Optimal Splines for Rigid Motion Systems: Benchmarking and Extensions

B. Demeulenaere*, J. De Caigny†, G. Pipeleers†, J. De Schutter†, J. Swevers†

*Atlas Copco Airpower NV - Airtec Division
Boomsesteenweg 957, B-2610 Wilrijk - Belgium

†Katholieke Universiteit Leuven
Mechanical Engineering Department
Celestijnenlaan 300B, B-3001 Leuven, Belgium

ABSTRACT

This paper illustrates the power and versatility of the convex programming framework for optimal spline synthesis that was developed in a companion paper. Two case studies concerning rigid motion systems illustrate the ability of the framework to improve upon recent (2005) literature results: (i) a numerical optimization study concerning kinematic optimization of uniform quintic splines for cam systems and (ii) an analytical study concerning time optimal quartic splines for motion systems driven by servomotors and subject to kinematic constraints. In a third study, the versatility of the framework is illustrated by generating time optimal and time-energy optimal motions for a rigid servomotor driven system under torque constraints. Based on these three case studies, the convex programming framework of the companion paper is extended with the following generic aspects: (i) bisection to generate time optimal motions, (ii) direct expression of upper and lower bounds on motor torque and (iii) a convex quadratic energy objective function for servomotor driven systems.
List of Figures

1. The quintic spline $\theta(\tau)$ and its first three derivatives, obtained as a solution of (5) by Qiu et al. [4].................. 8
2. Trajectories of ping $\theta^{(4)}(\tau)$ and puff $\theta^{(5)}(\tau)$ for solutions 1 (Qiu et al., solid line), 2 (dashed line) and 3 (dash-dotted line) indicated in Fig. 3. The thick vertical line at $\tau = 0.5$ indicates the infinite puff-value at this location................................................................. 9
3. Trade-off between peak value $C_p$ of puff and acceleration $C_a$ for quintic splines with peak velocity $C_v \leq C_{v,Qiu}$ and peak jerk $C_j \leq C_{j,Qiu}$ and complying with the (incomplete) boundary constraints (5d)–(5e). Results for $N = 7$ (dashed line) and $N = 350$ (solid line). The design of Qiu et al. [4] is indicated by the cross labeled 1. Table 1 summarizes the properties of the competing designs 1, 2 and 3................................................................. 12
4. Trajectories of ping $\theta^{(4)}(\tau)$ and puff $\theta^{(5)}(\tau)$ for solution 1 indicated in Fig. 3 (Qiu et al., solid line) and the solutions obtained through regularization of the solutions 2 (dashed line) and 3 (dash-dotted line). The thick vertical line at $\tau = 0.5$ indicates the infinite puff-value at this location................................................................. 13
5. Solutions of (11) for case 1 (Table 2): discrete solution of Lambrechts et al. (dashed line) and our bisection solution (solid line, endpoint marked by circle). ................................................................. 18
6. Solutions of (11) for case 2 (Table 2): discrete solution of Lambrechts et al. (dashed line) and our bisection solution (solid line, endpoint marked by circle). ................................................................. 19
7. Analytical solution of (15): $s(t)$ and its first two derivatives, as well as corresponding motor torque $M_m(t)$. The horizontal dashed lines in Figs. (b), (d) indicate the upper and lower limits expressed by (15d) and (15e), respectively................................................................. 23
8. (a) Relative error $\epsilon_r(g)$ between the analytical solution $T_a^\ast$ of (15) and the solution $T_b^\ast$ determined by bisection using $g$ internal knots. (b) Computational time required for the whole bisection process as a function of $g$. The solid lines mark linear regression lines................................................................. 23
9. Time-energy trade-off. $E_{\min}(T)/E_{\min}(T^\ast)$ (solid line) and $M_{rms}(T)/M_{rms}(T^\ast)$ (dashed line) as a function of $T/T^\ast$. Cases 1, 2, and 3 defined in section 4.4 and presented in Table 5 are indicated................................................................. 25
10. $s(t)$ and its first two derivatives, as well as corresponding motor torque $M_m(t)$ of time optimal (case 1, solid line) and time-energy optimal trajectories (case 2, dash-dotted line; case 3, dashed line) corresponding to the numerical data of Table 4. Table 5 summarizes the main properties. The endpoint of the trajectories is marked by a circle (case 1), diamond (case 2) or asterisk (case 3). The horizontal dashed lines in Figs. (b), (d) indicate the upper and lower limits expressed by (16d) and (16e), respectively................................................................. 26
List of Tables

1  Peak absolute values of the derivatives $\theta^{(q)}(\tau)$ of the various solutions 1 [4], 2 and 3 displayed in Fig. 3. Numbers between brackets indicate values after regularization. .......................... 12
2  Numerical data for solving (11). ................................................................. 17
3  Solutions for (11) (numerical data of Table 2) obtained by Lambrechts et al. (continuous: $T_{c,Lam}$ and discrete $T_{d,Lam}$) and our bisection approach ($T_{bise}$) as well as the relative improvement $\delta_t [%]$ defined by (13). ............................... 17
4  Numerical parameters defining problem (15). ....................................... 20
5  Required time $T$ as well as kinematic and dynamic properties of time optimal (case 1) and time-energy optimal trajectories (cases 2–3) corresponding to the numerical data of Table 4. ............................... 25
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.

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 increasing sequence $t_i, i = 0, \ldots, g + 1$ if $s(t)$ is a polynomial of degree $\leq k$ on each knot interval $[t_i, t_{i+1}]$ and (ii) $s(t)$ and its derivatives up to order $k - 1$ are continuous on $[0, T]$: $s(t) \in C^{k-1}[0, T]$. 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. The above definition implies that the $k$th order derivative of $s(t)$ may exhibit discontinuous jumps in each knot $t_i$. Such knots are termed active knots, as opposed to inactive (and hence, in fact, redundant) knots which do not feature such discontinuities. If all knots are equidistantly spaced, the spline is termed uniform.

For a given knot sequence $t_0, \ldots t_{g+1}$, all degree $k$ polynomial splines $s(t)$ constitute a vector space of dimension $g + k + 1$ [1]. This result implies that, for a given knot sequence, any $s(t)$ can be written as a unique linear combination of $g + k + 1$ basis spline functions $s_{k,j}(t)$, as for instance B-splines:

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

Scalars $d_j$ are termed the spline coefficients, and are grouped in $d \in \mathbb{R}^{g+k+1}$.

Optimizing polynomial splines ideally amounts to optimizing spline knots and spline coefficients simultaneously. However, such an approach generally results in optimization problems that are difficult to solve numerically [1], where this difficulty primarily stems from the nonlinear dependency of $s_{k,j}(t)$ on the spline knots. On the other hand, not considering the spline knots for optimization may result in severely suboptimal results.

In the companion paper [2], a convex programming framework was developed to synthesize optimal polynomial splines for rigid motion systems driven by cams or servomotors. This paper illustrates the power and versatility of this framework by benchmarking it against two case studies from literature and by further extending it to solve more complex but practically relevant motion synthesis problems. Before proceeding to the case studies considered here, the main concepts and results of the companion paper are briefly summarized below.

1.1 Convex Programming Framework for Spline Synthesis

The framework presented in [2] has the following three main characteristics.

(i) Since optimizing the spline knot locations directly generally results in optimization problems that are very difficult to solve numerically [1], the framework proposes an indirect approach to optimize spline knot locations: A large number of fixed, uniformly distributed candidate knots is provided, and optimizing the spline coefficients generally results in solutions where only few candidate knots are active.
(ii) If splines with an unacceptably high number of active knots result, one-norm regularization is used as an effective tool to favor solutions having fewer active knots.

(iii) In order to efficiently solve the resulting large-scale optimization problem to global optimality, only design objectives and constraints are allowed that result in convex programs. Convex programs constitute a class of nonlinear optimization problems for which, similarly as for linear optimization problems (or linear programs, LPs), the global optimum is guaranteed to be found efficiently and reliably using dedicated algorithms. Contrary to general nonlinear optimization problems, convex programs do not suffer from local optima, in which the optimization algorithm can get trapped. Formulating optimization problems as convex programs, therefore, has great advantages.

The framework’s implementation elaborated in [2] first converts spline \( s(t) \) to a dimensionless form \( \Theta(\tau) \), while the implementation for \( s(t) \) in physical dimensions is very similar. In [2], a spline parametrization inspired by [3] is preferred over B-splines, since the corresponding spline coefficients have a clear interpretation: they directly determine the piecewise-linear \( (k-1) \)st-order derivative \( \Theta^{(k-1)}(\tau) \) of degree \( k \) polynomial spline \( \Theta(\tau) \), and fully characterize the spline at boundary point \( \tau = 0 \). To enhance the numerical stability of evaluating \( \Theta(\tau) \) and its derivatives on the knots \( \tau_i \), this evaluation is performed in a recursive manner. The implementation of this recursive evaluation requires augmenting the vector \( d \) of spline coefficients to \( x \in \mathbb{R}^{k(g+2)} \):

\[
x = \left( \left( \Theta^{(0)} \right)^T \left( \Theta^{(1)} \right)^T \ldots \left( \Theta^{(k-1)} \right)^T \right)^T,
\]

where \( \Theta^{(q)} \) equals the \((g+2)\)-vector of all \( \Theta^{(q)}(\tau) \) values, \( i = 0, 1, \ldots, g + 1 \), and \((\cdot)^T\) denotes the matrix transpose. The recursive evaluation of \( \Theta(\tau) \) then translates into a set of linear equations in \( x \), concisely written as:

\[
G_{k,g} \cdot x = 0,
\]

where \( G_{k,g} \) is a sparse, full-rank matrix belonging to \( \mathbb{R}^{(k-1)(g+1) \times k(g+2)} \). Consequently, (3) leaves \((g+k+1)\) independent design parameters in \( x \): the spline coefficients.

In the companion paper [2], all optimization problems involved are elaborated in full technical detail, that is, in terms of the optimization variable \( x \) (2). For reasons of conciseness and readability, this level of detail is abandoned here by developing all optimization problems in a more abstract form, that is, in terms of the motion trajectory \( \Theta(\tau) \) or \( s(t) \) and its derivatives. Regaining the detailed problem formulation in terms of \( x \), requires substitution of the following relation:

\[
\Theta^{(q)}(\tau_i) = x(q(g+2)+i+1),
\]

and addition of the consistency constraint (3).

1.2 Contributions

This paper illustrates the power and versatility of the convex programming framework for motion synthesis of rigid motion systems. In particular: (i) it is shown that several previous results, obtained by ad-hoc analytical or numerical methods, can
all be obtained using the same generic framework; (ii) it is shown that previous results that were not recognized as convex optimization problems, are in fact convex. Hence, more efficient solution algorithms can be used to solve the problem (instead of a general-purpose solver for nonlinear problems); (iii) given the efficient solution of convex programs, it is possible to increase the number of knots considerably, while keeping the computation time limited, yielding a better numerical approximation of the global optimum; (iv) given the efficient solution of convex programs, it is possible to generate trade-off curves to investigate the effect of relaxing some of the constraints; (v) while fitting in the convex optimization framework, it is shown that previous results can easily and efficiently be extended to more complicated cases involving dynamic and energy considerations instead of only kinematic considerations; (vi) the framework is extended to deal with problems involving time optimal motions (instead of motions in a prescribed period of time).

The three considered examples deal with double-dwell motion trajectory design for rigid motion systems, in which a load must be moved from a lower dwell (period of standstill) at position $s = 0$ to an upper dwell at position $s = L$ in a period of time $T$.

The first example (Ex. 1, Sec. 2) concerns the kinematic optimization study carried out by Qiu et al. [4]. While the companion paper [2] (Sec. 5) showed that any of the kinematic spline optimization studies [4–8] can be represented, simplified and/or extended by the proposed convex optimization framework, the study by Qiu et al. was selected here for numerical benchmarking, since it clearly illustrates the importance of each of the three aforementioned main framework characteristics.

The second example (Ex. 2, Sec. 3) benchmarks the framework with respect to the recently published analytical method [9] for designing time optimal double-dwell motions for systems driven by servomotors. Time optimal motions are obtained by extending the framework with a bisection algorithm.

The third example (Ex. 3, Sec. 4) is not a literature benchmark but reveals the versatility of the framework by extending it for servomotor driven systems with (i) direct expression of upper and lower bounds on motor torque and (ii) a convex quadratic energy objective function. Both time and time-energy optimal control problems are considered.

The three proposed examples move from simple to more complicated in several respects: (i) first, only kinematic properties are optimized (Exs. 1 and 2), while Ex. 3 also considers dynamic aspects of the considered motion system; (ii) first, the motion time $T$ is prescribed (Ex. 1), while it is minimized afterwards (Exs. 2 and 3); (iii) while LPs are solved in Exs. 1 and 2, more general nonlinear convex programs, that is, convex quadratic programs are considered in Ex. 3.

All reported numerical optimization results are obtained using MOSEK\textsuperscript{1} as a numerical solver, running on a Pentium4@3.2GHz processor with 1 GB RAM.

2 Example 1: Benchmark Qiu et al. (2005)

Recently, Qiu et al. [4] published a method to optimize the coefficients\textsuperscript{2} of uniform B-splines in the area of cam design. The aim of this benchmark is designing a double-dwell curve that improves the kinematic characteristics of the frequently

\textsuperscript{1}www.mosek.com

\textsuperscript{2}Qiu et al. call the spline coefficients control points, which should not be confused with spline knots.
used modified trapezoidal law [10]. Results are developed here in the classical dimensionless form to be independent of the trajectory duration $T \, [s]$ and the motion amplitude $L \, [m]$ or [rad]: the dimensionless trajectory $\theta(\tau)$ moves the load from $\theta = 0$ at dimensionless time instant $\tau = 0$ to $\theta = 1$ at $\tau = 1$.

2.1 Results of Qiu et al. (2005)

Qiu et al. parameterize the double-dwell curve as a uniform quintic spline, implying that the spline knots are equidistantly spaced and that each spline segment is a fifth degree polynomial. Only half of the time axis is considered, since a symmetrical solution is anticipated:

$$\theta(\tau) = 1 - \theta(1 - \tau), \quad 0 \leq \tau \leq 0.5.$$  

The considered optimization problem amounts to

$$\begin{align*}
\text{minimize}_d & \quad \max_{\tau \in [0, 0.5]} |\theta^{(2)}(\tau)| \\
\text{subject to} & \quad |\theta^{(1)}(\tau)| \leq C_{v,mt} \\
& \quad |\theta^{(3)}(\tau)| \leq C_{j,mt} \\
& \quad \theta(0.0) = 0.0; \theta^{(1)}(0.0) = 0; \theta^{(2)}(0.0) = 0 \\
& \quad \theta(0.5) = 0.5; \theta^{(2)}(0.5) = 0
\end{align*}$$

where $C_{v,mt} = 2$ and $C_{j,mt} = 61.4$ refer to the peak absolute velocity and jerk of the modified trapezoidal law. Optimization variable $d$ collects the $N + 5$ spline coefficients of a uniform quintic ($k = 5$) spline with $N + 1$ equidistantly spaced knots $\tau_{Qiu,i} = i \cdot \delta \tau_{Qiu}$, where $i = 0, 1 \ldots, N$, and

$$\delta \tau_{Qiu} = \frac{0.50}{N}.$$  

As revealed by Eq. (1), $\theta(\tau)$ and its derivatives $\theta^{(q)}(\tau)$ are linear functions of the optimization variable $d$. Optimization problem (5) is convex, but nonlinear and nondifferentiable due to the absolute value function in the objective function and constraints. However, Qiu et al. do not recognize the convexity of their design problem and they solve it in the nonlinear and nondifferentiable formulation (5), although it can be reformulated as an LP based on insights from convex optimization theory (see Sec. 2.2). To solve (5), Qiu et al. propose an enhanced version of the Complex Search Method [11], a derivative-free algorithm for nonlinear optimization. The problem is turned into an unconstrained optimization problem by eliminating the linear equality constraints (5d)–(5e) and adding the remaining constraints (5b)–(5c) as penalty terms to the original objective (5a). The factors weighing the three objective function terms are automatically adjusted to ensure compliance with the

\footnote{\textsuperscript{3}This choice might have been inspired by the nondifferentiability of the problem.} 

\footnote{\textsuperscript{4}That is, the optimization variable $d$ is replaced by a new optimization variable $\tilde{d}$ with $N$ independent spline coefficients that represent a quintic spline that automatically complies with the five linear equality constraints (5d)–(5e).}
constraints (5b)–(5c). An optimized peak acceleration value of 4.869 is reported for \( N = 7 \), a 0.39\% reduction with respect to the 4.888 modified trapezoidal peak acceleration. \( N = 7 \) is selected, since “further improvement cannot be expected by increasing \( N \)”. No information is provided regarding required computational times.

Figure 1 shows the first three derivatives of the result for \( N = 7 \), reproduced based on the spline coefficients provided in [4], while Fig. 2 (solid line) focuses on the corresponding ping \( \theta^4(\tau) \) and puff \( \theta^5(\tau) \). Figure 2(a) clearly illustrates what can already be anticipated from the sharp cusp at \( \tau = 0.5 \) in Fig. 1(d): the resulting solution is not a quintic spline, since the ping features a discontinuity at \( \tau = 0.5 \). To correct for this problem an additional boundary condition \( \theta^4(0.5) = 0.0 \) would have been required. For the sake of solving identically the same problem as Qiu et al., it was decided not to correct for this problem.

Sampling the solution of Qiu et al. with a time increment

\[
\delta\tau = 0.5 / 350
\]

results in a peak acceleration \( C_{a,Qiu} = 4.8568 \) somewhat lower than the reported 4.869, while the peak jerk \( C_{j,Qiu} = 61.5374 \) slightly violates the \( C_{j,mt} = 61.4 \) constraint. This violation might be due to Qiu et al. using a coarser time increment in their optimization routine\(^5\). The peak absolute velocity \( C_{v,Qiu} \) equals 2.000, as required in the optimization.

\(^5\)Qiu et al. provide no information about the time increment with which \( \theta(\tau) \) and its derivatives are sampled to evaluate the objective function (5a) and the constraints (5b)–(5c).
Fig. 2. Trajectories of ping $\theta^{(4)}(\tau)$ and puff $\theta^{(5)}(\tau)$ for solutions 1 (Qiu et al., solid line), 2 (dashed line) and 3 (dash-dotted line) indicated in Fig. 3. The thick vertical line at $\tau = 0.5$ indicates the infinite puff-value at this location.

2.2 Reproduction of Benchmark Results

Since design problem (5) is convex, the solution of Qiu et al. shown in Figs 1 and 2 corresponds to the global optimum, within the accuracy limits of their solution approach (see above). Consequently, our spline optimization framework must be able to reproduce this result, where this reproduction involves the following steps:

(i) While Qiu at al. use the B-spline basis functions to parameterize the uniform spline $\theta(\tau)$, our framework adopts an alternative parametrization as briefly explained in Sec.1. In addition, $\theta(\tau)$ is evaluated in a recursive manner, which requires the use of $x$ (2) instead of $d$ (the vector of spline coefficients) as optimization variable, as well as the inclusion of equality constraints (3) in the optimization problem.

(ii) In our terminology, $N = 7$ corresponds to $g = N - 1 = 6$ internal knots. However, our framework does not suffice with the 8 equidistantly spaced knots $\tau_{Qiu,i} (i=0,1,\ldots,7)$ (6) as it only evaluates the spline and its derivatives at the candidate spline knots. Hence, by only considering the (very low number of) candidate knots provided by Qiu et al., the optimization framework would underestimate the peak jerk, since the peak jerk occurs in between these knot locations. In order to circumvent this problem, $\delta \tau$ is reduced to (7), which yields the following candidate knots $\tau_i$:

$$\tau_i = i \cdot \delta \tau, \quad i = 0, 1, \ldots, 350.$$  

This introduces a new problem: we now have $g = 349$ candidate internal knots instead of 6. In order to obtain a fair\(^6\)

\(^6\)Another detail important for obtaining an exact benchmark is that the upper limits $C_{r, \text{end}}$ and $C_{j, \text{end}}$ in (5b) and (5c) are replaced by $C_{r,Qiu}$ and $C_{j,Qiu}$ in
comparison with [4], only the candidate knots $\tau_i$ that coincide with the knots $\tau_{Qiu,i}$ of Qiu et al. should be allowed to be active. That is, all other candidate knots $\tau_i$ need to be “deactivated” beforehand by preventing a jump of the puff $\theta^{(5)}(\tau)$: 

$$\theta^{(5)}(\tau_i) = \theta^{(5)}(\tau_{i+1})$$

must hold for all $i \in I$, where set $I$ collects the $g-N+1$ indices $i (1 \leq i \leq g)$ of the spline knots that should be deactivated.

(iii) The added value of our framework is that the convexity of design problem (5) is both recognized and exploited to derive an equivalent formulation that is more easy to solve numerically. In fact, using insights and techniques similar to those used in the companion paper [2], design problem (5) can be reformulated as a linear program. The key to this reformulation is that every constraint of the form $|\theta^{(q)}(\tau_i)| \leq C$ is equivalent to two linear inequalities: $-C \leq \theta^{(q)}(\tau_i) \leq C$.

By combining steps (i), (ii) and (iii), our framework translates (5) into the following LP:

\[
\begin{align*}
\text{minimize}_{(x,w)} & \quad w \\
\text{subject to} & \quad -C_{v,Qiu} \leq \theta^{(1)}(\tau_i) \leq C_{v,Qiu}, & i = 0 \ldots g+1 \\
& \quad -w \leq \theta^{(2)}(\tau_i) \leq w, & i = 0 \ldots g+1 \\
& \quad -C_{j,Qiu} \leq \theta^{(3)}(\tau_i) \leq C_{j,Qiu}, & i = 0 \ldots g+1 \\
& \quad \theta(0.0) = 0.0; \theta^{(1)}(0.0) = 0.0; \theta^{(2)}(0.0) = 0.0 \\
& \quad \theta(0.5) = 0.5; \theta^{(2)}(0.5) = 0.0 \\
& \quad \theta^{(5)}(\tau_i) = \theta^{(5)}(\tau_{i+1}), & i \in I, \\
\end{align*}
\]

where $w$ is an auxiliary scalar variable to circumvent the absolute value in the objective function (5a). Recall that obtaining the detailed formulation of this optimization problem in terms of $x$, requires substitution (4) and addition of constraints (3).

Solving the LP (9) requires less than one CPU second and results in peak values $C_{v,LP} = 2.0000$ and $C_{j,LP} = 61.5374$ that comply with the constraints. A marginal improvement in peak acceleration is observed: $C_{a,LP} = 4.8563$ vs. $C_{a,Qiu} = 4.8568$. In addition, the resulting spline nearly coincides with the result shown in Figs 1 and 2, and hence, we conclude that our convex programming framework functions properly.

### 2.3 Improvement of Benchmark Results

Compared to Qiu et al., the added value of our framework stems from property (iii) mentioned in the previous subsection: as standard LP solvers are both fast and reliable, even for very large-scale problems, the LP reformulation (9) offers the flexibility to further extend and improve the results of Qiu et al. [4].

To illustrate this, we use our linear programming framework to investigate the relation between the peak values of acceleration and puff. In order to keep the results comparable with the results of Qiu et al., the uniform spline is sought for that has minimum peak puff subject to the constraints that the peak jerk be below $C_{j,Qiu}$, the peak velocity be below $C_{v,Qiu}$ and the
peak acceleration be below \((1 + \varepsilon) \cdot C_{a,Qiu}\). That is, the following LP is solved for various values of \(\varepsilon\):

\[
\begin{align*}
\text{minimize}_{(t,w)} & \quad w \\
\text{subject to} & \quad -C_{a,Qiu} \leq \theta^{(1)}(\tau_i) \leq C_{a,Qiu}, \quad i = 0 \ldots g + 1 \\
& \quad - (1 + \varepsilon) \cdot C_{a,Qiu} \leq \theta^{(2)}(\tau_i) \leq (1 + \varepsilon) \cdot C_{a,Qiu}, \quad i = 0 \ldots g + 1 \\
& \quad -C_{j,Qiu} \leq \theta^{(3)}(\tau_i) \leq C_{j,Qiu}, \quad i = 0 \ldots g + 1 \\
& \quad -w \leq \theta^{(5)}(\tau_i) \leq w, \quad i = 1 \ldots g + 1 \\
& \quad \theta(0.0) = 0.0; \quad \theta^{(1)}(0.0) = 0.0; \quad \theta^{(2)}(0.0) = 0.0 \\
& \quad \theta(0.5) = 0.5; \quad \theta^{(2)}(0.5) = 0.0 \\
& \quad \theta^{(5)}(\tau_i) = \theta^{(5)}(\tau_{i+1}), \quad i \in I, \\
\end{align*}
\]

where \(\tau_i\) is defined by (8). Solving this LP for \(\varepsilon = 0\) yields the results of Qiu et al., while the dashed line in Fig. 3 indicates the peak puff \(C_p\) and peak acceleration \(C_a = (1 + \varepsilon) \cdot C_{a,Qiu}\) if the LP (10) is solved for various values of \(\varepsilon\) (that is: each point on the dashed curve corresponds to the optimal solution of (10) for a certain \(\varepsilon\) value). The resulting curve stops at \(C_a = 4.857\) (labeled ‘1’), confirming that \(C_{a,Qiu}\) is the lowest possible acceleration value for \(N = 7\).

If, however, the set of constraints (10h) is dropped, any of the 351 time instants \(\tau_i\) (instead of just 8 of them) is allowed to be an active knot. By solving the corresponding optimization problem for various values of \(\varepsilon\), the solid line in Fig. 3 is constructed. In the terminology of Qiu et al., by dropping (10h), \(N\) increases from 7 to 350. In this way, the claim of Qiu et al., that “further improvement cannot be expected by increasing \(N\)” can be verified. The solid line in Fig. 3 shows that this claim is not justified: by increasing \(N\) from 7 to 350, it is possible to reduce \(C_a\) by 2.80% from \(C_{a,Qiu} = 4.8568\) to \(C_a = 4.7208\) at the cost, however, of a peak puff \(C_p\) that is almost four orders of magnitude higher.

More realistic results are indicated by the points 2 and 3 in Fig. 3. Comparison of point 2 with the solution of Qiu et al., point 1, indicates that increasing \(N\) from 7 to 350 allows us to obtain a trajectory with the same peak acceleration \(C_{a,Qiu}\), but a 40% lower peak puff \(C_p\) (13580) (Table 1). The point 3, on the other hand, shows that, compared to the solution of Qiu et al., for nearly the same level (22453) of peak puff \(C_p\), the peak acceleration \(C_a\) can be decreased by 1.07% to 4.805 (Table 1).

While the peak values \(C_a\) and \(C_p\) are significantly better than those of Qiu et al., the solutions 2 and 3 cannot be accepted ‘as is’ due to a large number (70-100) of active knots, as revealed by the frequent discontinuities in the puff trajectory, shown in Fig. 2(b). In order to overcome this problem, one-norm regularization, as discussed in [2], is required. By allowing a marginal 1% increase of \(C_p\), this process results in significantly better solutions, as illustrated by Fig. 4: the number of active internal knots is now limited to 9 (solution 2) or 11 (solution 3). For solutions 2 and 3, the major puff jumps occur at similar knot locations as for the benchmark solution 1. Furthermore, some smaller puff jumps (some of which hardly visible) occur in the middle of the \([0.0,0.50]\) interval. The corresponding peak values of the various derivatives are tabulated between brackets in Table 1 and differ hardly from the nonregularized values for \(q = 1,2,3\).
Table 1. Peak absolute values of the derivatives $\theta^{(q)}(\tau)$ of the various solutions 1 [4], 2 and 3 displayed in Fig. 3. Numbers between brackets indicate values after regularization.

<table>
<thead>
<tr>
<th></th>
<th>$q = 1 (C_v)$</th>
<th>$q = 2 (C_a)$</th>
<th>$q = 3 (C_j)$</th>
<th>$q = 4$</th>
<th>$q = 5 (C_p)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.000</td>
<td>4.857</td>
<td>61.54</td>
<td>1042.4</td>
<td>22453</td>
</tr>
<tr>
<td>2</td>
<td>2.000 (1.999)</td>
<td>4.857 (4.857)</td>
<td>61.54 (61.54)</td>
<td>922.0 (925.8)</td>
<td>13580 (13716)</td>
</tr>
<tr>
<td>3</td>
<td>2.000 (2.000)</td>
<td>4.805 (4.805)</td>
<td>61.54 (61.54)</td>
<td>1180.5 (1185.1)</td>
<td>22444 (22668)</td>
</tr>
</tbody>
</table>

Fig. 3. Trade-off between peak value $C_p$ of puff and acceleration $C_a$ for quintic splines with peak velocity $C_v \leq C_v^{Qiu}$ and peak jerk $C_j \leq C_j^{Qiu}$ and complying with the (incomplete) boundary constraints (5d)-(5e). Results for $N = 7$ (dashed line) and $N = 350$ (solid line). The design of Qiu et al. [4] is indicated by the cross labeled 1. Table 1 summarizes the properties of the competing designs 1, 2 and 3.
Fig. 4. Trajectories of ping $\theta^{(4)}(\tau)$ and puff $\theta^{(5)}(\tau)$ for solution 1 indicated in Fig. 3 (Qiu et al., solid line) and the solutions obtained through regularization of the solutions 2 (dashed line) and 3 (dash-dotted line). The thick vertical line at $\tau = 0.5$ indicates the infinite puff-value at this location.
2.4 Conclusion

The numerical results presented in this section clearly illustrate the importance of each of the three main framework characteristics: (i) Considering a large number of candidate knots: The results [4] can be improved by considering 350 instead of 7 knots, which contradicts the observation of Qiu et al. that “further improvement cannot be expected” by increasing the number of knots. The increased number of knots yields improvements that range from small (the absolute 2.80% decrease of the peak acceleration $C_a$ or the 1.07% $C_a$ decrease for identical peak puff) to significant (the 40% lower peak puff for identical $C_a$). (ii) One-norm regularization: The improved results would be meaningless if the great majority of these 350 internal knots were active, but they are not: through regularization, it is possible to find solutions that have a compact polynomial spline representation based on about ten nonuniformly spaced knots between 0 and 0.50. (iii) Formulation as a convex program: Qiu et al. solve their kinematic optimization problem as a general nonlinear program, while it is shown here that it can be formulated as an LP. The LP formulation is important in the presence of a large number of knots for such a large number is highly likely to be troublesome (long computational times, convergence problems) for the original nonlinear formulation [4].

---

7Note, however, that these values might be judged “large” compared to the 0.39% improvement with respect to the modified trapezoidal law that was reported by Qiu et al.
Example 2: Benchmark Lambrechts et al. (2005)

Recently, Lambrechts et al. [9] proposed an analytical method for designing ('planning') time optimal double-dwell motions for motion systems driven by servomotors. As explained in Sec. 1.2, this method was selected as an analytical benchmark example to verify the spline knot optimization capabilities of the convex programming framework.

3.1 Results of Lambrechts et al. (2005)

The overall aim of the study [9] is designing time optimal trajectories (that is, with minimal trajectory duration $T$) that do not overload the driving servomotor. Given user-specified values of the spline degree $k$ and lift $L$ [m] or [rad], the method generates polynomial splines of degree $k$ that are exact or approximate solutions of the following optimization problem:

$$
\begin{align*}
\text{minimize}_{s(t), T} & \quad T \\
\text{subject to} & \quad s(0.0) = 0.0; s(T) = L \\
& \quad s^{(m)}(0.0) = 0.0; s^{(m)}(T) = 0.0, \quad m = 1 \ldots k - 1 \\
& \quad -C_m \leq s^{(m)}(t) \leq C_m, \quad t \in [0, T], \quad m = 1 \ldots k
\end{align*}
$$

where $s(t) \in \mathcal{C}^{k-1}_{[0,T]}$ (that is: its spline knot locations $t_i$ and spline coefficients $d_j$) and the trajectory duration $T$ are the optimization variables. $C_m$ [m/s$^m$] or [rad/s$^m$] denotes a user-specified bound on the peak absolute value of $s^{(m)}(t)$. As opposed to the case study presented in Sec. 4, motor overloading is not prevented directly through bounds on the motor torque, but rather indirectly by imposing upper and lower limits (11d) on the higher derivatives of $s(t)$.

The contribution of [9] is the extension of known analytical methods for time optimal quadratic ($k = 2$) and cubic splines ($k = 3$) to quartic splines ($k = 4$). This development is driven by the observation that in typical servomotor driven systems 'fourth-order feedforward' significantly improves performance in terms of residual vibration. Two analytical algorithms are provided, a continuous version, in which the spline knots can occur at any time instant $t \in [0,T]$ and a discretized version, in which spline knots can only occur at integer multiples of the sample period $T_s$. This synchronization is important for the implementation of the planned trajectory into a digital feedforward control system.

Lambrechts et al. indicate that (i) while their method is very simple, it does not guarantee time optimality for $k = 4$ (although often a very good, approximate solution of (11) is found) and (ii) time optimality for $k = 4$ could be obtained but at the cost of a much more complex algorithm compared to the small gain with respect to a good sub-optimal solution.

3.2 Improvement of Benchmark Results

Our aim is to use the convex programming framework for solving (11) with $k = 4$ in order to reproduce, improve and generalize the analytical benchmark results by Lambrechts et al. A complication is, however, that this framework does not

---

8Feedforward computed from a trajectory that is continuous up to the jerk, as is a quartic spline.
allow direct minimization of \( T \) for it requires \( T \) to be a given, fixed value. A workaround is provided by \emph{bisection}, which solves (11) by solving, for a sequence of \emph{fixed} values of \( T \), the following special optimization problem in the optimization variable \( x \):

\[
\begin{align*}
\text{minimize} \quad & x \\
\text{subject to} \quad & s(0.0) = 0.0; \quad s(T) = L \quad (12b) \\
& s^{(m)}(0.0) = 0.0; \quad s^{(m)}(T) = 0.0, \quad m = 1 \ldots k - 1 \\
& -C_m \leq s^{(m)}(t_i) \leq C_m, \quad i = 0 \ldots g, \quad m = 1 \ldots k \\
\end{align*}
\]

where \( t_i = i \cdot T/(g + 1) \) and \( g \) denotes the number of internal knots provided to the optimization algorithm. Since the objective (12a) is identically zero, the optimal value of this \emph{feasibility problem}\(^9\) is either zero or \( \infty \), depending on whether a solution exists, or not, for the (linear) set of equations (12b)--(12d). Bisection solves the minimization problem (11) by solving a sequence of instances (corresponding to various values of \( T \)) of the feasibility problem (12) according to the following algorithm (see, e.g., [12]), where \([T_l, T_u]\) is a user-specified interval known to contain the optimal time \( T^* \) and \( \varepsilon \) [s] a tolerance on the calculation of \( T^* \):

\begin{itemize}
  \item \textbf{given} \( T_l \leq T^*, \quad T_u \geq T^*, \) tolerance \( \varepsilon > 0 \).
  \item \textbf{repeat} the following steps
    \begin{enumerate}
      \item \( T_{\text{trial}} := (T_l + T_u)/2 \)
      \item Solve the linear feasibility problem (12) for \( T = T_{\text{trial}} \).
      \item if (12) is feasible, \( T_u := T_{\text{trial}} \); else \( T_l := T_{\text{trial}} \).
    \end{enumerate}
  \item \textbf{until} \( T_u - T_l \leq \varepsilon \)
\end{itemize}

The interval \([T_l, T_u]\) is guaranteed to contain \( T^* \) in each step. In each iteration the interval is bisected (divided in two) so the length of the interval after \( v \) iterations is \( 2^{-v}(T_u - T_l) \). It follows that exactly

\[
\left\lceil \log_2 \left( \frac{(T_u - T_l)}{\varepsilon} \right) \right\rceil
\]

iterations are required\(^{10}\) before the algorithm terminates. Each step involves solving the feasibility problem (12), which is a linear program and can therefore be solved extremely fast.

As a numerical example, the analytical approach [9] and our approach are compared for two cases (each with \( k = 4 \)), of which the numerical constraint data are provided in Table 2. The bisection algorithm is run with \( g = 500, \ T_l = 0.5 \text{ s}, \ T_u = 2.5 \text{ s}; \ \varepsilon = 1 \text{e-}4 \text{ s}, \) such that 15 instances of the feasibility problem (12) need to be solved for solving (11). This requires a total CPU time of 16 s for case 1 and 6 s for case 2.

\footnote{by convention, infinity is returned as the optimal value if the problem is infeasible, that is, if no optimization variable can be found (exists) that satisfies all equality and inequality constraints [12].}

\footnote{\( \lceil \cdot \rceil \) denotes the ceil operator, that is, rounding to the nearest larger integer.}
<table>
<thead>
<tr>
<th>$L$ [m]</th>
<th>$C_1$ [m/s]</th>
<th>$C_2$ [m/s$^2$]</th>
<th>$C_3$ [m/s$^3$]</th>
<th>$C_4$ [m/s$^4$]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.00</td>
<td>1.50</td>
<td>5.00</td>
<td>50.0</td>
</tr>
<tr>
<td>2</td>
<td>1.00</td>
<td>3.00</td>
<td>15.00</td>
<td>100.0</td>
</tr>
</tbody>
</table>

Table 2. Numerical data for solving (11).

<table>
<thead>
<tr>
<th>$T_{c,Lam}$ [s]</th>
<th>$T_{d,Lam}$ [s]</th>
<th>$T_{bisec}$ [s]</th>
<th>$\delta_d$ [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.1167</td>
<td>1.1211</td>
<td>1.1166</td>
</tr>
<tr>
<td>2</td>
<td>0.8465</td>
<td>0.8516</td>
<td>0.8035</td>
</tr>
</tbody>
</table>

Table 3. Solutions for (11) (numerical data of Table 2) obtained by Lambrechts et al. (continuous: $T_{c,Lam}$ and discrete $T_{d,Lam}$) and our bisection approach ($T_{bisec}$) as well as the relative improvement $\delta_d$ [%] defined by (13).

The analytical solution is generated with the Matlab software made available\(^ {11}\) by Lambrechts. In determining the discrete solution, the sampling period is set to $T_{c,Lam}/(g + 1)$. $T_{c,Lam}$ and $T_{d,Lam}$ [s] denote the minimal time required by the continuous and discrete solution respectively [9].

In the first case, both approaches yield similar results, where the bisection result is marginally better. Defining the relative improvement $\delta_d$ [%] as

$$\delta_d = 100 \cdot \frac{T_{d,Lam} - T_{bisec}}{T_{d,Lam}}, \quad (13)$$

bisection yields a solution that is 0.40% faster (see Table 3)\(^ {12}\). Fig. 5 shows that this marginal improvement comes at the cost of a slightly more complicated trajectory, mainly revealed as the additional jerk and ping ‘bumps’ in the time interval $[0.46, 0.66]$ and two ping ‘spikes’ around $t = 0.15$ and $t = 0.97$. These results confirm the claim of Lambrechts et al. that their simple algorithm yields very good sub-optimal solutions.

---

\(^{11}\)This software can be downloaded from the Matlab Central File Exchange.

\(^{12}\)In the definition of $\delta_d$, $T_{bisec}$ is compared $T_{d,Lam}$, since the discrete solution is the one actually implemented in the control system.
A more pronounced difference, however, results if the bounds on velocity and acceleration (Table 2, case 2) are relaxed such that they are not active in the resulting solution. In that case, $\delta_d$ not only increases to 5.65% (see Table 3), but moreover, significantly different trajectories are obtained, as revealed by Fig. 6, because of the analytical solution’s unnecessary return to zero jerk halfway the cycle. This numerical example confirms the conjecture by Lambrechts et al. that the analytical algorithm only yields time optimality provided that a constant-velocity phase is present.
3.3 Conclusion

The main advantage of the approach [9] is its simplicity: no numerical optimization is required, only simple arithmetic, which makes it extremely suited for online implementation in a computer-controlled motion system. The fact that our bisection approach is able to reproduce or improve these intuitive results, confirms its power and versatility. Although it might be argued that the added computational complexity is not in proportion to the gain obtained\(^\text{13}\), such marginal improvements may nevertheless be worth the (limited) additional computational time, for instance in mass production environments where every millisecond reduction in cycle time counts.

Moreover, the true power of the bisection framework is in its versatility: (i) the symmetry requirement of [9] can be dropped, and (ii) the spline degree is not limited to four but can be arbitrarily high\(^\text{14}\).

---

\(^{13}\)depending on the problem data, the results of Lambrechts et al. are improved by up to 6%

\(^{14}\)Lambrechts et al. remark that, in principle, higher than fourth-order trajectories can be planned by means of their algorithm, but this is considered impractical due to the large increase in complexity.
4 Example 3: Dynamics and Time-Energy Optimality

This section illustrates that the motor torque can be directly constrained (instead of indirectly constraining the peak derivatives), and more generally, any nonlinear constraint or objective function that is convex in $\theta(\tau)$ and its derivatives can be considered.

The aim of the case study is to compute optimal double-dwell motions for a servomotor driven motion system that pushes boxes forward on a horizontal plane. The following constraints govern this system: lower and upper motor torque limits $M_{\text{min}}$ and $M_{\text{max}}$ [N-m] and an upper box speed limit $v_{\text{max}}$ [m/s]. While in [9] motor torque constraints are indirectly handled through bounded derivatives of $s(t)$, here the convex programming framework is extended to express motor torque constraints directly based on the system model provided in Sec. 4.1.

First, time optimal motions are considered, for which the time optimal solution is analytically known to be a spline that is not polynomial (Sec. 4.2). Considering time optimality therefore allows investigating the ability of the convex programming framework to approximate such nonpolynomial motions (Sec. 4.3).

Second, observing that time optimality often comes at significant energy cost, Sec. 4.4 investigates the trade-off between time and energy optimality. To this end, a second extension of the framework is introduced: a convex quadratic energy objective function. The corresponding discussion and numerical results complement the more qualitative discussion on convex quadratic programs provided in [2].

4.1 System Model

The boxes are pushed by a rod that is driven by a rotary servomotor through a pin/rack (transmission ratio $r$ [rad/m]). The maximum motor torque is $M_{\text{max}}$ [N-m], while guaranteeing a minimum contact force between the boxes and the rod is translated to a constraint that the motor torque always be above $M_{\text{min}} > 0$ [N-m].

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L$ [m]</td>
<td>0.5475</td>
</tr>
<tr>
<td>$M_{\text{max}}$ [N-m]</td>
<td>20</td>
</tr>
<tr>
<td>$M_{\text{min}}$ [N-m]</td>
<td>1</td>
</tr>
<tr>
<td>$v_{\text{max}}$ [m/s]</td>
<td>0.416666</td>
</tr>
<tr>
<td>$M_c$ [N-m]</td>
<td>2</td>
</tr>
<tr>
<td>$c$ [N-m-s/rad]</td>
<td>0.025</td>
</tr>
<tr>
<td>$I$ [kg-m$^2$]</td>
<td>0.008</td>
</tr>
<tr>
<td>$r$ [rad/m]</td>
<td>$200\pi$</td>
</tr>
</tbody>
</table>

Table 4. Numerical parameters defining problem (15).
The system is considered to be completely rigid, while all mass, inertia and friction are reduced to the motor shaft: the motor
'sees' an equivalent inertia \( I \) [kg-m^2], an equivalent viscous friction coefficient \( c \) [N-m-s/rad] and an equivalent Coulomb friction torque \( M_c \). Denoting the speed and acceleration of the boxes by \( \dot{s}(t) \) [m/s] and \( \ddot{s}(t) \) [m/s^2], respectively, the following first-order linear differential equation in \( \dot{s}(t) \) results:

\[
M_m(t) = I \cdot r \cdot \ddot{s}(t) + c \cdot r \cdot \dot{s}(t) + M_c,
\]

which is only valid provided that the motor torque \( M_m \) is nonnegative (which is in turn guaranteed by the nonnegative lower limit \( M_{\text{min}} \)).

### 4.2 Time Optimality: Analytical Solution

If it is imposed that \( s(t) \in C^1_{[0,T]} \), the following optimization problem in the optimization variables \( T \) and \( s(t) \) needs to be solved to compute the time optimal motion:

\[
\begin{align*}
\text{minimize}_{(s(t), T)} & \quad T \\
\text{subject to} & \quad s(0.0) = 0.0; \ s(T) = L \\
& \quad \dot{s}(0.0) = 0.0; \ \dot{s}(T) = 0.0 \\
& \quad 0 \leq \dot{s}(t) \leq v_{\text{max}}, \quad t \in [0, T] \\
& \quad M_{\text{min}} \leq I \cdot r \cdot \ddot{s}(t) + c \cdot r \cdot \dot{s}(t) + M_c \leq M_{\text{max}}, \quad t \in [0, T]
\end{align*}
\]

where the upper and lower bound on the motor torque (14) are directly expressed through the inequality constraint (15e), which is linear in \( \dot{s} \) and \( \ddot{s} \).

\( s(t) \) is required to belong to \( C^1_{[0,T]} \), since (15) has a simple analytical solution in that case. This solution consists of three stages (Fig. 7): first, the maximal motor torque \( M_{\text{max}} \) is applied to accelerate the boxes as fast as possible to their maximal speed \( v_{\text{max}} \). Second, the boxes move as long as possible at \( v_{\text{max}} \), the corresponding motor torque being \( c \cdot r \cdot v_{\text{max}} + M_c \). Finally, the minimal motor torque \( M_{\text{min}} \) is applied to decelerate the boxes from \( v = v_{\text{max}} \) to standstill as fast as possible.

Using the numerical parameters of Table 4, Appendix A shows that \( T^* = 1.8064 \) s are required for this time optimal solution. The three different stages imply that the time optimal solution is a spline with two internal knots (\( t_1^* = 0.1446; t_2^* = 1.1597 \)), where \( s(t) \) is given by the exponential trajectories (18)–(21) for the first stage (\( t \in [0,t_1^*] \)) and the third stage (\( t \in [t_2^*, T^*] \)). Consequently, the overall trajectory \( s(t) \) is a nonpolynomial spline.
4.3 Time Optimality: Numerical Solution

As in Ex. 2 bisection is required to solve (15). In every bisection step, the following feasibility problem is solved:

\[
\begin{align*}
\text{minimize} & \quad 0 & \quad (16a) \\
\text{subject to} & \\
T(0.0) &= L & (16b) \\
\dot{T}(0.0) &= 0 & (16c) \\
0 &\leq \dot{T}(t_i) \leq v_{\text{max}}, & i = 0 \ldots g & (16d) \\
M_{\min} &\leq I \cdot \ddot{T}(t_i) + c \cdot r \cdot \dot{T}(t_i) + M_c \leq M_{\max}, & i = 0 \ldots g & (16e)
\end{align*}
\]

where \( t_i = i \cdot T/(g+1) \) and \( g \) denotes the number of internal knots. This is a linear program and can hence be solved very efficiently. The bisection algorithm uses the following numerical values: \( T_l = 0.4 \) s, \( T_u = 2.4 \) s; \( \varepsilon = 1 \times 10^{-5} \) s.

In order to verify the accuracy with which the analytical result can be reproduced, the bisection is repeated for 50 logarithmically spaced values of \( g \), ranging between 32 and 7000, similarly as in the benchmark example discussed in Sec. 3.6 of the companion paper [2]. Figure 8(a) displays the relative error

\[
\varepsilon_r(g) = \frac{T_b^*(g) - T_a^*}{T_a^*}
\]

between the analytical solution \( T_a^* \) and the solution \( T_b^*(g) \) determined by bisection using \( g \) internal knots. This plot confirms what can be intuitively anticipated, that is, \( \varepsilon_r = O(1/g) \).\(^{15}\) The computational time for the whole bisection process, on the other hand, is \( O(g^{1.4}) \) in this particular case,\(^{16}\) and thus grows moderately as a function of \( g \). For instance, \( g = 1200 \) gives rise to \( T_b^* = 1.8075 \) s, that is, a mere relative error \( \varepsilon_r = 6.5 \times 10^{-4} \) (7 s of CPU time are required). Moreover, the corresponding trajectories of \( s(t) \) and its derivatives are nearly identical to those of the analytical solution.

Based on these observations, it is concluded that the convex programming framework is able to approximate nonpolynomial splines arbitrarily closely. Otherwise stated: the convex programming framework can yield an arbitrarily close polynomial spline approximation of a nonpolynomial spline. This approximation is not compact in the sense that in almost every knot in \([0,t_1^*]\) and \([t_1^*,T_u]\) a (small) jerk jump will occur, while bigger\(^{17}\) jumps will occur in the two knots that are closest to \( t_1^* \) and \( t_2^* \). Hence, it is not possible to accurately and compactly represent the result as a nonuniform polynomial spline with a small number of non-equidistantly spaced knots. Note that one-norm regularization is not applicable here, since the noncompactness is not due to the existence of several equivalent solutions, as in Ex. 1, but rather follows from the fact that the optimal solution is just not a polynomial spline.

\(^{15}\)The slope of the regression line in Fig.8(a) is minus one.

\(^{16}\)The slope of the regression line in Fig.8(\(b\)) is 1.4.

\(^{17}\)In this particular case, two orders of magnitude bigger.
Fig. 7. Analytical solution of (15): $s(t)$ and its first two derivatives, as well as corresponding motor torque $M_m(t)$. The horizontal dashed lines in Figs. (b), (d) indicate the upper and lower limits expressed by (15d) and (15e), respectively.

Fig. 8. (a) Relative error $\varepsilon_t(g)$ between the analytical solution $T_a^*$ of (15) and the solution $T_b^*$ determined by bisection using $g$ internal knots. (b) Computational time required for the whole bisection process as a function of $g$. The solid lines mark linear regression lines.
4.4 Time-Energy Optimality

If it is assumed that the motor current \( I_m \) is proportional to the motor torque \( M_m \), minimal energy consumption \( \int_0^T \dot{I}_m^2 \, dt \) of the driving motor is obtained if \( \int_0^T M_m^2 \, dt \) is minimized. Hence, time and energy optimality can be traded off by replacing the objective (16a) of the feasibility problem (16) by

\[
E = \sum_{i=1}^{g+1} (I \cdot r \cdot \dot{s}(t_i) + c \cdot r \cdot \ddot{s}(t_i) + M_c) \tag{17}
\]

An important consequence of using the objective function (17) is that the linear program turns into a convex quadratic program (QP). The convexity follows from basic properties of convex functions\(^\text{18}\) [12], and similar to LPs, convex QPs can be solved extremely efficiently using dedicated solvers.

For the problem data of Table 4, the trade-off between time and energy optimality is investigated by solving this optimization problem for a range of \( T \)-values between the previously determined time optimal value \( T_a^* = 1.8064 \) and \( T = 7.0 \) s, in increments of 0.02 s. Solving these 260 instances problem instances with \( g = 256 \) takes about 17 s (0.065 s on the average), thereby illustrating the efficiency with which the problem can be solved.

The solid line in Fig. 9 shows, as a function of \( T/T_a^* \), the relative minimal value of the objective (17), that is, \( E_{\text{min}}(T)/E_{\text{min}}(T_a^*) \). This line features a sharp decrease for \( T/T_a^* \) around 1. Three specific cases are considered in Fig. 9 and Table 5. **Case 1** is the time optimal solution, that is, the solution corresponding to \( T_a^* = 1.8064 \). **Case 2** is the solution corresponding to \( T/T_a^* = 1.10 \) (that is, if \( T \) is allowed to be 10% bigger than its minimum \( T_a^* \)), with a required energy reduction which is twice as big (21%). **Case 3** is the solution yielding an equal 34% decrease in energy consumption and increase in time \( T/T_a^* = 1.34 \).

The trajectories of \( s(t) \), its first two derivatives and the corresponding motor torque are displayed in Fig.10 for the three cases tabulated in Table 5 and discussed above. Allowing more time results in much less aggressive an actuator use, as can be seen from the inactive upper torque limit in cases 2–3. In case 3, even the upper speed limit is no longer active. While the time optimal trajectory was easy to find analytically, such analytic solutions do not exist for the more energy optimal cases 2-3, thereby illustrating the practical relevance of the convex programming framework.

The solid line in Fig. 9 features a minimum around \( T/T_a^* = 2.75 \), since two counteracting effects are present if \( T \) increases: the energy consumption due to the inertia and viscous friction torque decreases, while the energy consumption due to the Coulomb friction torque \( M_c \) increases linearly with \( T \). The relative root-mean-square motor torque \( M_{\text{rms}}(T)/M_{\text{rms}}(T_a^*) \), where \( M_{\text{rms}}(T) = \sqrt{E_{\text{min}}(T)/T} \), (dashed line in Fig. 9), on the other hand, decreases monotonically for the contribution due to the inertia and viscous friction torque decreases, while the Coulomb friction contribution is constant.

---

\(^{18}\) Since \( f(y) = y^2 \) is a convex function, (17) is convex in \( M_m(t_i) \) \( (M_m(t_i) = I \cdot r \cdot \dot{s}(t_i) + c \cdot r \cdot \ddot{s}(t_i) + M_c) \): \( E = \sum_{i=1}^{g+1} M_m(t_i)^2 \). Since composition with an affine mapping preserves the convexity of a function, \( E \) is convex in \( x \), where the affine dependency of \( M_m(t_i) \) on \( x \) follows from the affine dependency of \( \dot{s}(t_i) \) and \( \ddot{s}(t_i) \) on \( x \) (dimensional counterpart of Eq. (4)).
Table 5. Required time $T$ as well as kinematic and dynamic properties of time optimal (case 1) and time-energy optimal trajectories (cases 2–3) corresponding to the numerical data of Table 4.

<table>
<thead>
<tr>
<th>Case</th>
<th>$T$ [s]</th>
<th>$E$ [N-m]</th>
<th>$\max(\dot{s})$ [m/s]</th>
<th>$\min(\dot{s})$ [m/s]</th>
<th>$\max(\ddot{s})$ [m/s$^2$]</th>
<th>$\min(M_m)$ [N-m]</th>
<th>$\max(M_m)$ [N-m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.8064</td>
<td>133.6</td>
<td>0.417</td>
<td>-1.501</td>
<td>3.581</td>
<td>1</td>
<td>20.00</td>
</tr>
<tr>
<td>2</td>
<td>1.9870</td>
<td>105.0</td>
<td>0.417</td>
<td>-0.872</td>
<td>1.575</td>
<td>1</td>
<td>10.11</td>
</tr>
<tr>
<td>3</td>
<td>2.4151</td>
<td>88.6</td>
<td>0.322</td>
<td>-0.624</td>
<td>1.047</td>
<td>1</td>
<td>7.42</td>
</tr>
</tbody>
</table>

Fig. 9. Time-energy trade-off. $E_{\min}(T)/E_{\min}(T^*)$ (solid line) and $M_{\text{rms}}(T)/M_{\text{rms}}(T^*)$ (dashed line) as a function of $T/T^*$. Cases 1, 2, and 3 defined in section 4.4 and presented in Table 5 are indicated.

4.5 Conclusion

The framework is extended for servomotor driven systems with (i) direct expression of upper and lower bounds on motor torque and (ii) a convex quadratic energy objective function. First, the considered time optimal control problem with an analytical solution reveals that also for nonpolynomial splines, the framework has excellent spline knot optimization capabilities. Second, very intuitive numerical results are obtained for the considered time-energy optimal control problem which has no analytical solution. This example also complements the companion paper’s qualitative discussion on convex objectives and constraints (Sec. 5 in [2]) through a more detailed discussion about convex quadratic programs as well as numerical results.
Fig. 10. $s(t)$ and its first two derivatives, as well as corresponding motor torque $M_m(t)$ of time optimal (case 1, solid line) and time-energy optimal trajectories (case 2, dash-dotted line; case 3, dashed line) corresponding to the numerical data of Table 4. Table 5 summarizes the main properties. The endpoint of the trajectories is marked by a circle (case 1), diamond (case 2) or asterisk (case 3). The horizontal dashed lines in Figs. (b), (d) indicate the upper and lower limits expressed by (16d) and (16e), respectively.

5 Conclusion

Based on three case studies of increasing complexity, the power and versatility of the convex programming framework for optimal spline synthesis have been demonstrated, while several generic extensions were developed: (i) bisection to generate time optimal motions, (ii) direct expression of upper and lower bounds on motor torque and (iii) a convex quadratic energy objective function for servomotor driven systems.

Benchmarking with the numerical optimization study Qiu et al. [4] proved the importance of each of the three main framework characteristics: (i) Considering a large number of candidate knots significantly improves the benchmark results; (ii) formulation as a convex program is the key to efficiently and reliably solving the corresponding large-scale optimization problem and (iii) meaningful results were only obtained after one-norm regularization.

Benchmarking with the analytical study Lambrechts et al. [9] confirmed the potential of our indirect approach for finding optimal spline knot locations. Moreover, our framework is able to numerically improve these results, while also allowing for more general problem formulations.

Finally, the third example revealed (i) the possibilities to include dynamic and energy considerations and (ii) the framework’s capability to approximate optimal nonpolynomial spline trajectories arbitrarily closely. The presented numerical results furthermore demonstrate that moving from linear to nonlinear convex programs does not compromise computational
efficiency.

While this paper was devoted exclusively to rigid motion systems, current research focuses on further extending and experimentally validating the preliminary results [13] concerning flexible motion systems.

Acknowledgment

Bram Demeulenaere was a Postdoctoral Fellow of the Research Foundation - Flanders (FWO - Vlaanderen) until 2008 and has carried out part of 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


A Time Optimal Control Problem: Analytical Solution

This appendix discusses the analytical solution of (15) for \( s(t) \in C^1[0,T] \). The three stages, identified in Sec. 4.2, are solved in the following order: stage 1 – stage 3 – stage 2, using the shorthand notation \( A = \frac{1}{c_r} \) and \( t_c = \frac{1}{c} \).

First, \( M_{\text{max}} \) is applied to accelerate the boxes as fast as possible to their maximal speed \( v_{\text{max}} \). Equation (14) thus gives rise to the following first-order differential equation in \( \dot{s} \)

\[
I \cdot r \cdot \ddot{s} + c \cdot r \cdot \dot{s} = M_{\text{max}} - M_c
\]

with \( \dot{s}(0) = 0 \) as initial condition. The position and velocity trajectory of the corresponding solution are given by

\[
s_1(t) = A \cdot (M_{\text{max}} - M_c) \cdot t + t_c \cdot A \cdot (M_{\text{max}} - M_c) \left( \exp \left( -\frac{t}{t_c} \right) - 1 \right)
\]

(18)

\[
\dot{s}_1(t) = A \cdot (M_{\text{max}} - M_c) - A \cdot (M_{\text{max}} - M_c) \cdot \exp \left( -\frac{t}{t_c} \right)
\]

(19)

Given the numerical values of Table 4, this implies that a time \( t^*_1 = 0.1446 \) s is needed to reach \( v_{\text{max}} \). The distance traveled in this period of time is \( L_1 = 0.0324 \) m.

Second \( M_{\text{min}} \) is applied to decelerate the boxes as fast as possible from \( v_{\text{max}} \) to 0. Equation (14) thus gives rise to the following first-order differential equation in \( \dot{s} \)

\[
I \cdot r \cdot \ddot{s} + c \cdot r \cdot \dot{s} = M_{\text{min}} - M_c
\]

with \( \dot{s}(0) = v_{\text{max}} \) as initial condition. The position and velocity trajectory of the corresponding solution are given by

\[
s_3(t) = A \cdot (M_{\text{min}} - M_c) \cdot t - t_c \cdot (v_{\text{max}} - A \cdot (M_{\text{min}} - M_c)) \left( \exp \left( -\frac{t}{t_c} \right) - 1 \right)
\]

(20)

\[
\dot{s}_3(t) = A \cdot (M_{\text{min}} - M_c) + (v_{\text{max}} - A \cdot (M_{\text{min}} - M_c)) \cdot \exp \left( -\frac{t}{t_c} \right)
\]

(21)

Given the numerical values of Table 4, this implies that a time \( t^*_3 = 0.6467 \) s is needed to decelerate from \( \dot{s} = v_{\text{max}} \) to \( \dot{s} = 0 \). The distance traveled in this period of time is \( L_3 = 0.0922 \) m.

Finally, a distance \( L_2 = L - L_1 - L_3 = 0.4229 \) m needs to be covered during the constant speed phase (\( \dot{s} = v_{\text{max}} \)), which requires \( t^*_2 = \frac{L_2}{v_{\text{max}}} = 1.0151 \) s. The total travel time therefore amounts \( t^*_1 + t^*_2 + t^*_3 = 1.8064 \) s.