Explicit spline-based continuous-time MPC: a study on design and performance

Boris Rohaľ-Ilkiv, Martin Gulan, Peter Minarčík

Institute of Automation, Measurement and Applied Informatics; Faculty of Mechanical Engineering; Slovak University of Technology in Bratislava; Námestie slobody 17; 812 31 Bratislava; Slovakia (e-mail: boris.rohal-ilkiv@stuba.sk, martin.gulan∼, peter.minarcik∼).

Abstract: This paper presents a continuous-time model predictive control scheme based on B-spline functions used for signals and model approximation. The proposed controller offers two interesting advantages. First, it formulates the control signal as a continuous polynomial spline function, the nature of which is determined by its control polygon that is subject of optimization. Second, all continuous constraints assumed over prediction horizon are consistently transformed into constraints imposed on a finite number of elements of this control polygon. Using parametric quadratic programming we further show how to obtain an explicit representation of the proposed controller, which is known for its efficient online implementation. The featured simulation study demonstrates that by a suitable choice of number and position of knots of the spline function over the prediction horizon it is possible to substantially reduce the number of critical regions of the explicit controller while preserving control performance, and to mitigate the direct correlation between number of regions and chosen length of prediction horizon.

Keywords: explicit model predictive control, parametric programming, linear systems, B-spline functions, continuous-time control, efficient implementation

1. INTRODUCTION

Continuous-time model predictive control, i.e. CMPC, has been the subject of a long-term systematic research thanks to various advantages that it offers over standard discrete-time formulations of MPC. The latter, are however, vastly preferred in industrial applications (Ferreau et al., 2016), which, as pointed out by Pannocchia et al. (2015), can be attributed mainly to their computational nature. Typical approaches to continuous-time MPC are presented e.g. in Demircioğlu and Gawthrop (1991); Ronco et al. (1999); Wang (2009); Pannocchia et al. (2010). There are several arguments that motivated their development. First, a feasible set for discrete-time formulation of MPC problem has smaller volume than an analogous one for the continuous-time formulation (Pannocchia et al., 2015). Second, higher sampling rates in general lead to well-known phenomena related to numerical sensitivity and ill-conditioned problems, or give rise to nonminimum-phase zeros, etc. (Wang, 2009). Third, if we use a continuous-time model, the MPC design becomes essentially independent on the chosen sampling period. Fourth, characteristic for the CMPC formulation is a smoother control signal (Wang, 2009).

Following the work of Demircioğlu and Gawthrop (1991), in Rohaľ-Ilkiv (1997) the author proposed an alternative approach to the CMPC problem based on spline functions, specifically B-spline functions, known for their indisputable advantages in various fields, such as geometric modeling, interpolation and smoothing of signals, optimal trajectory planning, etc. The use of splines for optimal control has been also proposed, see e.g. Sun et al. (2000); Kano et al. (2003); Tabriz and Heydari (2014); Hilhorst et al. (2016); Matinfar and Dosti (2018). It can be shown that if we assume a finite number of B-spline basis functions, we may appropriately parametrize the control signal as a continuous spline with predefined degree of continuity, which can provide some new benefits in the problem formulation. Main design parameters in this procedure are order of the spline and placement of its knots. It is the number of these knots that essentially enables to reduce dimensionality of the optimization problem, i.e. length of the optimizer. By choosing a uniform distribution of the knots we constitute regular sampling time intervals at which the control law can determine a new direction of next polynomial segment of the control signal. In this light the above idea resembles the concept of "intermittent feedback control", successfully established for practical design of CMPC by Ronco et al. (1999). Another asset of the spline-based CMPC problem formulation is that it allows to guarantee intersample fulfilment of continuous constraints. To this end we may efficiently exploit conversion of the original continuous constraints assumed in CMPC problem to constraints active only on a finite number of elements of the so-called control polygon of the B-spline function.

This paper presents an explicit approach to the synthesis of CMPC proposed recently in Rohaľ-Ilkiv et al. (2019), using modern tools for parametric programming. In view of real-time feasible implementations—even on embedded computing hardware—the main motivation to obtain its explicit representation is to further reduce the online com-

⋆ The authors gratefully acknowledge the contribution of the Slovak Research and Development Agency under the contracts APVV-18-0023, APVV-14-0399 and APVV-17-0006.

putational effort—a typical bottleneck of MPC. The paper is organized as follows. Section 2 provides an introduction and main notations to the theory of polynomial splines. Section 3 concisely presents solution to the B-spline based CMPC problem in the standard, implicit form. The explicit solution based on parametric programming is introduced in Sect. 4. Efficacy and some properties of such B-spline parametrization of control law are discussed in Sect. 5 via a simulation study. The paper concludes with final remarks and goals of future work.

2. SPLINE AND B-SPLINE FUNCTIONS

In this section we provide a brief summary of basic properties of B-spline functions, conversions between particular spline representations, as well as their essential shape properties. For a comprehensive theory of splines we refer the interested reader e.g. to de Boor (1978); Schumaker (1981); Piegl and Tiller (1995); Höllig and Hörner (2013).

2.1 Basic definitions

Let us first state some formal definitions.

Definition 1. Let $(\xi_0 =)0 < \xi_1 < \ldots < \xi_q < T_x(= \xi_{q+1})$ be the subdivision of a closed finite-time interval $[0, T_x]$ by q distinct (time) points. A function $s(t)$, defined on the interval $[0, T_x]$, is called a spline function of the order $r > 0$ (degree $r-1$) and the defect d_{ef} if the following two conditions hold:

- in each open interval $]\xi_i, \xi_{i+1}[\,, i = 0, \ldots, q$, $s(t)$ is a polynomial of degree $\leq r-1$;
- has continuous derivatives up to the order $r - d_{\mathrm{ef}} - 1$ in the open interval $]0, T_x[$.

The points $\xi_i, i = 1, \ldots, q$ are referred to as interior knots or breakpoints of the spline function. For each fixed set $\xi = (\xi_1, \ldots \xi_q)$ of the knots, the class of splines is a linear space of functions with dimension

$$z = (\alpha_1 + \alpha_2 + \ldots + \alpha_q) + r, \tag{1}$$

where α_i denotes a multiplicity (or defect d_{ef}) of the knot ξ_i. Let $P_{r,\xi,\alpha}$ denote the linear space of spline functions for $\alpha = (\alpha_1, \ldots \alpha_q)$. If all interior knots ξ_i, are simple, i.e. $\alpha_i = 1, i = 1, \ldots, q$, then $P_{r,\xi,\alpha} = \mathcal{C}^{r-2}[0, T_x]$. In order to perform computations with splines, one must first choose a suitable representation, in which any member of $P_{r,\xi,\alpha}$ can be written as a unique linear combination of properly chosen z basis functions such that Defn. 1 is satisfied. A common choice is to use B-spline functions.

Definition 2. A B-spline function $M_{i,r,\xi}(t)$ of order $r > 0$, with knots $\xi_i, \ldots, \xi_{i+r}$, can be defined using the following recurrence relation:

$$M_{i,1,\xi}(t) := \begin{cases} 1 & \text{if } \xi_i \leq t < \xi_{i+1}, \\ 0 & \text{otherwise,} \end{cases}$$

$$M_{i,k,\xi}(t) := \frac{\xi_{i+k} - t}{\xi_{i+k} - \xi_{i+1}} M_{i+1,k-1,\xi}(t) + \frac{t - \xi_i}{\xi_{i+k-1} - \xi_i} M_{i,k-1,\xi}(t), \quad \text{for } k = 2, \ldots, r,$$

where the two fraction terms are interpreted as zero whenever $\xi_{i+k} - \xi_{i+1} = 0$ and $\xi_{i+k-1} - \xi_i = 0$, respectively.

From Defn. 2 one can observe that $M_{i,r,\xi}(t), i = \ldots 0, 1 \ldots$ (i) have a local support, (ii) are positive on their supports and (iii) form a partition of unity. Every function $s(t)$ satisfying Defn. 1 then has a unique representation (the Curry-Schoenberg theorem):

$$s(t) = \sum_{i=1}^{z} c_i M_{i,r,\xi}(t) = \mathbf{m}(t)^{\mathsf{T}} \mathbf{c}, \tag{2}$$

with $\mathbf{m}(t) = [M_{1,r,\xi}(t), \ldots, M_{z,r,\xi}(t)]^{\mathsf{T}}$, $\mathbf{c} = [c_1, \ldots, c_z]^{\mathsf{T}}$, where $M_{i,r,\xi}(t)$ or shortly $M_i(t)$, $i = 1, \ldots, z$, denote base functions of the spline space $P_{r,\xi,\alpha}$, and c_i denotes the i-th B-spline coefficient of $s(t)$. They are commonly referred to as control coefficients or control points, and the collection $\{c_i\}_{i=1}^{z}$ of all control points is referred to as control polygon of the spline. In the following we will assume the spline represented as a linear combination of basis B-spline functions (2) as the approximation function. The spline design parameters thus are:

- order r and defect d_{ef} of the spline;
- number q and location ξ of its knots;
- and control coefficients (control polygon) $\{c_i\}_{i=1}^{z}$.

Note that if the order and the knots of the spline function are fixed, the approximation problem becomes a linear one since the spline function is linear in the unknown B-spline coefficients $\mathbf{c}$, as follows from (2). Depending on the type of approximation, these coefficients can usually be easily calculated as the solution of an overdetermined system of linear equations. However, number and shape of B-spline basis functions must be fixed a priori.

2.2 Conversion from B- to pp-representation

A polynomial spline can be, by definition, written as:

$$s(t) = p_i(t) := \sum_{j=1}^{r} p_{ij} (t - \xi_i)^{j-1}, \; t \in [\xi_i, \xi_{i+1}], \quad i = 0, \ldots, q, \tag{3}$$

where $p_i(t)$ are polynomial pieces or segments which represent the spline $s(t)$ on interval $[0, T_x]$. Relation (3) is called a piecewise polynomial (pp-)representation of spline $s(t)$. Clearly, the pp-representation of spline $s(t)$ is completely determined by the $(q+1)r$-dimensional vector of polynomial coefficients $\mathbf{p} = [p_{01}, \ldots, p_{0r}, \cdots, p_{q1}, \ldots, p_{qr}]^{\mathsf{T}}$. Given the B-representation (2) of the spline $s(t)$, the vector $\mathbf{p}$ of its pp-representation can be easily computed according to:

$$\mathbf{p} = \mathbf{P}\,\mathbf{c}, \tag{4}$$

where rows of matrix $\mathbf{P}$ can be obtained by differentiation of vector $\mathbf{m}^{\mathsf{T}}(t)$ in knots $\{\xi_i\}_{i=1}^{q}$:

$$\mathbf{P} = \left[\frac{1}{(j-1)!} \, \mathbf{m}^{[j-1]^{\mathsf{T}}}(t) \Big|_{t=\xi_i} \right]_{j=1,\ldots,r}^{i=0,\ldots,q}$$

with

$$\mathbf{m}^{[j-1]^{\mathsf{T}}}(t) = [M_1^{[j-1]}(t), \ldots, M_z^{[j-1]}(t)],$$

where notation $f^{[i]}(t)$ stands for the i-th derivative of $f(t)$ with respect to t, with $f^{[0]}(t) \equiv f(t)$; see de Boor (1978); Bartels et al. (1987) for details.

Given the vector $\mathbf{p}$, the conversion from pp-representation to B-representation can be performed as follows:

$$\mathbf{c} = \mathbf{P}_{\mathrm{L}}^{-1} \mathbf{p},$$

which is more difficult because of left inverse of matrix $\mathbf{P}$; but if it is a priori known that the approximated function lies in $P_{r,\xi,\alpha}$ for a certain r, ξ, α, then $\mathbf{P}_{\mathrm{L}}^{-1}$ can be determined uniquely.

2.3 Shape properties of B-spline curves

The fundamental shape properties of B-spline curves may be summarized using the following theorem. The reader is referred to Höllig and Hörner (2013) for details and proof.

Theorem 1. (Shape properties of B-spline curves). Let $s(t)$ be a B-spline curve of order r over the knot sequence ξ. Then the following properties hold:

(i) in general there is no endpoint interpolation;
(ii) for $\xi_i \le t < \xi_{i+1}$, $s(t)$ lies in the convex hull of the r control points $c_{i-r+1}, \dots, c_i$;
(iii) local control: for $t \in [\xi_i, \xi_i + 1]$ the curve is independent of c_j for $j < i - r + 1$ and $j > i$;
(iv) if $r-1$ control points coincide, then the spline curve passes through this point and is tangent to the control polygon;
(v) if $r-1$ control points are on a line, then the spline curve touches this line;
(vi) if r control points are on a line L, then $s(t) \in L$ for $\xi_i \le t < \xi_{i+1}$, i.e. an entire segment of the curve $s(t)$ coincides with L;
(vii) derivative:
$$s'(t) = \sum_{i=1}^{z-1} \bar{c}_i M_{i,r-1,\xi}(t) \tag{5}$$
with $\bar{c}_i = \dfrac{r-1}{\xi_{i+r-1} - \xi_i}(c_i - c_{i+1})$;
(viii) if $r-1$ knots $t = \xi_{i+1} = \ldots = \xi_{i+r-1}$ coincide, then $s(t) = c_i$, i.e. the spline curve passes through a control point and is tangent to the control polygon.

3. SPLINE-BASED CONTINUOUS-TIME MPC FORMULATION

For ease of presentation, let us consider a continuous-time single-input single-output (SISO) system described near a given operating point by the following linear time-invariant (LTI) model:
$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t), \tag{6a}$$
$$y(t) = \mathbf{C}\mathbf{x}(t), \tag{6b}$$
where elements of the state vector $\mathbf{x}(t) \in \mathbb{R}^{r_u}$ (r_u – order of the spline input signal) can be directly calculated from the spline derivatives of input/output signals using a technique of spline filtration elaborated in Rohaľ-Ilkiv (1997). Considering a time window given by $\tau \in [t_k, t_k + T_\mathrm{h}]$, where t_k is the current time and T_h denotes the prediction horizon, the predicted state at time $t_k + \tau$ can be obtained by solving the differential equation (6a) as follows:
$$\mathbf{x}(t_k + \tau) = e^{\mathbf{A}\tau}\mathbf{x}(t_k) + \int_0^\tau e^{\mathbf{A}(\tau-\gamma)}\mathbf{B}u(t_k + \gamma)\mathrm{d}\gamma. \tag{7}$$
Now, if we employ the B-spline functions expansion (2) to approximate the control action as:
$$u(t) \equiv s_u(t) = \mathbf{m}(t)^\mathsf{T}\mathbf{c}_u, \quad u(t) \in P_{r_u,\xi,\alpha}, \tag{8}$$
we can substitute it into the prediction equation (7), which thus becomes parametrized in $\mathbf{c}_u$:
$$\mathbf{x}(t_k + \tau) = e^{\mathbf{A}\tau}\mathbf{x}(t_k) + \boldsymbol{\Gamma}(\tau)\mathbf{c}_u \tag{9}$$
$$\text{with } \boldsymbol{\Gamma}(\tau) = \int_0^\tau e^{\mathbf{A}(\tau-\gamma)}\mathbf{B}\mathbf{m}(\gamma)^\mathsf{T}\mathrm{d}\gamma.$$
Applying the quasi-infinite horizon approach of Chen and Allgöwer (1998) to guarantee the closed-loop stability, our goal is to solve the following finite-horizon continuous-time MPC problem:
$$\min_{u(\cdot)} \int_{t_k}^{t_k+T_\mathrm{h}} \Big(\|\mathbf{x}(t_k+\tau|t_k)\|^2_{\mathbf{Q}_x} + u^2(t_k+\tau|t_k)w_u(\tau) \Big)\mathrm{d}\tau + \|\mathbf{x}(t_k+T_\mathrm{h}|t_k)\|^2_{\mathbf{Q}_\mathrm{h}}, \tag{10a}$$
$$\text{s.t. } u^{\min} \le u(t+\tau|t) \le u^{\max}, \ \ \tau \ge t, \tag{10b}$$
$$\dot{u}^{\min} \le \dot{u}(t+\tau|t) \le \dot{u}^{\max}, \ \ \tau \ge t, \tag{10c}$$
$$\mathbf{x}(t_k+T_\mathrm{h}|t_k) \in \Omega. \tag{10d}$$
In the quadratic cost (10a), let us understand all signals as deviations from their respective reference values, $\mathbf{Q}_x \succeq 0$ and $w_u(\tau) > 0$ denote the state weighting matrix and the input weighting function, respectively. In the constraints (10b)–(10c), $(u^{\min}, u^{\max})$ and $(\dot{u}^{\min}, \dot{u}^{\max})$ denote bounds imposed on control signal and its derivative, respectively. In addition, stability and recursive feasibility are ensured by assuming a terminal cost weighted with $\mathbf{Q}_\mathrm{h} \succeq 0$ in (10a), and by assuming a terminal set[1] constraint (10d).

As outlined in Rohaľ-Ilkiv et al. (2019), the CMPC problem (10) can be tackled with the B-spline function parametrization, which leads to the following formulation:
$$\min_{\mathbf{c}_u(k)} \ \mathbf{c}_u(k)^\mathsf{T}\mathbf{H}\mathbf{c}_u(k) + 2\mathbf{c}_u(k)^\mathsf{T}\mathbf{G}\mathbf{x}(t_k), \tag{11a}$$
$$\text{s.t. } \mathbf{c}_u^{\min} \le \mathbf{c}_u(k) \le \mathbf{c}_u^{\max}, \tag{11b}$$
$$\mathbf{c}_{u,\Delta}^{\min} \le \mathbf{A}_\Delta\mathbf{c}_u(k) \le \mathbf{c}_{u,\Delta}^{\max}, \tag{11c}$$
$$\mathbf{x}_\Omega^{\min} \le \mathbf{A}_\Omega\mathbf{c}_u(k) \le \mathbf{x}_\Omega^{\max}, \tag{11d}$$
$$\mathbf{m}(0)^\mathsf{T}\mathbf{c}_u(k) = \mathbf{m}(T)^\mathsf{T}\mathbf{c}_u(k-1), \tag{11e}$$
$$\dot{\mathbf{m}}(0)^\mathsf{T}\mathbf{c}_u(k) = \dot{\mathbf{m}}(T)^\mathsf{T}\mathbf{c}_u(k-1), \tag{11f}$$
$$\dot{\mathbf{m}}(T_\mathrm{h})^\mathsf{T}\mathbf{c}_u(k) = 0, \tag{11g}$$
where the reformulated cost (11a) with matrices[2] $\mathbf{H}$ and $\mathbf{G}$ has been obtained by substituting (8) and (9) into (10a). Note that the continuous-time constraints (10b) and (10c) had to be reformulated to suitable finite-dimensional forms given by (11b) and (11c), respectively. These must however guarantee the so-called intersample behavior of the spline control signal, which is achieved by a proper bounding of its control polygon $\mathbf{c}_u(k)$. Taking into account the basic shape properties of B-splines, outlined in Sect. 2.3, (11b) represent amplitude constraints (10b) of the control signal $u(t)$ on horizon $[t_k, t_k + T_\mathrm{h}]$, where $\mathbf{c}_u^{\min}$ and $\mathbf{c}_u^{\max}$ are min and max values of the spline control coefficients $\mathbf{c}_u(k)$ computed using B-spline approximation of given $u^{\min}$ and $u^{\max}$ values. Using the relation (5), (11c) approximate the constraints (10c) on derivative of the control signal $\dot{u}(t)$, where $\mathbf{c}_{u,\Delta}^{\min}$ and $\mathbf{c}_{u,\Delta}^{\max}$ are min and max values of the differences of the spline control coefficients, which can, together with entries of matrix $\mathbf{A}_\Delta$, be computed based on location of spline knots. Next, (11d) represent terminal constraints transformed from (10d) using prediction model (9); where

[1] This set has to be invariant under a local state feedback, virtually acting for $\tau \in [t_k + T_\mathrm{h}, \infty[$ and feasible with (10b)–(10c). One may use e.g. a low-complexity invariant set computation procedure proposed by Rohaľ-Ilkiv (2004).

[2] The matrices $\mathbf{H}$ and $\mathbf{G}$ can be calculated as:
$$\mathbf{H} := \int_{t_k}^{t_k+T_\mathrm{h}} \boldsymbol{\Gamma}(\tau)^\mathsf{T}\mathbf{Q}_x\boldsymbol{\Gamma}(\tau)\mathrm{d}\tau + \int_{t_k}^{t_k+T_\mathrm{h}} \mathbf{m}(\tau)w_u(\tau)\mathbf{m}(\tau)^\mathsf{T}\mathrm{d}\tau + \boldsymbol{\Gamma}(T_\mathrm{h})^\mathsf{T}\mathbf{Q}_h\boldsymbol{\Gamma}(T_\mathrm{h}),$$
$$\mathbf{G} := \int_{t_k}^{t_k+T_\mathrm{h}} \boldsymbol{\Gamma}(\tau)^\mathsf{T}\mathbf{C}^\mathsf{T}w_y(\tau)\mathbf{C}e^{\mathbf{A}\tau}\mathrm{d}\tau + \boldsymbol{\Gamma}(T_\mathrm{h})^\mathsf{T}\mathbf{Q}_\mathrm{h}e^{\mathbf{A}T_\mathrm{h}}.$$
using numerical integration.

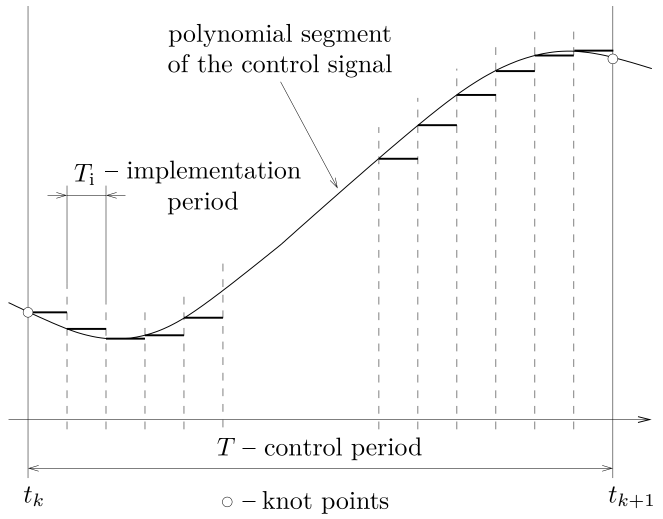


Fig. 1. Generation of the spline control signal.

$\mathbf{x}_\Omega^{\min}$, $\mathbf{x}_\Omega^{\max}$ denote vectors bounding the terminal set Ω. In order to keep the spline function $s_u(t)$ in the space $P_{r_u,\xi,\alpha}$ for given (r_u, ξ, α), the equality constraints (11e)–(11f) are employed to enforce continuity between the implemented and the projected spline control signal at the beginning of the prediction horizon $[t_k, t_k + T_\mathrm{h}]$. Finally, the constraint (11g) is added to improve stability of control by requiring a zero derivative of the projected spline control signal at the end of the prediction horizon $[t_k, t_k + T_\mathrm{h}]$. Naturally, problems (10) and (11) are not exactly equivalent, because the original constraints (10b)–(10d) are replaced with the constraints (11b)–(11g) imposed on the spline control coefficients, following the argumentation of basic spline shape properties listed in Sect. 2.3. These properties, however, imply only sufficient conditions in general, hence bringing some degree of conservatism to the problem (11). For more information on this open research topic and ideas how to reduce this conservatism we refer the interested reader e.g. to Carnicer and Delgado (2010); Loock et al. (2015).

From the optimization perspective, the spline-based CMPC problem (11) is a quadratic program (QP), which can be rewritten in the following simplified form:

$$\min_{\mathbf{c}_u(k)} \ \frac{1}{2}\mathbf{c}_u(k)^\mathsf{T}\mathbf{H}\mathbf{c}_u(k) + \mathbf{c}_u(k)^\mathsf{T}\mathbf{G}\mathbf{x}(t_k), \tag{12a}$$

$$\text{s.t.} \ \mathbf{A}_\mathrm{ineq}\mathbf{c}_u(k) \le \mathbf{b}_\mathrm{ineq}, \tag{12b}$$

$$\mathbf{A}_\mathrm{eq}\mathbf{c}_u(k) = \mathbf{b}_\mathrm{eq}. \tag{12c}$$

Solving QP (12) implicitly for a current state $\mathbf{x}(t_k)$ yields a vector of optimal control coefficients $\mathbf{c}^\star = [c_1^\star, \ldots, c_z^\star]^\mathsf{T} \in \mathbb{R}^z$. According to (4), $\mathbf{c}^\star(k)$ is subsequently converted to the pp-representation of the optimal spline control signal $s_u^\star(\tau)$, $\tau \in [t_k, t_k + T_\mathrm{h}]$ given by its polynomial coefficients $\mathbf{p}^\star(k)$. From problem (11) it is clear that we are looking at the control signal $u(t)$ from the viewpoint of a selected distance T between knot points of the spline function $s_u(t)$. In real-time control, the distance T corresponds to a control period, in which the first polynomial segment of the optimal spline input signal, i.e. $s_u^\star(\tau)$, $\tau \in [t_k, t_k + T]$, is applied for control. This is usually performed with a much shorter "implementation" period $T_\mathrm{i} \ll T$; see Fig. 1. This means we calculate the values of control signal $s_u(t_k + \tau)$ for $\tau = iT_\mathrm{i}$, $i = 1, \ldots, n$, $T = nT_\mathrm{i}$, and implement them for control using a common zero-order hold.

We remark that during the prediction horizon $[t_k, t_k + T_\mathrm{h}]$ the distance T between knots of the projected spline input signal $s_u(\tau)$, $\tau \in [t_k, t_k + T_\mathrm{h}]$ can be selected as βT, with $\beta = 1, 2, \ldots$, introducing a property comparable with the well-known move blocking techniques used in MPC. We also remark that the entire design procedure can be easily extended to multi-input multi-output (MIMO) and time-varying systems.

4. EXPLICIT SOLUTION OF SPLINE-BASED CMPC

Following the practical implementation aspects discussed in the introductory section, let us now present an explicit solution of the B-spline based CMPC problem (11), in contrast to the considerably more expensive computation in the implicit fashion given by (12). As shown in Bemporad et al. (2002), this can be achieved first by recasting a QP-based control problem such as (12) as a parametric quadratic program (pQP); which in our case takes the form:

$$\min_{\mathbf{c}_u} \ \frac{1}{2}\mathbf{c}_u^\mathsf{T}\mathbf{H}\mathbf{c}_u + \mathbf{c}_u^\mathsf{T}\bar{\mathbf{G}}\boldsymbol{\theta}, \tag{13a}$$

$$\text{s.t.} \ \bar{\mathbf{A}}_\mathrm{ineq}\mathbf{c}_u \le \bar{\mathbf{b}}_\mathrm{ineq} + \bar{\mathbf{B}}_\mathrm{ineq}\boldsymbol{\theta}, \tag{13b}$$

$$\bar{\mathbf{A}}_\mathrm{eq}\mathbf{c}_u = \bar{\mathbf{b}}_\mathrm{eq} + \bar{\mathbf{B}}_\mathrm{eq}\boldsymbol{\theta}, \tag{13c}$$

$$\boldsymbol{\theta} \in \Theta, \tag{13d}$$

with objective function and constraints parametrized (the problem data marked by $\bar{\cdot}$) with a vector of parameters

$$\boldsymbol{\theta} = \left[\mathbf{x}^\mathsf{T}, \mathbf{c}_u^{\mathrm{prev}\,\mathsf{T}}, u^\mathrm{ref}\right]^\mathsf{T},$$

in which presence of control coefficients $\mathbf{c}_u^\mathrm{prev}$, obtained in previous time instant, stems from constraints (11e)—(11f) which ensure continuity (in amplitude and first derivative) between the previous and projected spline control signal. Since we want the controller to keep its reference tracking capability, the parameter vector $\boldsymbol{\theta}$ also contains the input reference calculated as $u^\mathrm{ref} = -\left(\mathbf{C}\mathbf{A}^{-1}\mathbf{B}\right)^{-1} y^\mathrm{ref}$.

Problem (13) can be solved using the technique of (multi-)parametric programming, which enables us to precompute the solution $\mathbf{c}_u^\star(\boldsymbol{\theta})$ for all feasible values of the parameter $\boldsymbol{\theta}$, explicitly (offline), as a continuous and piecewise affine (PWA) function in the following form:

$$\mathbf{c}_u^\star(\boldsymbol{\theta}) := \begin{cases} \mathbf{F}_1\boldsymbol{\theta} + \mathbf{g}_1 & \text{if } \boldsymbol{\theta} \in \mathcal{R}_1, \\ \vdots & \vdots \\ \mathbf{F}_{n_\mathrm{reg}}\boldsymbol{\theta} + \mathbf{g}_{n_\mathrm{reg}} & \text{if } \boldsymbol{\theta} \in \mathcal{R}_{n_\mathrm{reg}}, \end{cases} \tag{14}$$

defined over n_reg polyhedral regions $\mathcal{R}_i$ in the parameter space, given as convex intersections of finitely many closed halfspaces, i.e.

$$\mathcal{R}_i = \{\boldsymbol{\theta} \mid \mathbf{R}_i\boldsymbol{\theta} \le \mathbf{r}_i\}.$$

The collection of the so-called critical regions $\{\mathcal{R}_i\}_{i=1}^{n_\mathrm{reg}}$ is referred to as a partition of the set of feasible parameters.

Online effort to implement the explicit spline-based CMPC controller hence reduces to a simple function evaluation, as per (14), where most of the time is spent on point location, that is determining which region $\mathcal{R}_i$ the current parameter $\boldsymbol{\theta}(k)$ resides in, by checking its defining inequalities. The result is the current optimal value of the control polygon $\mathbf{c}_u^\star(k) \equiv \mathbf{c}_u^\star(\boldsymbol{\theta}(k))$ of spline input signal. Analogously to the implicit (online) solution approach described in Sect. 4, the receding horizon implementation strategy—characteristic for MPC—is then realized by determining the parameters of the spline's pp-representation via (4) and applying only its first polynomial segment in a way illustrated in Fig. 1. The procedure is repeated at each sampling instant, thereby closing the feedback loop. The control law (14) clearly results in the same closed-loop behaviour as the implicit solution to QP (12).

The potential of the explicit representation of spline-based CMPC controller may be fully exploited namely in the context of real-time implementations on low-cost embedded hardware and in applications with higher sampling rates.

5. SIMULATION STUDY

In this section we demonstrate the basic features of design and performance of the proposed explicit spline-parameterized CMPC controller via a selected numerical example, in which we aim to control a continuous-time, linear non-minimum phase SISO stochastic system given in the Laplace domain as:

$$\begin{aligned} y(s) = \frac{(-15s+20)}{(s^3+4.5s^2+9s+10)}u(s)+ \\ + \frac{(0.2+0.07s)}{(s^3+4.5s^2+9s+10)}e(s), \end{aligned} \tag{15}$$

where $e(s)$ is a uniformly distributed noise with standard deviation $\sigma = 0.15$. This system is not very challenging in terms of control but can quite well illustrate the basic concept. Now, by applying the spline-based online identification procedure introduced in Rohal'-Ilkiv (1997), it can be transformed to a phase variable state-space form, such as in (7), with matrices:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & -0.457 \\ 1 & 0 & -0.799 \\ 0 & 1 & -1.138 \end{bmatrix}, \ \mathbf{B} = \begin{bmatrix} 0.912 \\ 0.596 \\ 0.333 \end{bmatrix}, \text{ and } \mathbf{C} = [0 \ 0 \ 1],$$

which was then used to formulate the spline-based CMPC problem (11), assuming values of parameters listed in Tab. 1a. The choice of spline design parameters, having an important impact on the shape of the spline function (order, defect, number and position of knots), was guided by common rules for signal approximation. Higher orders and a large number of knots imply high flexibility but may also result in overfitting. Conversely, lower orders and a small number of knots may result in over-smoothing behaviour.

By using the MATLAB based tools of YALMIP (Löfberg, 2004) and Multi-Parametric Toolbox (MPT, Herceg et al. (2013)), the spline-based CMPC problem (11) was recast and solved [3] as a parametric QP (13). For $T_\mathrm{h} = 6\,\mathrm{s}$, $z = 5$ and other design parameters (see the first row in Tab. 1b), the explicit solution was obtained in $\sim 2\,\mathrm{s}$ and consists of a PWA optimizer defined over a partition with 50 critical regions in space $\mathbb{R}^9$.

The controller was tested in 5 simulation setups, with the same objective of tracking predefined step changes of the output reference $y^{\mathrm{ref}}(t)$, to demonstrate the impact of horizon length and number of interior knots within a control period on the controller's complexity, in terms of number of optimized variables and number of critical regions. The obtained results are presented in Fig. 2 and Tab. 1b. The control performance is assessed also in terms of the mean squared control error. As indicated in Tab. 1b for a wide range of lengths of the prediction horizon, $T_\mathrm{h} = 6T$–$24T$, there is not a usually obvious relation between the choice of horizon length and an increase in number of optimized variables z and critical regions n_reg. An unrealistic scenario was also shown—choice of a very short prediction horizon, $T_\mathrm{h} = 6T$, when the controller apparently failed to track the reference and a large set-point change even lead to a temporary violation of input amplitude constraint (see Fig. 2a at $\sim 3500 T_\mathrm{i} = 700\,\mathrm{s}$). In terms of complexity reduction, an interesting observation is a very low number of controller's regions which have to be stored in the hardware memory. As shown in the simulation with fixed horizon (in our case $T_\mathrm{h} = 24T$), with an increasing number q of interior knots located within the control period T we can observe a proportional increase in number of optimized variables as well as in number of regions. In parallel, one may also observe an increased activity of the control input, in particular at a set-point change; see Fig. 2b, which also suggests that a larger number of interior knots does not necessarily lead to a smaller steady-state error—the smallest one is achieved already for two knots ($z = 5$). Table 1b moreover compares number of constraints and average execution times of the implicit [4] and the explicit implementation of the proposed controller, where the latter is expectedly faster, although it performs point location in higher dimensions of the parameter space.

Note also, that in all simulation runs the input weighting function $w_u(t)$ was set as zero, which sufficed to ensure a stabilizing control. This observation is a by-product of the applied conditions on continuity and zero derivative of the spline at the end of prediction horizon, (11e) and (11g).

6. CONCLUSION

This paper presented design, implementation and simulation analysis of an effective approach to the synthesis of explicit spline-based continuous-time model predictive control. It formulates the control input signal as a spline function which satisfies user-defined constraints—even between samples. The explicit controller design is built upon the elements of the spline function's control polygon, which—in comparison with the standard discrete-time formulation—mitigates the immediate relationship between the problem complexity (e.g. in terms of number of optimized variables or number of controller's critical regions) and the necessary length of prediction horizon. Longer prediction horizons, in turn, are then beneficial for contractivity and stability of the proposed explicit control law.

One might have also observed that the presented approach is similar to the flatness-based approach since it essentially rewrites the continuous-time optimal control problem as a tractable optimization problem using either the knowledge of the applied model transition or the flatness properties of the nominal controllable LTI system.

Future work of authors aims at robustness of the proposed control scheme and its potential extension to the nonlinear MPC framework.

[3] All computations were performed on a 2.8 GHz i5 core CPU with 8 GB of RAM. The pQP problem was solved using the enumeration-based approach of Herceg et al. (2015) available in MPT3.

[4] Obtained by solving (12) with MATLAB's `quadprog` solver.

Table 1. (a) Values of parameters used in simulation; (b) overview of controller design and implementation properties.

(a)

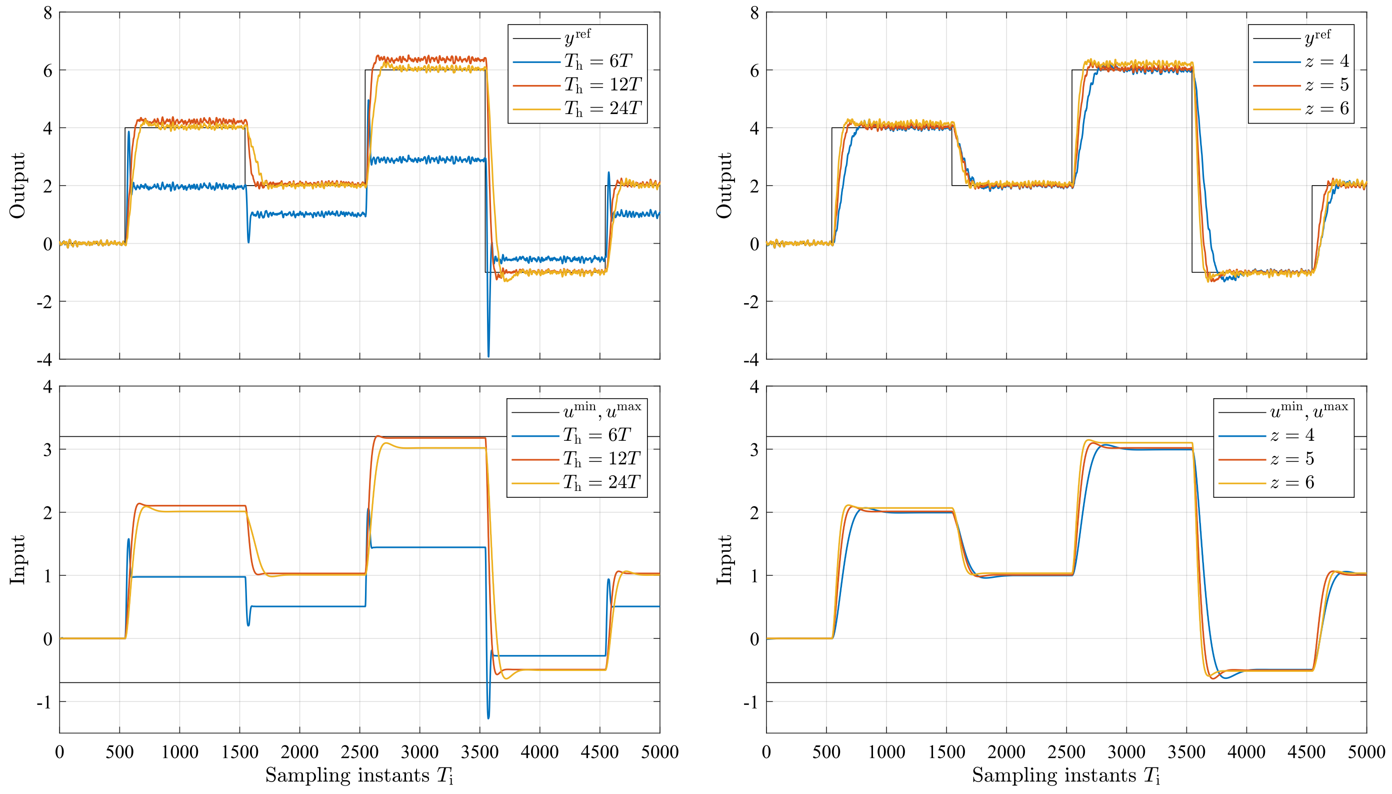
Parameter	Value
implementation period[†]	$T_\mathrm{i} = 0.2\,\mathrm{s}$
control period	$T = 5T_\mathrm{i} = 1\,\mathrm{s}$
prediction horizon	$T_\mathrm{h} = \{6T, 12T, 24T\}$
spline order	$r_u = 3$
spline defect	$d_\mathrm{ef} = 1$
weighting functions	$w_u(\tau) = 0,\ w_y(\tau) = 1$
input amplitude bounds	$u^\mathrm{min} = -0.7$
	$u^\mathrm{max} = 3.2$
input derivative bounds	$\dot{u}^\mathrm{min} = -3.0$
	$\dot{u}^\mathrm{max} = 3.0$

[†] corresponds to sampling time

(b)

				Solution						
		Location of interior		implicit		explicit				
T_h	q	knots $\{\xi\}_{i=1}^{q}$	z	n_con	t_ex	n_con	n_θ	n_reg	t_ex	msce[†]
$6T$	2	$\{2T, 4T\}$	5	24/3	1.85	42/3	9	50	0.17	1.641
$12T$	2	$\{4T, 8T\}$	5	24/3	1.83	42/3	9	57	0.20	0.578
$24T$	1	$\{12T\}$	4	20/3	1.60	36/3	8	7	0.05	1.382
$24T$	2	$\{8T, 16T\}$	5	24/3	1.66	42/3	9	25	0.11	0.917
$24T$	3	$\{6T, 12T, 18T\}$	6	28/3	1.63	48/3	10	69	0.22	0.788

q – number of interior knots within T_h, z – number of optimized variables,
n_con – number of inequality/equality constraints, n_θ – number of parameters,
n_reg – number of critical regions, t_ex – average controller execution time [ms]
[†] mean squared control error



(a) fixed z, various T_h (b) fixed T_h, various z

Fig. 2. Simulation results obtained with the explicit spline-based CMPC used to control the system described by (15).

REFERENCES

Bartels, R.H., Beatty, J.C., and Barsky, B.A. (1987). *An introduction to splines for use in computer graphic and geometric modeling*. Morgan Kaufman Publishers, Inc., San Francisco, USA.

Bemporad, A., Morari, M., Dua, V., and Pistikopoulos, E.N. (2002). The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1), 3–20.

Carnicer, J. and Delgado, J. (2010). Mean distance from a curve to its control polygon. In *Mathematical Methods for Curves and Surfaces*, 81–92. Springer Berlin Heidelberg, Germany.

Chen, H. and Allgöwer, F. (1998). A quasi-infinite horizon nonlinear predictive control scheme with guaranteed stability. *Automatica*, 34(10), 1205–1217.

de Boor, C. (1978). *A Practical Guide to Splines*. Springer-Verlag, New York, USA.

Demircioğlu, H. and Gawthrop, P.J. (1991). Continuous-time generalised predictive control. *Automatica*, 27(1), 55–74.

Ferreau, H.J., Almér, S., Peyrl, H., Jerez, J.L., and Domahidi, A. (2016). Survey of industrial applications of embedded model predictive control. In *2016 European Control Conference*, 601–601. Aalborg, Denmark.

Herceg, M., Kvasnica, M., Jones, C., and Morari, M. (2013). Multi-Parametric Toolbox 3.0. In *2013 European Control Conference*, 502–510. Zürich, Switzerland.

Herceg, M., Jones, C.N., Kvasnica, M., and Morari, M. (2015). Enumeration-based approach to solving parametric linear complementarity problems. *Automatica*, 62, 243–248.

Hilhorst, G., Lambrechts, E., and Pipeleers, G. (2016). Control of linear parameter-varying systems using B-splines. In *55th Conference on Decision and Control*, 3246–3251. Las Vegas, USA.

Höllig, K. and Hörner, J. (2013). *Approximation and Modeling with B-Splines*. SIAM, Philadelphia, USA.

Kano, H., Egerstedt, M., Nakata, H., and Martin, C.F. (2003). B-splines and control theory. *Applied Mathematics and Computation*, 145(2), 263–288.

Löfberg, J. (2004). YALMIP: a toolbox for modeling and optimization in MATLAB. In *43th IEEE International Symposium on Computer Aided Control System Design*, 284–289. Taipei, Taiwan.

Loock, W., Pipeleers, G., and Swevers, J. (2015). B-spline parameterized optimal motion trajectories for robotic systems with guaranteed constraint satisfaction. *Mechanical Sciences*, 6(2), 163–171.

Matinfar, M. and Dosti, M. (2018). Solving linear optimal control problems using cubic B-spline quasi-interpolation. *Matematika*, 34(2), 313–324.

Pannocchia, G., Rawlings, J.B., Mayne, D.Q., and Mancuso, G.M. (2015). Whither discrete time model predictive control? *IEEE Transactions on Automatic Control*, 60(1), 246–252.

Pannocchia, G., Rawlings, J.B., Mayne, D.Q., and Marquardt, W. (2010).On computing solutions to the continuous time constrained linear quadratic regulator. *IEEE Transactions on Automatic Control*, 55(9), 2192–2198.

Piegl, L. and Tiller, W. (1995). *The NURBS Book*. Springer Berlin Heidelberg, Germany.

Rohaľ-Ilkiv, B. (1997). One approach to continuous-time predictive control. *IFAC Proceedings Volumes*, 30(21), 107–114. 2nd IFAC Workshop on New Trends in Design of Control Systems, Smolenice, Slovak Republic.

Rohaľ-Ilkiv, B. (2004). A note on calculation of polytopic-invariant feasible sets for linear continuous-time systems. *Annual Reviews in Control*, 28(1), 59–64.

Rohaľ-Ilkiv, B., Gulan, M., and Minarčík, P. (2019). Implementation of continuous-time MPC using B-spline functions. In *22nd International Conference on Process Control*, 222–227. Šrbské Pleso, Slovakia.

Ronco, E., Arsan, T., and Gawthrop, P.J. (1999). Open-loop intermittent feedback control: practical continuous-time GPC. *IEE Proceedings - Control Theory and Applications*, 146(5), 426–434.

Schumaker, L.L. (1981). *Spline functions: basic theory*. John Wiley & Sons, New York, USA.

Sun, S., Egerstedt, M., and Martin, C. (2000). Control theoretic smoothing splines. *IEEE Transactions on Automatic Control*, 45(12), 2271–2279.

Tabriz, Y. and Heydari, A. (2014). Generalized B-spline functions method for solving optimal control problems. *Computational Methods for Differential Equations*, 2(4), 243–255.

Wang, L. (2009). *Model Predictive Control System Design and Implementation Using MATLAB*. Springer-Verlag London, UK.