# Optimal Splines for Rigid Motion Systems: A Convex Programming Framework

**B. Demeulenaere[+], G. Pipeleers[*], J. De Caigny[*], J. Swevers[*], J. De Schutter[*], L. Vandenberghe[†]**

[+]Atlas Copco Airpower NV - Airtec Division
Boomsesteenweg 957, B-2610 Wilrijk - Belgium

[*]Katholieke Universiteit Leuven
Mechanical Engineering Department
Celestijnenlaan 300B, B-3001 Leuven, Belgium

[†]University of California Los Angeles
Electrical Engineering Department
68-119 Engineering IV, Los Angeles, CA 90095, USA

## ABSTRACT

This paper develops a general framework to synthesize optimal polynomial splines for rigid motion systems driven by cams or servomotors. This framework is based on numerical optimization and has three main characteristics: (i) spline knot locations are optimized through an indirect approach based on providing a large number of fixed, uniformly distributed candidate knots; (ii) in order to efficiently solve the corresponding large-scale optimization problem to global optimality, only design objectives and constraints are allowed that result in convex programs and (iii) one-norm regularization is used as an effective tool for selecting the better (that is, having fewer active knots) solution if many equally optimal solutions exist. The framework is developed and validated based on a double dwell benchmark problem for which an analytical solution exists.

**List of Figures**

## List of Tables

## 1 Introduction

A core design problem in motion systems driven by cams and servomotors is the synthesis of an optimal motion trajectory, for which splines constitute an often chosen parametrization. Spline curves consist of several segments, blended together under strict continuity requirements at various interconnection points, the *spline knots*. Depending on the segment type, distinction is made between polynomial, rational, trigonometric, . . . splines. Polynomial splines constitute the main focus of this paper and are rigorously defined in Sec. 1.1.

*Polynomial splines* entered the area of cam design in the early 1980s (see the references provided in [1]) and were quickly recognized as being more flexible than combining and modifying standard analytical curves, especially for complex motion tasks with a large array of constraints [2]. Splines have currently replaced polynomials as the mathematical descriptors of valve motion about anywhere in automotive industry [3].

Splines constitute an attractive parametrization for they allow the designer (i) to locally control the curve and (ii) directly impose its smoothness. *Local control* implies the possibility to concentrate the effects of a change within a specific region of the curve [4]. *Smoothness* is quantified here through the number $m$ of continuous derivatives: a curve belonging to $\mathcal{C}^m_{[0,T]}$ has derivatives that are continuous on the interval $[0,T]$ up to the $m$th-order derivative. It is a common rule of thumb that motion trajectories should be at least $\mathcal{C}^2$ (acceleration must be continuous). This rule, dating back to the pioneering work by Neklutin [5], is termed the *fundamental law of cam design* by Norton [6].

For polynomial splines, smoothness is directly controlled through the *spline degree k*. Besides $k$, the designer also needs to carefully select the locations of the spline knots and the corresponding *spline coefficients*[1], which is a difficult problem for complex motion tasks. It is, for instance, in general not clear how many knots are required, nor what their optimal location is [4]. Several interactive software tools are available, such as *DynaCam* [6], which give the designer direct feedback regarding the chosen knot locations in terms of the spline and its derivatives. The success of such tools, however, depends to a large extent on the skills and experience of the designer.

A more automated approach results if the spline knots are determined based on numerical optimization [4, 7–13] for it provides a mathematical framework in which many design objectives and constraints can be included. Numerical optimization looks very attractive, since it seems to suffice to push a button to automatically obtain a solution that optimizes the design objective while complying with all design constraints. Unfortunately, despite the advances in algorithmic development and computational power, this is generally not the case. Optimization algorithms often need good initialization, may require a lot of tuning, can be very sensitive to problem scaling and may yield rather poor suboptimal solutions if the algorithm gets stuck in a local optimum, if converging at all.

Problems with local optima are generally attributed to the nonlinearity of the problem. It is, indeed, true that if the objective and constraints are *linear* functions of the design variable $x$, a linear optimization problem (or *linear program*, LP) results, of which the global optimum is guaranteed to be found efficiently and reliably using dedicated algorithms. Contrary to what is generally believed, however, nonlinearity does not mean that all is lost: in fact, the great watershed in optimization is not between linearity and nonlinearity, but between convexity and nonconvexity [14]. That is, a particular class of nonlinear

[1] Spline knots, coefficients and degree are rigorously defined in Sec. 1.1.

optimization problems exist, called *convex programs*, which share with LPs the guarantee to efficiently find the global optimum (Sec. 2.1). Formulating a nonlinear optimization problem as a convex program therefore has great advantages.

The aim of this paper is to develop a framework in which polynomial spline optimization for *rigid* motion systems is formulated as such a convex program. To this end, spline knots are optimized *indirectly* by providing a *large* number of *fixed*, uniformly distributed candidate knots and only optimizing the corresponding spline coefficients (Sec. 2). The second essential characteristic of the framework, its formulation as a convex program, is developed in two stages: Sec. 3 first develops a linear programming formulation, while Sec. 5 subsequently adds several nonlinear objectives and constraints that give rise to a nonlinear, yet convex, program (further convex extensions are discussed in the companion paper [15]). Sec. 4 introduces a third essential characteristic, that is, one-norm regularization as an effective tool for selecting the better (that is, having fewer active knots) solution if many equally optimal solutions exist.

Before proceeding to the framework development, polynomial splines are formally defined and the main contributions of this work are presented.

## 1.1 Polynomial Spline Basics

A motion trajectory $s(t)$, defined on a finite time interval $[0,T]$, is a polynomial *spline* of *degree* $k \geq 0$, having as *knots* the strictly[2] increasing sequence $t_i$, $i = 0,1,\ldots,g+1$ if [16]:

$s(t)$ is a polynomial of degree $\leq k$ on each knot interval $[t_i, t_{i+1}]$:

$$s_{[t_i,t_{i+1}]} \in \mathcal{P}_k,\ i = 0,1,\ldots,g, \tag{1}$$

with $\mathcal{P}_k$ the set of polynomials of degree $\leq k$.

$s(t)$ and its derivatives up to order $k-1$ are continuous on $[0,T]$:

$$s(t) \in \mathcal{C}^{k-1}_{[0,T]}. \tag{2}$$

It is assumed that $t_0 = 0$ and $t_{g+1} = T$; the knots $t_i$, $i = 1,\ldots,g$, are the $g$ *internal* knots. Internal knot $t_i$ is called *active* if the $k$th-order derivative of $s(t)$ is discontinuous at $t_i$, while at an *inactive* internal knot, the $k$th-order derivative of $s(t)$ is continuous. The definition (1)–(2) implies that the fundamental law of cam design dictates the use of at least *cubic* ($k = 3$) splines. Figure 1 shows a cubic spline $\theta(\tau)$ and its first three derivatives as a function of (dimensionless, see further) time $\tau$, where $0 \leq \tau \leq 2\pi$. The spline consists of 3rd-degree polynomial segments and, as allowed by (2), features discontinuous jumps of the 3rd-order derivative at all spline knots, except for the knots at $\tau = 1.0$ and $\tau = 5.0$. The former knots are active knots, as opposed to the redundant, inactive knots $\tau = 1.0$ and $\tau = 5.0$.

## 1.2 Contributions

The presented work is a framework that combines old and new ideas. Our particular contributions are the following.

[2] If the knot sequence is increasing, but not strictly increasing (that is, coincident knots are present), the continuity condition (2) has to be relaxed.

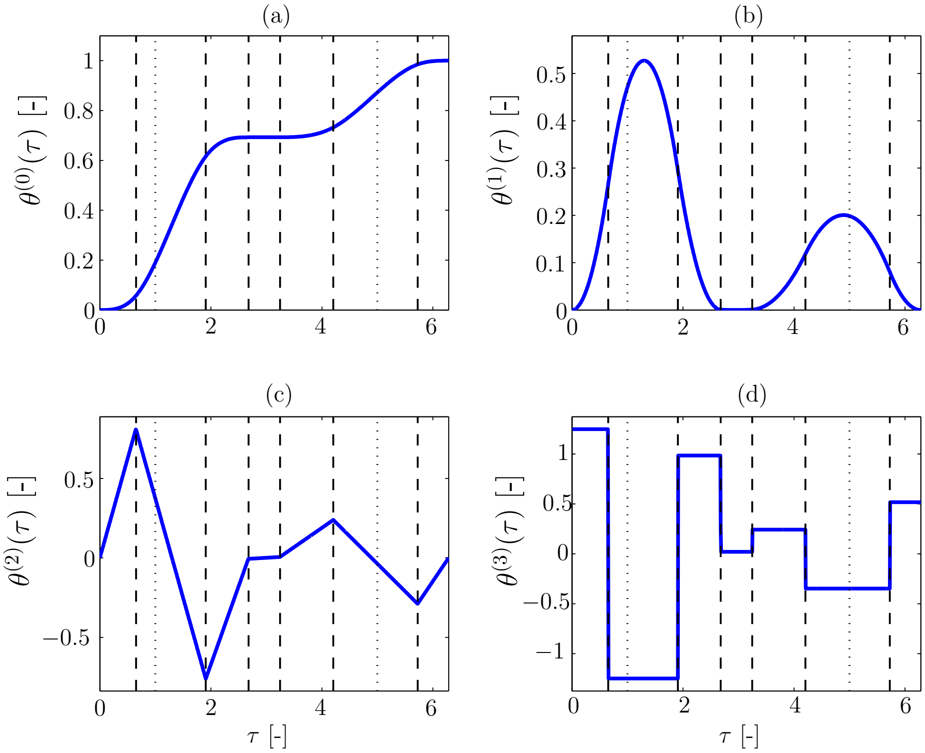

Fig. 1. An example of a cubic spline with knot sequence $\{0.0, 0.64, 0.65, 1.00, 1.91, 2.68, 3.25, 4.21, 5.00, 5.72, 6.28\}$. Active knots are indicated with a dashed vertical line, while dotted vertical lines denote the (redundant) inactive knots $\tau = 1.00$ and $\tau = 5.00$.

(i) Simplifying spline optimization by providing a set of fixed candidate knots and only optimizing spline coefficients is not a new idea [4, 7–11, 13]. Contrary to our approach, however, the number of knots was previously always kept limited (generally 5–20, as opposed to a few hundreds or thousands here), even if the effect of increasing the number of knots was investigated [7, 9, 11].

Convergence, in terms of the number of knots, is carefully investigated here for the analytical benchmark problem introduced in Sec. 3.1. It is shown that increasing the number of knots far beyond a few tens may be rewarding.

(ii) Increasing the number of knots requires careful formulation of the corresponding optimization problem to keep it numerically tractable and hence goes way beyond merely increasing some number in an algorithm. To this end, only design objectives and constraints are allowed that give rise to convex programs. The important role of convexity as the key criterion making an optimization problem 'easy' or 'difficult' seems to be little known in the area of spline optimization for motion systems. While Sections 3 and 4 focus on linear programs, a subclass of convex programs, Section 5 makes the extension to convex programs by reviewing the known optimization methods [4, 7–13] thereby showing that some of these methods can be extended or simplified based on insights from convex optimization theory.

(iii) An undesirable consequence of the use of a large number of knots is that in more complicated case studies counter-intuitive splines with an unnecessarily high number of active knots result (Sec. 4). To the best of our knowledge, this phenomenon, seemingly typical for purely kinematic optimization studies, is reported here for the first time. It is interpreted as the occurrence of a set of solutions all giving rise to an equivalent value of the objective function. The use of one-norm regularization to efficiently deal with this problem is, although known in areas such as function approximation and signal processing, new in the area of splines for motion curve design.

## 2 Polynomial Spline Optimization

This section addresses general topics in polynomial spline optimization relevant for this paper. Consider the problem of finding a polynomial spline $s(t)$ of degree $k$ that solves the general optimization problem:

$$\begin{aligned} \text{minimize}_{s(t)} \quad & f_0(s(t)) && \text{(3a)} \\ \text{subject to} \quad & f_v(s(t)) \leq 0, \; v = 1, \dots, n_f && \text{(3b)} \\ & h_v(s(t)) = 0, \; v = 1, \dots, n_g && \text{(3c)} \end{aligned}$$

where $f_0$ denotes the design objective to be minimized, while $f_v$ and $h_v$ describe, respectively, the inequality and equality constraints to be satisfied. The optimization problem (3) represents a fairly large class of optimization problems involving splines. Whether this problem is numerically easy or difficult to solve (convex or nonconvex) depends on (i) whether the spline knot sequence is given or not and (ii) the mathematical properties of the functions $f_0$, $f_v$ and $h_v$.

### 2.1 Fixed Knot Spacing

For a given knot sequence $t_0, \dots, t_{g+1}$, all splines $s(t)$ that comply with the definition (1)–(2) constitute a vector space of dimension $g+k+1$ [16]. This result implies that any $s(t)$ can be written as a unique linear combination of $g+k+1$ *basis* spline functions $s_{k,j}(t)$ that only depend on $k$ and the knot sequence:

$$s(t) = \sum_{j=1}^{g+k+1} d_j s_{k,j}(t). \tag{4}$$

Hence, given $k$ and the knot sequence, the spline is completely determined by the *spline coefficients* $d_j$, such that (3) reduces to an optimization problem in $d \in \mathbb{R}^{g+k+1}$:

$$\begin{aligned} \text{minimize}_{d} \quad & f_0(s(t,d)) && \text{(5a)} \\ \text{subject to} \quad & f_v(s(t,d)) \leq 0, \; v = 1, \dots, n_f && \text{(5b)} \\ & h_v(s(t,d)) = 0, \; v = 1, \dots, n_g && \text{(5c)} \end{aligned}$$

where $d$ collects all $d_j$ and is added as an argument in $s(t,d)$ to indicate relation (4). If $f_0$, $f_v$ and $h_v$ are linear functions of $s(t,d)$, the above optimization problem is a linear program. LPs are easy to solve: problems with hundreds of variables and thousands of constraints can be solved –in the sense of finding the *global* optimum– on a small desktop computer in a matter of seconds. If the problem is *sparse*[3], or has some other exploitable structure, problems with tens or hundreds of thousands of variables and constraints can still be solved [17].

[3] Sparse problems have constraint functions that depend on few variables only.

While the problem (5) is easy to solve if $f_0$, $f_v$ and $h_v$ are linear, it is generally difficult to solve otherwise, due to the existence of an unknown number of local optima. A notable exception, however, occurs if $f_0$ and $f_v$ are *convex*[4] nonlinear functions and $h_v$ is linear. In that case, the resulting optimization problem is convex [17], which guarantees that the global optimum be found fast and reliably. The convex optimization framework for spline optimization is first developed in Sec. 3 as an LP, while nonlinear, convex extensions are subsequently discussed in Sec. 5.

### 2.2 Variable Knot Spacing

While a fixed knot sequence may give rise to a tractable linear or convex optimization problem, it is generally (unless an incremental improvement of an existing design is to be found) not clear beforehand what a ’good’ number of knots is, nor what their location should be. Therefore, it is important that also the knot sequence itself be considered for optimization, as in the automotive cam studies [4, 12] and the cam design software *VENTIL* [12]. Unfortunately, doing so gives rise to nonlinear, nonconvex optimization problems that are difficult to solve.

In fact, in the area of spline optimization for function approximation, directly optimizing the knot sequence is known as a notoriously difficult optimization problem [16]. Similar observations have been made in the area of spline optimization for motion curves. Sandgren and West [4] observed that keeping the spline knots fixed “allowed the optimization routine to quickly locate a reasonable solution in all cases”, while including the knot spacing into the optimization variable gave rise to a fourfold computational time and yet only similar results. Nevertheless, Sandgren and West conclude that “in general, the added flexibility provided by allowing the knot sequence to be altered is important and should be considered.”

In order to circumvent the fundamental problems with variable knot sequences, the framework presented here adopts an *indirect* approach for knot optimization. Instead of considering the knot locations $t_i$ directly as optimization variables, the knot locations $t_i$, $i = 0, 1, \ldots, g+1$ are fixed beforehand, but using a special distribution: the knots are chosen equidistantly[5] as

$$t_i = i \cdot \delta t,\ i = 0, 1, \ldots, g+1 \text{ where } \delta t = \frac{T}{g+1}, \tag{6}$$

and $g$ is a large number, typically a few hundreds or even thousands. Choosing the knots equidistantly implies that the resulting spline $s(t)$ is a *uniform* spline.

The proposed indirect approach has two important consequences for the formulation of the optimization problem. First, while only the spline coefficients need to be optimized (the main advantage of this approach), there is a large number of them. Hence, the objective function $f_0$ and constraint functions $f_v$ and $h_v$ need to be carefully selected to obtain a tractable optimization problem, being either an LP (Sec. 3) or a nonlinear convex program (Sec. 5).

Second, providing a large number $g$ of candidate internal knots $t_i$ implies that the $k$th-order derivative can exhibit a discontinuous jump at each of these locations, which may result in an unnecessarily high number of active knots (see the solid line in Fig. $6(d)$, discussed further on). This phenomenon can, however, be avoided if *one-norm regularization* is applied

[4] A formal definition of convexity is provided in Appendix A.

[5] Also nonuniform distributions as in [4] can be considered.

(see Sec. 4), such that the resulting number of active knots is very small. Hence, while our indirect approach optimizes a uniform spline with a large number of equidistant candidate knots, it returns a non-uniform spline with a small number of non-equidistantly spaced active knots, as numerically illustrated in Secs. 3–4.

### 2.3 Spline Parametrization

Since our indirect approach is based on fixing the knot sequence beforehand, a set of basis functions $s_{k,j}(t)$, as defined in (4) can be chosen. While *B-spline* basis functions are often preferred for their numerical stability [16], an alternative parametrization, inspired by the work of Kwakernaak and Smit [18] is opted for here. This parametrization is in our opinion more intuitive to mechanical engineers and has proven to be numerically stable for all spline optimization problems considered in this work. It is derived in three steps:

First, in order to improve scaling, the proposed representation for a $k$th-degree polynomial spline is developed in *dimensionless form*, that is, for $\theta(\tau)$, where $\tau$ ($0 \leq \tau \leq 2\pi$) is the dimensionless time defined by

$$\tau = t \cdot \frac{2\pi}{T} \;, \tag{7a}$$

while $\theta(\tau)$ denotes a dimensionless version of $s(t)$:

$$\theta(\tau) = \frac{1}{L} \cdot s\left(\tau \cdot \frac{T}{2\pi}\right) \;, \tag{7b}$$

with $L$ a characteristic measure that determines the scale of the motion trajectory. The set of equidistant knots $\tau_i$ is defined by

$$\tau_i = i \cdot \delta\tau,\; i = 0, 1, \ldots, g+1 \text{ where } \delta\tau = \frac{2\pi}{g+1}. \tag{8}$$

In dimensionless form, (4) amounts to

$$\theta(\tau) = \sum_{j=1}^{g+k+1} d_j \theta_{k,j}(\tau), \tag{9}$$

where the *basis spline functions* $\theta_{k,j}(\tau)$ represent the dimensionless counterparts of the basis spline functions $s_{k,j}(t)$.

Second, a particular set of dimensionless basis spline functions $\theta_{k,j}(\tau)$ is chosen. This set is constructed starting from the observation that the definition (1)–(2) of $\theta(\tau)$ as a degree $k$ polynomial spline implies that its $(k-1)$st-order derivative $\theta^{(k-1)}(\tau)$ is continuous and piecewise-linear[6]. Hence, $\theta^{(k-1)}(\tau)$ is completely determined by its values at the knot locations $\tau_i$ (see Fig.2) and can therefore be simply parameterized as

$$\theta^{(k-1)}(\tau) = \sum_{i=0}^{g+1} \theta^{(k-1)}(\tau_i) \cdot \beta_i(\tau), \tag{10}$$

[6] Figure 1 illustrates this for a cubic $k = 3$ spline: its $(k-1)$st (2nd) order derivative is piecewise-linear.

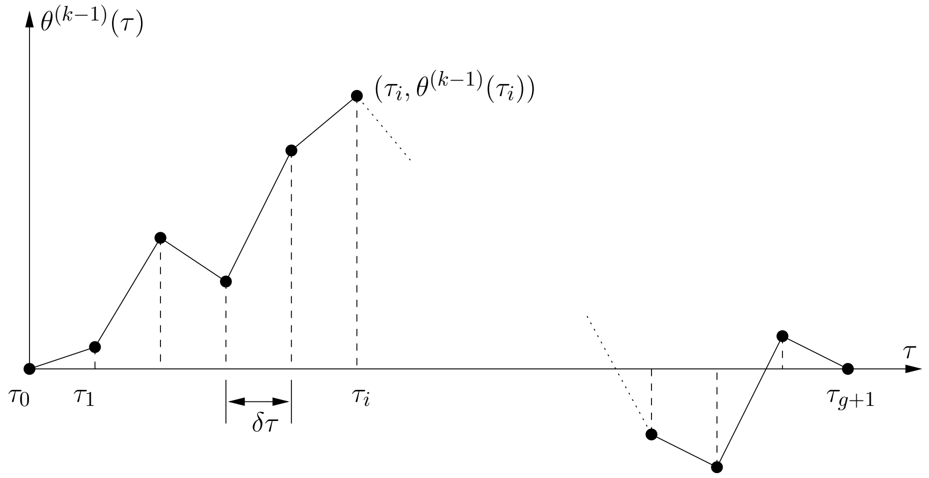

Fig. 2. Piecewise-linear parametrization of $\theta^{(k-1)}(\tau)$, where $\delta\tau = 2\pi/(g+1)$.

where $\beta_i(\tau)$ denotes the *hat function*, further discussed in Appendix B. To obtain $\theta(\tau)$ from (10), $\theta^{(k-1)}(\tau)$ needs to be integrated $k-1$ times that are uniquely determined by fixing values for:

$$\theta^{(q)}(0), \quad q = 0, \ldots, k-2 \,. \tag{11}$$

As a result, every degree $k$ polynomial spline $\theta(\tau)$ is uniquely determined by the following vector $d$ of $g+k+1$ parameters:

$$d = \Big(\theta^{(k-1)}(0)\ \theta^{(k-1)}(\tau_1)\ \cdots\ \theta^{(k-1)}(\tau_{g+1})\ \Big|\ \theta^{(0)}(0)\ \cdots\ \theta^{(k-2)}(0)\Big)^T \,. \tag{12}$$

where $(\cdot)^T$ denotes the matrix transpose. The elements $d_j$ of $d$ correspond to the spline coefficients of $\theta(\tau)$ (Eq. (9)) for the following set of $g+k+1$ basis spline functions $\theta_{k,j}(\tau)$:

$$j = 1,2,\ldots,g+2 : \theta_{k,j}(\tau) \text{ equals the hat function } \beta_{j-1}(\tau)\text{, integrated} \tag{13a}$$

$$k-1 \text{ times with zero integration constants}$$

$$j = g+3,\ldots g+k+1 : \theta_{k,j}(\tau) = \frac{\tau^{(j-g-3)}}{(j-g-3)!} \,. \tag{13b}$$

The preference for this set of basis functions stems from the clear physical interpretation of the corresponding spline coefficients (12): the first $g+2$ coefficients directly determine the piecewise-linear function $\theta^{(k-1)}(\tau)$, whereas the remaining coefficients fully characterize the spline at boundary point $\tau = 0$.

Third, to enhance the numerical stability of evaluating $\theta(\tau)$ and its derivatives at the knots $\tau_i$, this *evaluation* is performed in a *recursive* manner. That is, instead of computing $\theta^{(q)}(\tau_i)$ from Eq. (9), it is computed by the following recursive relation, which is only valid for the equidistant knot spacing defined by (8), and follows from elementary calculus ($i = 1,\ldots,g+1$; $q = 0,\ldots,k-2$):

$$\theta^{(q)}(\tau_i) = \left[\sum_{j=0}^{k-q-2} \theta^{(q+j)}(\tau_{i-1}) \cdot \frac{(\delta\tau)^j}{j!}\right] + \frac{(k-q-1)\cdot(\delta\tau)^{k-q-1}}{(k-q)!} \cdot \theta^{(k-1)}(\tau_{i-1}) + \frac{(\delta\tau)^{k-q-1}}{(k-q)!} \cdot \theta^{(k-1)}(\tau_i) \,. \tag{14}$$

This recursion writes $\theta^{(q)}(\tau_i)$ as a *linear* combination of the $k-q$ derivatives $\theta^{(q)}(\tau_{i-1}),\ldots,\theta^{(k-1)}(\tau_{i-1})$ evaluated at the previous time instant $\tau_{i-1}$, as well as the $(k-1)$st-order derivative $\theta^{(k-1)}(\tau_i)$ evaluated at the considered time instant $\tau_i$. The implementation of recursive evaluation (14) requires augmenting the vector $d$ of spline coefficients to $x \in \mathbb{R}^{k\cdot(g+2)}$:

$$x = \left( \left(\Theta^{(0)}\right)^T \; \left(\Theta^{(1)}\right)^T \ldots \left(\Theta^{(k-1)}\right)^T \right)^T, \tag{15}$$

where $\Theta^{(q)}$ equals the $(g+2)$-vector of all $\theta^{(q)}(\tau_i)$ values, $i = 0,1,\ldots,g+1$. The linear recursion (14) can then be written compactly as

$$G_{k,g} \cdot x = 0, \tag{16}$$

where $G_{k,g}$ is a full-rank matrix belonging to $\mathbb{R}^{(k-1)(g+1)\times k(g+2)}$ and only depending on the spline degree $k$ and the number $g$ of equidistant internal knots. Equation (16) imposes $(k-1)(g+1)$ linearly independent, linear equations on the $k(g+2)$ elements of $x$, resulting in $k+g+1$ independent degrees of freedom: the spline coefficients $d$ defined by (12).

## 3 Linear Programming Framework

Based on the spline parametrization of Sec. 2.3 and the analytical benchmark problem introduced in Sec. 3.1, this section develops the convex optimization framework as a linear program (nonlinear, convex extensions are discussed in Sec. 5). After defining the optimization variables (Sec. 3.2), constraints (Sec. 3.3) and objective (Sec. 3.4), the framework's linear programming structure is highlighted in Sec. 3.5. Section. 3.6 provides numerical validation based on the analytical benchmark problem.

### 3.1 Analytical Benchmark Problem

The considered benchmark problem arises in the area of *double-dwell* motion trajectory design. In dimensionless form (7), double-dwell curves move a load from a lower dwell (period of standstill) at position $\theta = 0$ at $\tau = 0$ to an upper dwell at position $\theta = 1$ at $\tau = 2\pi$. A whole body of literature exists on the design of such curves, of which well-known analytical examples include the cycloidal curve, modified trapezoidal, modified sine and a great many others [6]. In this context it is known that in general a smoother curve, that is, a curve with a higher degree of continuity, results in less excitation of the structural dynamics of the driven mechanism[7], while on the other hand a higher degree of continuity results in higher peak values of velocity, acceleration and jerk. Hence, the main design objective is to synthesize curves of a prescribed smoothness that realize an optimal trade-off between peak values of velocity, acceleration and jerk. In the context of this paper, the considered analytical benchmark problem aims at finding a $\mathcal{C}^m$-continuous double-dwell curve $\theta(\tau)$ with minimal peak value of the $(m+1)$st-order derivative $\theta^{(m+1)}(\tau)$, which is the first discontinuous yet still finite derivative. More specifically,

[7] provided that the driven mechanism is sufficiently stiff (Although a motion system may be assumed to be rigid in the design phase, it is never infinitely rigid in practice.)

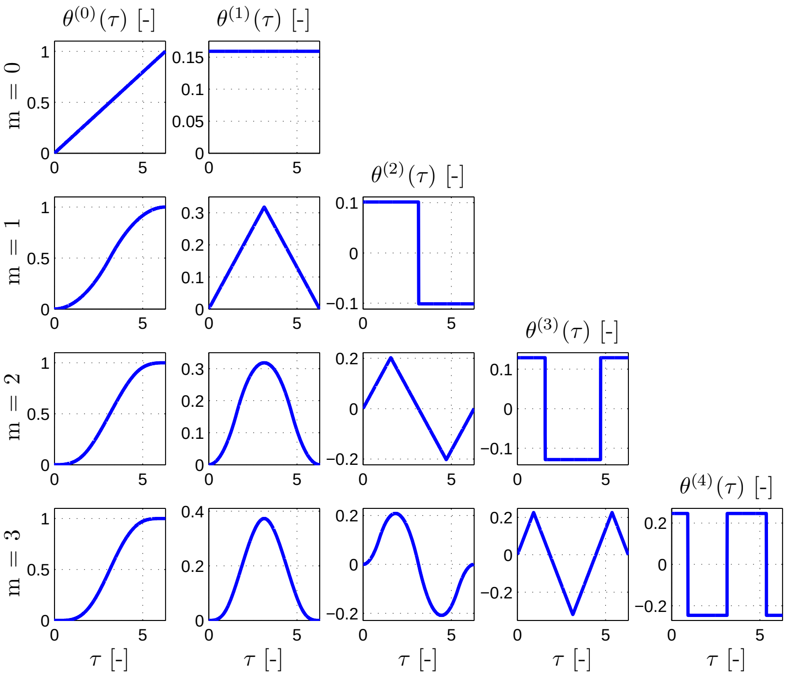

Fig. 3. Analytical benchmark problem solutions for $m = 0, 1, 2, 3$. The solution is a polynomial spline of degree $m + 1$ with $g = m$ internal knots.

$\theta(\tau)$, $t \in [0, 2\pi]$ must satisfy

$$\theta(\tau) \in C^m_{[0,2\pi]} \tag{17a}$$

$$\theta(0) = 0; \quad \theta(2\pi) = 1 \tag{17b}$$

$$\theta^{(q)}(0) = 0;\ \theta^{(q)}(2\pi) = 0,\ q = 1, \ldots, m \tag{17c}$$

$$\max_{\tau \in [0,2\pi]} |\theta^{(m+1)}(\tau)| \text{ minimized} \tag{17d}$$

For example, a designer wanting to find a double-dwell curve that complies with the fundamental law of cam design and minimizes follower vibration, will be eager to know the solution of the above problem for $m = 2$, since it results in the cubic spline that minimizes peak jerk.

The analytical solution of (17) is known from [19] and is illustrated in Fig. 3 for $m = 0, 1, 2, 3$. These solutions can be represented as polynomial splines of degree $k = m + 1$ with $g = m = k - 1$ internal knots [19]. Table 1 provides the knot locations $\tau^*_{mj}$, $j = 0, 1, \ldots, m + 1$ and the peak absolute values of the finite-valued derivatives.

| m | $\tau^*_{k,j}$, $j=0,1,\ldots,k$ | $\max(\lvert\theta^{(1)}\rvert)$ | $\max(\lvert\theta^{(2)}\rvert)$ | $\max(\lvert\theta^{(3)}\rvert)$ | $\max(\lvert\theta^{(4)}\rvert)$ |
|---|---|---|---|---|---|
| 0 | $2\pi\cdot\{0,1\}$ | $1/(2\pi)$ | $\infty$ | $\infty$ | $\infty$ |
| 1 | $2\pi\cdot\{0,\frac{1}{2},1\}$ | $2/(2\pi)$ | $4/(2\pi)^2$ | $\infty$ | $\infty$ |
| 2 | $2\pi\cdot\{0,\frac{1}{4},\frac{3}{4},1\}$ | $2/(2\pi)$ | $8/(2\pi)^2$ | $32/(2\pi)^3$ | $\infty$ |
| 3 | $2\pi\cdot\{0,z,\frac{1}{2},1-z,1\}$ | $2.34/(2\pi)$ | $8.23/(2\pi)^2$ | $79.53/(2\pi)^3$ | $384/(2\pi)^4$ |

Table 1. Analytical benchmark problem solutions for $m=0,1,2,3$. The solution is a polynomial spline of degree $m+1$ with $g=m$ internal knots. The scalar z equals $(2-\sqrt{2})/4$.

### 3.2 Optimization Variables

Given that $\mathcal{C}^m$-continuity is required, the spline degree $k$ is selected as $m+1$. The number of knots $g$ is chosen to be large, typically a few hundreds or even thousands. To adopt the numerically stable recursive implementation (14), vector $x$ (15) is used as optimization variable instead of $d$ (12), whereby the set of linear equality constraints (16) must be included in the optimization problem. Definition (15) of $x$ allows expressing $\theta^{(q)}(\tau_i)$ as

$$\theta^{(q)}(\tau_i) = x(q(g+2)+i+1) \tag{18a}$$

$$= a_{q,i}^T \cdot x, \tag{18b}$$

where $a_{q,i} \in \mathbb{R}^{k(g+2)}$ is a selection vector of which all elements are zero, except the element with index $q(g+2)+i+1$, which equals one. Shorthand notation (18b) allows us to express the constraints and objective criterion in a compact and neat manner.

### 3.3 Constraints

The choice of $x$, defined by (15), as the optimization variable implies that the linear set of equations defined by (16) needs to be imposed to guarantee that $x$ represents a polynomial spline of degree $k=m+1$. While $G_{m+1,g}$ is typically a large matrix, it is also very sparse. The sparsity can be understood from (14) which relates every element of $x$ to at most $m+2$ other elements of $x$. Hence, while $G$ features $\mathcal{O}(m^2g^2)$ elements, only $\mathcal{O}(m^2g)$ are nonzero, which is important given that $m$ is small but $g$ is large.

A second set of constraints follows from the boundary constraints (17b)–(17c):

$$a_{0,0}^T \cdot x = 0, \qquad a_{0,g+1}^T \cdot x = 1 \tag{19a}$$

$$a_{q,0}^T \cdot x = 0 \quad (q=1,\ldots,m), \qquad a_{q,g+1}^T \cdot x = 0 \quad (q=1,\ldots,m) \tag{19b}$$

or, more concisely

$$A_{BC} \cdot x = b_{BC}, \tag{20}$$

where $A_{BC} \in \mathbb{R}^{2(m+1)\times(m+1)(g+2)}$ is large and sparse and $b_{BC} \in \mathbb{R}^{2(m+1)}$. Equation (20) represents a set of $2(m+1)$ *linear* equality constraints in $x$. The fact that constraints such as (17b)–(17c) can so easily be expressed in terms of $x$ constitutes one of the major advantages of considering the set of variables defined by (15).

### 3.4 Objective Function

The objective in the analytical benchmark problem is to minimize

$$\max_{\tau\in[0,2\pi]} |\theta^{(m+1)}(\tau)| \, , \tag{21}$$

which is equivalent to minimizing

$$\max_i |\theta^{(m+1)}(\tau_i)| = \max_i \left| \frac{\theta^{(m)}(\tau_i) - \theta^{(m)}(\tau_{i-1})}{\delta\tau} \right| \, , \tag{22a}$$

$$= \max_i \left| \frac{a^T_{m,i} - a^T_{m,i-1}}{\delta\tau} \cdot x \right| \, . \tag{22b}$$

Given the piecewise-linear, continuous parametrization of $\theta^{(m)}(\tau)$, the backward difference used in (22a) constitutes an exact expression for the left derivative of $\theta^{(m)}(\tau)$, as well as an unambiguous definition of the discontinuous function $\theta^{(m+1)}(\tau)$. Using a forward difference (and hence defining $\theta^{(m+1)}(\tau)$ as the right derivative of $\theta^{(m)}(\tau)$) yields equivalent results.

Objective function (21) is convex, but as it is piecewise-linear, minimizing this function directly does not give rise to a linear program. However, as detailed in [17], the problem of minimizing (21) can be transformed into the following *equivalent linear program*:

$$\text{minimize}_{(x,w)} \quad w \tag{23a}$$

$$\text{subject to} \qquad -w \le \frac{a^T_{m,i} - a^T_{m,i-1}}{\delta\tau} \cdot x \le w, \; i = 1,\ldots,g+1. \tag{23b}$$

where $w$ is a scalar auxiliary optimization variable. This is an optimization problem in $(x,w)$ with a linear objective function $w$ and $2(g+1)$ inequality constraints, all of them linear in $w$ and $x$.

|   | $\varepsilon_p$ [%] |   |
|---|---|---|
| m | $g = 1000$ | $g = 999$ |
| 0 | 1.74e-14 | 1.74e-14 |
| 1 | 9.98e-05 | 9.59e-13 |
| 2 | 2.99e-04 | 4.28e-12 |
| 3 | 9.71e-04 | 7.91e-04 |

Table 2. Relative error $\varepsilon_p$ [%] between the numerically obtained peak absolute value of $\theta^{(k)}(\tau)$ and the analytical values of Table 1 for $g = 1000$ and $g = 999$.

### 3.5 Resulting Linear Program

Putting everything together, the analytical benchmark problem can be translated in the following optimization problem, where the designer choices are the continuity level $m$ and the number $g$ of (uniformly distributed) internal knots:

$$\text{minimize}_{(x,w)} \quad w \tag{24a}$$

$$\text{subject to} \quad -w \leq \frac{a_{m,i}^T - a_{m,i-1}^T}{\delta\tau} \cdot x \leq w, \; i = 1, \ldots, g+1 \tag{24b}$$

$$G_{m+1,g} \cdot x = 0, \tag{24c}$$

$$A_{BC} \cdot x = b_{BC}. \tag{24d}$$

(24) constitutes a linear program for the objective and constraints are linear functions of the optimization variable $(x, w)$. In the case of cubic splines ($m = 2$), for instance, choosing $g = 1000$ gives rise to a moderate to large-scale linear program involving 3007 variables, 2008 linear equality constraints and 2002 linear inequality constraints. All matrices involved are, however, sparse, such that dedicated sparsity-exploiting linear programming algorithms can still guarantee finding the global optimum in a matter of seconds. Here, the simplex and interior-point algorithms of the MOSEK[8] toolbox are used.

### 3.6 Numerical Results

This section investigates whether the proposed framework is able to reproduce the analytical results given Sec. 3.1. The overall answer to this question is 'yes'. In fact, Fig. 3 was made, not by implementing the analytical solution, but by plotting numerical solutions of (24) for $g = 1000$ and $m = 0 \ldots 3$. Solving the corresponding LPs took a mere 1.34 CPU seconds on a laptop equipped with a Pentium4@3.2GHz processor and 1GB RAM.

While the algorithm is allowed to put an active knot in any of the $g$ available internal knots, Fig. 3 reveals that only $m$ internal

[8] www.mosek.com

knots are active[9], as in the analytical solution. Table 2, on the other hand, verifies whether the obtained peak absolute values of $\theta^{(m+1)}(\tau)$ agree with the analytical values of Table 1. To this end the relative error $\varepsilon_p$ is defined as:

$$\varepsilon_p = \frac{|\theta^{(k)*} - \theta^{(k)+}|}{\theta^{(k)+}},$$

where $\theta^{(k)*}$ and $\theta^{(k)+}$ represent the maxima of the absolute values of $\theta^{(k)}(\tau)$ resulting from the numerical and analytical solutions, respectively. For $m = \{1,2\}$, the reported relative errors $\varepsilon_p$ are small but significant for $g = 1000$, while negligible for $g = 999$. Such distinctions between the two considered $g$ values do not occur for $m = \{0,3\}$.

This dependence on $g$ follows from the fact that $g$ determines whether the analytical knots $\tau^*_{mj}$ belong to the set of available knots $\tau_i$. For $m = \{1,2\}$, Eq. (8) implies that the $\tau_i$-set includes the analytical knots for odd $g$, while not for even $g$. For $m = \{0,3\}$, on the other hand, the value of $g$ is irrelevant: (i) $m = 0$ represents a trivial case for no internal knots are present; (ii) for $m = 3$ the knots $2\pi\{z, 1-z\}$ can never be part of the $\tau_i$-set for $z$ (defined in Table 1) is an irrational number[10]. If the $\tau_i$-set includes *all* analytical knots, $\varepsilon_p$ is numerically zero and the exact solution has been found to numerical precision (Table 2: $m = \{1,2\}$ and odd $g$). In the other case, the error can be made arbitrarily small by increasing $g$. The latter claim is not formally proved but numerically verified for $m = 3$ by solving (24) for 50 logarithmically spaced (even) $g$ values between 10 and 1000. Figure 4 shows that the relative error $\varepsilon_p$ globally (but not monotonically) decreases for increasing $g$ and drops just below 0.0001% for $g = 1000$. The decrease is linear on a double logarithmic scale implying that $\varepsilon_p$ drops two orders of magnitude for $g$ increasing one.

Figures $5(a-b)$ provide additional insight in these results by showing how the numerically computed knots $\hat{\tau}_{m2}$ and $\hat{\tau}_{m3}$[11] vary as a function of $g$ for $m = 3$. The solid lines indicate error bounds, which are computed based on the conjecture that if the $\tau_i$-set does not contain the analytical knot $\tau^*_{mj}$, the computed $\hat{\tau}_{mj}$-value is the element of the $\tau_i$-set lying closest to $\tau^*_{mj}$. $\hat{\tau}_{mj}$ therefore belongs to the interval

$$\left[\tau^*_{mj} - \frac{\delta\tau}{2}, \tau^*_{mj} + \frac{\delta\tau}{2}\right] = \left[\tau^*_{mj} - \frac{\pi}{g+1}, \tau^*_{mj} + \frac{\pi}{g+1}\right]. \tag{25}$$

implying error bounds equal to $\pm\frac{\pi}{g+1}$. Figures $5(a-b)$ provide numerical evidence that this conjecture is true. Observe that the numerically computed knot $\hat{\tau}_{m3}$ always lies on the error bound, since for any even $g$ the analytical value $\tau^*_{m3} = 0.5 \cdot 2\pi$ lies in the middle of the interval between the two nearest available knots. Figures $5(c-d)$ provide similar information as Figs. $5(a-b)$ but for 50 logarithmically spaced *odd* $g$ values between 11 and 1001. In this case $\tau^*_{m3} = 0.5 \cdot 2\pi$ belongs to the $\tau_i$-set and is always chosen by the numerical algorithm.

[9] As can be seen by counting the number of jumps in $\theta^{(m+1)}(\tau)$ for each of the considered $m$.

[10] Observe that the factor $1/(g+1)$ in (8) is a *rational* number for integer $g$.

[11] The values of $\hat{\tau}_{mj}$ are numerically found by detecting jumps of $\theta^{(m+1)}(\tau)$ in the solution of (24).

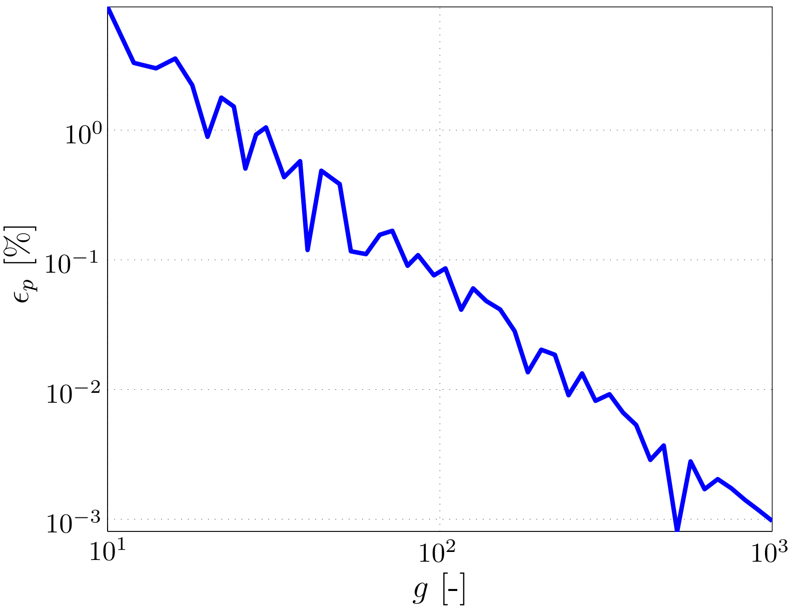

Fig. 4. Relative error $\varepsilon_p$ [%] on the obtained peak absolute value of $\theta^{(4)}(\tau)$ for $m = 3$ and 50 logarithmically spaced even $g$-values between 10 and 1000.

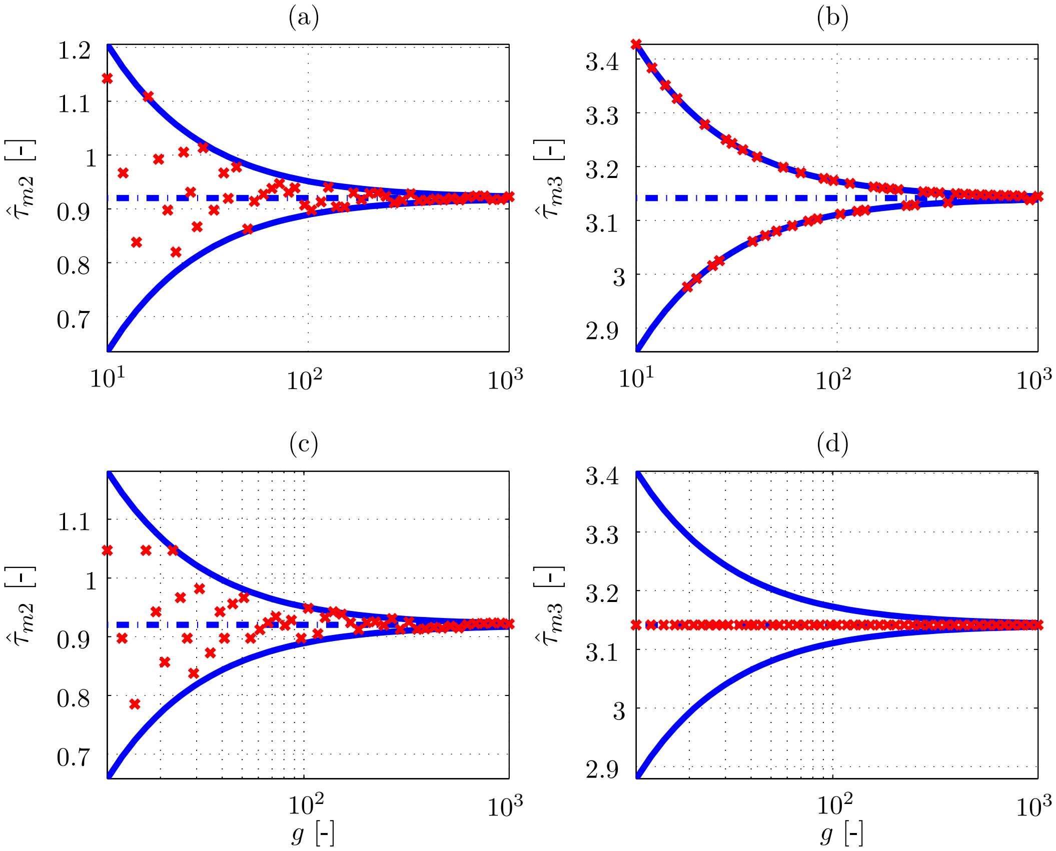

Fig. 5. Numerically computed knots $\hat{\tau}_{m2}$ and $\hat{\tau}_{m3}$ ('x') for $m = 3$ and 50 logarithmically spaced even or odd $g$-values between 10 and 1001. $(a)$ $\hat{\tau}_{m2}$ for even values of $g$; $(b)$ $\hat{\tau}_{m3}$ for even values of $g$; $(c)$ $\hat{\tau}_{m2}$ for odd values of $g$; $(d)$ $\hat{\tau}_{m3}$ for odd values of $g$. The analytical values $\tau^*_{m2}$ and $\tau^*_{m3}$ are indicated by the dash-dotted line, while the solid lines indicate the error bounds derived from (25).

## 4 One-Norm Regularization

The numerical results of the previous section reveal that, while the numerical algorithm is allowed to put an active knot in any of the $g$ available internal knots, only very few of them effectively exhibit a $\theta^{(m+1)}(\tau)$ jump. Such 'tidy' solutions do no always result: Sec. 4.1 shows that making the analytical benchmark slightly more complicated gives rise to very 'dirty' solutions featuring an unnecessarily high number of active knots, as revealed by the $\theta^{(m+1)}(\tau)$ trajectory nervously banging back and forth between its upper and lower limit (solid line in Fig. $6(d)$). However, it is also shown (Sec. 4.2) that such solutions can be cleaned up very efficiently and, quite importantly, at minimal cost through a technique called *one-norm regularization*. One-norm regularization constitutes the third key feature of the present framework and is indispensable to deal with more complex trajectory design problems.

### 4.1 Analytical Benchmark Problem Revisited

In order to make the analytical benchmark problem more complicated, two *precision points* are added, that is, points $(\tau_{p,j}, \theta_{p,j})$ through which $\theta(\tau)$ must pass:

$$\theta(\tau_{p,j}) = \theta_{p,j}, \tag{26}$$

where

$$(\tau_{p,1} = 1.26, \theta_{p,1} = 0.32), \quad (\tau_{p,2} = 3.77, \theta_{p,2} = 0.70). \tag{27}$$

These precision points were chosen in an arbitrary way while making sure that they do not lie on the analytical solution presented in Sec. 3.1 and Fig. 3.

Taking into account that the time instants $\tau_{p,j}$ do not necessarily coincide with the available knot locations $\tau_i$, the framework enforces (26) by requiring that $\hat{\theta}(\tau)$, the linear interpolation of $\theta(\tau)$ between the knots $\tau_i$, pass through the prescribed points:

$$\hat{\theta}(\tau_{p,j}) = \theta_{p,j}. \tag{28}$$

Expressing (28) in terms of $x$, with the subscript $j$ dropped, gives rise to

$$\left[a_{0,\underline{p}}^T + \frac{a_{0,\overline{p}}^T - a_{0,\underline{p}}^T}{\delta\tau} \cdot (\tau_p - \tau_{\underline{p}})\right] \cdot x = \theta_p, \tag{29}$$

where[12]

$$\underline{p} = \lfloor \frac{\tau_p}{\delta\tau} \rfloor,$$
$$\overline{p} = \underline{p} + 1.$$

[12] The floor operator $\lfloor x \rfloor$ rounds $x$ to the nearest integer less than or equal to $x$.

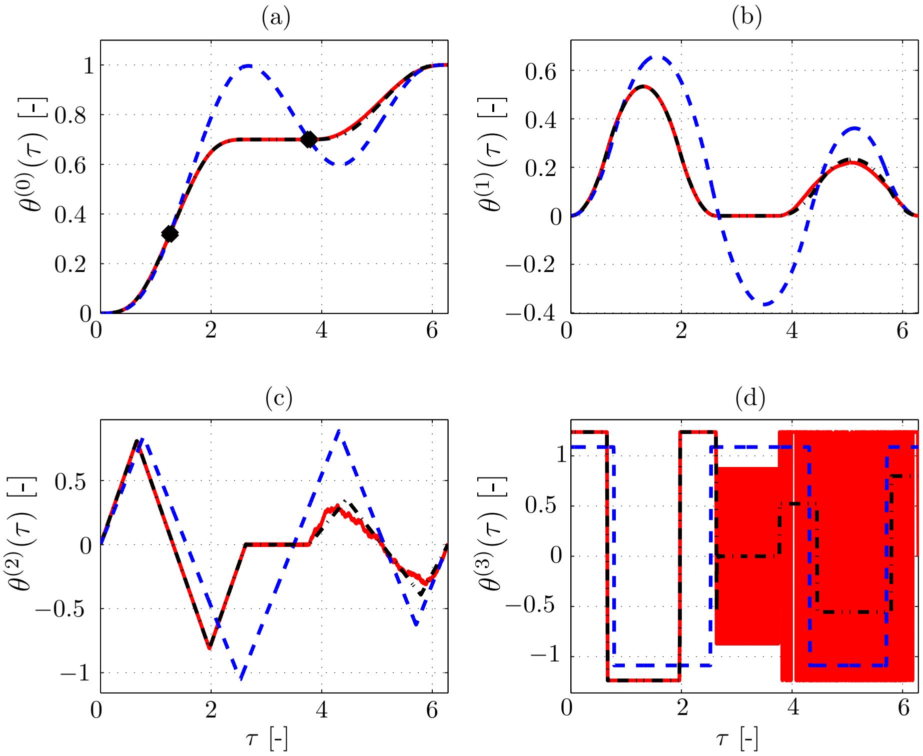

Fig. 6. Analytical benchmark problem revisited $(g = 1000, m = 2)$: non-monotonous solution (dashed line), monotonous solution without one-norm regularization (solid line), monotonous solution with one-norm regularization (dash-dotted line). For $\tau \leq 2.64$, the latter two solutions coincide. The circles in Fig. 6$(a)$ indicate the precision points (27).

For every considered precision point $(\tau_p, \theta_p)$, Eq. (29)[13] constitutes a linear equality constraint in $x$. Numerically solving (24) for $(g = 1000, m = 2)$ and the precision points (27) gives rise to the dashed line in Fig. 6. Compared to the solution of the analytical benchmark (Fig. 3), the number of active internal knots has doubled from two to four, but is still limited. However, the solution is not monotonous, which might be undesirable in many applications of double dwell motions, because it gives rise to unnecessarily large peak values of velocity and acceleration (see Fig. 3), causing increased wear and inertial load.

Monotonicity is enforced by imposing that $\theta^{(1)}(\tau_i) \geq 0, i = 0, 1, \ldots, g+1$:

$$a_{1,i}^T \cdot x \geq 0,\; i = 0, 1, \ldots, g+1. \tag{30}$$

Adding the monotonicity constraint has a spectacular effect on the solution, as shown by the solid line in Fig. 6. The solution is monotonous and passes through the required precision points, but features a large number of active knots in the $[2.64, 6.28]$ interval as revealed by the jerk $\theta^{(3)}(\tau)$, which bangs nervously between $\pm w^* = \pm 1.237$ in this part of the motion cycle, where $w^*$ is the optimal value of objective function (21).

Further interpretation of this unacceptable solution provides the clue to improving upon it. The velocity and acceleration profiles reveal that in the $[2.64, 3.77]$ interval, a standstill is aimed for. The extremely nervous jerk profile is consistent

[13] Being based on the linear interpolation (28), (29) is an approximation of the exact constraint (26). However, given that $\delta\tau$ is typically very small, the approximation is sufficiently accurate. If judged necessary (26) can be expressed *exactly* as a linear, yet more tedious to derive, equality constraint in $x$.

with the observed standstill, but does not look very intuitive nor sensible to a designer. Our conjecture is that the designer's obvious guess (that is, the same trajectory but with zero jerk for $\tau \in [2.64, 3.77]$) and the algorithm's result are equally optimal but different solutions of the considered optimization problem: both give rise to the same optimal peak jerk $w^* = 1.237$ and comply with the imposed set of constraints[14]. The challenge is now to force the numerical algorithm to choose, in the set of all optimal $x^*$, an $x^*$ that looks sensible to a designer. Finding such a solution that does not feature any unnecessary jerk jumps (active knots) can be done through the one-norm regularization introduced in Sec. 4.2.

### 4.2 One-Norm Regularization

If our conjecture of the existence of a whole set of equivalent $x$-vectors is true, it should be possible to find a sensible solution $x$ by solving the following optimization problem $(m = 2)$

$$\begin{aligned}
&\text{minimize}_x \quad f_s(x) && \text{(31a)}\\
&\text{subject to } \; -w^* \leq \frac{a_{m,i}^T - a_{m,i-1}^T}{\delta\tau} \cdot x \leq w^*, \; i = 1, \dots, g+1 && \text{(31b)}\\
&\qquad G_{m+1,g} \cdot x = 0, && \text{(31c)}\\
&\qquad A_{BC} \cdot x = b_{BC} && \text{(31d)}\\
&\qquad A_p \cdot x = b_p && \text{(31e)}\\
&\qquad a_{1,i}^T \cdot x \geq 0, \; i = 0, 1, \dots, g+1, && \text{(31f)}
\end{aligned}$$

where (31e) concisely expresses the precision point constraints (29). $w^*$ is the optimal value of objective function (21), that is, the maximum absolute value of $\theta^{(m+1)}$ obtained by solving the non-regulated linear program (24) supplemented with the precision point constraints (29) and the monotonicity constraint (30). This optimization problem seeks, in the set of all feasible $x$ that give rise to the optimal $w^*$, the particular $x$ that minimizes $f_s(x)$, an objective function that measures the 'lack of sensibility' of a solution. Given that solutions with fast switching of $\theta^{(m+1)}(\tau)$ are judged not very sensible, it makes sense to obtain smoother solutions by penalizing some norm of its backward difference, $\theta^{(m+2)}(\tau)$. The particular norm chosen here is the one norm, that is,

$$f_s(x) = \sum_{i=2}^{g+1} \left| \theta^{(m+2)}(\tau_i) \right| = \sum_{i=2}^{g+1} \left| \frac{\theta^{(m+1)}(\tau_i) - \theta^{(m+1)}(\tau_{i-1})}{\delta\tau} \right|. \tag{32}$$

It is well-known [17] in the area of function approximation that one-norm minimization is likely to yield sparse solutions, that is, solutions with only few nonzero components. Translated to the present problem, sparsity implies solutions with few nonzero $\theta^{(m+2)}(\tau_i)$, that is, few $\theta^{(m+1)}(\tau)$-jumps. In other words, few active knots.

Numerically solving the linear program (31) with (32) as the objective function[15] for the precision points (27) and $(g =$

[14] This conjecture does not contradict the convexity of the problem: convexity guarantees that the globally optimal objective value $f_0^*$ be found, but there may be many vectors $x^*$ for which $f_0(x^*) = f_0^*$. Which of these $x^*$ is actually obtained depends on the particular algorithm used.

[15] While in the form (32), the one-norm constitutes a nonlinear, nondifferentiable function due to the presence of the absolute value, it can easily be transformed into a linear objective function through the introduction of auxiliary variables and additional linear inequality constraints, similar to the transformation discussed in Sec. 3.4. More details are provided in, e.g., [17].

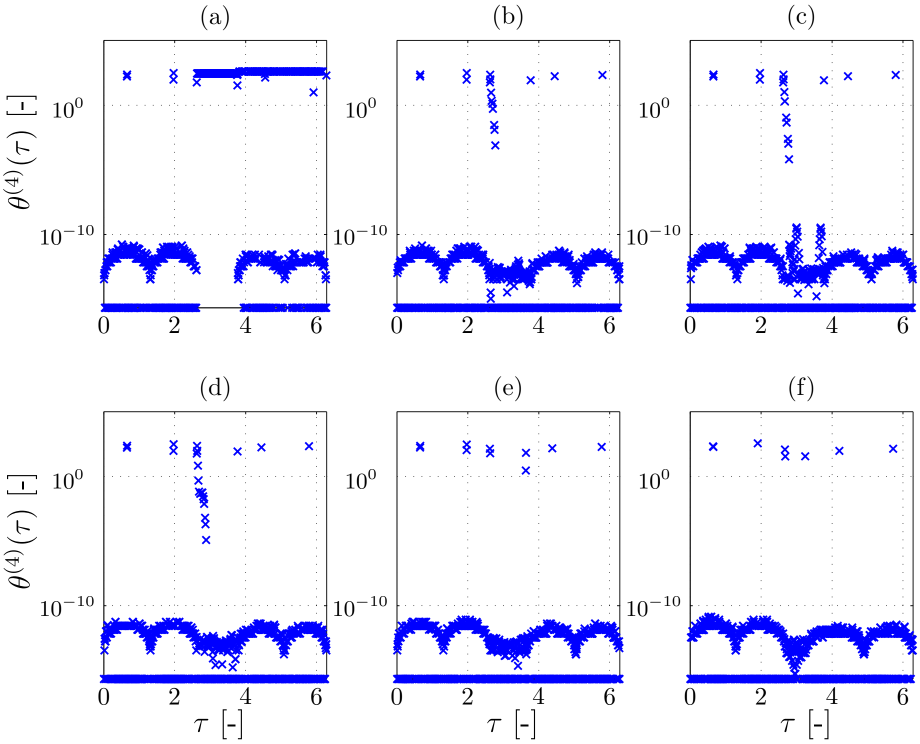

Fig. 7. Analytical benchmark problem revisited $(g = 1000, m = 2)$: jerk derivative on logarithmic scale for monotonous solution without one-norm regularization $(a)$, monotonous solution with one-norm regularization $(b)$, monotonous solutions with relaxed one-norm regularization: $\varepsilon = 1e-12$ $(c)$, $\varepsilon = 1e-8$ $(d)$, $\varepsilon = 1e-4$ $(e)$, $\varepsilon = 1e-2$ $(f)$.

$1000, m = 2)$ gives rise to the dash-dotted line in Fig. 6. The improvement is remarkable and confirms the conjecture: the obtained solution is sensible, featuring zero jerk for $\tau \in [2.64, 3.77]$ and a mere six internal knots.

Figure 7 further analyzes the effect of the one-norm minimization by plotting $\theta^{(m+2)}(\tau_i)$ on a logarithmic scale as a function of $\tau$. Figure $7(a)$ focuses on the original monotonous solution (solid line of Fig. 6), while Fig. $7(b)$ shows the solution of (31). The difference between the two figures confirms the far lower number of knots observed in Fig. 6. However, Fig. $7(b)$ also reveals that the solution of (31) is not as 'clean' as suggested by the six apparent jerk jumps in Fig. $6(d)$: although most of the knots feature a $\theta^{(m+2)}(\tau_i)$ that is numerically zero (below $10^{-10}$), some clearly nonzero $\theta^{(m+2)}(\tau_i)$ values appear in pairs, while others $\theta^{(m+2)}(\tau_i)$ are in some "gray zone" ranging from $10^{-5}$ to 10. To further investigate this effect, the constraint (31b) is relaxed to:

$$-(1+\varepsilon)\cdot w^* \leq \frac{a_{m,i}^T - a_{m,i-1}^T}{\delta\tau}\cdot x \leq (1+\varepsilon)\cdot w^*,\ i = 1,\ldots,g+1 \tag{33}$$

That is, slightly higher values for the peak jerk are allowed, so as to obtain even more sparse solutions. Figure $7(c-f)$ show the results obtained for $\varepsilon = \{1e-12, 1e-8, 1e-4, 1e-2\}$. It is clear that the solutions do get sparser, at the small cost of a slightly higher peak jerk.

## 5 Convex Objectives and Constraints

This section provides several nonlinear framework extensions that turn the linear programming framework into a more general convex programming framework. In doing so, the known optimization methods [4, 7–13] are reviewed, thereby showing that some of these methods can be extended or simplified based on insights from convex optimization theory.

(i) Instead of minimizing the peak absolute value (21) of the $(m+1)$st-order (or any other) derivative of $\theta(\tau)$, also other functions of $\theta(\tau)$ and its derivatives can be minimized. If a weighted combination (weights $W_j$) of mean-square values is considered,

$$\sum_{j=0}^{m+1} \frac{W_j}{2\pi} \int_0^{2\pi} \left(\theta^{(j)}(\tau)\right)^2 \cdot \mathrm{d}\tau \tag{34}$$

a convex quadratic program follows[16], while a (convex) second-order cone program results if a weighted combination of root-mean-square values is considered:

$$\sum_{j=0}^{m+1} W_j \sqrt{\frac{1}{2\pi} \int_0^{2\pi} \left(\theta^{(j)}(\tau)\right)^2 \cdot \mathrm{d}\tau}. \tag{35}$$

Mermelstein and Acar [10] developed, as an improvement of [8], a method to minimize the mean-square jerk (that is, an objective of type (34), with all $W_j = 0$, except $W_3 = 1$) of a spline subject to a number of prescribed positions, velocities and/or higher derivatives at specific time instants. For this equality-constrained, convex quadratic program, a complicated solution procedure was developed of which the implementation ”requires a symbolic mathematics software application, as the calculations involved are lengthy” [10]. This solution procedure relies on the fact that the solution of an equality-constrained, convex quadratic program can be found by solving a set of linear equations. There is, however, no need to develop such a complicated procedure, since any freely available convex solver (such as the matlab-based software packages SeDuMi [20] and SDPT3 [21] or the Python-based software CVXOPT[17]) will do the job.

(ii) Nguyen and Kim [13] designed cam profiles based on the idea of a smoothing spline. That is, given some existing nominal trajectory $\theta_n(\tau)$ (e.g., a trapezoidal trajectory), a spline $\theta(\tau)$ of odd degree $k$ is sought that has minimal *roughness*[18] and simultaneously satisfies (i) given boundary constraints of the general type (17b)–(17c), as well as (ii) user-specified upper limits $S_j \geq 0$ on the sum-of-squares deviation of $\theta^{(j)}(\tau)$ with respect to $\theta_n^{(j)}(\tau)$:

$$\sum_{i=0}^{p} \left( \frac{\theta^{(j)}(\tau_{p_i}) - \theta_n^{(j)}(\tau_{p_i})}{W_{ji}} \right)^2 \leq S_j, \quad j = 0, \ldots, k-1 \tag{36}$$

where the $(p+1)$ time instants $\tau_{p_i}$ and the relative weights $W_{ji}$ of the various time instants must be chosen by the user. Minimizing the roughness (a convex quadratic function), subject to boundary constraints (linear equality constraints) and the

[16] Convex quadratic programs are discussed more thoroughly in the companion paper [15].

[17] `http://www.ee.ucla.edu/~vandenbe/cvxopt/`

[18] Roughness is defined as the mean square of the $(k-1)/2$'th-order derivative and hence, corresponds to a convex quadratic objective function of the general type (34) (34).

convex quadratic constraints (36) gives rise to a convex program, more specifically a *quadratically constrained quadratic program*. The convexity was, however, not recognized nor exploited by Nguyen and Kim: instead of using one of the aforementioned standard convex solvers, a dedicated quasi-Newton method was developed, while also the guarantee to obtain a global optimum was not recognized.

(iii) While nonlinear (but convex) extensions are required for the framework to reproduce the results [10, 13], the original linear programming framework suffices to reproduce the results [11] (kinematic optimization example) and [7].

Yoon and Rao [7] minimized the peak acceleration of a cubic spline subject to prescribed positions, velocities and accelerations at specific time instants. This problem can be directly formulated as an LP in the framework of Secs. 3–4. Linear programming is, however, not mentioned in [7], nor is any information about the method with which the optimization problem is solved.

Qiu et al. [11], on the other hand, minimized the peak acceleration of a quintic $(k = 5)$ spline subject to boundary constraints and upper limits on peak velocity and peak absolute jerk. While, as shown in the companion paper [15], this problem can be reformulated as an LP, Qiu et al. solved it as a general nonlinear program. The companion paper [15] furthermore shows that considering a large number of knots (350 instead of 7, as did Qiu et al.) leads to significantly improved results thereby illustrating the benefit of formulating linear/convex programs whenever possible.

(iv) All aforementioned studies [7, 8, 10, 13] and [11] (kinematic optimization example) are methods to optimize purely *kinematic* properties of a spline with prescribed knots. The basic framework of Secs. 3–4, along with its nonlinear extensions (34) and (36) is able to simplify (no need to write dedicated numerical solution procedures; proof of global optimality follows from convexity), and extend any of these results. Extensions are twofold. First of all, the convexity of the presented framework allows considering a large number of knots, which pays off as shown in the companion paper [15] for the case study [11]. Second, other objectives such as (35) can be considered, while the companion paper further generalizes the framework such that it can also deal with (i) time optimality; (ii) upper and lower bounds on motor torque for servomotor driven systems; (iii) the time-energy optimality trade-off in servomotor driven systems.

(v) Preserving convexity implies that objectives and constraints related to cam geometry [4, 9, 12] or cam driving torque ( [11], dynamic optimization example) cannot be included in the framework. The main problem with cam geometry is the complex geometric relation between the motion curve and the corresponding cam profile. The main problem with cam driving torque is that it involves products of velocity and acceleration (inertia part) or velocity and displacement (spring part). Still, however, a kinematically optimized motion law can provide a very good initialization for a nonconvex optimization routine involving the cam shape, since many geometric and dynamic aspects of cam design can be expressed approximately in terms of purely kinematic properties. Minimizing peak acceleration, for instance, results in a spline that is a good starting value for finding a spline that minimizes Hertzian pressure. Minimizing

$$\max_{\tau}\{|\theta^{(2)}(\tau)| \quad | \quad \theta^{(2)}(\tau) \leq 0\}$$

on the other hand, can be cast as an LP and is in general beneficial for minimizing return spring size.

(vi) While cam geometry cannot be directly included in the convex programming framework, load vibration can, provided that it is described by a linear set of differential equations. Preliminary efforts along these line are described in [22].

## 6 Discussion

The present paper develops a general framework to synthesize optimal polynomial splines for rigid motion systems driven by cams or servomotors. This framework is based on numerical optimization and has three main characteristics: (i) spline knot locations are optimized through an indirect approach based on providing a large number of fixed, uniformly distributed candidate knots; (ii) in order to efficiently solve the corresponding large-scale optimization problem to global optimality, only design objectives and constraints are allowed that result in convex programs and (iii) one-norm regularization is used as an effective tool for selecting the better (that is, having fewer active knots) solution if many equally optimal solutions exist.

One-norm regularization as a means of controlling the number of active knots does not allow specifying the number of active knots beforehand. The authors, however, feel that this is only a minor restriction: the more important consideration is that the number of knots is limited, not whether it is exactly four, six or ten. The use of one-norm regularization is to a large extent inspired by the idea of (smoothing) *regularization*. In mathematics, the term *regularization* is associated with making a function more ’regular’ or smooth. Regularization is, for instance, commonly used in approximation and fitting problems. In regularized approximation the goal is to find a vector $x$ that is small (if possible) and also makes the residual $Ax-b$ small, where $A$ and $b$ denote given data [17]. This goal is achieved by solving the optimization problem

$$\text{minimize} ||Ax-b||+\gamma||x||, \tag{37}$$

where the norms may be different and $\gamma > 0$ is a user-defined parameter. As $\gamma$ varies over $(0,\infty)$, the solution of (37) traces out an optimal trade-off curve. If both norms are equal to the two-norm, *Tikhonov regularization* results. If, on the other hand, the one-norm is selected for $||x||$, a sparse solution $x$ is likely to be found, that is, a solution with only few nonzero $x_i$. For an elaborate discussion on this topic, the reader is referred to [17].

In fact, the whole framework laid out in the present paper can be thought of as a variation of *basis pursuit* [23], another well-known concept from function approximation. In basis pursuit, there is a very large number of basis functions and the goal is to find a good fit of some given data as a linear combination of a small number of the basis functions. The term basis pursuit was coined since a much smaller basis is selected from a given over-complete basis. A commonly used heuristic to finding such a sparse description is to minimize an objective of the general type (37), using a one-norm for $||x||$. In the present framework, one can think of the $g+2$ equidistantly spaced available knots as defining the over-complete basis and the one-norm minimization of $\theta^{(m+2)}(\tau_i)$ as the heuristic to select a smaller basis, defined by much fewer non-uniformly distributed knots. The use of the one-norm as a sparsity-promoting function also underlies a new sensing/sampling paradigm, termed *compressive sampling* [24], as well as total variation reconstruction in image processing [25].

The current framework has an interpretation as an optimal control problem in which a piecewise-linear control signal is applied to a series of $m$ integrators. This interpretation may be beneficial if also nonconvex nonlinear functions need to be considered for efficient, structure-exploiting algorithms exist that are able to find reasonable local optima for nonconvex optimal control problems [26].

The framework was developed for the spline basis presented in Sec. 2.3, while an implementation based on the numerically more stable and more commonly used B-spline basis is also already available. Preliminary numerical experiments with the latter implementation revealed identical solutions but longer computational times. Which basis is more appropriate for elastic (instead of rigid) motion systems, is currently under investigation.

While the current paper already provides substantial numerical evidence of the efficiency of the proposed framework, more complicated numerical benchmarks and extensions are considered in the companion paper [15]. Current research focuses on further extending and experimentally validating the preliminary results [22] concerning follower vibration.

## Acknowledgment

The authors wish to acknowledge S. Vandewalle and P. Dierckx of K.U.Leuven's Computer Science Department and M. Diehl of the K.U.Leuven Optimization in Engineering Center (OPTEC) for their pertinent remarks on the subject of this paper. Bram Demeulenaere was a Postdoctoral Fellow of the Research Foundation - Flanders (FWO - Vlaanderen) until 2008 and has carried out this research as a visiting scholar (2004–2005) at the Electrical Engineering Dept. of the University of California, Los Angeles. The support of the following projects is gratefully acknowledged: FWO projects G.0446.06 and G.0462.05; Research Council K.U.Leuven, CoE EF/05/006 Optimization in Engineering (OPTEC); the Belgian Programme on Interuniversity Attraction Poles, initiated by the Belgian Federal Science Policy Office.

## References

[1] Tsay, D., and Huey Jr., C., 1993. "Application of rational B-splines to the synthesis of cam-follower motion programs". *Transactions of the ASME, Journal of Mechanical Design,* **115**, pp. 621–626.

[2] Tsay, D., and Huey Jr., C., 1988. "Cam motion synthesis using spline functions". *Transactions of the ASME, Journal of Mechanisms, Transmissions, and Automation in Design,* **110**, pp. 161–165.

[3] Mosier, R., 2000. "Modern cam design". *Int. J. Vehicle Design,* **23**((1/2)), pp. 38–55.

[4] Sandgren, E., and West, R., 1989. "Shape optimization of cam profiles using a B-spline representation". *Transactions of the ASME, Journal of Mechanisms, Transmissions, and Automation in Design,* **111**, pp. 195–201.

[5] Neklutin, C., 1952. "Designing cams for controlled inertia and vibration". *Machine Design*, pp. 143–160.

[6] Norton, R., 2002. *Cam Design and Manufacturing Handbook.* Industrial Press, Inc.

[7] Yoon, K., and Rao, S., 1993. "Cam motion synthesis using cubic splines". *Transactions of the ASME, Journal of Mechanical Design,* **115**, pp. 441–446.

[8] Wang, L., and Yang, Y., 1996. "Computer aided design of cam motion programs". *Computers in Industry,* **28**, pp. 151–161.

[9] Kim, J., Ahn, K., and Kim, S., 2002. "Optimal synthesis of a spring-actuated cam mechanism using a cubic spline".

*Proceedings of the Institution of Mechanical Engineers: Part C–Journal of Mechanical Engineering Science,* **216**, pp. 875–883.

[10] Mermelstein, S., and Acar, M., 2004. "Optimising cam motion using piecewise polynomials". *Engineering with Computers,* **19**, pp. 241–254.

[11] Qiu, H., Lin, C.-J., Li, Z.-Y., Ozaki, H., Wang, J., and Yue, Y., 2005. "A universal optimal approach to cam curve design and its applications". *Mechanism and Machine Theory,* **40**, pp. 669–692.

[12] Hartwig, K.-H., 2006. "Kurvengetriebe zur steuerung des ladungswechsels in verbrennungsmotoren (cam mechanisms that drive the charge cycle in combustion engines)". *Proc. VDI-Getriebetagung 2006, VDI-Berichte 1966,* Fulda, Germany,, pp. 29–43.

[13] Nguyen, V., and Kim, D., 2007. "Flexible cam profile synthesis method using smoothing spline curves". *Mechanism and Machine Theory,* **42**, pp. 825–838.

[14] Rockafellar, R., 1993. "Lagrange multipliers and duality". *SIAM Review,* **35**, pp. 183–283.

[15] Demeulenaere, B., De Caigny, J., Swevers, J., and De Schutter, J. "Optimal splines for rigid motion systems: Benchmarking and extensions". *Transactions of the ASME, Journal of Mechanical Design*. Submitted for Publication.

[16] Dierckx, P., 1993. *Curve and Surface Fitting with Splines*. Oxford University Press.

[17] Boyd, S., and Vandenberghe, L., 2004. *Convex Optimization*. Cambridge University Press.

[18] Kwakernaak, H., and Smit, J., 1968. "Minimum vibration cam profiles". *J. of Mech. Engr. Sci.,* **10**, pp. 219–227.

[19] Van de Straete, H. J., and De Schutter, J., 1999. "Optimal variable transmission ratio and trajectory for an inertial load with respect to servo motor size". *Transactions of the ASME, Journal of Mechanical Design,* **121**, pp. 544–551.

[20] Sturm, J., 1999. "Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones". *Optimization Methods and Software,* **11–12**, pp. 625–653. Special issue on Interior Point Methods (CD supplement with software).

[21] Tutuncu, R., Toh, K., and Todd, M., 2003. "Solving semidefinite-quadratic-linear programs using sdpt3". *Mathematical Programming Ser. B,* **95**, pp. 189–217.

[22] Demeulenaere, B., De Caigny, J., Swevers, J., and De Schutter, J., 2007. "Dynamically compensated and robust motion system inputs based on splines: A linear programming approach". In Proc. of American Control Conference, pp. 5011–5018.

[23] Chen, S., Donoho, D., and Saunders, M., 1998. "Atomic decomposition by basis pursuit". *SIAM Journal on Scientific Computing,* **20**(1), pp. 33–61.

[24] Candès, E., and Wakin, M., 2008. "An introduction to compressive sampling". *IEEE Signal Processing Magazine,* **25**(2), pp. 21–30.

[25] Rudin, L., Osher, S. J., and Fatemi, E., 1992. "Nonlinear total variation based noise removal algorithms". *Physica D,* **60**, pp. 259–268.

[26] Betts, J., 2001. *Practical Methods for Optimal Control Using Nonlinear Programming*. SIAM, Philadelphia.

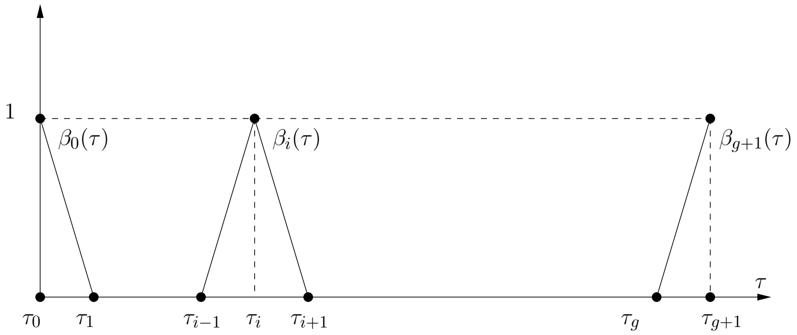

Fig. 8. The hat functions $\beta_0(\tau)$, $\beta_i(\tau)$ $(1 \leq i \leq g)$ and $\beta_{g+1}(\tau)$.

## A Convex Functions and Sets

A set $S \subseteq \mathbb{R}^n$ is convex if the line segment between any two points in $C$ lies in $C$, that is, if for any $x,y \in S$ and $\theta \in [0,1]$, we have

$$\theta x + (1-\theta)y \in S. \tag{38}$$

A function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is convex if its domain $\mathbf{dom} f$ is a convex set and if for all $x,y \in \mathbf{dom} f$ and $\theta \in [0,1]$, we have

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta) f(y). \tag{39}$$

Geometrically, this inequality means that the line segment between $(x; f(x))$ and $(y; f(y))$, which is the chord from $x$ to $y$, lies above the graph of $f$. Some examples (on $\mathbb{R}$) include: (i) $f(x) = x^2$ is convex; (ii) $f(x) = \log 1/x$ is convex ($\mathbf{dom} f = \{x | x > 0\}$); $f(x) = 1/x$ is convex ($\mathbf{dom} f = \{x | x > 0\}$).

## B The Hat Function

The hat function $\beta_i(\tau)$ is defined by $(1 \leq i \leq g)$

$$\beta_i(\tau) = \begin{cases} 0 & \tau \leq \tau_{i-1} \\ \frac{\tau-\tau_{i-1}}{\tau_i-\tau_{i-1}} & \tau_{i-1} \leq \tau \leq \tau_i \\ \frac{-\tau+\tau_i}{\tau_{i+1}-\tau_i}+1 & \tau_i \leq \tau \leq \tau_{i+1} \\ 0 & \tau \geq \tau_{i+1} \end{cases}.$$

$\beta_0$ is defined as

$$\beta_0(\tau) = \begin{cases} 0 & \tau \leq \tau_0 \\ \frac{-\tau+\tau_0}{\tau_1-\tau_0}+1 & \tau_0 \leq \tau \leq \tau_1 \\ 0 & \tau \geq \tau_1 \end{cases}.$$

$\beta_{g+1}$ is defined as

$$\beta_{g+1}(\tau) = \begin{cases} 0 & \tau \leq \tau_g \\ \frac{\tau - \tau_g}{\tau_{g+1} - \tau_g} & \tau_g \leq \tau \leq \tau_{g+1} \\ 0 & \tau \geq \tau_{g+1} \end{cases} .$$