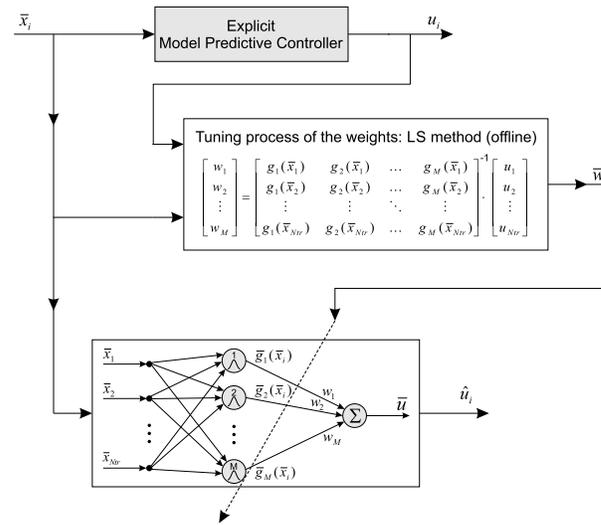


Fig. 18. Scheme to obtain approximated explicit MPC with RBF NN.

It can be shown that when a quadratic performance index is considered along with binary variables, overlapping regions may arise. In the overlapping regions, the control action has to be chosen carefully as to minimize the value of the performance objective. The enormous number of the regions in the explicit MPC decreases the applicability for real systems since the online searching among the regions can take a long time. To mitigate this issue, we propose to use the RBF type NN to approximate the explicit MPC controller. The RBF NNs have similar properties as the MLP networks but they have simpler structure and the back-propagation learning is not required [20]. Fig. 17 shows the structure of the RBF NN which we will use to approximate the explicit MPC. The input–output training set consists of the state vector samples and the corresponding control actions. The RBF NN approximation of explicit MPC is shown in Fig. 18. Our NN applies Gaussian basis functions with the center (c_j) and a width (σ) parameters. The same σ will be used for all neurons since, usually, the approximation is not particularly sensitive to the values of σ . The basis functions are defined by $g_j(\bar{x}_i) = \exp[-(\|\bar{x}_i - \bar{c}_j\|^2/2\sigma^2)]$. We tune only the weights in the output layer and so the weights can be calculated simply

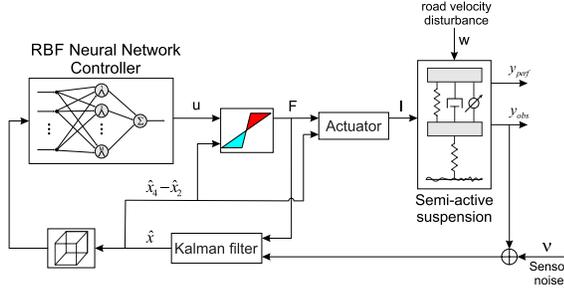


Fig. 19. NN control system with Kalman filter for the semiactive suspension.

using the Moore–Penrose pseudoinverse offline

$$\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_M \end{bmatrix} = \begin{bmatrix} g_1(\bar{x}_1) & \dots & g_M(\bar{x}_1) \\ g_1(\bar{x}_2) & \dots & g_M(\bar{x}_2) \\ \vdots & \ddots & \vdots \\ g_1(\bar{x}_{Ntr}) & \dots & g_M(\bar{x}_{Ntr}) \end{bmatrix}^{-1} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{Ntr} \end{bmatrix}. \quad (37)$$

After training of the NN the approximated control action can be represented in a nonlinear state-feedback form

$$u_k = e^{-\frac{\|\bar{x}_k - \bar{c}_1\|^2}{2\sigma^2}} w_1 + e^{-\frac{\|\bar{x}_k - \bar{c}_2\|^2}{2\sigma^2}} w_2 + \dots + e^{-\frac{\|\bar{x}_k - \bar{c}_M\|^2}{2\sigma^2}} w_M \quad (38)$$

which can be compared with the polynomial nonlinear state-feedback controller mentioned at the end of Section IV

$$u_k = a_0 + a_1^T x_k + a_2^T x_k^2 + \dots + a_n^T x_k^n \quad (39)$$

where $a_0 \in \mathbb{R}$ and $a_i \in \mathbb{R}^m$, $m = 4$ for $i = 1, \dots, n$. The NN control system with Kalman filter is shown in Fig. 19.

More possibilities exist to generate the training set, such as uniform, Chebyshev center, and grid-based sampling. Sampling may happen either in the set of the polyhedron regions or inside the whole state constraints. In the last case, the combined controller has to be used to generate the training control actions. For example, outside of the polyhedron regions the LQ controller can be a good choice. If the NN controller is trained in the whole constraint set, then a unified controller can be used for the suspension. However, if the polyhedral regions provide training points of the control action, then we have two possibilities for the final controller: the combined explicit MPC-based NN/LQ controller or alone the explicit MPC-based NN controller. The uniform sampling means the training set is generated uniformly inside the state constraints or only inside the set of the polyhedral regions. In this case, some regions may not contribute in generation of the training set at all, while other regions may provide many training points. The Chebyshev center sampling eliminates this problem and takes one training point from each of the polyhedrons. The Chebyshev center of a polyhedron is the center of the largest inscribed ball (Fig. 14). The grid sampling can be performed either in the whole state constraint set or only in the polyhedral set. The latter is supported by the MPT toolbox but it is a very time consuming task if we have lot of polyhedron regions. The algorithm creates first a rectangular grid and then it throws away the training points which do not belong to polyhedron. The MPT toolbox also allows to generate the Chebyshev centers of each controller region. Large amount of the training points is needed to cover the large

number of polyhedra. However, not all the training points can be used as center point of the Gaussian basis functions (c_j) because it would result in too many neurons, increasing the computing time unnecessarily. Good approximation can also be reached if the training points are revised according to some clustering algorithms and the center points of the Gaussian basis functions are ordered to these cluster center points. We will use the most known clustering procedure, the so-called *K-means clustering* for the Chebyshev-based sampling. In case of uniform sampling, the cluster centers will be chosen from the uniformly distributed training points so that we take every n th training point. Center points will be also training points for this case but for *K-means clustering* it is not ensured since it depends on which method is applied to determine center of the clusters. The grid point-based training set is not investigated in this article although we investigated earlier another type of the approximation, namely, the singular value decomposition fuzzy approximation [38]–[40], which only works on rectangular grid training points, however, it did not yield the expected results. The uniformly distributed training points provides better excitation for the projected state variables than the grid points. The above discussion also shows that many possibilities exist to generate, handle training points and to design different explicit NN-based controller. Before presenting the simulation results of the NN controller, the essential steps of the *K-means clustering* algorithm are summarized shortly based on [20] and the help of *k means* in MATLAB.

- 1) Choose K cluster center randomly.
- 2) Cluster the training points according to their distance from the cluster points, e.g., by considering the squared Euclidean or absolute ℓ_1 distance.
- 3) Determine the new cluster centers. For example, define each center as the mean of the points in that cluster or each centroid is the component-wise median of the points in that cluster.
- 4) Reassign the training points to the new centers and continue the procedure from 1 to 3 until $\min \sum_{k=1}^K \sum \text{dist}(x_i, c_k)$ does not change, where c_k denotes the cluster centers and x_i denotes the training points or a given iteration number has reached.
- 5) This is the step of *online updates* where points are individually reassigned if the sum of distances are reduced. After each reassignment the cluster centers are recalculated and each iteration consists of one pass all the points.

It is not easy to find the global minimum for *K-means clustering*. The following dictionary explains the names and notions which we will use thereafter.

- 1) N and N_{tr} denote the prediction horizon and the number of training points, respectively.
- 2) *hard constr.* and *soft constr.* tell that hard or soft constraints were used to derive the explicit MPC controller.
- 3) *Chebyshev*, *Cheb.* mean that the Chebyshev centers provide the training points.
- 4) ℓ_1 , *abs.*, absolute tell that the ℓ_1 distance is used to determine the cluster centers.

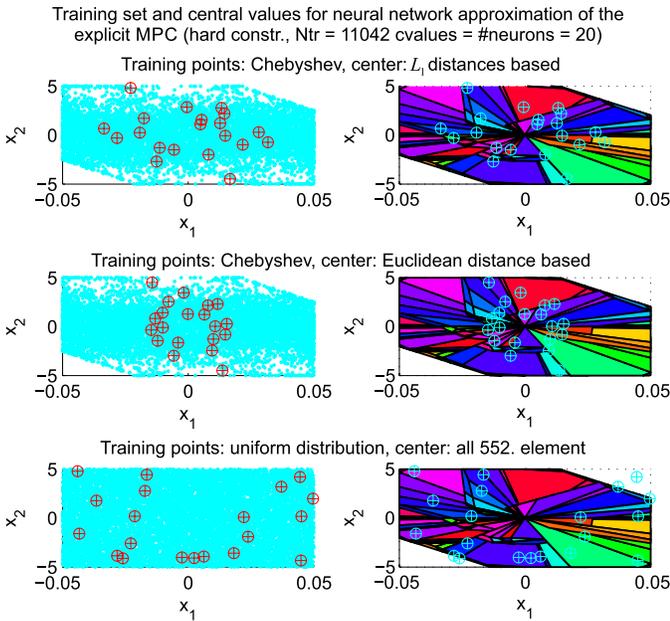


Fig. 20. Example of the training points and centers of the Gaussian activation functions of the neurons.

- 5) Euclidean, Euc say that the Euclidean distance is used to determine the cluster centers.
- 6) *unif.* denotes that the training set is generated using uniform distribution and we take, for example, all 50th elements to create center points for the neurons.
- 7) *cvalues.* means the center values for basis function of the neurons.
- 8) *of.* means output feedback: e.g., MPCof means output-feedback MPC controller (i.e., MPC controller with Kalman filter).

Furthermore, in the lack of MPC control action for a given state we use the LQ control action. This statement is valid for the training set and for the control system as well.

Fig. 20 compares three types of generating training sets and creating cluster centers on the state slice $x_1 - x_2$. Generating the training points according to uniform distribution and selecting the center values uniformly from that set provide the spreadest covering on the slice. Training points are generated inside the interval of the variables as 1-D vectors with uniform distribution. Then these vectors are put next to each other to create higher dimensional distribution. One can observe that the training points and the center points of the Gaussian basis functions can lay outside of MPC control regions. The Chebyshev center-based sampling with Euclidean distance-based cluster center shows narrow covering. The Chebyshev center-based sampling with absolute (l_1) distance falls between the two previous mentioned techniques regarding the spreading of the cluster centers, i.e., the mean value of the distance between cluster center and its nearest cluster center. The explicit MPC with hard or soft constraints and with prediction horizon $N = 5$ will be approximated with NN controller. In case of hard constraints the number of regions is 11042 and in the case of soft constraints it is 43 245. To be able to compare the NNs based on different sampling and clustering methods these numbers were considered

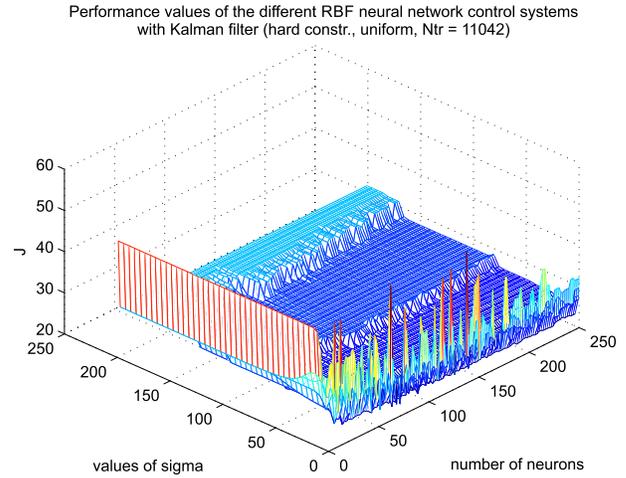


Fig. 21. Performance values of the different control systems with respect to σ and the number of neurons.

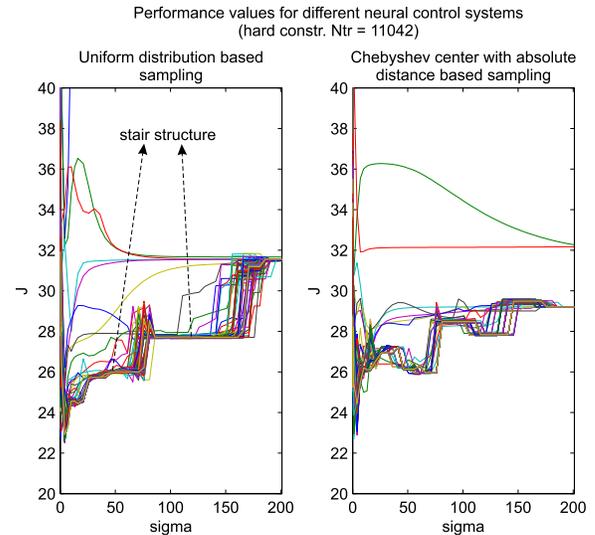


Fig. 22. Performance values of the different control systems projected to values σ .

to be the number of training points. After defining of the training set and the centers of the Gaussian basis functions in the neurons many NNs were designed, where the number of neurons and the value of σ was changed. The σ is varied according to $[0.2:0.2:0.8 \ 1:3:13 \ 16:5:201]$, while the number of neurons is changed according to $[2:2:250]$ in MATLAB notation. It is known that the RBF NNs are not very sensible to the value of σ and usually some heuristical methods based on the cluster points are used to choose them. Here, σ was chosen based on the performance curve of the NNs. Fig. 21 depicts performance values of the different NNs when 20-s long road velocity disturbance were applied to the suspension system. Only the performance surface of the uniform distribution-based RBF NNs is presented because the other type of sampling and clustering methods yield very similar stair structure view (Fig. 22). Notice that the performance has cost character according to (25a). We tried many different design techniques regarding training and center points to create NN controller. All of them exhibited

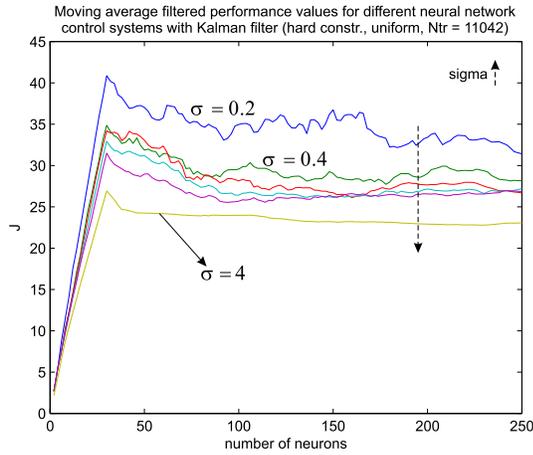


Fig. 23. Performance values of the different control systems projected to number of neurons.

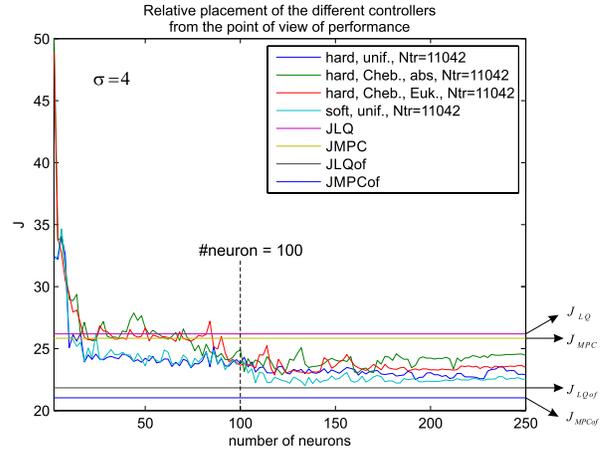


Fig. 25. Comparison of performances of the NNs based on different sampling and clustering techniques.

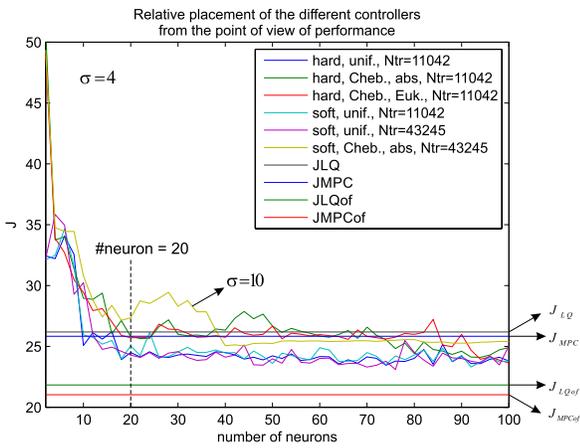


Fig. 24. Comparison of performances of the NNs based on different sampling and clustering techniques.

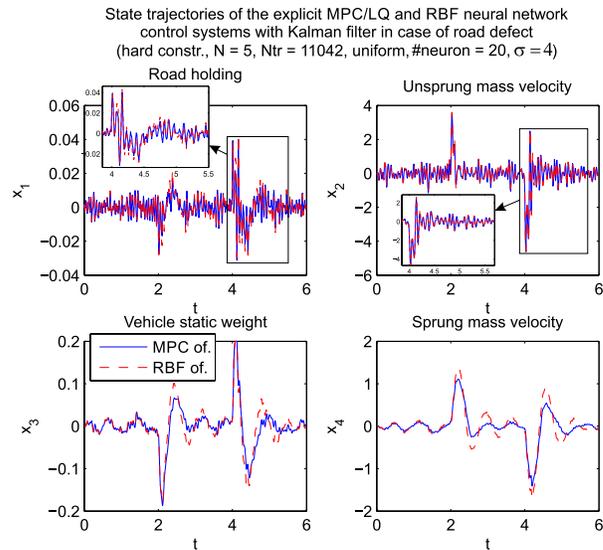


Fig. 26. State trajectories of the explicit MPC/LQ and RBF NN control systems.

property that if the value σ was increased, then the curves of the performance decreases up to a certain σ , and later they would increase again. Fig. 23 shows only the performance curves which are obtained in the decreasing phase. To make the comparison easier, a moving average filter was used with window size 15. Except of the NN type parameterized by soft constraints, Chebyshev center training set, absolute distance (ℓ_1)-based clustering, and $N_{tr} = 43245$ we obtained that the other type of the NNs have the minimal performance curve at $\sigma = 4$, while the exception type of the NNs has the optimum at $\sigma = 10$. These σ values are used in Figs. 24 and 25. The reason why the NN with Chebyshev center training set, Euclidean distance (ℓ_2)-based clustering, and $N_{tr} = 43245$ is not presented is that in this case the Euclidean clustering had convergence problems in case of *K-means clustering*. Because of the very large computation time and to fulfill our goal, namely, that number of neurons in the approximator do not exceed 100, the soft constraints-based NNs with $N_{tr} = 43245$ were calculated only up to 100 neurons (Fig. 24). However, as can be observed in Fig. 25, performance curves of the Chebyshev-based NNs start to decrease just at 100. Therefore, we present the results for the other NN up to 250 neurons. The figures show that under 100 neurons the uniform

distribution-based NN approximators are the preferred ones but over 100 there do not exist significant differences between the approximators. We also depicted performance curves of the LQ, MPC without and LQ, MPC with Kalman filter controllers to compare the relative placement of the performances. It can be seen well that none of approximators can reach performance curves of the LQ output feedback and MPC output-feedback controllers (see J_{LQof} and J_{MPCof} , respectively) but they are better than the state-feedback controller (see J_{LQ} and J_{MPC} , respectively) which probably may thank to the Kalman filters since the NN controller approximators were trained for the state-feedback controllers. On the basis of the figures, 20 neurons seem to be a good choice for the uniform distributed sampling-based NN, and 100 neurons for the Chebyshev with Euclidean distance-based network. Based on this latter choice one can conclude to uniform distributed sampling-based NN at 100 neurons because they have very similar performance value at 100. Next, we investigate only these two types of NNs. In Fig. 26, we compare the designed

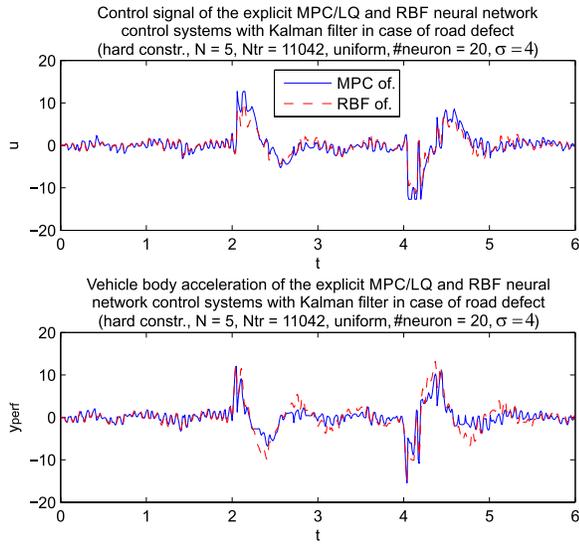


Fig. 27. Control signal and vehicle body acceleration of the explicit MPC/LQ and RBF NN control systems.

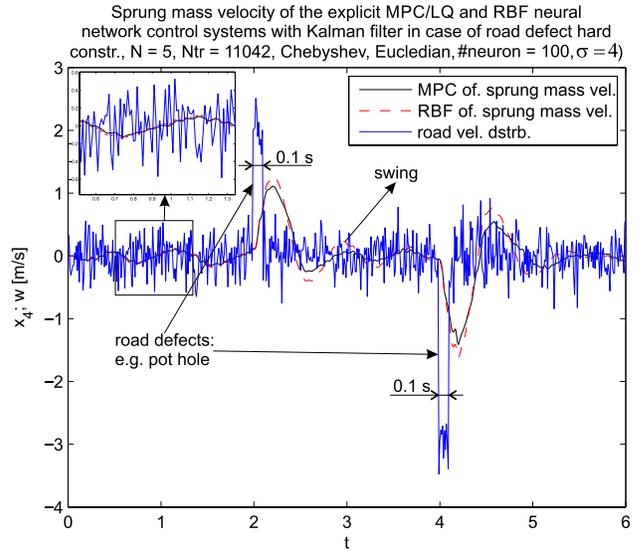


Fig. 29. Sprung mass velocity of the explicit MPC/LQ and RBF NN control systems.

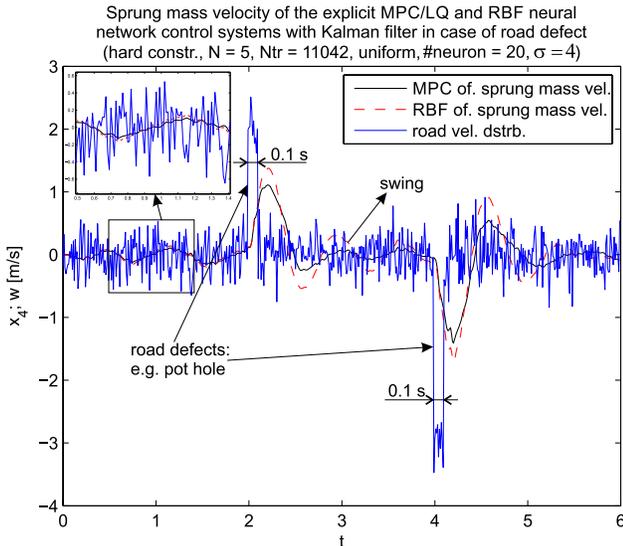


Fig. 28. Sprung mass velocity of the explicit MPC/LQ and RBF NN control systems.

output-feedback NN and explicit MPC/LQ controllers when beside of the normal road velocity disturbance bigger road defects (e.g., pot hole) also beats the tire. The impulse is 0.1-s long and it happens first in the positive direction then in the negative direction. Both controllers eliminate the disturbance in effective way and the controllers behave similarly, but the NN controller can return only after longer swing to the normal state which can also be observed on the sprung mass velocity in Fig. 28. The control signal and the performance outputs are also presented in Fig. 27. The saturation constraint limits the control signal at the second impulse. Essentially, the NN with Chebyshev sampling and Euclidean clustering in Fig. 29 provides similar results as the NN with uniformly distributed sampling and clustering in Fig. 28. Therefore, only the sprung mass velocity was depicted. NNs contain bigger oscillation in the sprung mass velocity than the

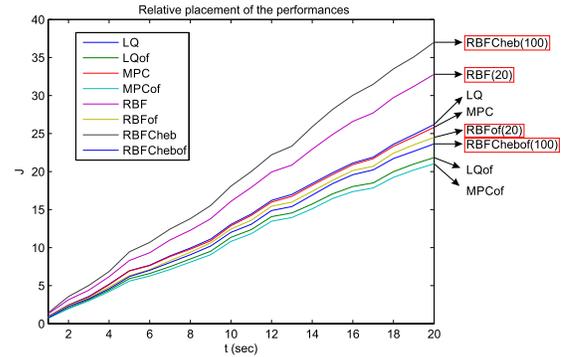


Fig. 30. Change of the performance values with respect to time during 20 s.

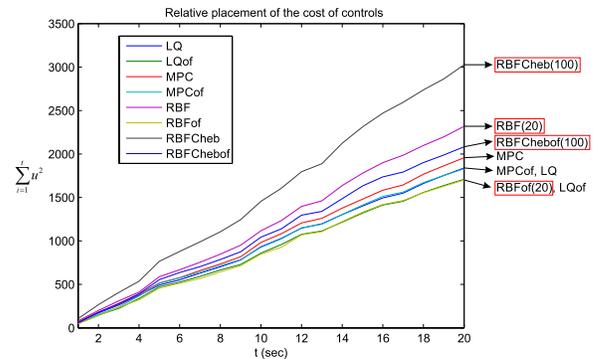


Fig. 31. Change of the cost of control with respect to time during 20 s.

output-feedback explicit MPC/LQ control system. For NNs larger number of neurons (100 instead of 20) causes better disturbance attenuation. The longer oscillation can come from the fact that the NN approximator can estimate only the training points. The authors investigated that similar results can be reached using training points 5000. After certain number of training points the approximation does not get better and the oscillation remains also. Improvement of the oscillation in the NN approximator control system can be an interesting research area later on. Figs. 30 and 31 show the change of

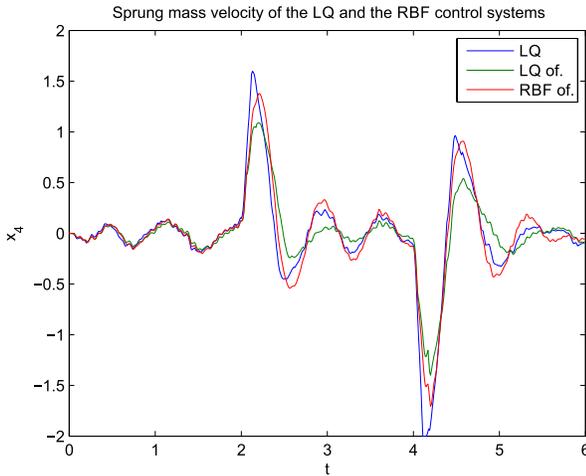


Fig. 32. Sprung mass velocity of the LQ, LQG, and RBF control systems.

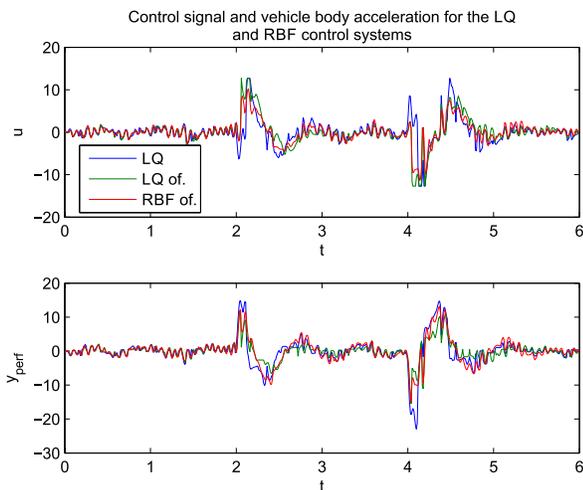


Fig. 33. Control signal and vehicle body acceleration for the LQ, LQG, and RBF control systems.

the performance values and change of the cost of control with respect to the time during 20 s. The NN approximators cannot provide the performance properties, which is ensured with the output feedback and explicit MPC/LQ (Fig. 30). It is an interesting result that the NN approximator may give better performance than the state-feedback controllers. Note that the explicit MPC/LQ output-feedback controller reached better performance values than the LQG controller, which can increase life duration of the mechanical elements in the car. In the control energy, the RBF NN control system and the LQG controller consumed the lowest energy, which is also an interesting result, as shown in Fig. 31. Finally, Figs. 32 and 33 show the transient for LQ, LQG, and RBF NN suspension control systems. The above presented results also show if the performance function (25a) is applied, then the LQG controller with the clipped strategy provide satisfying control properties for the semiactive suspension and furthermore it has simple controller structure.

VII. CONCLUSION

The explicit MPC is a promising method to increase the practical applicability of the MPC to such real systems,

where the time consuming online optimization is not allowed because fast control action is required. The optimal MPC control does not have a linear state-feedback form. To make sure the controller covers the whole constraint set, soft constraints and a combined MPC/LQ setup can be applied. Two main disadvantages of the explicit MPC are the exponential blow-up of the number of regions with increasing the prediction horizon and the requirements of the full state measurement. This paper provided a solution for both problems in case of the quarter car semiactive suspension model. We have shown that the Kalman filter is an appropriate choice to estimate the states from the measurement of the suspension deflection. Afterward, we designed the RBF NN approximator with Gaussian basis functions to replace the explicit MPC/LQ controller to avoid the time consuming searching among the regions. The NN control was a nonlinear state-feedback controller. This paper presented a systematic method to design NN approximator and investigated the efficiency of many types of the training set generation and of the clustering algorithm. The NN approximator was analyzed with the designed Kalman filter together in the complete control system. The results showed that the NN approximator works with Kalman filter adequately, but the disturbance attenuation is slower than in case of explicit MPC/LQ controller with Kalman filter. Furthermore, the RBF NN control system cannot reach the performance of the explicit MPC/LQ and LQG control systems but it may provide better performance than the state-feedback controllers. It was also shown that the explicit MPC/LQ controller has smaller performance (cost) values than the LQG controller. Cost of the control in case of the NN controller and the LQG controller were the lowest.

The goal of this paper was to develop and analyze a novel explicit MPC-based controller design method for the semiactive suspension. The presented ideas and investigation methods can be applied to similar nonlinear systems with constraints. The advantage of employing an explicit solution is two fold. First, the domain of the explicit solution allows to *directly* generate only meaningful samples, i.e., those which are feasible in (33). On the other hand, with the MIQP formulation one would need to grid the state space (whose feasibility boundaries are not known in the MIQP approach) and solve the MIQP problem just to see if the sample is feasible. As a consequence, one would solve many problems which might be infeasible, increasing the computational overhead of the learning scheme. The second advantage of basing the same generation on the explicit solution is that one can use its properties (e.g., relative volumes of individual regions) to generate more samples for parts of the state space which are more likely to be active in practice. Finally, one can easily combine the MIQP approach with the explicit solution as follows: use the explicit solution for a narrower region of the state space and use the MIQP approach to obtain a NN approximation in the remainder of the space. By doing so we can trade off complexity for suboptimality. This paper also showed that it can happen that a simpler control structure can also ensure the required expectations. To sum up, this paper built and analyzed a complete explicit MPC-based NN control system with Kalman filter. Proof of the quality

of the approximation, decreasing of the oscillation in case of bigger road defects and making better the performance issues for NN control system are topics of future research.

REFERENCES

- [1] C. Papageorgiou and M. C. Smith, "Positive real synthesis using matrix inequalities for mechanical networks: Application to vehicle suspension," *IEEE Trans. Control Syst. Technol.*, vol. 14, no. 3, pp. 423–435, May 2006.
- [2] A. Molina-Cristobal, C. Papageorgiou, G. T. Parks, M. C. Smith, and P. J. Clarkson, "Multi-objective controller design: Evolutionary algorithms and bilinear matrix inequalities for a passive suspension," in *Proc. Int. Fed. Autom. Control*, 2006, pp. 386–391.
- [3] M. Z. Q. Chen, C. Papageorgiou, F. Scheibe, F.-C. Wang, and M. C. Smith, "The missing mechanical circuit element," *IEEE Circuits Syst. Mag.*, vol. 9, no. 1, pp. 10–26, Mar. 2009.
- [4] F. Borrelli, *Constrained Optimal Control of Linear and Hybrid Systems* (Lecture Notes in Control and Information Sciences). New York, NY, USA: Springer-Verlag, 2003.
- [5] M. Sznajder and M. J. Damborg, "Suboptimal control of linear systems with state and control inequality constraints," in *Proc. 26th IEEE Conf. Decision Control*, vol. 26, Dec. 1987, pp. 761–762.
- [6] D. Chmielewski and V. Manousiouthakis, "On constrained infinite-time linear quadratic optimal control," *Syst. Control Lett.*, vol. 29, no. 3, pp. 121–130, Nov. 1996.
- [7] L. Chisci and G. Zappa, "Fast algorithm for a constrained infinite horizon LQ problem," *Int. J. Control*, vol. 72, no. 11, pp. 1020–1026, 1999.
- [8] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, Jan. 2002.
- [9] P. O. M. Scokaert and J. B. Rawlings, "Constrained linear quadratic regulation," *IEEE Trans. Autom. Control*, vol. 43, no. 8, pp. 1163–1169, Aug. 1998.
- [10] P. Grieder, F. Borrelli, F. Torrisi, and M. Morari, "Computation of the constrained infinite time linear quadratic regulator," *Automatica*, vol. 40, no. 4, pp. 701–708, Apr. 2004.
- [11] L. H. Csekő, M. Kvasnica, and B. Lantos, "Analysis of the explicit model predictive control for semi-active suspension," *Periodica Polytech.*, vol. 54, nos. 1–2, pp. 41–58, 2010.
- [12] P. Tøndel, T. A. Johansen, and A. Bemporad, "Evaluation of piecewise affine control via binary search tree," *Automatica*, vol. 39, no. 5, pp. 945–950, May 2003.
- [13] C. N. Jones, P. Grieder, and S. V. Raković, "A logarithmic-time solution to the point location problem for parametric linear programming," *Automatica*, vol. 42, no. 12, pp. 2215–2218, Dec. 2006.
- [14] F. J. Christophersen, M. Kvasnica, C. N. Jones, and M. Morari, "Efficient evaluation of piecewise control laws defined over a large number of polyhedra," in *Proc. Eur. Control Conf. (ECC)*, 2007, pp. 2360–2367.
- [15] M. Kvasnica and M. Fikar, "Performance-lossless complexity reduction in explicit MPC," in *Proc. 49th IEEE Conf. Decision Control*, Dec. 2010, pp. 5270–5275.
- [16] M. Kvasnica, I. Rauová, and M. Fikar, "Simplification of explicit MPC feedback laws via separation functions," in *Proc. IFAC World Congr.*, 2011, pp. 5383–5388.
- [17] M. Kvasnica, J. Löfberg, M. Herceg, Ľ. Čirka, and M. Fikar, "Low-complexity polynomial approximation of explicit MPC via linear programming," in *Proc. Amer. Control Conf.*, Baltimore, MD, USA, Jun./Jul. 2010, pp. 4713–4718.
- [18] J. Lygeros, "Fast approximate explicit model predictive control," Automatic Control Lab., ETH Zürich, Zürich, Switzerland, Tech. Rep.
- [19] E. F. Camacho and C. B. Alba, *Model Predictive Control*. New York, NY, USA: Springer-Verlag, 2004.
- [20] G. Horváth, Ed., *Neural Networks and Their Applications* (in Hungarian). Budapest, Hungary: Műegyetem Press, 1998.
- [21] N. Giorgetti, A. Bemporad, H. E. Tseng, and D. Hrovat, "Hybrid model predictive control application towards optimal semi-active suspension," *Int. J. Control*, vol. 79, no. 5, pp. 521–533, Feb. 2006.
- [22] C. Poussot-Vassal, "Robust linear parameter varying multivariable automotive global chassis control," Ph.D. dissertation, Dept. Control Syst. GIPSA-Lab., Grenoble Inst. Technol., Isère, France, 2008.
- [23] S. Guo, S. Yang, and C. Pan, "Dynamic modeling of magnetorheological damper behaviors," *J. Intell. Mater. Syst. Struct.*, vol. 17, no. 1, pp. 3–14, Jan. 2006.
- [24] O. Sename, A.-L. Do, C. Poussot-Vassal, and L. Dugard, "Some LPV approaches for semi-active suspension control," in *Proc. Amer. Control Conf.*, Montreal, QC, Canada, Jun. 2012, pp. 1567–1572.
- [25] R. Rajamani, *Vehicle Dynamics and Control*. New York, NY, USA: Springer-Verlag, 2006.
- [26] H. E. Tseng and J. K. Hedrick, "Semi-active control laws—Optimal and sub-optimal," *Veh. Syst. Dyn.*, vol. 23, no. 1, pp. 545–569, 1994.
- [27] B. Lantos, *Theory and Design of Control Systems II* (in Hungarian). Budapest, Hungary: Akadémia Press, 2003.
- [28] E. D. Sontag, "Nonlinear regulation: The piecewise linear approach," *IEEE Trans. Autom. Control*, vol. 26, no. 2, pp. 346–358, Apr. 1981.
- [29] W. P. M. H. Heemels, J. M. Schumacher, and S. Weiland, "Linearity complementarity systems," *SIAM J. Appl. Math.*, vol. 60, no. 4, pp. 1234–1269, 2000.
- [30] A. J. van der Schaft and J. M. Schumacher, "Complementarity modeling of hybrid systems," *IEEE Trans. Autom. Control*, vol. 43, no. 4, pp. 483–490, Apr. 1998.
- [31] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, Mar. 1999.
- [32] M. Kvasnica, P. Grieder, and M. Baotić. (2004). *Multi-Parametric Toolbox (MPT)*. [Online]. Available: <http://control.ee.ethz.ch/~mpt/>
- [33] P. Grieder, M. Kvasnica, M. Baotić, and M. Morari, "Stabilizing low complexity feedback control of constrained piecewise affine systems," *Automatica*, vol. 41, no. 10, pp. 1683–1694, Oct. 2005.
- [34] T. Geyer, F. D. Torrisi, and M. Morari, "Optimal complexity reduction of polyhedral piecewise affine systems," *Automatica*, vol. 44, no. 7, pp. 1728–1740, Jul. 2008.
- [35] A. Bemporad and C. Filippi, "Suboptimal explicit receding horizon control via approximate multiparametric quadratic programming," *J. Optim. Theory Appl.*, vol. 117, no. 1, pp. 9–38, Apr. 2003.
- [36] A. Ulbig, S. Oлару, D. Dumur, and P. Boucher, "Explicit solutions for nonlinear model predictive control: A linear mapping approach," in *Proc. Eur. Control Conf.*, 2007, pp. 3295–3302.
- [37] C. N. Jones and M. Morari, "Approximate explicit MPC using bilevel optimization," in *Proc. Eur. Control Conf.*, Budapest, Hungary, Aug. 2009, pp. 2396–2401.
- [38] Y. Yam, "Fuzzy approximation via grid point sampling and singular value decomposition," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 27, no. 6, pp. 933–951, Dec. 1997.
- [39] B. Lantos, *Fuzzy Systems and Genetic Algorithms*. Budapest, Hungary: Műegyetem Press, 2002.
- [40] L. H. Csekő, "GA-based multiobjective optimization and optimal fuzzy approximation with applications to control engineering," (in Hungarian), M.S. thesis, Dept. Control Eng. Inf. Technol., Budapest Univ. Technol. Econ., Budapest, Hungary, 2001.



Lehel Huba Csekő received the M.S. degree in electrical engineering with a specialization in control engineering and robotics from the Budapest University of Technology and Economics (BUTE), Budapest, Hungary, in 2001. He was Ph.D student in the Control Engineering and Robotics Laboratory, BUTE, until 2004.

He has been working with the Hungarian Subsidiary Company of the Siemens AG, Munich, Germany, since 2004, and performs parallel Ph.D. research at BUTE. He is currently with Siemens electric car project as a Model Based Software Engineer. His current research interests include robust control, nonlinear systems, linear parametric varying control, model predictive control, vehicle suspension control, and soft computing.



Michal Kvasnica was born in 1977. He received the Diploma degree in chemical engineering from the Slovak University of Technology in Bratislava, Bratislava, Slovakia, and the Ph.D. degree in electrical engineering from the Swiss Federal Institute of Technology in Zurich, Zurich, Switzerland.

He has been an Associate Professor with the Slovak University of Technology in Bratislava since 2011. He has co-authored and is a Developer of the MPT Toolbox for explicit model predictive control. His current research interests include model

predictive control, modeling of hybrid systems, and development of software tools for control.



Béla Lantos received the M.S. degree in control engineering from the Ilmenau Institute of Technology, Ilmenau, Germany, in 1965, the M.S. degree in applied mathematics from the Budapest University of Sciences, Budapest, Hungary, in 1972, and the Ph.D. and D.Sc. degrees in control engineering and robotics from the Hungarian Academy of Sciences, Budapest, in 1976 and 1994, respectively.

He was a Full Professor and Leader of the Control Engineering and Robotics Laboratory with the Budapest University of Technology and Economics, Budapest, from 1994 to 2011, where he is currently a Professor Emeritus. He has co-authored the book entitled *Nonlinear Control of Vehicles and Robots* (Springer, 2011). His current research interests include optimum theory, robot control and identification, dexterous hands, predictive control, 3-D image processing, soft computing, and vehicle control.

Regionless Explicit Model Predictive Control of Active Suspension Systems With Preview

Johan Theunissen, Aldo Sorniotti , Member, IEEE, Patrick Gruber , Saber Fallah , Marco Ricco, Michal Kvasnica, and Miguel Dhaens

Abstract—Latest advances in road profile sensors make the implementation of preemptive suspension control a viable option for production vehicles. From the control side, model predictive control (MPC) in combination with preview is a powerful solution for this application. However, the significant computational load associated with conventional implicit model predictive controllers is one of the limiting factors to the widespread industrial adoption of MPC. As an alternative, this article proposes an explicit model predictive controller (e-MPC) for an active suspension system with preview. The MPC optimization is run offline, and the online controller is reduced to a function evaluation. To overcome the increased memory requirements, the controller uses the recently developed regionless e-MPC approach. The controller is assessed through simulations and experiments on a sport utility vehicle demonstrator with controllable hydraulic suspension actuators. For frequencies <4 Hz, the experimental results with the regionless e-MPC without preview show a $\sim 10\%$ reduction of the root-mean-square (RMS) value of the vertical acceleration of the sprung mass with respect to the same vehicle with a skyhook controller. In the same frequency range, the addition of preview improves the heave and pitch acceleration performance by a further 8 to 21%.

Index Terms—Active suspension, preview, regionless explicit model predictive control, ride comfort.

I. INTRODUCTION

THE performance benefits of active suspension systems that account for the road profile ahead have been investigated and demonstrated by several authors ([1]–[4]). Preview strategies for controllable suspensions are typically based on a feedforward disturbance compensation and a state feedback contribution. An industrial benchmark is the integrated

Manuscript received November 19, 2018; revised February 21, 2019 and May 1, 2019; accepted June 6, 2019. Date of publication July 12, 2019; date of current version February 10, 2020. This work was supported in part by Tenneco Automotive Europe and in part by the Slovak Research and Development Agency (Projects APVV 15-0007 and VEGA 1/0585/19). (Corresponding author: Aldo Sorniotti.)

J. Theunissen is with Simmanco, 3060 Korbeek-Dijle, Belgium (e-mail: johan.theunissen@simmanco.com).

A. Sorniotti, P. Gruber, S. Fallah, and M. Ricco are with the University of Surrey, Guildford GU2 7XH, U.K. (e-mail: a.sorniotti@surrey.ac.uk; p.gruber@surrey.ac.uk; s.fallah@surrey.ac.uk; m.ricco@surrey.ac.uk).

M. Kvasnica is with the Slovak University of Technology, 811 07 Bratislava, Slovakia (e-mail: michal.kvasnica@stuba.sk).

M. Dhaens is with Tenneco Automotive Europe BVBA, 3800 Sint-Truiden, Belgium (e-mail: mdhaens@Tenneco.com).

Color versions of one or more of the figures in this article are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIE.2019.2926056

TABLE I
REDUCTION (IN %) OF THE RMS VALUES OF THE HEAVE ACCELERATION, DUE TO MPC AND PREVIEW

Ref.	Susp. type	Compared to the passive set-up		Compared to another controller type		
		MPC w/o preview	MPC w/ preview	Ctrl. type	MPC w/o preview	MPC w/ preview
[22]	Active	n.a.	25 – 60%	Skyhook/ LQR	n.a.	n.a.
[23]	Active	37 – 38 %	43%		0 – 15%	7 – 20%
[24]	Semiactive	n.a.	n.a.	Skyhook	18%	40%
[25]	Active	n.a.	28%	Skyhook	n.a.	20%
[26]	Semiactive	n.a.	14%	LQR + preview	n.a.	n.a.
[27]	Leveling	n.a.	40 – 90%		n.a.	10 – 73%
[28]	Leveling	n.a.	> 50%	LQR	n.a.	n.a.
[36]	Semiactive	n.a.	n.a.		< 5%	n.a.

Note: Only [23] includes a comparison based on experimental results.

feedforward-feedback scheme by Mercedes-Benz for ride height adjustment through hydraulic active suspension actuators ([5], [6]).

A wide range of preview suspension controllers has been proposed in the literature, including feedforward compensators [7], fuzzy logic controllers [8], gain scheduled controllers [9], and neural network implementations [10]. Linear quadratic regulators (LQRs) and linear quadratic Gaussian controllers are frequently adopted optimal control strategies for preview suspensions, because of their simple formulations and the common assumption of linear suspension dynamics ([11]–[17]). H_∞ and H_2/H_∞ controllers can deal with model uncertainties, external disturbances and parameter variations, e.g., the sprung mass variation depending on the vehicle load condition ([18]–[21]).

The idea of accounting for future disturbances from the road ahead and for system or actuator constraints fits well with the model predictive control (MPC) philosophy. Hence, several authors, e.g., [22]–[31], proposed MPC implementations for preview suspension systems. Table I indicates the ride comfort benefits of various MPC suspension control implementations from the literature without and with preview, with respect to the passive vehicle and the same vehicle with a more conventional controller, such as an LQR or skyhook. To the best of our knowledge, the published work to date focused on conventional implicit model predictive control (i-MPC) implementations, in which the optimization is run online. This, in turn, requires

significant computational power and makes industrial implementations relatively difficult. As a consequence, most of the studies are limited to simulation-based assessments. The very few papers with experimental results either use high-performance processors [24] or very long sample times, i.e., 30 ms [23], to allow real-time implementation of the controllers.

To facilitate the industrial adoption of MPC for active suspension control with preview, this article proposes an e-MPC approach ([32], [33]). With e-MPC the optimization problem is solved offline for an assigned range of operating conditions. The first output of the optimal control sequence is stored as an “explicit” function of the states, and the online algorithm is reduced to a simple function evaluation. Hence, e-MPC requires a limited amount of online computational power compared to i-MPC, while providing similar performance and ability to handle constraints. On the other hand, the challenges of e-MPC are the increased design complexity and memory requirements. The latter issue is significantly mitigated by the recently developed theory of regionless e-MPC ([34], [35]). Region-based e-MPCs—but not regionless e-MPCs—have already been implemented in simulation ([36]–[39]) on semiactive and active suspensions without preview. In a few cases, they have also been preliminarily experimentally validated ([40], [41]). However, to the best of the knowledge of the authors, e-MPC has not been proposed so far for preview suspension control.

In summary, the contributions of this article are as follows.

- 1) The e-MPC formulation for active suspension systems with preview.
- 2) The adoption of the regionless e-MPC approach for suspension control. This facilitates the implementation at shorter time steps with respect to i-MPC, and reduces the memory requirements in comparison with the traditional region-based e-MPC.

The proof-of-concept regionless e-MPC algorithm is assessed through vehicle simulations and preliminary experimental tests on a vehicle demonstrator equipped with four commercially available active suspension actuators.

II. INTERNAL MODEL FORMULATION

This article proposes a decentralized controller, i.e., based on an independent controller for each vehicle corner (see also Section III-D). As a consequence, a quarter car (QC) model (see Fig. 1) is used for the internal model of the MPC formulation

$$\begin{aligned} m_1 \ddot{x}_1 + k_1 (x_1 - x_2) + c_1 (\dot{x}_1 - \dot{x}_2) + u_a &= 0 \\ m_2 \ddot{x}_2 + k_1 (x_2 - x_1) + k_2 (x_2 - w_0) \\ + c_1 (\dot{x}_2 - \dot{x}_1) + c_2 (\dot{x}_2 - \dot{w}_0) - u_a &= 0 \end{aligned} \quad (1)$$

where u_a is the actual force generated by the actuator; m_1 and m_2 are the sprung and unsprung masses; k_1 and c_1 are the vertical suspension stiffness and residual damping of the passive components; k_2 and c_2 are tire stiffness and damping; x_1 and x_2 are the sprung and unsprung mass displacements; and w_0 is the vertical displacement of the tire contact patch. For simplicity, the implementation of this article assumes $c_2 \approx 0$.

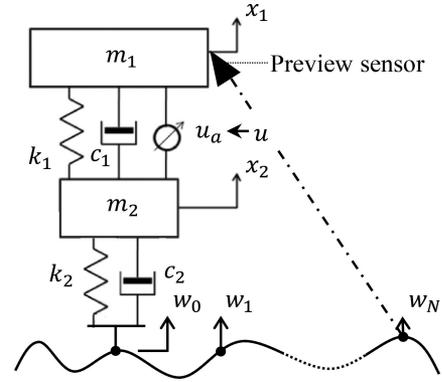


Fig. 1. QC model, including the hydraulic actuator, the road profile model and preview capability.

Usually, the main nonlinearity of a suspension system is due to the characteristic of the passive damper, which is absent in the specific plant. Moreover, the damping resulting from other passive suspension components (e.g., the bushings) is very small, so that c_1 can be considered negligible. Hence, the hypothesis of using a linear model in (1) is deemed acceptable.

The hydraulic suspension actuator, installed in the strut assembly, is modeled as a first order transfer function

$$\frac{u_a(s)}{u(s)} = \frac{1}{s\tau + 1} \quad (2)$$

where u is the actuator force demand, i.e., the system control input, and τ is the time constant of the transfer function.

The previous equations can be re-written into a continuous time state-space formulation

$$\begin{aligned} \dot{x}_{QC}(t) &= A_{QC}x_{QC}(t) + B_{QC}u(t) + E_{QC}w_0(t) \\ y_{QC}(t) &= C_{QC}x_{QC}(t) + D_{QC}u(t) \end{aligned} \quad (3)$$

where x_{QC} and y_{QC} are the state and output vectors; A_{QC} , B_{QC} , C_{QC} , and D_{QC} are the system matrices; E_{QC} is the road disturbance matrix; and t is time. $y_{QC}(t)$ contains the acceleration of the sprung mass \ddot{x}_1 .

The e-MPC uses a state feedback law. Hence, its performance depends on the accuracy and appropriate selection of the measured or estimated states. In the specific implementation, $x_{QC}(t) = [x_1 \ \dot{x}_1 \ x_1 - x_2 \ \dot{x}_1 - \dot{x}_2 \ u_a]^T$, i.e., x_{QC} contains the position and speed of the sprung mass, the suspension displacement and deflection rate, and the actual actuator force.

In the controller implementation, the estimates of x_1 and \dot{x}_1 are computed by high-pass filtering and integrating the vertical acceleration measurements of the vehicle body, through an algorithm already implemented on production vehicles with the same active suspension system of this article. $x_1 - x_2$ is obtained from the direct measurement of the active suspension actuator displacement and consideration of the suspension installation ratio, i.e., the ratio between the actuator displacement and the relative vertical displacement between the sprung and unsprung masses [42]. $\dot{x}_1 - \dot{x}_2$ is calculated through

differentiation of $x_1 - x_2$ with the hybrid smooth derivative method [43]. u_a is estimated from the measurements of the compression and rebound chamber pressures.

For preview control, the vertical road profile is modeled through a shift register, which is represented in discrete time form as

$$\begin{bmatrix} w_0(k+1) \\ w_1(k+1) \\ w_2(k+1) \\ \vdots \\ w_{N-1}(k+1) \\ w_N(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} w_0(k) \\ w_1(k) \\ w_2(k) \\ \vdots \\ w_{N-1}(k) \\ w_N(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} y_r(k) \quad (4)$$

where k indicates the current time step. With simplified notations (4) can be rewritten as

$$\hat{w}(k+1) = [0 \quad A_{r,d}] \hat{w}(k) + E_{r,d} y_r(k) \quad (5)$$

where $\hat{w} = [w_0 \dots w_N]^T$ is the vector of the road system states, i.e., the road profile heights ahead of the tire, which consists of N points (see Fig. 1) equally spaced according to the time step Δt of the internal model; $A_{r,d}$ is the shift model matrix; $y_r(k) = w_N(k+1)$ is the disturbance input provided by the preview sensor measurement; and $E_{r,d}$ is the road system disturbance matrix.

By augmenting the state vector to $x(t) = [x_1 \quad \dot{x}_1 \quad x_1 - x_2 \quad \dot{x}_1 - \dot{x}_2 \quad u_a \quad \hat{w}]^T$, applying zero-order-hold discretization of the QC model (3) to obtain the system matrices $A_{QC,d}$, $B_{QC,d}$, $C_{QC,d}$, $D_{QC,d}$, and $E_{QC,d}$, and integrating the QC model with the road model (5), the complete vehicle-actuator-road system, indicated by the subscripts s in the remainder, reads

$$\begin{aligned} x(k+1) &= \begin{bmatrix} A_{QC,d} & E_{QC,d} & 0 \\ 0 & 0 & A_{r,d} \end{bmatrix} x(k) + B_{QC,d} u(k) \\ &+ \begin{bmatrix} 0 \\ E_{r,d} \end{bmatrix} y_r(k) \\ y(k) &= [C_{QC,d} \quad 0] x(k) + D_{QC,d} u(k) \end{aligned} \quad (6)$$

which can be simplified into

$$\begin{aligned} x(k+1) &= A_{s,d} x(k) + B_{QC,d} u(k) \\ &+ E_{s,d} y_r(k) \\ y(k) &= C_{s,d} x(k) + D_{QC,d} u(k). \end{aligned} \quad (7)$$

III. CONTROLLER FORMULATION

A. System Prediction

Given the initial state, $x(k)$, and the system in (7), the predicted output, \hat{y} , is calculated as

$$\begin{aligned} \hat{y} &= \begin{bmatrix} C_{s,d} A_{s,d} \\ C_{s,d} A_{s,d}^2 \\ \vdots \\ C_{s,d} A_{s,d}^p \end{bmatrix}_{p \times 1} x(k) \\ &+ \begin{bmatrix} C_{s,d} B_{QC,d} & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ C_{s,d} A_{s,d}^{p-1} B_{QC,d} & \dots & C_{s,d} B_{QC,d} & D_{QC,d} \end{bmatrix}_{p \times (c+1)} \hat{u} \end{aligned} \quad (8)$$

with

$$\hat{y} = \begin{bmatrix} y(k+1) \\ \vdots \\ y(k+p) \end{bmatrix}, \quad \hat{u} = \begin{bmatrix} u(k) \\ \vdots \\ u(k+c) \end{bmatrix} \quad (9)$$

where p and c are the number of steps corresponding to the prediction and control horizons, and \hat{u} is the control input over c , i.e., the vector of optimization variables. (8) can be shortened to

$$\hat{y} = \Lambda x(k) + \Theta_u \hat{u}. \quad (10)$$

The state predictions, \hat{x} , are computed with a similar method

$$\hat{x} = \Psi x(k) + \Omega_u \hat{u} \quad (11)$$

with

$$\hat{x} = \begin{bmatrix} x(k+1) \\ \vdots \\ x(k+p) \end{bmatrix} \quad (12)$$

where Λ , Θ_u , Ψ , and Ω_u are the resulting matrices.

B. Constrained Optimization and mp-QP Problem Formulation

A generic model predictive controller finds the optimal sequence of control inputs, \hat{u} , that minimizes a cost function, J_{MPC} , which depends on \hat{y} , \hat{x} , and \hat{u}

$$\begin{aligned} \min_{\hat{u}} J_{MPC} &= \min_{\hat{u}} (\hat{y}^T Q_1 \hat{y} + \hat{x}^T Q_2 \hat{x} + \hat{u}^T R \hat{u}) \\ \text{s.t. } &(x(k+i), u(k+i)) \in \mathcal{F}, \quad i = 0, \dots, p \end{aligned} \quad (13)$$

where Q_1 , Q_2 , and R are weight matrices, \mathcal{F} is a full-dimensional polyhedral set of appropriate dimensions, i is an integer, and p is the number of prediction steps, which defines the prediction horizon.

By substituting the formulations of the output and state predictions (respectively (10) and (11)) into (13), eliminating the terms not depending on \hat{u} , and dividing by 2, the optimization

problem becomes

$$\begin{aligned} \min_{\hat{u}} & \left(\frac{1}{2} \hat{u}^T (\Theta_u^T Q_1 \Theta_u + \Omega_u^T Q_2 \Omega_u + R) \hat{u} \right. \\ & \left. + x(k)^T (\Lambda^T Q_1 \Theta_u + \Psi^T Q_2 \Omega_u) \hat{u} \right) \\ \text{s.t. } & P \hat{u} \leq M_1 + M_2 x(k). \end{aligned} \quad (14)$$

The typical quadratic programming (QP) format is obtained through the simplification of (14)

$$\begin{aligned} \min_{\hat{u}} & \left(\frac{1}{2} \hat{u}^T H \hat{u} + x(k)^T F \hat{u} \right) \\ \text{s.t. } & P \hat{u} \leq M_1 + M_2 x(k) \end{aligned} \quad (15)$$

where H , F , P , M_1 , and M_2 are constant matrices. The initial states of the system are included in $x(k)$, the parameter vector.

A conventional i-MPC would execute an online optimization at each time step for a given value of $x(k)$, which is replaced by x in the remainder for the sake of brevity, and the control law would be implicitly obtained by the QP solver. In the e-MPC case, the optimization is performed offline, i.e., the QP problem is solved for the whole range of x , which explicitly generates $u = u(x)$. The optimization problem becomes a multiparametric QP (mp-QP) problem, generally described as the minimization of the objective function with the constraints defined in (15).

C. Objective Function

The key objective for ride comfort enhancement is the minimization of the vertical acceleration of the sprung mass. Additionally, the optimal solution has to consider the limitation of actuator displacement, chassis motion and wheel hop [44]. Hence, this article uses a cost function penalizing \ddot{x}_1 , $x_1 - x_2$, x_1 , and $x_2 - w_0$. The control effort u is also included to limit the actuation power consumption. The discrete form of the performance index to be minimized J_{MPC} is

$$\begin{aligned} J_{MPC} = & \sum_{i=1}^p \left(\rho_1 \ddot{x}_1(k+i)^2 + \rho_2 (x_1(k+i) - x_2(k+i))^2 \right. \\ & \left. + \rho_3 x_1(k+i)^2 + \rho_4 (x_2(k+i) \right. \\ & \left. - w_0(k+i))^2 \right) + \sum_{i=0}^c \rho_5 u(k+i)^2 \end{aligned} \quad (16)$$

where the factors ρ_i are the objective function weights, which define Q_1 , Q_2 , and R in (13). In the specific implementation, the constraints are related to the actuator force and suspension displacement.

D. Decentralized Controller

To reduce the e-MPC generation time, memory requirements and implementation complexity, a decentralized control architecture is adopted, with one independent e-MPC at each vehicle corner. In fact, each QC-actuator-road model inherits only

$5 + (N + 1)$ mp-QP parameters. In contrast, a centralized suspension controller would have to be based on a seven-degree-of-freedom (7-DOF) model to consider the vertical dynamics of the unsprung masses, the heave, pitch, and roll dynamics of the sprung mass, the actuator dynamics, and the road model for each corner. This would result in a considerably larger problem, with $18 + 4(N + 1)$ mp-QP parameters.

E. Regionless e-MPC

In the e-MPC implementation, the solution of the mp-QP problem in (15) is computed offline. The solution is the function $\hat{u}^*(x)$, which is piecewise affine and maps the parameter vector onto the sequence of optimal control inputs. The e-MPC uses only the control input at the first time step, i.e., $u(x) = [I \ 0 \ \dots \ 0] \hat{u}^*(x)$, and the online implementation reduces to a simple function evaluation.

In the region-based e-MPC [33], the explicit representation of the control action is a piecewise affine state feedback law, defined by a partitioning of the state-space into m polyhedral critical regions

$$u(x) = \begin{cases} L_1 x + l_1, & S_1 x \leq s_1 \\ \vdots & \vdots \\ L_m x + l_m, & S_m x \leq s_m \end{cases} \quad (17)$$

where L_i , l_i , S_i , and s_i are constant matrices that are stored in the control hardware. The benefit of this method is the reduction of the online computational requirements with respect to the more common i-MPC. On the downside, the method yields increased memory requirements, especially for systems with a large number of parameters, and significant offline calculations. The first point is a major issue of the region-based method applied to preview suspension control, in particular, if multiple preview points (i.e., e-MPC parameters) are included in the model in (4).

To mitigate the weaknesses of the region-based e-MPC, this article adopts the recently proposed regionless e-MPC approach, described in [34] and [35]. The method does not need to compute or store the critical regions, defined by S_i and s_i . In fact, in the offline process all the possible active sets $\{A_1, \dots, A_{N_R}\}$ that can be locally optimal are considered through the extensive enumeration method in [45], where N_R is the number of regions. A linear program based on the Karush–Kuhn–Tucker conditions is solved to determine the feasibility of the candidate active sets. For each locally optimal active set the solution is

$$\hat{u}^* = -H^{-1} (F^T x + P_{A_i}^T \lambda^*) \quad (18)$$

where P_{A_i} includes only the rows of P indexed by the set of active constraints, and λ^* represents the dual variables given by

$$\lambda^* = Q(A_i) x + q(A_i) \quad (19)$$

with

$$Q(A_i) = -(P_{A_i} H^{-1} P_{A_i}^T)^{-1} (M_{2A_i} + P_{A_i} H^{-1} F^T) \quad (20)$$

$$q(A_i) = -(P_{A_i} H^{-1} P_{A_i}^T)^{-1} M_{1A_i} \quad (21)$$



Fig. 2. ACOCAR vehicle demonstrator with preview sensor.

where M_{1A_i} and M_{2A_i} contain only the rows of M_1 and M_2 corresponding to the active set A_i . The maps of $Q(A_i)$ and $q(A_i)$ are generated offline and stored in the controller together with H^{-1} , F , P , M_1 , and M_2 .

In the online implementation of the regionless controller, (18)–(19) are used to calculate \hat{u}^* , by finding the optimal active set for the current x from the list of locally optimal active sets. In particular, the optimal active set must fulfil the conditions

$$\begin{aligned} \lambda^* &\geq 0 \\ P\hat{u}^* &< M_1 + M_2x. \end{aligned} \quad (22)$$

The details of the online algorithm are reported in [45]. The resulting control action is identical to the one generated by the region-based e-MPC, i.e., the regionless and region-based implementations bring exactly the same results.

IV. CONTROL SYSTEM IMPLEMENTATION

A. Vehicle Demonstrator

The decentralized controller was implemented on a sport utility vehicle (SUV) demonstrator (see Fig. 2) with a hydraulic active suspension system—the Tenneco Monroe intelligent suspension, ACOCAR. At each vehicle corner, a pump pressurizes the hydraulic circuit of the actuator and thereby inputs energy into the system. The pressure level in the hydraulic chamber is modulated through the currents of the base and piston valves of the actuator, which is installed in parallel to an air spring. This actuation system mainly targets roll, pitch and primary ride control (see [46], [47] for the definition of primary ride), i.e., it is designed for input frequencies <4 Hz, but usually causes degradation of the secondary ride comfort performance, i.e., for frequencies >4 Hz. The vehicle demonstrator has a double wishbone suspension on the front axle, and a multilink suspension system on the rear axle, with installation ratios of 0.7 and 0.76.

The relevant sensor set consists of the following:

- 1) three vertical acceleration sensors installed on the sprung mass, two of them located in proximity of the front bumper, and one in proximity of the rear bumper;
- 2) a three-degree-of-freedom (3-DOF) inertial measurement unit;
- 3) suspension displacement sensors;
- 4) a preview sensor, i.e., the solid state LiDAR XenoTrack, mounted on the roof of the car.

A three-dimensional (3-D) model of the road ahead is constructed (i.e., a rolling carpet), and only the road profile heights directly in front of the wheels are sent to the e-MPCs. The accuracy and robustness of the preview road profile signal was guaranteed via appropriate high-pass filtering of the sensor outputs, a compensation algorithm of the sprung mass motion, and experimental tests to obtain the synchronization lag values.

All controllers and state estimators were installed on the dSPACE MicroAutoBox II system of the vehicle, which has a 16 MB flash memory. The regionless e-MPCs were integrated into the ACOCAR suspension control software framework to interface with the hardware. A low-level actuator management system calculates the reference currents for the compression and rebound valves, as well as the pump reference speed, as functions of u and $\dot{x}_1 - \dot{x}_2$. The current driver modules of the production suspension system feed the actuator valves and pumps.

B. Model Validation

Measurements of the ACOCAR vehicle demonstrator response on a four-poster test rig were used for the validation of two simulation models: a) a 7-DOF model for control system assessment, implemented in MATLAB/Simulink. This model considers heave, roll and pitch of the sprung mass, and vertical displacement of each unsprung mass, and includes a simplified model of the actuation system dynamics; and b) the internal e-MPC model, i.e., the QC model described in Section II.

The four-poster test rig was set up to emulate a typical ISO C–D ride comfort assessment road [48]. During the experiments, a fixed current of 0.4 A was applied to the piston and base valves of the actuators to maximize the size of the valve orifices, and, thus, achieve minimum damping.

The reported experimental values were calculated from the vertical acceleration and actuator displacement measurements, by using the state estimator of the ACOCAR suspension system. The time domain results were converted into the frequency domain, and are shown in Fig. 3 in terms of power spectral densities (PSDs). A good match between the 7-DOF model and the real vehicle is observed up to ~ 15 Hz, which is in line with the model bandwidth. In particular, the 7-DOF model captures the resonance peak of the sprung mass at ~ 1 – 1.5 Hz, and those of the unsprung masses at ~ 10 – 12 Hz.

The e-MPC internal model in (1) was validated in a similar way, i.e., the front and the rear QC model outputs were compared with the experimental displacements of the suspension top mounts and wheels. A good level of modeling accuracy was achieved also in this case, as shown in Fig. 4.

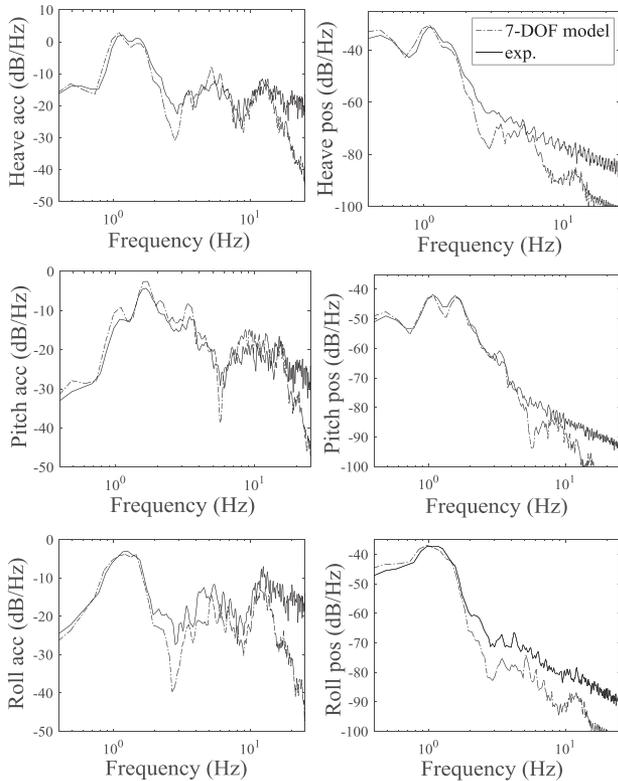


Fig. 3. Example of experimental validation of the 7-DOF model along the ride comfort road profile emulated on the four-poster test rig: PSDs of sprung mass accelerations and positions.

The e-MPC internal actuator model in (2) was validated with actuator test rig data. For example, Fig. 5 shows the time histories of the force demand, measured force and simulated force for step-in and step-out force demand tests. A good match was achieved with $\tau = 50$ ms, despite the simplicity of the model formulation.

C. Explicit Controller Generation and Implementation

According to the internal model formulation in (7), discretized at $\Delta t = 10$ ms, each controller is based on 8 mp-QP parameters, i.e., the four states of the QC model, one state for the actuator, and three states ($N = 2$) for the road profile ahead according to (4).

Simulations on a ride comfort road and a speed bump were carried out to evaluate the independent and combined effects of p , i.e., the prediction horizon, c , i.e., the control horizon, and N , i.e., the number of preview points. It was verified that in the specific test scenarios the increase of p and c brings significant benefits. Therefore, p and c were assigned relatively large values, respectively, 8 and 6. On the contrary, N was tuned to be as low as possible, to reduce the required flash memory size (which strongly varies with N) without significantly affecting comfort. At 50 km/h, the selected parametrization corresponds to a ~ 0.3 m look ahead distance and a > 1 m prediction distance.

An inequality constraint was applied to the actuator force magnitude, i.e., < 9000 N. The tuning of the cost function

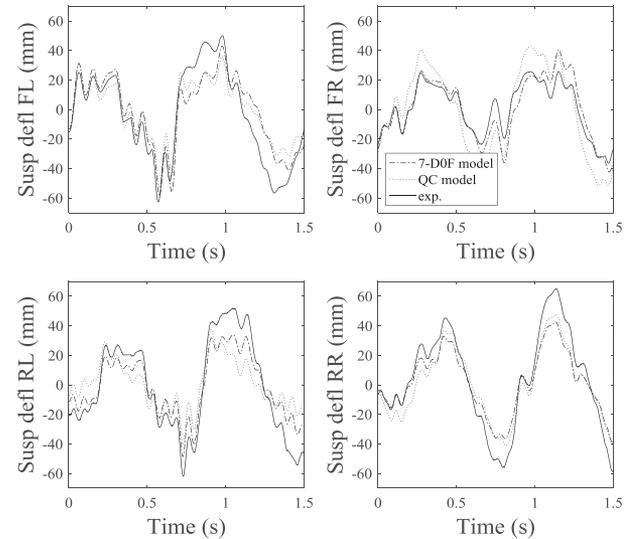


Fig. 4. Example of experimental validation of the 7-DOF and QC models along the ride comfort road profile emulated on the four-poster test rig: time histories of suspension deflections. The subscripts FL, FR, RL, and RR indicate the front left, front right, rear left, and rear right corners.

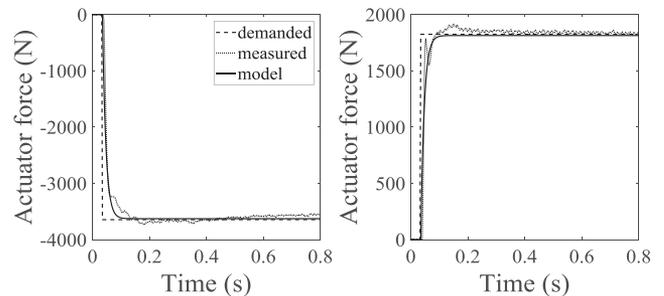


Fig. 5. Example of experimental validation of the e-MPC internal actuator model for step-in and step-out force demand tests.

(16) prioritized the reduction of the vertical acceleration and displacement of the sprung mass, by choosing greater values for ρ_1 and ρ_3 relative to ρ_2 and ρ_5 . ρ_4 was only used for a preliminary feasibility check in simulation, targeting the wheel hop reduction, while it was set to 0 in the experiments as wheel hop was not observed.

The mp-QP problems for the active suspension system with and without preview were solved with a custom version of the multiparametric toolbox 3 [49] that included the regionless solver RLENUMPQP. The solution was considered over a bounded partition of the state-space, with the following limits: ± 0.1 m in body displacement; ± 0.5 m/s in body velocity; ± 0.15 m in suspension displacement; ± 4 m/s in suspension velocity; and ± 0.15 m in road displacement.

Table II gives the comparison of the region-based and regionless algorithms, in terms of solution generation time and corresponding memory requirements, where the reduction of the latter is of the essence for the industrial implementation of the algorithm. In particular, the industrial partners of this article specified an upper limit of 1 MB memory to ensure applicability to a production-ready suspension system. As indicated by

TABLE II

EXPLICIT SOLUTION GENERATION TIME AND MEMORY DEMAND OF THE REGION-BASED AND REGIONLESS E-MPC APPROACHES

Method	Generation time	Memory demand
Region-based, w/o preview	3 s	~600 kB
Regionless, w/o preview	3 s	~30 kB
Region-based, w/ preview	140 s	~30 MB
Regionless, w/ preview	21 s	~1 MB

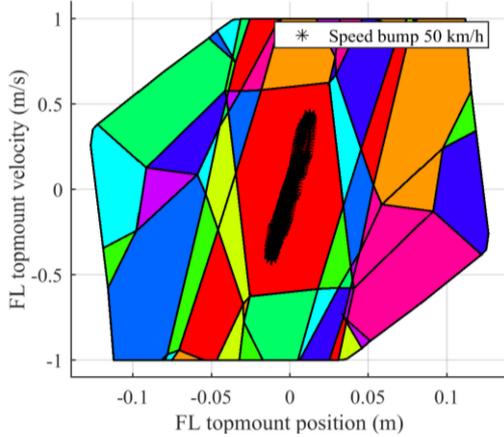
Fig. 6. Critical regions on the $\dot{x}_1(x_1)$ plane for the e-MPC with preview tested on the vehicle demonstrator.

Table II, the regionless e-MPC achieves this specification for both configurations, with and without preview, which is an important outcome of this article. To meet the 1 MB memory specification at the vehicle level for the system with preview, and verify the system robustness with respect to the modeling uncertainty, the same regionless explicit solution was implemented on the front and rear suspensions, despite a marginal difference in their parameters. In contrast, the traditional region-based e-MPC solution obtained with the ENUMQP solver meets the memory specification only for the nonpreview version, and significantly exceeds the limit when preview is included. Moreover, the online algorithm of the regionless e-MPC required, on average, only 0.09 ms run time (with a maximum of 0.2 ms) on the dSpace platform during a typical test. The short computation times therefore allow the implementation of the controller at almost any time-step used in automotive applications.

With the regionless approach, the regions do not need to be calculated, but they can be reconstructed and visualized *a posteriori*. For the specific preview controller, the solution is a set of affine functions over 1099 polyhedral regions. Figs. 6 and 7 show two-dimensional (2-D) slices over the multidimensional state-space. Such representation of the explicit control law allows the formal analysis of the stability and robustness properties of the resulting controller. The figures also report the operating points of the system along a speed bump at 50 km/h. The analysis of the actual operating points of the vehicle in real maneuvers is useful to understand whether specific portions of the e-MPC control law can be adopted to formulate a simplified rule-based controller.

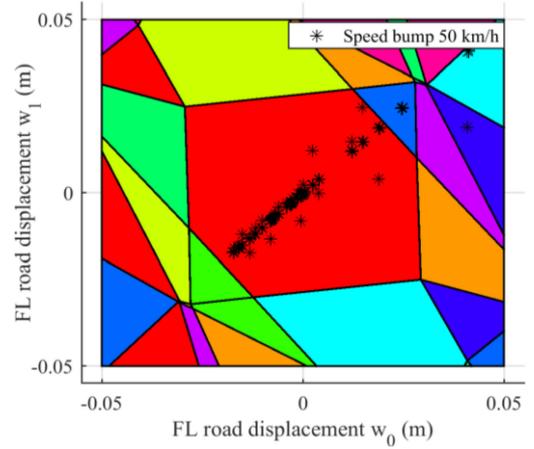
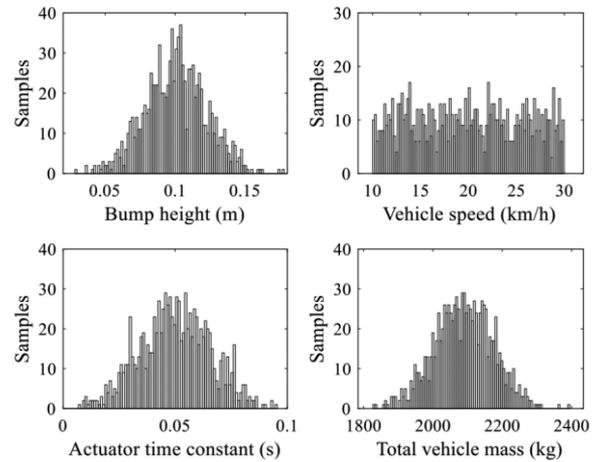
Fig. 7. Critical regions on the $w_1(w_0)$ plane, for the e-MPC with preview tested on the vehicle demonstrator.

Fig. 8. Distribution of the randomly selected parameters of the Monte Carlo analysis.

D. e-MPC Stability

From a theoretical viewpoint, the closed-loop stability of the proposed e-MPC can be achieved by including the term $x^T(k+p)Zx(k+p)$ into the objective function (16) via (15), where Z is the solution of the algebraic Riccati equation for the system in (7), along with the constraint $x(k+p) \in \Theta$, where Θ is a positive invariant set for the system. However, stability can also be achieved by appropriately choosing the state and input weighting coefficients ρ_i in (16). Typically, selecting the state weights significantly larger than the input weight ρ_5 helps to achieve a stable behavior of the closed-loop system, which is the tuning method used here.

In this article, the stability of the controller was verified through Monte Carlo simulations. The e-MPC strategy was tested in 1000 challenging scenarios, each set up with a different vehicle mass, speed and actuator response time. The simulations were performed over a 1-m long speed bump with a height that was also changed between runs. Fig. 8 shows the distribution of the randomly selected values of the four parameters. The

controller was considered stable if the suspension deflections at each corner of the 7-DOF model did not exceed 5 mm, 3 s after the front axle hit the bump. Stability was achieved in all cases.

E. Benchmark Controller: Centralized Skyhook

A centralized skyhook algorithm [51], already implemented and tested on the case study vehicle demonstrator, was used as the experimental benchmark for the decentralized e-MPCs. In the skyhook approach, the total sprung mass reference heave force F_h , antipitch moment M_p , and antiroll moment M_r , are calculated as

$$\begin{bmatrix} F_h \\ M_p \\ M_r \end{bmatrix} = \begin{bmatrix} c_h & 0 & 0 \\ 0 & c_p & 0 \\ 0 & 0 & c_r \end{bmatrix} \begin{bmatrix} \dot{x}_{s,est} \\ \dot{\theta}_{est} \\ \dot{\varphi}_{est} \end{bmatrix} \quad (23)$$

where c_h , c_p , and c_r are the skyhook damping coefficients for the heave, pitch and roll motions; and $\dot{x}_{s,est}$, $\dot{\theta}_{est}$, and $\dot{\varphi}_{est}$ are the estimated heave, pitch and roll rates of the sprung mass. The matrix form of (23) is $F_{sh} = c_{sh} V_{est}$, where F_{sh} is the vector of the total skyhook force and moments, c_{sh} is the matrix of the skyhook coefficients, and V_{est} is the vector including the three speeds in (23). In addition, $F_{sh} = L u_{c,sh}$, where $u_{c,sh}$ is the vector of the skyhook actuation forces at the four corners, i.e., the outputs of the controller, and L is the matrix with the coefficients to calculate the resulting force and moments acting on the sprung mass. The terms of L include the geometric vehicle parameters, e.g., the front and rear semi-wheelbases h_F and h_R ; and track widths t_F and t_R . In the controller implementation, a pseudoinverse formulation is used to obtain the control action vector $u_{c,sh}$

$$u_{c,sh} = [u_{FL} \quad u_{FR} \quad u_{RL} \quad u_{RR}]^T = (L^T L)^{-1} L^T c_{sh} V_{est} \quad (24)$$

with

$$L = \begin{bmatrix} -1 & -1 & -1 & -1 \\ h_F & h_F & -h_R & -h_R \\ \frac{t_F}{2} & -\frac{t_F}{2} & \frac{t_R}{2} & -\frac{t_R}{2} \end{bmatrix} \quad (25)$$

where the notations FL , FR , RL , and RR indicate the front left, front right, rear left, and rear right corners.

V. RESULTS

A. Simulation Results

The 7-DOF vehicle model was used for the virtual validation of the controllers along a ride comfort track, at a constant speed of 60 km/h. The analysis involved the regionless e-MPC implementations, including and excluding preview, and their performance comparison with the passive vehicle, i.e., the case study SUV without active suspensions. The simulations with the controllers were based on realistic data of next-generation suspension actuators with higher bandwidth than those installed on the real vehicle demonstrator, and under the hypothesis of perfect synchronization of the preview input with the actual road profile at the wheels. This set-up was chosen to assess

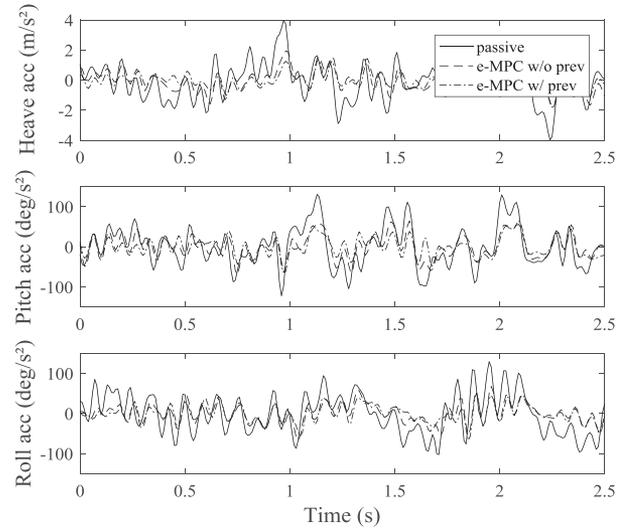


Fig. 9. Time domain plots of heave, pitch and roll accelerations obtained on a simulated section of the ride comfort road at 60 km/h.

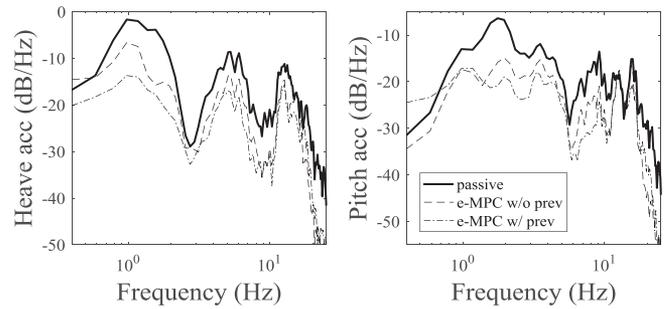


Fig. 10. Simulation of a ride comfort road at 60 km/h: PSDs of the heave and pitch accelerations.

TABLE III
RMS VALUES OF THE SPRUNG MASS ACCELERATIONS FOR THE SIMULATED RIDE COMFORT ROAD AT 60 KM/H

	Mode	Pass.	e-MPC w/o preview	e-MPC w/ preview
Non freq. weighted	Heave (m/s ²)	1.00	0.53 (-47%)	0.34 (-35%)
	Pitch (rad/s ²)	0.63	0.34 (-47%)	0.27 (-19%)
	Roll (rad/s ²)	0.63	0.32 (-48%)	0.26 (-21%)
Freq. weighted	Heave (m/s ²)	0.73	0.39 (-46%)	0.28 (-29%)
	Pitch (rad/s ²)	0.29	0.13 (-55%)	0.11 (-13%)
	Roll (rad/s ²)	0.34	0.17 (-51%)	0.12 (-27%)

Note: The % variations are with respect to the system in the column to the immediate left.

the medium-to-long-term potential of the e-MPC preview technology.

Fig. 9 reports the time histories of the heave, pitch, and roll accelerations for a section of the run. In particular, the passive set-up has a 3.96 m/s² peak heave acceleration, which is reduced to 1.91 and 1.26 m/s² for the e-MPCs without and with preview. Fig. 10 shows the results in terms of PSD profiles of the heave and pitch accelerations, while Table III reports the

root-mean-square (RMS) values of the vehicle body accelerations $a_{i,\text{RMS}}$ for heave, pitch and roll, calculated as

$$a_{i,\text{RMS}} = \left(\int_{f_1}^{f_2} \text{PSD}_i(f) df \right)^{0.5} \quad (26)$$

where f is the frequency, and f_1 and f_2 are the boundaries of the considered frequency range. In the PSD plots, the benefits of the controllers are evident for the 0–15 Hz range. This confirms the appropriateness of the e-MPC designs for improving both primary ride and secondary ride. In particular, the e-MPC without preview reduces the $a_{i,\text{RMS}}$ values by more than 45% with respect to the passive vehicle, while the introduction of preview brings a further improvement, ranging from 19 to 35% depending on the considered acceleration.

The table also includes the RMS values of the heave, pitch and roll accelerations of the vehicle sprung mass, after the application of frequency weighting functions according to [52]. In particular, the heave acceleration is weighted more in the 4–8 Hz frequency band than in the other frequency ranges. The overall improvements brought by the e-MPCs are similar to those without frequency weighting and consistent with the results in Table I, which confirms the all-around effectiveness of the proposed controllers.

As the actuation dynamics represent an unmatched uncertainty in the system, ride comfort road simulations at 60 km/h were run to assess robustness with respect to the actuator time constant τ , which was varied up to 300 ms (six times the value for the available hydraulic actuators), while keeping the e-MPC tuned for the nominal τ . The results show that the controllers without and with preview perform significantly better than the passive set-up, and the active setup with preview always provides the best performance.

B. Experimental Results

The performance of the e-MPCs (excluding and including preview) was experimentally tested with the ACOCAR vehicle demonstrator (see Section IV-A) and compared to the car with the active skyhook controller (Section IV-E) and a passive suspension set-up. The passive set-up was obtained by applying fixed currents to the actuator valves to achieve a suspension tuning that is close to the one of the passive version of the SUV.

The experiments consisted of two tests carried out on the public roads of Sint Truiden (Belgium).

- 1) *Test 1*: Driving over a short wavelength speed bump with a height of 5 cm and a length of 0.4 m, at approximately 30 km/h.
- 2) *Test 2*: Driving over a long wavelength speed bump with a height of 15 cm and a length of 2.5 m, at approximately 50 km/h, which causes significantly higher accelerations than test 1.

The tests were repeated several times to verify the reliability of the measurements. Fig. 11 shows the time history of the heave position and pitch angle of the vehicle body for test 2. The results confirm the reduction of the sprung mass motion when negotiating the bump. For instance, the passive and skyhook

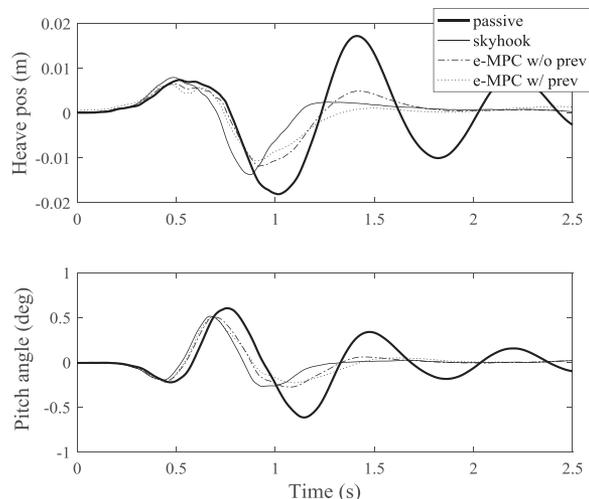


Fig. 11. Experimental results for test 2: Time domain plots of the heave position and pitch angle.

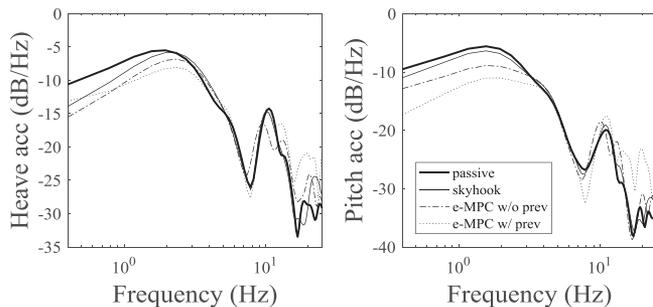


Fig. 12. Experimental results for test 1: PSDs of the heave and pitch accelerations.

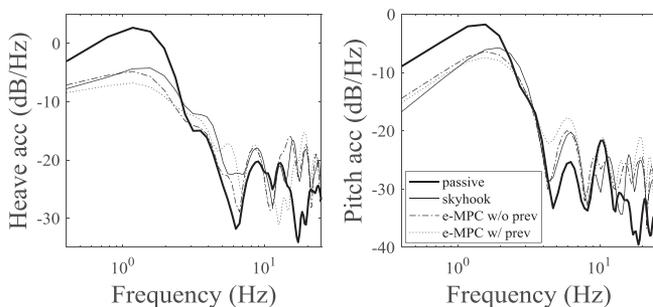


Fig. 13. Experimental results for test 2: PSDs of the heave and pitch accelerations.

set-ups have heave displacements of -0.018 and -0.014 m at the first negative oscillation peak. For the e-MPCs without and with preview, these values are reduced to -0.011 and -0.010 m.

Figs. 12 and 13 show the PSD results in the frequency domain. Tables IV and V report the RMS values of the heave and pitch accelerations of the vehicle sprung mass without and with frequency weighting, up to 15 Hz, i.e., well beyond the

TABLE IV
RMS VALUES OF THE SPRUNG MASS ACCELERATIONS DURING TEST 1

	Mode	Freq. range	Pass.	Skyhook	e-MPC w/o preview	e-MPC w/ preview
Non freq. weighted	Heave (m/s ²)	0-4 Hz	0.86	0.82 (-4%)	0.74 (-11%)	0.66 (-10%)
		0-15 Hz	0.95	0.92 (-3%)	0.83 (-9%)	0.80 (-4%)
	Pitch (rad/s ²)	0-4 Hz	0.81	0.75 (-8%)	0.61 (-17%)	0.49 (-21%)
0-15 Hz		0.85	0.80 (-7%)	0.68 (-14%)	0.57 (-16%)	
	a_p (m/s ²)	0-15 Hz	0.51	0.48 (-6%)	0.43 (-10%)	0.39 (-9%)
Freq. weighted	Heave (m/s ²)	0-15 Hz	0.67	0.65 (-3%)	0.62 (-5%)	0.63 (+2%)
	Pitch (rad/s ²)	0-15 Hz	0.51	0.50 (-1%)	0.36 (-27%)	0.29 (-21%)
	a_p (m/s ²)	0-15 Hz	0.34	0.33 (-3%)	0.29 (-12%)	0.27 (-7%)

Note: The % variations are with respect to the system in the column to the immediate left.

TABLE V
RMS VALUES OF THE SPRUNG MASS ACCELERATIONS DURING TEST 2

	Mode	Freq. range	Pass.	Skyhook	e-MPC w/o preview	e-MPC w/ preview
Non freq. weighted	Heave (m/s ²)	0-4 Hz	1.60	0.87 (-45%)	0.78 (-11%)	0.69 (-12%)
		0-15 Hz	1.61	0.93 (-42%)	0.84 (-9%)	0.74 (-12%)
	Pitch (rad/s ²)	0-4 Hz	0.99	0.68 (-31%)	0.63 (-7%)	0.58 (-8%)
0-15 Hz		1.00	0.70 (-30%)	0.66 (-6%)	0.63 (-4%)	
	a_p (m/s ²)	0-15 Hz	0.76	0.47 (-38%)	0.43 (-9%)	0.38 (-12%)
Freq. weighted	Heave (m/s ²)	0-15 Hz	0.82	0.55 (-32%)	0.50 (-10%)	0.47 (-5%)
	Pitch (rad/s ²)	0-15 Hz	0.70	0.39 (-43%)	0.41 (+4%)	0.37 (-8%)
	a_p (m/s ²)	0-15 Hz	0.43	0.27 (-37%)	0.26 (-4%)	0.24 (-8%)

Note: The % variations are with respect to the system in the column to the immediate left.

bandwidth of the specific actuators. The roll acceleration results are omitted, as roll motion was not excited by these tests.

As expected, given the relatively low bandwidth of the specific actuators, the controlled set-ups mainly improve primary ride, i.e., the range of 0–4 Hz. For example, in this frequency range, the e-MPC without preview improves the RMS heave acceleration performance without frequency weighting in both tests by 11% compared to the skyhook. The addition of preview reduces the RMS accelerations by a further 10% and 12% in tests 1 and 2. The e-MPC without preview reduces the pitch accelerations by 17% and 7% in the two tests compared to the skyhook, while the preview adds a further benefit, i.e., 21% in test 1 and 8% in test 2.

The results are confirmed over the 0–15 Hz frequency band. For instance, the RMS values of the heave acceleration in the two tests are 0.95 and 1.61 m/s² for the passive set-up, while the e-MPC with preview reduces the values to 0.80 and 0.74 m/s². In the same frequency range, the heave acceleration performance of the e-MPC with preview is consistently better than that of the e-MPC without preview; 4% improvement during test 1 and 12% improvement during test 2. Similarly, the preview reduces the RMS of the pitch motion by 16% and 4%. Moreover, the e-MPC without preview consistently outperforms the skyhook algorithm, e.g., by 9% in terms of heave acceleration. An important conclusion is that despite the decentralized architecture of the implemented e-MPCs, the associated vehicle performance improvement is evident also in terms of pitch acceleration. In fact, the RMS values of pitch acceleration with the skyhook

controller are 14% and 6% higher than with the e-MPC without preview. This result is particularly remarkable considering that the skyhook controller includes a term directly targeting the pitch dynamics. Also, the level of technology maturity of its implementation on the vehicle demonstrator is significantly higher than that of the proposed e-MPCs.

In general, the RMS values of the heave and pitch frequency weighted accelerations of the vehicle sprung mass in the 0–15 Hz frequency range tend to generate more limited controller benefits in comparison with the non-weighted results. This is mainly due to the actuator bandwidth, and the fact that the frequency weighting functions were not accounted for in the cost function (16) nor in the tuning of the e-MPC parameters, which is the subject of future work. Nevertheless, the e-MPCs still show considerable benefits over the skyhook.

As a summary of the performance benefit, **Tables IV and V** also include the vibration total value a_v i.e., an indicator that combines vibrations in multiple directions [52]

$$a_v = (k_h^2 a_{h,RMS}^2 + k_p^2 a_{p,RMS}^2)^{0.5} \quad (27)$$

where $a_{h,RMS}$ and $a_{p,RMS}$ are the RMS heave and pitch accelerations. k_h and k_p are the multiplying factors, both set to 0.4. In test 1, the skyhook reduces the a_v indicator based on the frequency weighted accelerations by only 3% with respect to the passive vehicle, while the e-MPCs without and with preview outperform the production skyhook controller by 12% and a further 7%. In test 2, despite the already excellent performance of the skyhook, which provides a 37% improvement over the passive case, the e-MPCs without and with preview further reduce the vibration total value by 4% and 8%. Such preliminary experimental benefits are aligned with the literature in **Table I**, which is mainly based on simulation results, and encourage the further industrial development of regionless e-MPC for active suspension control.

VI. CONCLUSION

To the best of our knowledge, for the first time this article implemented a regionless e-MPC strategy for an active suspension system with and without preview. The activity allows the following conclusions.

- 1) The internal QC models of the decentralized e-MPC architecture provide a sufficiently good match with the experimental data, and can be considered simple yet appropriate formulations for suspension control design.
- 2) The regionless e-MPC with preview based on a QC model brings a memory requirement reduction by a factor of ~ 30 , compared to the corresponding region-based e-MPC.
- 3) The e-MPC simulation results with hydraulic actuators along a ride comfort road show reductions of the RMS values of the sprung mass accelerations in excess of 45% relative to the passive car, and a further benefit (up to 35%) is achieved with the addition of preview.
- 4) The preliminary experimental results along two speed bump road inputs on a vehicle demonstrator with active suspension actuators show that, compared to the more

conventional skyhook, the e-MPC without preview improves primary ride performance—with reductions of primary ride vehicle body accelerations ranging from 7 to 17%. The addition of preview further reduces primary ride accelerations by 8 to 21%. All the evaluated e-MPC implementations improve the vibration total value in the 0–15 Hz frequency range, which indicates their overall ride comfort enhancement capability.

Future developments will include the systematic optimization of the tuning parameters of the proposed controllers, and the assessment of centralized control approaches based on regionless e-MPC technology.

REFERENCES

- [1] E.K. Bender, "Optimum linear preview control with application to vehicle suspension," *J. Basic Eng.*, vol. 90, no. 2, pp. 213–221, 1968.
- [2] Y. Iwata and M. Nakano, "Optimum preview control of vehicle air suspension," *Bull. Jpn. Soc. Mech. Eng.*, vol. 19, no. 138, pp. 1485–1489, 1976.
- [3] Y. Iwata and M. Nakano, "Optimal preview control of vehicle suspension," *Bull. Jpn. Soc. Mech. Eng.*, vol. 19, no. 129, pp. 265–273, 1976.
- [4] W. A. H. Oraby, M. A. Aly, S. M. El-Demerdash, and A. M. Selim, "Influence of active suspension preview control on the vehicle lateral dynamics," SAE Tech. Paper 2007-01-2347, 2007.
- [5] R. Streiter, "Active preview suspension system ABC prescan in the F700," *ATZ Worldwide*, vol. 110, pp. 4–11, 2008.
- [6] A. Schindler, "Neue Konzeption und erstmalige Realisierung eines aktiven Fahrwerks mit Preview-Strategie." Ph.D. thesis, Karlsruhe Inst. Technol., Karlsruhe, Germany, 2009.
- [7] M. Kaldas *et al.*, "Preview enhanced rule-optimized fuzzy logic damper controller," SAE Tech. Paper 2014-01-0868, 2014.
- [8] A. Desikan and V. Kalachelvi, "Design for a semi-active preview suspension system using fuzzy-logic control and image processing techniques," presented at the IEEE Int. Conf. Cyber Technol. Automat., Control Intell. Syst., 2015.
- [9] A. M. A. Soliman and D. A. Crolla, "Limited bandwidth active suspension employing wheel base preview," SAE Tech. Paper 2001-01-1063, 2001.
- [10] F. Jun-ping, Y. Shao-yi, and Z. Lan-chun, "Research on wheelbase preview control for vehicle semi-active suspension based on neural networks," presented at the IEEE Int. Symp. Intell. Inf. Technol. Appl., 2009.
- [11] M. Tomizuka, "Optimal preview control with application to vehicle suspension," *J. Dyn. Syst., Meas., Control*, vol. 98, no. 3, pp. 309–315, 1976.
- [12] A. Hac, "Optimal linear preview control of active vehicle suspension," in *Proc. IEEE Conf. Decis. Control*, 1990, pp. 2779–2784.
- [13] M. A. H. van der Aa, J. H. E. A. Muijderland, and F. E. Veldpaus, "Constrained optimal control of semi-active suspension systems with preview," *Veh. Syst. Dyn.*, vol. 28, no. 4/5, pp. 307–323, 1997.
- [14] D. Martinus, B. Soenarko, and Y. Y. Nazaruddin, "Optimal control design with preview for semi-active suspension on a half-vehicle model," in *Proc. IEEE Conf. Decis. Control*, 1996, pp. 2798–2803.
- [15] A. G. Thompson and C. E. M. Pearce, "Performance index for a preview active suspension applied to a quarter-car model," *Veh. Syst. Dyn.*, vol. 35, no. 1, pp. 55–66, 2001.
- [16] G. Prokop and R.S. Sharp, "Performance enhancement of limited bandwidth active automotive suspensions by road preview," *IEEE J. Control Theory Appl.*, vol. 142, no. 2, pp. 140–148, Mar. 1995.
- [17] H. J. Kim, H. S. Yang, and Y. P. Park, "Improving the vehicle performance with active suspension using road-sensing algorithm," *J. Comput. Struct.*, vol. 80, no. 18–19, pp. 1569–1577, 2002.
- [18] A. Akbari and B. Lohmann, "Output feedback H_∞/GH_2 preview control of active vehicle suspensions: A comparison study of LQG preview," *J. Veh. Syst. Dyn.*, vol. 48, no. 12, pp. 1475–1494, 2010.
- [19] R. S. Prabakar, C. Sujatha, and S. Narayanan, "Optimal semi-active preview control response of a half car vehicle model with magnetorheological damper," *J. Sound Vib.*, vol. 326, no. 3–5, pp. 400–420, 2009.
- [20] S. Ryu, Y. Park, and M. Suh, "Ride quality analysis of a tracked vehicle suspension with a preview control," *J. Terramechanics*, vol. 48, no. 6, pp. 409–417, 2011.
- [21] P. Li, J. Lam, and K. Chung Cheung, "Multi-objective control for active vehicle suspension with wheelbase preview," *J. Sound Vib.*, vol. 333, no. 21, pp. 5269–5282, 2014.
- [22] R. K. Mehra, J. N. Amin, K. J. Hedrick, C. Osorio, and S. Gopalasamy, "Active suspension using preview information and model predictive control," in *Proc. IEEE Int. Conf. Control Appl.*, 1997, pp. 860–865.
- [23] M. D. Donahue and J. K. Hedrick, "Implementation of an active suspension, preview controller for improved ride comfort," 2002.
- [24] M. Ahmed and F. Svaricek, "Preview optimal control of vehicle semi-active suspension based on partitioning of chassis acceleration and tire load spectra," in *Proc. IEEE Eur. Control Conf.*, 2014, pp. 1669–1674.
- [25] S. De Bruyne, H. Van der Auweraer, and J. Anthonis, "Preview control of a constrained hydraulic active suspension system," in *Proc. IEEE Conf. Decis. Control*, 2012, pp. 4400–4405.
- [26] S.M. Savaresi *et al.*, "Optimal Strategy for Semi-Active Suspensions and Benchmark," in *Semi-Active Suspension Control Design for Vehicles*, Amsterdam, The Netherlands: Elsevier, 2010, ch. 5.
- [27] C. Göhrle *et al.*, "Design and vehicle implementation of preview active suspension controllers," *IEEE Trans. Control Sys. Technol.*, vol. 22, no. 3, pp. 1135–1142, May 2014.
- [28] C. Göhrle *et al.*, "Road profile estimation and preview control for low-bandwidth active suspension systems," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 5, pp. 2299–2310, Oct. 2015.
- [29] M.Q. Nguyen, M. Canale, O. Senane, and L. Dugard, "A model predictive control approach for semi-active suspension control problem of a full car," in *Proc. IEEE Conf. Decis. Control*, 2016, pp. 721–726.
- [30] L. Xu, H. E. Tseng, and D. Hrovat, "Hybrid model predictive control of active suspension with travel limits and nonlinear tire contact force," in *Proc. Amer. Control Conf.*, 2016, pp. 2415–2420.
- [31] K. Çalışkan, R. Henze, and F. Küçükay, "Potential of road preview for suspension control under transient road inputs," *IFAC Proc.*, vol. 49, no. 3, pp. 117–122, 2016.
- [32] A. Bemporad, M. Morari, V. Dua, and E.N. Pistikopoulos, "The explicit solution of model predictive control via multiparametric quadratic programming," in *Proc. Amer. Control Conf.*, 2000, vol. 2, pp. 872–876.
- [33] A. Bemporad, F. Borrelli, and M. Morari, "Model predictive control based on linear programming – The explicit solution," *IEEE Trans. Autom. Control*, vol. 47, no. 12, pp. 1974–1985, Dec. 2003.
- [34] F. Borrelli, M. Baotic, J. Pekar, and G. Stewart, "On the computation of linear model predictive control laws," *Automatica*, vol. 46, no. 6, pp. 1035–1041, 2010.
- [35] M. Kvasnica, B. Takács, J. Holaza, and S. Di Cairano, "On region-free explicit model predictive control," in *Proc. IEEE Conf. Decis. Control*, 2015, pp. 3669–3674.
- [36] N. Giorgetti, A. Bemporad, and H. E. Tseng, "Hybrid model predictive control application towards optimal semi-active suspension," *Int. J. Control*, vol. 79, no. 5, pp. 521–533, 2006.
- [37] L.H. Csekő, M. Kvasnica, and B. Lantos, "Analysis of the explicit model predictive control for semi-active suspension," *Periodica Polytechnica Elect. Eng.*, vol. 54, no. 1-2, pp. 41–58, 2010.
- [38] M. Canale, M. Milanese, C. Novara, and Z. Ahmad, "Semi-active suspension control using "Fast" model predictive techniques," *IEEE Trans. Control Syst. Technol.*, vol. 14, no. 6, pp. 1034–1046, Nov. 2006.
- [39] T. Paschedag, A. Giu, and C. Seatzu, "Constrained optimal control: An application to semiactive suspension systems," *Int. J. Sys. Sci.*, vol. 47, no. 7, pp. 797–811, 2010.
- [40] J. Theunissen *et al.*, "Explicit model predictive control of active suspension systems," in *Proc. Int. Conf. Adv. Veh. Powertrains*, 2017, pp. 1–19.
- [41] G. Montanez, D. Patino, and D. Mendez, "Comparison of model predictive control techniques for active suspension," in *Proc. Int. Conf. Appl. Electron.*, 2015, pp. 157–160.
- [42] W.F. Milliken and D.L. Milliken, *Race Car Vehicle Dynamics*, Warrendale, PA, USA: SAE Int., 1995.
- [43] S. Fallah, A. Sorniotti, and P. Gruber, "A novel robust optimal active control of vehicle suspension systems," *IFAC Proc.*, vol. 47, no. 3, pp. 11213–11218, 2014.
- [44] D. Hrovat, "Survey of advanced suspension developments and related optimal control applications," *Automatica*, vol. 33, no. 10, pp. 1781–1817, 1997.
- [45] J. Drgona, M. Klauco, F. Janeczek, and M. Kvasnica, "Optimal control of a laboratory binary distillation column via regionless explicit MPC," *Comput. Chem. Eng.*, vol. 96, pp. 139–148, 2017.
- [46] C. Jablonowski, V. Underberg, and M. Paefgen, "The chassis of the all new Audi Q7," in *Proc. 6th Int. Munich Chassis Symp.*, 2015, pp. 37–49.

- [47] M. Anderson and D. Harty, "Unsprung mass with in-wheel motors - myths and realities," in *Proc. 10th Int. Symp. Adv. Veh. Control*, Loughborough, U.K., Aug. 22–26, 2010, pp. 261–266.
- [48] *Mechanical Vibration - Road Surface Profiles - Reporting of Measured Data*, ISO 8608:2016, 2016.
- [49] M. Kvasnica, "Multi-Parametric Toolbox 3," [Online]. Available: <http://people.ee.ethz.ch/~mpt/3/>. Accessed: Jul. 12, 2019.
- [50] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.
- [51] D. Karnopp, M.J. Crosby, and R.A. Harwood, "Vibration control using semi-active force generators," *J. Eng. Ind.*, vol. 96, no. 2, pp. 619–626, 1974.
- [52] *Mechanical Vibration and Shock - Evaluation of Human Exposure to Whole-Body Vibration - Part 1: General Requirements*, ISO 2631-1, 1997.



Johan Theunissen received the M.Sc. degree in electro-mechanical engineering from the K.U. Leuven, Leuven, Belgium, in 2008, and the Ph.D. degree in automotive engineering from the University of Surrey, Guildford, U.K., in 2019.

He recently founded Simmanco, an engineering company focusing on mechatronics and automotive. Earlier in his career, he worked for Ford Motor Company, Flanders' DRIVE and Tenneco Automotive Europe, as a Project Engineer and a Project Manager.



Aldo Sorniotti (M'12) received the M.Sc. degree in mechanical engineering and the Ph.D. degree in applied mechanics from the Politecnico di Torino, Turin, Italy, in 2001 and 2005, respectively.

He is a Professor in Advanced Vehicle Engineering with the University of Surrey, Guildford, U.K., where he coordinates the Centre for Automotive Engineering. His research interests include vehicle dynamics control and transmission systems for electric and hybrid electric vehicles.



Patrick Gruber received the M.Sc. degree in motorsport engineering and management from Cranfield University, Cranfield, U.K., in 2005, and the Ph.D. degree in mechanical engineering from the University of Surrey, Guildford, U.K., in 2009.

He is a Reader in advanced vehicle systems engineering with the University of Surrey. His research interests include vehicle dynamics and tire dynamics with special focus on friction behavior.



Saber Fallah received the B.Sc. degree from the Isfahan University of Technology, Isfahan, Iran, in 2001, the M.Sc. degree from Shiraz University, Shiraz, Iran, in 2004, and the Ph.D. degree from Concordia University, Montreal, QC, Canada, in 2010, all in mechanical engineering.

He is a Senior Lecturer of Vehicle and Mechatronic Systems with the University of Surrey, Guildford, U.K. His research interests include vehicle dynamics and control, electric and hybrid electric vehicles, intelligent vehicles, and vehicle system design and integration.



Marco Ricco received the M.Sc. degree in mechanical engineering from the Politecnico di Torino, Turin, Italy, in 2018. He is currently working toward the Ph.D. degree in automotive engineering with the University of Surrey, Guildford, U.K.

His research interests include electric vehicles, vehicle dynamics and vehicle testing.



Michal Kvasnica received the Diploma in process control from the Slovak University of Technology (STUBA), Bratislava, Slovakia, in 2000, and the Ph.D. degree in electrical engineering from the Swiss Federal Institute of Technology, Zurich, Switzerland, in 2008.

He is an Associate Professor (docent) of Automation with STUBA. His research interests include decision making and control supported by artificial intelligence, embedded optimization and control, security and safety of cyber-physical systems, and control of human-in-the-loop systems.



Miguel Dhaens received the M.Sc. degree in electro-mechanical engineering from KIH, Ostend, Belgium.

He is Engineering Manager of the Global Research Ride Performance Team of Tenneco, and responsible for defining the research road map and coordinating the global research activities of Tenneco's Ride Performance business, which includes vehicle dynamics, damping solutions, mechatronics, material science, manufacturing technologies and predictive tools. Before joining Tenneco, he worked with the Formula One team of Toyota Motorsport GmbH, Germany, as Manager of the engine, testing and advanced strategies teams, and with Flanders' DRIVE as R&D Manager.



Complexity reduction in explicit MPC: A reachability approach

Michal Kvasnica*, Peter Bakaráč, Martin Klaučo

Slovak University of Technology in Bratislava, Slovakia



ARTICLE INFO

Article history:

Received 25 May 2018

Received in revised form 3 October 2018

Accepted 2 December 2018

Available online xxxx

Keywords:

Model predictive control

Complexity

Reachability

Parametric optimization

Mixed-integer optimization

ABSTRACT

We propose to reduce the complexity of explicit MPC controllers by removing regions that will never be reached during the closed-loop evolution from a given set of initial conditions. The identification of such regions is done by solving a reachability analysis problem, formulated as a mixed-integer feasibility program. The procedure directly accounts for possible discrepancies between the prediction model and the actual plant dynamics by, among other things, considering a case where state measurements are affected by an unknown, but bounded measurement noise. The result of the procedure is the reduction of explicit MPC complexity without sacrificing closed-loop performance.

© 2018 Elsevier B.V. All rights reserved.

1. Introduction

Since its inception [1], Explicit Model Predictive Control (MPC) has proved to be a successful concept to design feedback controllers which enable fast and cheap implementation of optimization-based control in many applications, such as in automotive, aerospace, and process industries, see, e.g., [2–4]. In explicit MPC the optimal receding-horizon (RHC) feedback law is constructed as the explicit relation between the state measurements and the associated optimal control actions [5]. The relation, which for a rich class of MPC problems takes the form of a piecewise affine (PWA) function, is constructed by parametric optimization [6,7]. The on-line implementation of explicit MPC controllers then reduces to a mere function evaluation that can be performed fast even on hardware with modest computational resources.

The implementation complexity of explicit MPC, which entails both the required computational resources as well as the amount of memory required to store the PWA feedback law, is directly proportional to the total number of controller's regions. The number of regions, however, often exceeds practical limits. Therefore significant attention is devoted to reducing the complexity of explicit MPC controllers. Numerous approaches have been proposed in the literature to reduce the number of regions by, e.g., exploiting the geometry of explicit MPC solutions [8], tessellation techniques [9], convex liftings [10], bilevel optimization [11], or by using move blocking techniques [12].

In this paper we propose a novel method which removes from the explicit MPC controller those regions which are not reachable [13], in the closed-loop sense, from a given set of initial conditions \mathcal{L} . To illustrate the reasoning, consider the control of an object along a 1-dimensional axis. The states of the system are composed of the object's position and its velocity, i.e., $x = [p, v]^T$. The control objective is to manipulate the object's acceleration (which is the control input), such that the object moves from the current position to the origin. Once the origin is reached, the control system is stopped, the object is physically moved to a different location, and the control system is restarted. In such a scenario the set \mathcal{L} of initial conditions for the closed-loop system is given by $\mathcal{L} = \{(p, v) \mid v = 0\}$. Clearly, while moving the object from the initial position to the origin, a non-zero speed will be attained. Hence, during the closed-loop evolution, the states leave the set \mathcal{L} . However, not all of the controller's regions may be reached during the closed-loop response. Our objective is to determine, in a rigorous fashion, which regions could be reached if the closed-loop evolution starts from an arbitrary point in the set \mathcal{L} . If a particular critical region is determined to never be reachable, it can be removed from the definition of the explicit MPC feedback law, decreasing its complexity. This idea is shown graphically in Fig. 1. Note that the same result cannot be simply achieved by taking \mathcal{L} as the set of initial conditions for which the *open-loop* MPC problem is solved, as it would render the controller undefined for states outside of \mathcal{L} .

To the authors' best knowledge, this idea first appeared in [14]. There the authors suggested to identify reachable regions by performing numerical closed-loop simulations for several individual initial conditions. However, since only a finite number of initial conditions is investigated, one could, falsely, remove a region that could be reached by a different initial condition, not among the

* Corresponding author.

E-mail addresses: michal.kvasnica@stuba.sk (M. Kvasnica), peter.bakarac@stuba.sk (P. Bakaráč), martin.klauco@stuba.sk (M. Klaučo).

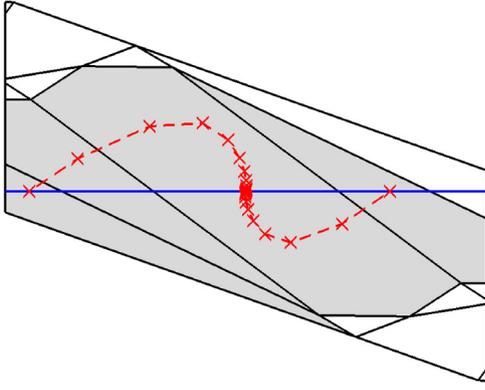


Fig. 1. Illustration of the idea. The red crosses depict the discrete-time closed-loop evolution of system's states starting from two distinct initial conditions. The regions in gray can be reached during the closed-loop operation, while the regions in the white color will never be reached. The objective is to identify which regions could be reached if a whole set \mathcal{L} of initial conditions (depicted as the solid blue line) is analyzed.

ones analyzed. To remedy such a drawback, the authors in [14] propose to employ interpolation, which naturally leads to loss of performance. In this paper we improve upon the aforementioned approach in two ways. First, we show how to analyze a whole convex set of initial conditions \mathcal{L} . Hence, our approach never removes regions that could be reached from some $x(0) \in \mathcal{L}$ and, as a consequence, the simplified controller offers the same closed-loop performance as the original (complex) feedback. Second, our procedure can be extended to cover cases where the state measurements are affected by an unknown noise, and to analyze scenarios when the MPC controller, designed for a linear prediction model, controls a system with a different dynamics. The decision about which regions are reachable is formulated as a reachability problem and solved as a feasibility mixed-integer problem that entails the optimality conditions of the underlying MPC feedback.

2. Preliminaries

We consider the control of linear discrete-time systems described by the state-space representation

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

with the state vector $x \in \mathcal{X} \subset \mathbb{R}^{n_x}$ and the input vector $u \in \mathcal{U} \subset \mathbb{R}^{n_u}$ with \mathcal{X} and \mathcal{U} being polytopical constraint sets that contain the origin in their respective interiors. The constrained finite-time optimal control problem is given by

$$U_{ol}^* = \arg \min_{x_N^T Q_N x_N + \sum_{k=0}^{N-1} x_k^T Q_x x_k + u_k^T Q_u u_k} \quad (2a)$$

$$\text{s.t. } x_{k+1} = Ax_k + Bu_k, \quad k = 0, \dots, N-1, \quad (2b)$$

$$x_k \in \mathcal{X}, \quad k = 0, \dots, N-1, \quad (2c)$$

$$u_k \in \mathcal{U}, \quad k = 0, \dots, N-1, \quad (2d)$$

$$x_N \in \mathcal{X}_f, \quad (2e)$$

where x_k and u_k are, respectively, predictions of states and inputs at the k th step of the prediction horizon (denoted by N). Moreover, $Q_N = Q_N^T \geq 0$, $Q_x = Q_x^T \geq 0$ and $Q_u = Q_u^T > 0$ denote weighting matrices, and $\mathcal{X}_f \subseteq \mathcal{X}$ is the terminal set. Finally, $U_{ol}^* = [u_0^T, \dots, u_{N-1}^T]^T$ denotes the open-loop sequence of optimal control moves obtained by solving (2) for a particular initial condition x_0 . After introducing the substitution $x_{k+1} = A^k x_0 + \sum_{i=0}^{k-1} A^{k-i-1} B u_i$, the open-loop profile of predicted states, i.e., $X_{ol} =$

$[x_0^T, \dots, x_N^T]^T$, can be compactly written as $X_{ol} = \Gamma x_0 + \Psi U_{ol}$, and (2) can be rewritten [6] into

$$U_{ol}^*(x_0) = \arg \min_{U_{ol}} \frac{1}{2} U_{ol}^T P U_{ol} + x_0^T Q U_{ol} \quad (3a)$$

$$\text{s.t. } G U_{ol} \leq w + E x_0, \quad (3b)$$

$$x_0 \in \mathcal{K}, \quad (3c)$$

which is a strictly convex parametric quadratic program (QP) with x_0 being the parameters, and $\mathcal{K} \subset \mathbb{R}^{n_x}$ is the set of parameters of interest (typically, $\mathcal{K} = \mathcal{X}$).

It is well known (see, e.g., [1,6]) that the explicit representation of U_{ol}^* as a function of the initial condition x_0 can be obtained by solving (3) using parametric optimization. Then, $U_{ol}^*(x_0)$ is a piecewise affine function, i.e.,

$$U_{ol}^*(x_0) := F_i x_0 + g_i \text{ if } x_0 \in \mathcal{R}_i, \quad i = 1, \dots, R, \quad (4)$$

where $F_i \in \mathbb{R}^{n_u \times n_x}$, $g_i \in \mathbb{R}^{n_u}$, and R denotes the total number of polytopical critical regions \mathcal{R}_i , described by

$$\mathcal{R}_i = \{x_0 \mid H_i x_0 \leq h_i\}. \quad (5)$$

The receding-horizon feedback law is obtained by calculating the open-loop sequence U_{ol}^* for a particular initial condition $x_0 = x(t)$ at each sampling step, but only employing its first element, i.e., u_0^* , as the closed-loop control action. Hence, the RHC feedback law $\kappa: \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_u}$ is given by

$$\kappa(x(t)) = \underbrace{\begin{bmatrix} I & 0 & \dots & 0 \end{bmatrix}}_{\phi} U_{ol}^*(x(t)). \quad (6)$$

In this paper we study the evolution of the system in (1) subject to MPC feedback law $u(t) = \kappa(x(t))$, i.e.,

$$x(t+1) = Ax(t) + B\kappa(x(t)), \quad (7)$$

In particular, we investigate the closed-loop trajectories $X_{cl}^M = [x(0)^T, \dots, x(M)^T]^T$ over a finite number M of time steps. Note that the system in (7) is nonlinear because, in general, the feedback law $\kappa(\cdot)$ is a piecewise affine function due to (4).

Remark 2.1. The receding horizon nature of the feedback law in (6) causes, in general, a mismatch between the closed-loop state trajectory X_{cl}^M of (7) and the open-loop predictions X_{ol} even when the prediction model (2b) is identical to the dynamics of the controlled system in (7), see, e.g., [15, Section 13.1].

3. Problem statement

We aim at reducing the complexity of explicit representations of MPC feedback laws in (6) by removing critical regions of (4) that will be unreachable when the evolution of the closed-loop system (7) starts from some $x(0) \in \mathcal{L}$, where $\mathcal{L} \subset \mathbb{R}^{n_x}$ is a known set.

Definition 3.1. The set of states $\mathcal{S} \subseteq \mathbb{R}^{n_x}$ is called *reachable from the set \mathcal{L} by the system in (7) under the RHC feedback (6)* if there exists an initial state $x(0) \in \mathcal{L}$ such that $x(M) \in \mathcal{S}$ for some finite M , where $x(i)$ is the i th element of the closed-loop sequence governed by (7). If no such $x(0) \in \mathcal{L}$ exists, the set \mathcal{S} is called *unreachable*.

If some critical regions of (4) are determined to be unreachable, they can be removed from (4), hence reducing the memory and computational resources required to implement such a controller. The difficulty of determining the reachability/unreachability status of a particular critical region stems from the fact that one needs to analyze all possible initial conditions from the set \mathcal{L} . Moreover, each initial condition has a different optimal control action associated to it via (2).

Remark 3.2. It is important to realize that there is a fundamental difference between the set \mathcal{L} and the set \mathcal{K} in (3c). The former is the set of the initial conditions for the closed-loop system, i.e., $x(0) \in \mathcal{L}$, while \mathcal{K} is the set of initial conditions for the open-loop problem, i.e., $x_0 \in \mathcal{K}$. As can be seen in Fig. 1, it is expected that the closed-loop states $x(t)$ leave the set \mathcal{L} for some $t > 0$. Hence, setting $\mathcal{K} \equiv \mathcal{L}$ would not lead to the same result as the procedure of this paper. Instead, it would force the explicit MPC controller to be undefined for states outside of \mathcal{L} since \mathcal{K} is the set of parameters for which (3) is solved.

The reachability analysis problem can be formally stated as follows:

Problem 3.3. Given the critical regions \mathcal{R}_i of the PWA feedback law (4) and a polytopic set \mathcal{L} of initial conditions for the closed-loop system (7), determine, for each region, whether it is reachable by the closed-loop system (7) under the RHC feedback (6), i.e., find $x(0) \in \mathcal{L}$ such that $x(M) \in \mathcal{R}_i$ for some finite non-negative integer M , or determine that no such $x(0) \in \mathcal{L}$ exists.

Remark 3.4. Any critical region \mathcal{R}_i that has a non-empty intersection with \mathcal{L} is trivially reachable, according to Definition 3.1, for $M = 0$. Such a detection can be performed at the price of solving a single linear program for each region. We also pronounce as reachable any critical region that intersects the terminal set \mathcal{X}_f to avoid dealing with $M \rightarrow \infty$ due to asymptotic convergence inside of \mathcal{X}_f if the terminal set is composed of several (e.g., nested) critical regions.

4. Main results

The reachability of a particular critical region \mathcal{R}_i by the MPC feedback law (6) in exactly j steps is identical to the existence of an initial condition $x(0) \in \mathcal{L}$ such that the final element of the evolution of the system in (7), i.e., $x(j) = Ax(j-1) + B\kappa(x(j-1))$, satisfies $x(j) \in \mathcal{R}_i$. We recall that $\kappa(x(k)) = \Phi U_{oi}^*(x(k))$ via (6) where $U_{oi}^*(x(k))$ is the open-loop optimizer of (2) for the particular initial condition $x(k)$, $k = 0, \dots, j-1$. The difficulty of determining the reachability status lies in the fact that $U_{oi}^*(x(k))$ depends on $x(k)$ and therefore attains different values at different time steps.

Technically, the reachability of \mathcal{R}_i in exactly j steps can be stated as

$$\text{find } x(0) \quad (8a)$$

$$\text{s.t. } x(0) \in \mathcal{L}, \quad (8b)$$

$$x(j) \in \mathcal{R}_i, \quad (8c)$$

$$x(k+1) = Ax(k) + B\Phi U_{oi}^*(x(k)), \quad (8d)$$

$$U_{oi}^*(x(k)) = \arg \min_{U_{oi}(k)} 1/2 U_{oi}(k)^T P U_{oi}(k) + x(k)^T Q U_{oi}(k) \quad (8e)$$

$$\text{s.t. } GU_{oi}(k) \leq w + Ex(k), \quad x(k) \in \mathcal{K} \quad (8f)$$

where constraints (8d)–(8f) are enforced for $k = 0, \dots, j-1$. In what follows we first state main theoretical results before providing a computationally tractable formulation of (8).

Lemma 4.1. If (8) is feasible for some $j \in [0, \dots, M]$, then the critical region \mathcal{R}_i is reachable from \mathcal{L} under the RHC feedback in the sense of Definition 3.1. If the problem is infeasible for all $j = 0, \dots, M$, then \mathcal{R}_i is unreachable from \mathcal{L} in up to M steps.

Proof. Follows directly from Definition 3.1, from the fact that $\kappa(x(k)) = \Phi U_{oi}^*(x(k))$ as in (6), and since (8e)–(8f) represents the open-loop optimizer to (2) with $x(k)$ as the initial condition. ■

Lemma 4.2. Let \mathcal{R}_i with $\mathcal{R}_i \cap (\mathcal{X}_f \cup \mathcal{L}) = \emptyset$ be given and let the tuning parameters of the MPC problem (2) be such that the controller forces the state of the closed-loop system (7) to enter a positively invariant terminal set \mathcal{X}_f in, at most, M time steps for all feasible initial conditions $x(0)$. If (8) is infeasible for all $j = 0, \dots, M$ then \mathcal{R}_i is unreachable for all time, i.e., $\nexists k > 0$ such that $x(k) \in \mathcal{R}_i$.

Proof. Infeasibility of (8) for all $j = 0, \dots, M$ implies unreachability of \mathcal{R}_i in up to M steps. Since the terminal set \mathcal{X}_f is assumed to be positively invariant, the MPC setup (2) is assumed to be tuned in a way that \mathcal{X}_f is reached in at most M steps, and because only the critical regions outside of \mathcal{X}_f are considered (cf., Remark 3.4), it follows that after at most M steps the states of the closed-loop system (7) enter \mathcal{X}_f and stay in the set for all future time instants. Therefore if \mathcal{R}_i is unreachable in, at most, M steps, it will not be reachable in $M+k$ steps for an arbitrary $k > 0$. ■

Remark 4.3. In practice, it is difficult to derive (or even to estimate) M directly from parameters of the open-loop MPC problem in (2). A practical approach to detect M would be to grid the state-space, perform numerical closed-loop simulations, and detect the largest number of steps the closed-loop response requires to reach the terminal set. Alternatively, an a-priori upper bound on M can be set and be interpreted as the worst acceptable liveness of the controller. Then infeasibility of (12) implies that the corresponding critical region is not reached during the closed-loop evolution, or that \mathcal{R}_i could be reached, but after an unacceptably long time.

As noted in Section 3, any regions that are unreachable from the given set of initial conditions \mathcal{L} may be safely removed from the PWA feedback law (4), thus reducing its memory storage as well as the induced amount of computational operations required to evaluate (4) for given state measurements. Although the reachability status of each critical region is determined off-line it is desired, from a practical point of view, to be able to solve (8) swiftly, especially for controllers with many critical regions. Therefore in the next sections we review various ways of formulating and solving (8).

Remark 4.4. In practice, infeasibility of (8) can be caused by aspects not related to reachability of \mathcal{R}_i , e.g., by running into a time limit in the solver or encountering numerical difficulties. Therefore we suggest to soften the hard constraint (8c), i.e., to replace it by $H_i x(j) \leq h_i + s_i$ where H_i, h_i are the defining half-spaces of the i th critical region (see (5)), $s_i \in \mathbb{R}^{n_i}$ is a vector of non-negative slack variable, and n_i denotes the number of constraints of the i th region. Moreover, the feasibility objective in (8a) should be replaced by $\min 1^T s_i$. Such a modification renders (8) always feasible. If, however, $s_{i,\ell}^* > 0$ for some of its element(s), then the original hard constraint (8c) could not be satisfied without relaxing it, i.e., $s_{i,\ell}^* > 0$ in the modified problem implies infeasibility of (8) with hard constraints in (8c).

4.1. Formulation of (8) as a mixed-integer optimization problem using KKT conditions

The main difficulty of determining feasibility of (8) stems from the fact that it is a bilevel optimization problem with a feasibility outer objective (8a) and an associated inner optimization problem (8e)–(8f), which is equivalent to (3). These two problems are coupled by the open-loop optimizer $U_{oi}^*(x(k))$ from the lower problem, which in turn depends on $x(k)$ from the upper problem. A standard way of solving bilevel problems of the form (8) is to replace the inner problem by its Karush–Kuhn–Tucker (KKT)

conditions [16, Section 5.5.3]:

$$PU_{0i}^*(k) + Q^T x(k) + G^T \lambda(k) = 0, \quad (9a)$$

$$GU_{0i}^*(k) \leq w + Ex(k), \quad (9b)$$

$$\lambda(k) \geq 0, \quad (9c)$$

$$\lambda_i(k)(G_i U_{0i}^*(k) - w_i - E_i x(k)) = 0, \quad (9d)$$

where (9a) is the stationarity condition, (9b) represents primal feasibility, (9c) is the dual feasibility, and (9d) stands for the complementary slackness condition, which is imposed for $i = 1, \dots, n_c$, where n_c is the number of rows of G . Moreover, G_i denotes the i th row of the corresponding matrix. Since the inner problem (8e)–(8f) is a strictly convex parametric QP, the KKT conditions (9) are necessary and sufficient [16, Section 5.5.3]. However, they are nonlinear due to product between the Lagrange multipliers λ and the decision variables U_{0i}^* in (9d).

Such a nonlinearity can be worked around by realizing that for (9d) to hold, either $\lambda_i(k) = 0$ or $G_i U_{0i}^*(k) - w_i - E_i x(k) = 0$. One can introduce binary indicators $\delta_i(k) \in \{0, 1\}$ and $\gamma_i(k) \in \{0, 1\}$ for $k = 0, \dots, j-1$ and $i = 1, \dots, n_c$ such that

$$(\delta_i(k) = 1) \Leftrightarrow (\lambda_i(k) = 0), \quad (10a)$$

$$(\gamma_i(k) = 1) \Leftrightarrow (G_i U_{0i}^*(k) - w_i - E_i x(k) = 0). \quad (10b)$$

By applying standard rules of propositional logic [17], the equivalences in (10) can be furthermore rewritten into a set of inequalities that are linear in the decision variables $\lambda_i(k)$, $U_{0i}^*(k)$, $\delta_i(k)$, and $\gamma_i(k)$:

$$-Z(1 - \delta_i(k)) \leq \lambda_i(k) \leq Z(1 - \delta_i(k)), \quad (11a)$$

$$-Z(1 - \gamma_i(k)) \leq G_i U_{0i}^*(k) - w_i - E_i x(k) \leq Z(1 - \gamma_i(k)), \quad (11b)$$

where Z is a sufficiently large constant. It is trivial to verify that if $\delta_i(k) = 1$ in (11a), then $\lambda_i(k) = 0$ is the only feasible value. If $\delta_i(k) = 0$, then (11a) is inactive. Similar reasoning holds for (11b). Then the complementarity slackness condition (9d) can be equivalently written as the propositional logic statement of the form $\delta_i(k) \vee \gamma_i(k)$ (i.e., either the i th Lagrange multiplier is zero, or the i th constraint is active). Such a statement can be equivalently written as $\delta_i(k) + \gamma_i(k) \geq 1$. Therefore the KKT conditions (9) can be equivalently written as a combination of (9a)–(9c), together with (11) and $\delta_i(k) + \gamma_i(k) \geq 1$ replacing (9d); and will be denoted by $KKT(x(k), U_{0i}^*(k), \lambda(k), \delta(k), \gamma(k)) \leq 0$ in the sequel. Then the bilevel optimization problem (8) can be equivalently written as

$$\text{find } x(0) \quad (12a)$$

$$\text{s.t. } x(0) \in \mathcal{L}, \quad (12b)$$

$$x(j) \in \mathcal{R}_i, \quad (12c)$$

$$x(k+1) = Ax(k) + B\Phi U_{0i}^*(k), \quad (12d)$$

$$KKT(x(k), U_{0i}^*(k), \lambda(k), \delta(k), \gamma(k)) \leq 0, \quad (12e)$$

where constraints (12d) and (12e) are imposed for $k = 0, \dots, j-1$. Since \mathcal{L} and \mathcal{R}_i are assumed to be polytopes, and because (12e) can be cast as a set of mixed-integer inequalities as per (9a)–(9c) and (11), problem (12) for a finite j is a mixed-integer feasibility problem in decision variables $x(0), \dots, x(j)$ with $x(k) \in \mathbb{R}^{n_x}$, $U_{0i}^*(0), \dots, U_{0i}^*(j-1)$ with $U_{0i}^*(k) \in \mathbb{R}^{n_{u_i}}$, $\lambda(k) \in \mathbb{R}^{n_c}$, and binary decision variables $\delta(k) \in \{0, 1\}^{n_c}$ and $\gamma(k) \in \{0, 1\}^{n_c}$ for $k = 0, \dots, j-1$.

4.2. Formulation of (8) as a mixed-integer optimization problem using the explicit optimizer

An alternative way of formulating and solving (8) as a mixed-integer problem is to employ the explicit representation of the optimizer to (8e)–(8f), readily available in (4). By doing so,

(8e)–(8f) can be removed and (8d) is replaced by

$$x(k+1) = Ax(k) + B\Phi(F_i x(k) + g_i) \text{ if } x(k) \in \mathcal{R}_i, \quad (13)$$

for $i = 1, \dots, R$ where R denotes the number of critical regions. Note that (13) involves IF/THEN rules. These can be tackled in the propositional logic framework as follows. First, we equivalently rewrite (13) into

$$[x(k) \in \mathcal{R}_i] \Rightarrow [x(k+1) = Ax(k) + B\Phi(F_i x(k) + g_i)]. \quad (14)$$

Subsequently, we introduce binary variables $\theta_i(k) \in \{0, 1\}$ for $i = 1, \dots, R$ and $k = 0, \dots, j$ such that $\theta_i(k) = 1$ if and only if $x(k) \in \mathcal{R}_i$. Since the union of critical regions of the parametric solution to a strictly convex QP of the form (3) is convex, this is equivalent to (see [18, Section 3.1])

$$H_i x(k) - h_i \leq Z(1 - \theta_i(k)), \quad \sum_{i=1}^R \theta_i(k) = 1, \quad (15)$$

where H_i and h_i define the half-spaces of the i th critical region in (5), and Z is a sufficiently large positive constant. Finally, the implication in (14) is equivalent to

$$-Z(1 - \theta_i(k)) \leq x(k+1) - Ax(k) - B\Phi(F_i x(k) + g_i) \leq Z(1 - \theta_i(k)), \quad (16)$$

by exploiting continuity of the explicit optimizer in (4) along boundaries of critical regions.

Therefore (8) can be equivalently reformulated by replacing (8d) by (16) and (8e)–(8f) by (15). As all constraints are linear, the reformulated problem is again a mixed-integer feasibility program. However, unlike the formulation in (12) which has a total of $2n_c$ binary variables for each step $k = 0, \dots, j$ (specifically, $\delta(k)$ and $\gamma(k)$), the formulation of this section features R binary variables $\theta(k)$ for each step of the reachability analysis. In practice, $R \gg n_c$ and therefore the formulation of Section 4.1 is typically superior, from a computational point of view, to the approach based on an explicit optimizer.

4.3. Formulation of (8) as a set-based reachability problem

Another option is to approach (8) as a set-based reachability problem. Specifically, denote by

$$\mathcal{S}(j) = \{x(j) \mid x(0) \in \mathcal{L}, x(k+1) = Ax(k) + B\kappa(x(k)), k = 0, \dots, j-1\} \quad (17)$$

the set of states of the system in (1) that are reachable from the set of initial conditions \mathcal{L} under the MPC feedback law κ in (6) in j steps. Clearly, if $\mathcal{S}(j) \cap \mathcal{R}_i = \emptyset$ for all $j = 0, \dots, M$, then the i th critical region is unreachable and can therefore be removed from the explicit optimizer (4). The difficulty of characterizing the sets $\mathcal{S}(j)$ stems from the fact that κ is a PWA function, therefore computing (17) entails solving a forward reachability problem for PWA systems [19, Chapter 6.3]. Specifically, the dynamics of the closed-loop system in (7) subject to the PWA controller (4) and (6) is given by the autonomous system

$$x(k+1) = \underbrace{(A + B\Phi F_i)}_{\tilde{A}_i} x(k) + \underbrace{\Phi g_i}_{\tilde{f}_i} \text{ if } x(k) \in \mathcal{R}_i, \quad (18)$$

where F_i, g_i are the parameters of the explicit optimizer in (4), and Φ is as in (6). Let $\mathcal{S}(0) = \{\mathcal{L}\}$ and denote by $\mathcal{I}_\ell(k) = \{i \mid \mathcal{R}_i \cap \mathcal{S}_\ell(k) \neq \emptyset\}$ the index set of critical regions that have a non-empty intersection with the ℓ th element of $\mathcal{S}(k)$ for $k = 0, \dots, j-1$. Then

$$\mathcal{S}(k+1) = \{\tilde{A}_i \circ (\mathcal{R}_i \cap \mathcal{S}_\ell(k)) + \tilde{f}_i\}_{\forall i \in \mathcal{I}_\ell(k), \forall \ell \in \{1, \dots, |\mathcal{S}(k)|\}}, \quad (19)$$

where $\tilde{A} \circ \mathcal{P} + \tilde{f} = \{\tilde{A}x + \tilde{f} \mid x \in \mathcal{P}\}$ is the one-step forward reachable set of the autonomous system $x(t+1) = \tilde{A}x(t) + \tilde{f}$, and

$|\mathcal{S}(k)|$ is the cardinality of the set. We remark that if $\mathcal{P} = \mathcal{R}_i \cap \mathcal{S}_\ell(k)$ is a polytope (as is the case for polytopic critical regions (5) and polytopic set of initial conditions \mathcal{L}), each element of $\mathcal{S}(k+1)$ will be a polytope as well [15, Section 5.4.11]. Therefore the set $\mathcal{S}(j)$ can be computed by propagating the sets in (19) forwards in time for $k = 0, \dots, j-1$. The downside of this approach is that the cardinality of $\mathcal{S}(k)$ increases, in the worst case exponentially, with k and with the number of critical regions.

5. Complexity reduction

Next we show how to employ the reachability procedures reported in Section 4 to reduce complexity of explicit MPC feedback laws in (4) and (6). Denote by $\mathcal{I} \subseteq \{1, \dots, R\}$ the index set of the critical regions \mathcal{R}_i of (4) that are reachable¹ from \mathcal{L} . Consider the reduced optimizer

$$\tilde{U}_{01}^*(x) = F_i x + g_i \text{ if } x \in \mathcal{R}_i, \forall i \in \mathcal{I}, \quad (20)$$

created from (4) by retaining only the critical regions indexed by \mathcal{I} . Let

$$\tilde{\kappa}(x) = [I \quad 0 \quad \dots \quad 0] \tilde{U}_{01}^*(x) \quad (21)$$

be the corresponding reduced RHC feedback law.

Lemma 5.1. *Let the set \mathcal{L} of initial conditions for the closed-loop system (7) be given, along with parameters of the MPC problem (2). Assume that these parameters are chosen such that the RHC feedback (6) with the original (complex) optimizer (4) drives the states of the closed-loop system to the terminal set \mathcal{X}_f in finite time. Then the closed-loop profile of (7) under the original complex feedback $\kappa(\cdot)$ is equivalent to the closed-loop profile under the reduced feedback $\tilde{\kappa}(\cdot)$ for all $x(0) \in \mathcal{L}$.*

Proof. Follows directly from the fact that the reduced optimizer $\tilde{U}_{01}^*(\cdot)$ contains all reachable critical regions of $U_{01}^*(\cdot)$. Hence there is no $x(0) \in \mathcal{L}$ which would generate a closed-loop sequence (7) that would enter any critical region not indexed by \mathcal{I} . ■

A consequence of Lemma 5.1 is that the reduced feedback (21) is equivalent, in the closed-loop sense, to the original (complex) strategy (6). Under such an equivalence the complexity of the explicit MPC optimizer (4) can be reduced as follows. Assume the explicit representation of $U_{01}^*(x)$ in (4) is available. For each critical region \mathcal{R}_i with $i = 1, \dots, R$, determine reachability of \mathcal{R}_i per the procedures of Section 4. Then the index set \mathcal{I} of reachable critical regions is obtained by solving R decision problems described previously. Afterwards, only the regions indexed by \mathcal{I} are retained and the unreachable critical regions are removed from the explicit representation of the RHC controller.

6. Extensions

6.1. State measurements affected by measurement noise

The reachability approach to complexity reduction in explicit MPC presented in Section 4 can be further extended to cases where the state measurements are affected by an unknown, but bounded, measurement noise $\xi \in \mathcal{E}$ with \mathcal{E} being a polytope. In this case, (7) becomes

$$x(t+1) = Ax(t) + B\kappa(x(t) + \xi(t)). \quad (22)$$

¹ Recall that, by Remark 3.4, any critical region that intersects either \mathcal{L} or \mathcal{X}_f is considered reachable. The reachability status of the remaining critical regions needs to be checked per the procedures of Section 4.

Notice that ξ enters only into the argument of the feedback law (6), which is evaluated for noisy state measurements $x(t) + \xi(t)$, but the noise does not directly affect the open-loop predictions (2b).

Determining reachability/unreachability of a particular critical region \mathcal{R}_i in the presence of measurement noise amounts to determining whether there exists a sequence $\{\xi(0), \dots, \xi(M-1)\}$ which, when employed as a measurement noise, will drive the system (22) to \mathcal{R}_i in, at most, M steps.

Such a decision problem can be formulated by extending the formulation in (12) into the form

$$\text{find } x(0) \quad (23a)$$

$$\text{s.t. } x(0) \in \mathcal{L}, \quad (23b)$$

$$x(j) \in \mathcal{R}_i, \quad (23c)$$

$$x(k+1) = Ax(k) + B\Phi U_{01}^*(k), \quad (23d)$$

$$\text{KKT}(x(k) + \xi(k), U_{01}^*(k), \lambda(k), \delta(k), \gamma(k)) \leq 0, \quad (23e)$$

$$\xi(k) \in \mathcal{E}, \quad (23f)$$

where (23d)–(23f) are imposed for $k = 0, \dots, j-1$. The KKT conditions in (23e) implicitly define the optimal open-loop sequence associated to the noisy state measurements and can be converted into a set of linear inequalities involving binary variables using the propositional logic framework as set forth in Sections 4.1 and 4.2.

Lemma 6.1. *Let \mathcal{R}_i with $\mathcal{R}_i \cap (\mathcal{X}_f \cup \mathcal{L}) = \emptyset$ be given. If the mixed-integer feasibility problem (23) is feasible for some $j \in [0, \dots, M]$, then the critical region \mathcal{R}_i is reachable from \mathcal{L} under the RHC feedback. If the problem is infeasible for all $j = 0, \dots, M$, then \mathcal{R}_i is unreachable from \mathcal{L} .*

Proof. Follows along the lines of proofs of Lemmas 4.1 and 4.2. Specifically, if (23) is feasible for some j , then there exists a $x(0) \in \mathcal{L}$, along with a sequence of measurement noises $\{\xi(0), \dots, \xi(j-1)\}$, such that $x(j) \in \mathcal{R}_i$. If no such initial condition and/or sequence of noise values could be found, the region is unreachable in finite time. If, moreover, reachability of the terminal set \mathcal{X}_f in M steps is enforced by a proper tuning of the MPC problem in (2), it follows from Lemma 4.2 that infeasibility of (23) for all $j \in [0, \dots, M]$ implies unreachability of \mathcal{R}_i by $x(k)$ for any $k > 0$. ■

Remark 6.2. Critical region \mathcal{R}_i is declared reachable by Lemma 6.1 if there exists a sequence of noise variables, i.e., $\{\xi(0), \dots, \xi(j-1)\}$, such that $x(j) \in \mathcal{R}_i$ for some $0 \leq j \leq M$. This sequence corresponds to the worst-case behavior of the closed-loop system. Employing a different sequence of noise values can render the region unreachable. However, since the noise is assumed to be a random variable, the worst case needs to be considered. On the other hand, if (23) is infeasible for all $j = 0, \dots, M$, then \mathcal{R}_i is unreachable regardless of a choice of the noise with $\xi(j) \in \mathcal{E}$ holds.

6.2. Nonlinear systems

So far we have considered the case where the dynamics of the closed-loop system in (7) is identical to that of the prediction model in (2b). The results can be further generalized as follows. Consider that the closed-loop dynamics (7) is driven by

$$x(t+1) = f(x(t), u(t)), \quad (24)$$

where f is an arbitrary vector field. Notice, however, that we assume that $u(t) = \kappa(x(t))$ with κ as in (6), i.e., the system is controlled by an MPC strategy designed for a linear prediction model in (2b). By replacing (8d) by

$$x(k+1) = f(x(k), \Phi U_{01}^*(x(k))), \quad (25)$$

and after treating (8e)–(8f) using the propositional logic framework of Sections 4.1 and 4.2, problem (8) becomes a mixed-integer nonlinear program (MINLP). Importantly, Lemmas 4.1 and 4.2 still apply, i.e., infeasibility of (8) with (8d) replaced by (25) implies unreachability of a particular critical region even under the nonlinear dynamics in (24). This extension therefore allows to study the behavior of linear MPC in conjunction with nonlinear systems. One obvious application is in the spirit of complexity reduction as set forth in Section 5 where one removes from the explicit MPC feedback law the regions that are unreachable by the nonlinear system.

The procedure can also be used to analyze the safety of linear MPC applied to control nonlinear systems. Specifically, denote by $\mathcal{F} = \cup_i \mathcal{R}_i$ the feasible set of the explicit optimizer (4) (we remark that for strictly convex QP formulations (3) \mathcal{F} is a polytope), and let $\mathcal{E} = \mathbb{R}^{n_x} \setminus \mathcal{F}$ be the complement to \mathbb{R}^{n_x} , i.e., the set of states outside of \mathcal{F} . We remark that $\mathcal{E} = \{\mathcal{E}_n\}$ is composed of a finite number of polyhedra \mathcal{E}_n [15, Section 5.5.1] since \mathcal{F} is a polytope. Then finite-time reachability of any element of \mathcal{E} (verified by solving (8) with \mathcal{R}_i replaced by \mathcal{E}_n in (8c)) implies a lack of recursive feasibility guarantees of the linear MPC controller (2) in conjunction with the nonlinear dynamics (24). Such an a-posteriori analysis allows the control designer to base the control synthesis on linear models and check if the safety guarantees (i.e., infinite-time constraint satisfaction) extend to a nonlinear dynamics.

It should be noted that MINLP formulations arising from such an extension are notoriously difficult to solve. However, if the nonlinear vector field f in (24) is piecewise affine, i.e., $f(x(t), u(t)) := A_i x(t) + B_i u(t) + g_i$ if $(x(t), u(t)) \in \mathcal{D}_i$ with polyhedral regions of validity \mathcal{D}_i , then the resulting feasibility optimization problem can be cast as a mixed integer linear program that can be solved rather efficiently using off-the-shelf tools, see [18, Section 3.1] for details.

6.3. Reachability-based explicit MPC solver

One of the advantages of the KKT-based reachability analysis of Section 4.1 is that (12) can determine reachability of an arbitrary set \mathcal{R}_i (i.e., not necessarily just that of a critical region) by employing only the *implicit* representation of the MPC feedback law, cf. (8e)–(8f) and (12e). As a consequence, the *full explicit* MPC solution as in (4) is *not* required to determine reachability of \mathcal{R}_i .

This observation can be used to design an explicit MPC solver that never generates unreachable regions in the first place, i.e., no post-processing of Section 5 is required. To do so, we propose to extend the enumeration method for parametric QPs (3), originally suggested in [20]. Here, instead of approaching critical regions (5) as geometric sets, the regions are equivalently captured by their underlying active sets, i.e., the index sets of constraints that are active at the optimum. Specifically, the primal feasibility constraints in (9b) are split into

$$G_{\mathcal{A}_i} U_{ol}^* = w_{\mathcal{A}_i} + E_{\mathcal{A}_i} x, \quad (26a)$$

$$G_{\mathcal{N}_i} U_{ol}^* < w_{\mathcal{N}_i} + E_{\mathcal{N}_i} x, \quad (26b)$$

where $\mathcal{A}_i \subseteq \{1, \dots, n_c\}$ (we recall that n_c is the number of constraints in (3b)) is the index set of active constraints and $\mathcal{N}_i = \{1, \dots, n_c\} \setminus \mathcal{A}_i$ are the inactive constraints. To determine all optimal active sets, the authors of [20] have proposed to enumerate all possible combinations thereof, followed by a branch-and-cut pruning of the tree of possible active sets. Specifically, if a particular active set \mathcal{A}_i is determined to be infeasible, so will be any other active set $\mathcal{A}_{i'}$ with $\mathcal{A}_{i'} \supset \mathcal{A}_i$, cf. [20, Section 3.2]. Such a feasibility-based pruning allows to keep the size of the exploration tree (which would otherwise explode exponentially with the problem size) under control.

If the parametric QP (3) originates from an MPC problem (2) where, moreover, only a limited subset of initial conditions $x(0) \in$

\mathcal{L} is considered, then the procedure of [20] can be further improved as follows. Note that feasibility of (26a) is a necessary condition for the inclusion in (12c) to hold. Next, replace $x(j) \in \mathcal{R}_i$ in (12c) by $G_{\mathcal{A}_i} U_{ol}^*(x(j)) = w_{\mathcal{A}_i} + E_{\mathcal{A}_i} x(j)$ from (26a). Clearly, infeasibility of (12) then implies that \mathcal{A}_i is not a feasible active set. Thus the corresponding critical region \mathcal{R}_i is unreachable either in finite time (Lemma 4.1) or in infinite time (Lemma 4.2). Subsequently, we have by [20, Section 3.2] that any other active set $\mathcal{A}_{i'} \supset \mathcal{A}_i$ will be infeasible, and, in consequence, the associated region $\mathcal{R}_{i'}$ will be unreachable and therefore need to be constructed and included into (4). In short, we are proposing to extend the feasibility test of [20, eq. (16)] by including (12) such that regions that are not reachable from a given set of initial conditions $x(0) \in \mathcal{L}$ are not generated during the construction of the explicit solution (4). Doing so has two upsides. First, the size of the exploration tree is limited further, allowing the enumeration-based approach to perform faster. Second, such an approach directly generates the simple solution without having to perform, a-posteriori, the analysis of Section 5.

7. Examples

7.1. Illustrative 2D example

We consider the control of pendulum, which consists of a ball attached to the ceiling by a rope. The dynamics of the system is modeled by the discrete-time linear system

$$x(t+1) = \begin{bmatrix} 0.5403 & -0.8415 \\ 0.8415 & 0.5403 \end{bmatrix} x(t) + \begin{bmatrix} -0.4597 \\ 0.8415 \end{bmatrix} u(t) \quad (27)$$

where $x \in \mathbb{R}^2$ is the state vector (composed of the ball's position and its velocity), and the control input $u \in \mathbb{R}$ is the acceleration of the ball. The states and the input of (27) are constrained by $\begin{bmatrix} -10 \\ 10 \end{bmatrix} \leq x \leq \begin{bmatrix} 10 \\ 10 \end{bmatrix}$ and $-1 \leq u \leq 1$. The penalties were chosen as $Q_x = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $Q_u = 1$, and Q_N given as the solution to the algebraic Riccati equation, \mathcal{X}_t designed as the LQR set, and the prediction horizon was set to $N = 8$. The explicit MPC solution in (4) was calculated with the MPT toolbox [21]. The solution consisted of 147 polytopic critical regions in the 2-dimensional space.

To reduce the complexity, we have first applied the procedure of Section 4.1 with the set of initial conditions for the closed-loop system $\mathcal{L} = \{x \mid x_2 = 0, -5 \leq x_1 \leq 5\}$ (i.e., the ball placed anywhere between ± 5 with zero velocity). The reachability of each critical region was tested using the procedures of Section 4.1–4.3. Specifically, the mixed-integer problems of Sections 4.1 and 4.2 were formulated by YALMIP [22] and solved by Gurobi. The set-based reachability method of Section 4.3 was implemented using the MPT toolbox. The runtimes² of respective methods are reported in Table 1. All methods provided the same answer, namely that only 23 out of the 147 regions are reachable by the closed-loop system with $x(0) \in \mathcal{L}$. Therefore the complexity of the simple controller (20)–(21) was reduced by a factor of 6.3 without sacrificing the closed-loop performance. The reachable/unreachable regions are depicted in Fig. 2, along with several illustrative closed-loop trajectories.

Then we have investigated how the reachability status of individual critical regions is affected when a bounded measurement noise is assumed per the scenario of Section 6.1. In particular, we have assumed that the noise is bounded by $\mathcal{E} = \{\xi \mid \begin{bmatrix} -0.25 \\ -0.25 \end{bmatrix} \leq \xi \leq \begin{bmatrix} 0.25 \\ 0.25 \end{bmatrix}\}$. The analysis was performed by solving the mixed-integer problems (23) with M set to $M = 20$. The analysis took 4.5 s. Out of the 147 regions of the original feedback, 31 were reachable

² All computations were performed on a 3.5 GHz Intel Core i7 CPU with Matlab R2017b, YALMIP R20180926, Gurobi 7.5.2 and MPT 3.3 beta.

Table 1
Runtimes of various reachability analysis methods for the 2D oscillator.

Method	Runtime
Section 4.1	1.9 s
Section 4.2	8.2 s
Section 4.3	30.0 s

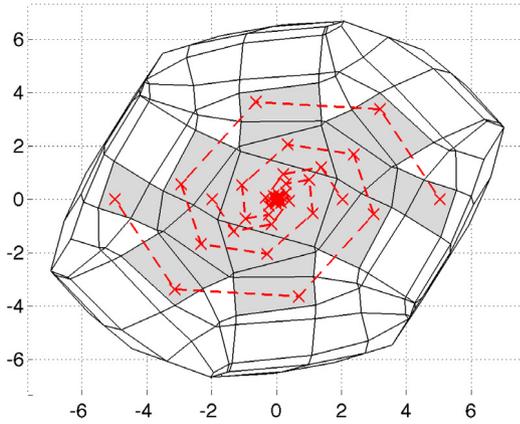


Fig. 2. The 23 reachable (gray) and 124 unreachable (white) critical regions for the pendulum example. The red dashed lines represent closed-loop profiles for four different initial conditions. Only the reachable regions are retained by the simpler optimizer (20). Notice that not all of the reachable regions are activated by the four example closed-loop trajectories. However, they could be reached should a suitable initial condition from the set \mathcal{L} be selected.

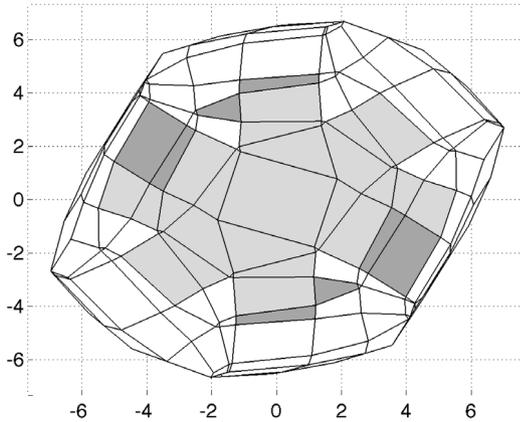


Fig. 3. 31 reachable regions under noisy measurements (light gray = regions reachable under zero noise, dark gray = regions reachable with non-zero noise).

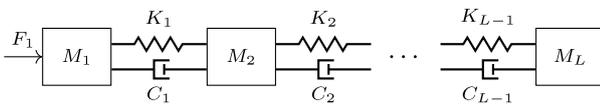


Fig. 4. The mass spring damper system.

under noisy measurements. In particular, the index set of regions that were reachable under the noisy measurements included all the regions that were reachable with zero noise, plus additional 8 regions that could be reached should an unfavorable realization of the noise occur. The newly added reachable regions are depicted in dark color in Fig. 3.

7.2. Mass spring damper system

Next we analyze a system composed of $L = 5$ masses interconnected by springs and dampers, depicted in Fig. 4. Each mass

is represented by two states (position p_i and velocity \dot{p}_i), thus the overall system has 10 states. The control input is the external force F_1 applied to the left-most mass. The linearized dynamics is given by

$$M_1 \ddot{p}_1 - C_1(\dot{p}_2 - \dot{p}_1) - K_1(p_2 - p_1) = F_1, \quad (28)$$

for the left-most mass,

$$M_i \ddot{p}_i - C_i(\dot{p}_{i+1} - \dot{p}_i) + C_{i+1}(\dot{p}_i - \dot{p}_{i-1}) - K_i(p_{i+1} - p_i) + K_{i-1}(p_i - p_{i-1}) = 0, \quad (29)$$

for the intermediate masses, and

$$M_L \ddot{p}_L - C_{L-1}(\dot{p}_L - \dot{p}_{L-1}) + K_{L-1}(p_L - p_{L-1}) = 0, \quad (30)$$

for the right-most mass. For this system, the MPC problem (2) was constructed with $N = 5$, $\mathcal{U} = \{F_1 \mid |F_1| \leq 5\}$, $\mathcal{X} = \{x \mid |p_i| \leq 10, |\dot{p}_i| \leq 20, i = 1, \dots, L\}$, $Q_x = I_{10 \times 10}$, $Q_u = 0.001$, Q_N given as the LQR penalty, and \mathcal{X}_f as the LQR set. Numerical values of constants were set to $M_i = 10$ kg, $C_i = 10$ N m⁻¹, and $K_i = 20$ kg s⁻¹, $i = 1, \dots, L$. Sampling time of 0.2 s was chosen to discretize (28)–(30).

The parametric QP (3) was subsequently solved using the enumerative solver of [20], implemented in the MPT toolbox. After 62 s we have obtained the explicit optimizer (4) defined over 1715 regions in the 10-dimensional state space. Complexity of this solution was subsequently reduced by applying the procedures of Sections 4 and 5. Specifically, we have assumed $\mathcal{L} = \{x \mid \dot{p}_i = 0, i = 1, \dots, L\}$, i.e., masses starting from an arbitrary position with zero velocity. The analysis of Section 4.1 has shown that only 87 regions are reachable from \mathcal{L} , therefore the complexity of the explicit MPC controller can be reduced by a factor of 20 without sacrificing performance. The analysis took 283 s (notice that the problem has 5 times more states and 10 times more regions as the case in Section 7.1).

Finally, we have augmented the extensive enumeration solver of [20] as proposed in Section 6.3 to include reachability-based pruning of active sets. This allowed to solve the parametric QP in just 33 s, yielding directly just the 87 reachable regions without the need to apply the a-posteriori analysis of Sections 4 and 5.

8. Conclusion

We have proposed to apply reachability analysis to remove from a given explicit MPC controllers the regions which are not reachable from a given set of initial conditions. The reachability problem was formulated for three different scenarios. The third one is the most versatile one, since it allows for a mismatch between the predicted open-loop trajectory and the actual closed-loop response, and accounts for measurement noise as well. Each reachability problem was formulated as a mixed-integer feasibility problem which entails Karush–Kuhn–Tucker optimality conditions to express reachability under the optimal MPC feedback.

The main advantage of the proposed approach is that the reachability/unreachability status of a particular critical region can be determined without having to know the full explicit solution, as the procedure of Section 4.1 is based on the implicit representation of the MPC feedback law. This opens up the possibility to combine the proposed method with a parametric programming solver to directly generate a simple solution, without the need to construct the full (complex) explicit optimizer first, as reported in Section 6.3. An another application of the presented procedure, described in Section 6.2, is to analyze safety and liveness properties of RHC controllers where the open-loop optimization problem is solved numerically at every sampling instant. Then the critical regions can be interpreted as the sets of unsafe conditions which the controller is supposed to avoid. If such a set is determined to be reachable, then the controller fails the safety check. Similarly, when checking liveness, one can employ the suggested procedure to verify that a certain set of states is reachable in a given number of steps.

Acknowledgments

The authors gratefully acknowledge the contribution of the Slovak Research and Development Agency under the project APVV 15-0007, the contribution of the Scientific Grant Agency of the Slovak Republic under grant 1/0585/19, and the Research & Development Operational Programme, Slovakia for the project University Scientific Park STU in Bratislava, ITMS 26240220084, supported by the Research & Development Operational Programme, Slovakia funded by the ERDF.

References

- [1] A. Bemporad, M. Morari, V. Dua, E.N. Pistikopoulos, The explicit linear quadratic regulator for constrained systems, *Automatica* 38 (1) (2002) 3–20.
- [2] S. Di Cairano, Diana Yanakiev, Alberto Bemporad, Ilya V Kolmanovsky, Davor Hrovat, Model predictive idle speed control: Design, analysis, and experimental evaluation, *IEEE Trans. Control Syst. Technol.* 20 (1) (2012) 84–97.
- [3] S. Di Cairano, H. Park, I. Kolmanovsky, Model Predictive Control approach for guidance of spacecraft rendezvous and proximity maneuvering, *Internat. J. Robust Nonlinear Control* 22 (12) (2012) 1398–1427.
- [4] E. Pistikopoulos, N. Diangelakis, R. Oberdieck, M. Papatthanasidou, I. Nascu, M. Sun, PAROC—An integrated framework and software platform for the optimisation and advanced model-based control of process systems, *Chem. Eng. Sci.* 136 (2015) 115–138.
- [5] D. de la Pena, D. Ramírez, E. Camacho, T. Alamo, Explicit solution of min–max MPC with additive uncertainties and quadratic criterion, *Systems Control Lett.* 55 (4) (2006) 266–274.
- [6] F. Borrelli, *Constrained Optimal Control of Linear and Hybrid Systems*, in: *Lecture Notes in Control and Information Sciences*, vol. 290, Springer-Verlag, 2003.
- [7] R. Oberdieck, N. Diangelakis, E. Pistikopoulos, Explicit model predictive control: a connected-graph approach, *Automatica* 76 (2017) 103–112.
- [8] M. Kvasnica, M. Fikar, Clipping-based complexity reduction in explicit MPC, *IEEE Trans. Automat. Control* 57 (7) (2012) 1878–1883.
- [9] F. Scibilia, S. Oлару, M. Hovd, Approximate explicit linear MPC via delaunay tessellation, in: *Control Conference (ECC), 2009 European, IEEE, 2009*, pp. 2833–2838.
- [10] N. Nguyen, M. Gulan, S. Oлару, P. Rodriguez-Ayerbe, Convex lifting: Theory and control applications, *IEEE Trans. Automat. Control* 63 (5) (2018) 1243–1258.
- [11] C. Jones, M. Morari, Polytopic approximation of explicit model predictive controllers, *IEEE Trans. Automat. Control* 55 (11) (2010) 2542–2553.
- [12] R. Shekhar, J. Maciejowski, Robust variable horizon MPC with move blocking, *Systems Control Lett.* 61 (4) (2012) 587–594.
- [13] M. Helwa, Z. Lin, M. Broucke, On the necessity of the invariance conditions for reach control on polytopes, *Systems Control Lett.* 90 (2016) 16–19.
- [14] A. Alessio, A. Bemporad, A survey on explicit model predictive control, in: *Nonlinear Model Predictive Control*, Springer, 2009, pp. 345–369.
- [15] F. Borrelli, A. Bemporad, M. Morari, *Predictive control for linear and hybrid systems*, Cambridge University Press, 2017.
- [16] S. Boyd, L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.
- [17] H.P. Williams, *Model Building in Mathematical Programming*, third ed., John Wiley & Sons, 1993.
- [18] A. Bemporad, M. Morari, Control of systems integrating logic, dynamics, and constraints, *Automatica* 35 (3) (1999) 407–427.
- [19] F.D. Torrisi, *Modeling and Reach-Set Computation for Analysis and Optimal Control of Discrete Hybrid Automata* (Dr. sc. thesis), ETH Zurich, Zürich, Switzerland, 2003.
- [20] A. Gupta, S. Bhartiya, PSV. Nataraj, PSV Nataraj A novel approach to multiparametric quadratic programming, *Automatica* 47 (9) (2011) 2112–2117.
- [21] M. Herceg, M. Kvasnica, C. Jones, M. Morari, Multi-parametric toolbox 3.0, in: *2013 European Control Conference, 2013*, pp. 502–510.
- [22] J. Löfberg, YALMIP : A Toolbox for Modeling and Optimization in MATLAB. In *Proc. of the CACSD Conference, Taipei, Taiwan, 2004*. Available from <http://users.isy.liu.se/johanl/yalmip/>.

SÚBOR 5 VYBRANÝCH NAJVÝZNAMNEJŠÍCH CITÁCIÍ

doc. Ing. Michal Kvasnica, Dr.sc.

Bratislava, 2021

Citovaná práca: [ADC] **Kvasnica, M.** [60%] – Löfberg, J. [20%] – Fikar, M. [20%]: Stabilizing polynomial approximation of explicit MPC. *Automatica*, č. 10, zv. 47, str. 2292–2297, 2011. IF: 5,541 (**24 citácií**)

Citujúca práca: **SCI(z): Bayat, F. – Johansen, T. A.: Multi-Resolution Explicit Model Predictive Control: Delta-Model Formulation and Approximation.** *IEEE Transactions on Automatic Control*, č. 11, zv. 58, str. 2979-2984, 2013.

Multi-Resolution Explicit Model Predictive Control: Delta-Model Formulation and Approximation

Publisher: IEEE

Cite This

PDF

Farhad Bayat ; Tor Arne Johansen [All Authors](#)

9
Paper
Citations

662
Full
Text Views



Abstract

Document Sections

- I. Introduction
- II. Problem Formulation
- III. Approximation of the δ -eMPC
- IV. Conclusions

Authors

Figures

References

Abstract:

This paper deals with the explicit solution and approximation of the constrained linear finite time optimal control problem for systems with fast sampling rates. To this aim, the recently developed explicit model predictive control (eMPC) is reformulated and characterized using the δ -operator to enjoy its promising advantages compared to the time-shift operator. Using the proposed δ -model eMPC formulation, a systematic method is proposed for first designing a low-complexity approximate eMPC solution and then improving its closed loop action without first determining an exact optimal solution that might be of prohibitive complexity. It is shown that the stability and feasibility of the proposed sub-optimal solution is guaranteed.

Published in: [IEEE Transactions on Automatic Control](#) (Volume: 58 , Issue: 11, Nov. 2013)

Page(s): 2979 - 2984

INSPEC Accession Number: 13858192

Date of Publication: 24 April 2013

DOI: [10.1109/TAC.2013.2259982](#)

ISSN Information:

Publisher: IEEE

optimization with no need to compute the optimal explicit MPC first. However, it can be computationally expensive in certain cases, as an iterative solution of MILP problems in each step is required. Besides, only an a posteriori stability test is provided. in [22] a polynomial approximation of the optimal controller is presented which requires computation of stability tubes to ensure stability and feasibility. Recently, canonical PWA functions are employed in [23] to obtain an approximate eMPC suitable for implementing on chips. However, only a posteriori checks for stability are provided.

22. M. Kvasnica, J. Lofberg and M. Fikar, "Stabilizing polynomial approximation of explicit MPC", *Automatica*, vol. 47, no. 10, pp. 2292-2297, 2011.

[Show Context](#) [CrossRef](#) [Google Scholar](#)

Citovaná práca: [ADC] Grieder, P. [25%] – **Kvasnica, M.** [25%] – Baotic, M. [25%] – Morari, M. [25%]: Stabilizing low complexity feedback control of constrained piecewise affine systems. *Automatica*, vol. 41, issue 10, str. 1683–1694, 2005. IF: 5,541 (**43 citácií**)

Citujúca práca: **SCI(z): Bemporad, A. – Oliveri, A. – Poggi, T. – Storage, M.: Ultra-fast stabilizing model predictive control via canonical piecewise affine approximations. IEEE Transactions on Automatic Control, č. 12, zv. 56, str. 2883-2897, 2011.**

Ultra-Fast Stabilizing Model Predictive Control via Canonical Piecewise Affine Approximations

Publisher: IEEE

Cite This

PDF

Alberto Bemporad ; Alberto Oliveri ; Tomaso Poggi ; Marco Storage [All Authors](#)

64
Paper
Citations

4
Patent
Citations

1252
Full
Text Views



Abstract

Document Sections

I. Introduction

II. Model Predictive Controller

III. PWA Simplicial functions

IV. PWAS Approximations of MPC

V. Stability Analysis

Show Full Outline ▾

Authors

Abstract:

This paper investigates the use of canonical piecewise affine (PWA) functions for approximation and fast implementation of linear MPC controllers. The control law is approximated in an optimal way over a regular simplicial partition of a given set of states of interest. The stability properties of the resulting closed-loop system are analyzed by constructing a suitable PWA Lyapunov function. The main advantage of the proposed approach to the implementation of MPC controllers is that the resulting stabilizing approximate MPC controller can be implemented on chip with sampling times in the order of tens of nanoseconds.

Published in: [IEEE Transactions on Automatic Control](#) (Volume: 56 , Issue: 12, Dec. 2011)

Page(s): 2883 - 2897

INSPEC Accession Number: 12403999

Date of Publication: 07 April 2011

DOI: 10.1109/TAC.2011.2141410

ISSN Information:

Publisher: IEEE

As pointed out in [38], the intimate relationship between closed-loop MPC systems and piecewise affine (PWA) dynamical systems allows one use stability analysis tools developed for hybrid systems for checking closed-loop stability properties of (explicit) MPC. Stability methods for PWA systems based on piecewise quadratic Lyapunov functions and linear matrix inequality relaxations [39], [40] were applied in [10] for analyzing closed-loop stability of reduced-complexity explicit MPC controllers. Piecewise linear Lyapunov functions were considered in [40]–[41] [42] for stability of PWA closed-loop systems.

41. P. Grieder, M. Kvasnica, M. Baoti and M. Morari, "Stabilizing low complexity feedback control of constrained piecewise affine systems", *Automatica*, vol. 41, no. 10, pp. 1683-1694, 2005.

[Show Context](#) [CrossRef](#) [Google Scholar](#)

Citovaná práca: [ADC] **Kvasnica, M.** [80%] – Fikar, M. [20%]: Clipping-Based Complexity Reduction in Explicit MPC. *IEEE Transactions on Automatic Control*, č. 7, zv. 57, str. 1878–1883, 2012. ISSN: 0018-9286, IF: 5,625 (**24 citácií**)

Citujúca práca: **SCI(z): Airan, Astha – Bhushan, Mani – Bhartiya, Sharad: Linear Machine Solution to Point Location Problem.** *IEEE Transactions on Automatic Control*, č. 3, zv. 62, str. 1403-1410, 2017.

Linear Machine Solution to Point Location Problem

Publisher: **IEEE**

[Cite This](#)

[PDF](#)

Astha Airan ; Mani Bhushan ; Sharad Bhartiya [All Authors](#)

8
Paper
Citations

423
Full
Text Views



Abstract

Document Sections

I. [Introduction](#)

II. [Relevant Background](#)

III. [Linear Machine for Point Location](#)

IV. [Complexity Analysis](#)

V. [Examples](#)

[Show Full Outline](#) ▾

[Authors](#)

[Figures](#)

Abstract:

Linear machine has been recently proposed as an elegant solution for solving the point location problem arising in multi-parametric programming (mp-P) based online optimization. Linear machine associates a linear discriminant function with each polytopic region in the parametric space. The solution to the point location problem is then obtained by simply evaluating these discriminant functions and finding their maximum value. In this technical note, we rigorously establish the correctness of the linear machine generation procedure and identify a necessary condition for existence of linear machine. A modified procedure, involving systematic subdivision of the parametric space, is proposed when this condition is not satisfied. Analysis of complexity and storage requirements, along with computational experiments on a large sized example, indicate that linear machine can be an efficient tool for solving the point location problem.

Published in: [IEEE Transactions on Automatic Control](#) (Volume: 62 , Issue: 3, March 2017)

Page(s): 1403 - 1410

INSPEC Accession Number: 16706783

Date of Publication: 26 May 2016

DOI: [10.1109/TAC.2016.2573201](#)

tree height have also been proposed. An alternative method, labeled descriptor function approach [7], associates a parameter dependent affine function with each CR by exploiting the properties of the value function of the mp-P. It thus avoids storing the polyhedral regions. Other approaches for solving the point location problem include use of hash tables [8] and determination of low-complexity approximations of the CRs [9]–[10] [11] [12].

10. M. Kvasnica and M. Fikar, "Clipping-based complexity reduction in explicit MPC", *IEEE Trans. Autom. Control*, vol. 57, no. 7, pp. 1878-1883, Jul. 2012.

[Show Context](#) [Google Scholar](#)

Citovaná práca: [ADC] **Kvasnica, M.** [50%] – Hledík, J. [20%] – Rauová, I. [10%] – Fikar, M. [20%]:
Complexity reduction of explicit model predictive control via separation. *Automatica*, č. 6, zv. 49, str. 1776–1781, 2013. ISSN: 0005-1098, IF: 5,541 (**17 citácií**)

Citujúca práca: **SCI(z): Yan, Zheng – Wang, Jun: Nonlinear Model Predictive Control Based on Collective Neurodynamic Optimization. IEEE Transactions on Neural Networks and Learning Systems**, no. 4, vol. 26, pp. 840-850, 2015.

Nonlinear Model Predictive Control Based on Collective Neurodynamic Optimization

Publisher: **IEEE**

[Cite This](#)

[PDF](#)

Zheng Yan ; Jun Wang [All Authors](#)

26
Paper
Citations

1007
Full
Text Views



Abstract

Document Sections

- [I. Introduction](#)
- [II. Problem Formulation](#)
- [III. Background Information](#)
- [IV. Collective Neurodynamic Optimization](#)
- [V. Simulation Results](#)

[Show Full Outline](#) ▾

[Authors](#)

[Figures](#)

[References](#)

[Citations](#)

Abstract:

In general, nonlinear model predictive control (NMPC) entails solving a sequential global optimization problem with a nonconvex cost function or constraints. This paper presents a novel collective neurodynamic optimization approach to NMPC without linearization. Utilizing a group of recurrent neural networks (RNNs), the proposed collective neurodynamic optimization approach searches for optimal solutions to global optimization problems by emulating brainstorming. Each RNN is guaranteed to converge to a candidate solution by performing constrained local search. By exchanging information and iteratively improving the starting and restarting points of each RNN using the information of local and global best known solutions in a framework of particle swarm optimization, the group of RNNs is able to reach global optimal solutions to global optimization problems. The essence of the proposed collective neurodynamic optimization approach lies in the integration of capabilities of global search and precise local search. The simulation results of many cases are discussed to substantiate the effectiveness and the characteristics of the proposed approach.

Published in: [IEEE Transactions on Neural Networks and Learning Systems](#) (Volume: 26 , Issue: 4, April 2015)

Page(s): 840 - 850

INSPEC Accession Number: 15000622

Date of Publication: 15 January 2015

DOI: 10.1109/TNNLS.2014.2387862

ISSN Information:

Publisher: IEEE

problems, which can be solved very efficiently [5]– [6] [7]. The second category is explicit MPC using multiparametric nonlinear programming, in which the optimal control signals are computed offline as an explicit function of the state and reference vectors, so that online operations reduce to a simple function evaluation [8]– [9] [10].

10. M. Kvasnica, J. Hledík, I. Rauová and M. Fikar, "Complexity reduction of explicit model predictive control via separation", *Automatica*, vol. 49, no. 6, pp. 1776-1781, 2013.

[Show Context](#) [CrossRef](#) [Google Scholar](#)

Citovaná práca: [ADC] Cseko, L. [40%] – **Kvasnica, M.** [40%] – Lantos, B. [20%]: Explicit MPC-Based RBF Neural Network Controller Design With Discrete-Time Actual Kalman Filter for Semiactive Suspension. *IEEE Transactions on Control Systems Technology*, č. 5, zv. 23, str. 1736–1753, 2015. IF: 5,312 (**22 citácií**)

Citujúca práca: **SCI(z): Dominic, Shane – Shardt, Yuri A. W. – Ding, Steven X. – Luo, Hao: An Adaptive, Advanced Control Strategy for KPI-Based Optimization of Industrial Processes. IEEE Transactions on Industrial Electronics**, č. 5, zv. 63, str. 3252-3260, 2016.

An Adaptive, Advanced Control Strategy for KPI-Based Optimization of Industrial Processes

Publisher: **IEEE**

[Cite This](#)

[PDF](#)

Shane Dominic ; Yuri A. W. Shardt ; Steven X. Ding ; Hao Luo [All Authors](#)

11
Paper
Citations

677
Full
Text Views



Abstract

Document Sections

- I. Introduction
- II. A KPI-Based Control Strategy
- III. Dynamic Setpoint Optimization
- IV. Modeling of the KPI Cost Function
- V. Online Learning of the Time-Varying Behavior

[Show Full Outline](#) ▾

Authors

Figures

References

Abstract:

The need to deal with rapid change in an environmentally and economically friendly manner has led to renewed interest in data-driven, online process optimization. Although various methods, such as economic model predictive control (EMPC), are available to achieve this goal, they require that the process model be available and relatively accurate and that there be no process changes. Recently, the focus has shifted to using economic key performance indices (KPIs) to design supervisory controllers to regulate the process. In order to accomplish this, accurate models of the highly nonlinear KPIs are needed. A solution to this problem is to develop a two-step control strategy consisting of a static, offline component and a dynamic, online component. This paper proposes the use of a linear, BILIMOD method combined with a self-partitioning algorithm for the static component and gradient-based optimization method for the dynamic component. In order to deal with process changes, the static model parameters are updated. The proposed new controller strategy is tested on the wastewater treatment process. It is shown that the proposed method can quickly and effectively achieve the desired optimal point with minimal disturbance to the overall process.

Published in: [IEEE Transactions on Industrial Electronics](#) (Volume: 63 , Issue: 5, May 2016)

Page(s): 3252 - 3260

INSPEC Accession Number: 15904528

Date of Publication: 01 December 2015

DOI: [10.1109/TIE.2015.2504557](#)

ISSN Information:

Publisher: IEEE

As with all strategies, this approach consists of three key steps: 1) process modeling; 2) optimization; and 3) monitoring [11] [12]– [13]. Failure to consider these three tasks has led to malfunctioning control systems that could put the plant at risk. Process modeling and optimization pose many challenges, including the need to consider highly nonlinear systems, limited storage capabilities, and fast process fluctuations [14].

14. L. Cseko, M. Kvasnica and B. Lantos, "Explicit MPC-based RBF neural network controller design with discrete-time actual kalman filter for semiactive suspension", *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 5, pp. 1736-1753, Sep. 2015.

[▶ Show Context](#)

[View Article](#)

[Full Text: PDF \(7523KB\)](#)

[Google Scholar](#)