Mech. Sci., 6, 163–171, 2015
www.mech-sci.net/6/163/2015/
doi:10.5194/ms-6-163-2015

Mechanical Sciences
Open Access

B-spline parameterized optimal motion trajectories for robotic systems with guaranteed constraint satisfaction

W. Van Loock, G. Pipeleers, and J. Swevers

Department of Mechanical Engineering, Division PMA, KU Leuven, 3001 Leuven, Belgium

Correspondence to: W. Van Loock (wannes.vanloock@kuleuven.be)

Received: 8 May 2015 – Revised: 31 July 2015 – Accepted: 8 August 2015 – Published: 1 September 2015

Abstract. When optimizing the performance of constrained robotic system, the motion trajectory plays a crucial role. In this research the motion planning problem for systems that admit a polynomial description of the system dynamics through differential flatness is tackled by parameterizing the system's so-called flat output as a polynomial spline. Using basic properties of B-splines, sufficient conditions on the spline coefficients are derived ensuring satisfaction of the operating constraints over the entire time horizon. Furthermore, an intuitive relaxation is proposed to tackle conservatism and a supporting software package is released. Finally, to illustrate the overall approach and potential, a numerical benchmark of a flexible link manipulator is discussed.

1 Introduction

The computation of a constrained optimal motion trajectory is a challenging problem in control and has attracted researchers already for several decades. In the 1990s the concept of differential flatness (Fliess et al., 1995) arose, which allows characterizing all the state space trajectories and the corresponding input history by means of a particular set of outputs. Differentially flat systems encompass all linear, controllable systems and many nonlinear systems as well. It quickly gained popularity for solving optimal control problems since in this way, the integration of the system dynamics is avoided. Hence, the problem reduces to finding the best flat output that obeys the boundary conditions and the state and input constraints. To deal with the infinite dimensionality of this problem, a polynomial or spline parameterization for the flat output is often used. To impose state and input constraints classical approaches in the literature (Louembet et al., 2009; Milam et al., 2000) apply a sampling strategy. As a result the constraints are not guaranteed to be satisfied in between the samples such that post-analysis is required for critical constraints. The aim in this paper is to provide constraints that can guarantee constraint satisfaction.

For linear systems, several methods have been proposed in the literature to guarantee constraint satisfaction at all times. Henrion and Lasserre (2006) propose a polynomial parameterization for the flat output, hereby transforming the constrained motion planning problem into a polynomial nonnegativity problem. Subsequently, a sum-of-squares decomposition is sought for using semidefinite programming. Piecewise polynomials can allow for more freedom in the parameterization and the former approach can be straightforwardly extended by searching for sum-of-squares decompositions on the individual polynomial pieces. A similar strategy is followed by Louembet et al. (2010), where a sum-of-squares decomposition is searched for directly in the B-spline basis, but by doing so a conservative solution is determined. Suryawan et al. (2012) also adopt a piecewise polynomial parameterization, but in contrast to the sum-of-squares procedure of Louembet et al. (2010), the authors express the semi-infinite constraints by applying basis function segmentation and using the convex hull property of B-splines, leading to linear constraints. Such an approach yields only sufficient conditions and hence introduces conservatism, which can be quite severe (de Boor and Daniel, 1974).

For nonlinear systems, existing approaches resort to convex approximations of the feasible set. Louembet et al. (2010) require a polytopic inner approximation of the feasible set. Inevitably, this method introduces conservatism in the problem. Moreover, some feasible sets do not admit such a polytopic approximation, e.g. obstacle avoidance constraints. For nonlinear systems that admit a polynomial representation by differential flatness, Suryawan et al. (2012) propose

a strategy to impose the semi-infinite constraints by relying on the convex hull property of splines and only keeping the linear and cubic monomial terms in the polynomial expansion. This way the optimization problem amounts to a simple QP. It should be noted, however, that this approach results in overly conservative constraints. To the best of our knowledge, no adequate method exists for guaranteed constraint satisfaction in nonlinear systems.

This paper aims to develop an optimization approach with guaranteed constraint satisfaction over the entire time horizon for systems that admit a polynomial representation by differential flatness. For systems that are flat but do not have such a polynomial representation, the equations are transformed into polynomial form, either by simple manipulation, a change of variables or approximation. Similar to Suryawan et al. (2012), our method is based on the convex hull property of B-splines. However, we do not require basis function segmentation and additionally we propose an intuitive method to control the conservatism that is introduced.

Section 2 introduces the motion planning problem as well as the concept of differential flatness. The following section proposes a spline parameterization of the flat output and discusses various relevant properties of splines. In Sect. 4, two relaxation strategies are discussed that effectively reduce conservatism. The free end-time problem is discussed as well. Section 5 validates our approach on a numerical benchmark problem. Furthermore, as a complement to the paper, a supporting software tool is released to aid the user in formulating motion planning problems involving splines.

2 Problem formulation

Consider a system governed by the differential equation

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}), \boldsymbol{x}(0) = \boldsymbol{x}_0, \tag{1}$$

with states $\boldsymbol{x}(t) \in \mathbb{R}^{n_x}$ and inputs $\boldsymbol{u}(t) \in \mathbb{R}^{n_u}$. We are interested in finding the control law $\boldsymbol{u}(t)$, $t \in [0, t_\mathrm{f}]$ that steers the system from an initial state $\boldsymbol{x}_0$, at $t = 0$, to a terminal state $\boldsymbol{x}_{t_\mathrm{f}}$, at $t = t_\mathrm{f}$, and that minimizes a performance criterion $g(\boldsymbol{x}, \boldsymbol{u}, t_\mathrm{f})$. At the same time the control law must obey state and input constraints:

$$\boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \geq 0, \ \forall t \in [0, t_\mathrm{f}], \tag{2}$$

where the constraint function $\boldsymbol{h} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_h}$ are assumed to be polynomial in $\boldsymbol{x}$ and $\boldsymbol{u}$.

In this work, we assume the system Eq. (1) is differentially flat (Fliess et al., 1995). This means that there exists a set of variables, called the flat outputs, $\boldsymbol{y} \in \mathbb{R}^{n_u}$ of the form

$$\boldsymbol{y} = \boldsymbol{\phi}\left(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{u}^{(1)}, \ldots, \boldsymbol{u}^{(q)}\right)$$

such that

$$\boldsymbol{x} = \boldsymbol{\psi}_x\left(\boldsymbol{y}, \boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(r-1)}\right)$$

and

$$\boldsymbol{u} = \boldsymbol{\psi}_u\left(\boldsymbol{y}, \boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(r)}\right)$$

for some positive integers q, r. So for a flat system, there exists an algebraic relationship between the states and inputs, and the flat output and its derivatives. Aside from all linear, controllable systems also many nonlinear systems are differentially flat. For more details and a catalog of flat systems the interested reader is referred to Martin et al. (2003) and Lévine (2010).

Differential flatness is particularly interesting when solving optimal control problems since it avoids integration of the system dynamics Eq. (1), an often costly and numerically challenging step. Indeed, by formulating the problem from the first paragraph in terms of the flat output, we arrive at the following optimization problem:

$$\begin{aligned} \underset{\boldsymbol{y}(\cdot)}{\text{minimize}} \quad & g\left(\boldsymbol{\psi}_x\left(\boldsymbol{y}, \ldots, \boldsymbol{y}^{(r-1)}\right), \boldsymbol{\psi}_u\left(\boldsymbol{y}, \ldots, \boldsymbol{y}^{(r)}\right), t_\mathrm{f}\right) \\ \text{subject to} \quad & \boldsymbol{y}^{(j)}(0) = \boldsymbol{y}_0^{(j)}, \ j = 0, \ldots, r-1 \\ & \boldsymbol{y}^{(j)}(t_\mathrm{f}) = \boldsymbol{y}_{t_\mathrm{f}}^{(j)}, \ j = 0, \ldots, r-1 \\ & \boldsymbol{h}\left(\boldsymbol{\psi}_x\left(\boldsymbol{y}, \ldots, \boldsymbol{y}^{(r-1)}\right), \boldsymbol{\psi}_u\left(\boldsymbol{y}, \ldots, \boldsymbol{y}^{(r)}\right)\right) \\ & \geq 0, \forall t \in [0, t_\mathrm{f}], \end{aligned} \tag{3}$$

where the boundary conditions for the flat output, $\boldsymbol{y}_0^{(j)}$ and $\boldsymbol{y}_{t_\mathrm{f}}^{(j)}$, are readily determined from $\boldsymbol{x}_0$ and $\boldsymbol{x}_{t_\mathrm{f}}$.

In solving the above optimization problem, we still face two challenges: (i) instead of a finite set of variables, the optimization variable is a function $\boldsymbol{y}(\cdot)$ and (ii) the constraints must be enforced at all time instances. Therefore, the problem is infinite dimensional with infinitely many constraints. To cope with the infinite dimensionality a fixed parameterization is usually chosen for $\boldsymbol{y}(\cdot)$. As splines provide a good approximation for smooth functions (de Boor, 2001), we will use a polynomial spline parameterization for $\boldsymbol{y}$ in this paper resulting in an optimization problem with few optimization variables that can be solved efficiently. In addition, as shown in the following section, such a parameterization allows us to impose the semi-infinite constraints by a finite number of sufficient constraints provided that the maps $\boldsymbol{\psi}_x$ and $\boldsymbol{\psi}_u$ are polynomial.

3 B-spline parameterized solutions

Let $\boldsymbol{\kappa} = (\kappa_0, \ldots, \kappa_{m+1})$ be a strictly increasing vector of points, k be a positive integer, and $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_m)$ be a vector of integers with $0 \leq \nu_i \leq k-1$. Then, s is a polynomial spline of order k with break points $\boldsymbol{\kappa}$ and continuity conditions $\boldsymbol{\nu}$ if there exist polynomials $p_0, \ldots, p_l$ of order k such that

$$s(t) = p_i(t), \text{ for } \kappa_i \le t < \kappa_{i+1},\ i = 0, 1, \ldots, m-1$$
$$s(t) = p_m(t), \text{ for } \kappa_m \le t \le \kappa_{m+1},$$

and

$$p_{i-1}^{(j-1)}(\kappa_i) = p_i^{(j-1)}(\kappa_i) \text{ for } j = 1, \ldots, \nu_i,\ i = 1, \ldots, m.$$

The vector space of polynomial splines with given k, $\boldsymbol{\kappa}$ and $\boldsymbol{\nu}$ is denoted by $\Pi_{k,\kappa,\nu}$ and has dimension $n = (m+1)k - \sum_{i=1}^{m} \nu_i$. The normalized B-spline basis of order k, defined over the knot vector

$$\boldsymbol{t} = \left(\underbrace{\kappa_0, \ldots, \kappa_0}_{k}, \underbrace{\kappa_1, \ldots, \kappa_1}_{k-\nu_1}, \ldots, \underbrace{\kappa_m, \ldots, \kappa_m}_{k-\nu_m}, \underbrace{\kappa_{m+1}, \ldots, \kappa_{m+1}}_{k} \right)$$

is commonly used as a basis for this vector space as it has various useful properties: the basis functions are nonnegative, sum up to one (partition of unity) and have local (minimal) support (de Boor, 2001). It yields a stable evaluation of the functions and its derivatives. A spline $s \in \Pi_{k,\kappa,\nu}$ with B-spline basis $\boldsymbol{b}_s = (b_1, \ldots, b_n)$ and (B-spline) coefficients $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_n)$ is represented as

$$s(t) = \sum_{i}^{n} \sigma_i b_i(t) = \langle \boldsymbol{\sigma}, \boldsymbol{b}_s(t) \rangle.$$

The control polygon of the spline is the broken line with

$$c_i = \left(t_i^*, \sigma_i\right), i = 1, \ldots, n$$

as vertex sequence, where

$$t_i^* = \frac{t_{i+1} + \ldots + t_{i+k-1}}{k-1}, \forall i.$$

Figure 1 illustrates a fourth order spline and its control polygon, which can be regarded as an exaggerated version of the spline itself (de Boor, 2001).

The convex hull property of splines is essential for the further course of this paper and is repeated from de Boor (2001) for completeness:

Property 1 (Convex hull) *Let s be a polynomial spline of order k with knot vector $\boldsymbol{t}$. From the nonnegativity, partition of unity and local support property of the B-spline basis it follows immediately that the segment $s(t)$, $t \in [t_i, t_{i+1}]$ lies within the convex hull of its control points $c_{i-k+1}, \ldots, c_i$.*

The convex hull property is illustrated in Fig. 1. It follows immediately from Property 1 that for constants a and b.

$$a \le \boldsymbol{\sigma} \le b \Rightarrow a \le s(t) \le b, \forall t \in \left[\kappa_0, \kappa_{m+1}\right].$$

Thus, by constraining the spline's coefficients, semi-infinite bounds on the spline can easily be imposed. Furthermore, it is trivial to see that any polynomial function of splines is itself a spline. Moreover, its B-spline coefficients can be determined from the B-spline coefficients of its constituents using the sum and product properties detailed in Appendix A.

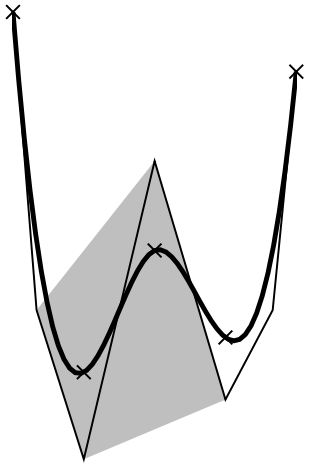

Figure 1. A continuous fourth order spline with five breaks indicated by the crosses. The spline's control polygon is the broken thin line. The gray area illustrates the convex hull property for points between the second and third break.

Now, let us apply Properties 1–3 to the optimization problem Eq. (3). Let $\boldsymbol{b}_{y_i}$, $i = 1, \ldots, n_u$ denote the B-spline basis for the ith flat output and $\boldsymbol{\gamma}_i$ the corresponding coefficients. Since $\boldsymbol{\psi}_x$, $\boldsymbol{\psi}_u$ and $\boldsymbol{h}$ are polynomial, we can determine the B-spline coefficients $\boldsymbol{\eta}_i(\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_{n_u})$ of the ith component of $\boldsymbol{h}(\boldsymbol{\psi}_x(\boldsymbol{y}, \boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(r-1)}), \boldsymbol{\psi}_u(\boldsymbol{y}, \boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(r)}))$. Then, an approximate solution for Eq. (3) is determined by solving

$$\begin{aligned} \underset{\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_{n_u}}{\text{minimize}} \quad & \widetilde{g}\left(\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_{n_u}, t_\mathrm{f}\right) \\ \text{subject to} \quad & \langle \boldsymbol{\gamma}_i, \boldsymbol{b}_{y_i}^{(j)}(0) \rangle = \left(\boldsymbol{y}_0^{(j)}\right)_i, \quad j = 0, \ldots, r-1,\ i = 1, \ldots, n_u, \\ & \langle \boldsymbol{\gamma}_i, \boldsymbol{b}_{y_i}^{(j)}(t_\mathrm{f}) \rangle = \left(\boldsymbol{y}_{t_\mathrm{f}}^{(j)}\right)_i, \quad j = 0, \ldots, r-1,\ i = 1, \ldots, n_u \\ & \boldsymbol{\eta}_i\left(\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_{n_u}\right) \ge 0, \quad i = 1, \ldots, n_h \end{aligned} \tag{4}$$

where, $\widetilde{g}$ denotes the result of the substitution of the spline parameterization in the objective function of Eq. (3).

4 Discussion

4.1 Reducing conservatism

Imposing a semi-infinite constraint on a polynomial spline through constraints on its B-spline coefficients yields only sufficient conditions and hence, the optimal value of Eq. (4) is an upper bound on the optimal value of Eq. (3). This conservatism is due to the distance between the control polygon of the spline and the spline itself. By representing the spline in a higher dimensional basis that includes the original one, the control polygon can be brought closer to the spline. Such

```
basis = BSplineBasis([0, tf], 4, 11); % Basis of degree 4 with 11 knots
y = BSpline.sdpvar(basis, [1, 1]);    % scalar (1x1) spline variable
dy = y.derivative(1);
ddy = y.derivative(2);
dddy = y.derivative(3);

pu = c0 + c1 * y;                     % Upper bound (l=1)
pl = -c0 + c1 * y;                    % Lower bound (l=1)

obj = y^2;
con = [y.f(0) == 0.8, y.f(tf) == -0.8, % z.f evaluates the spline
       dy.f(0) == 0, dy.f(tf) == -0.8,
       ddy.f(0) == -0.2, ddy.f(tf) == 0.2,
       dddy.f(0) == 0, dddy.f(tf) == 0,
       M*g*L/k * pu + I1/k * ddy + y <= pi/4, % semi-infinite constraints
       M*g*L/k * pl + I1/k * ddy + y >= -pi/4,
       M*g*L/k * pu + I1/k * ddy <= pi/16,
       M*g*L/k * pl + I1/k * ddy >= -pi/16];
sol = optimize(con, obj.integral());

y_opt = value(y);                     % Retrieve numerical solution for z
```

Listing 1. Example code for solving Eq. (6) with $l = 1$.

a basis can be derived by inserting knots, increasing the order[1] or a combination of both.

More precisely, let $s \in \Pi_{k,\kappa,\nu}$ with B-spline coefficients $\boldsymbol{\sigma}$. Let $\Pi_{k,\kappa,\nu} \subset \Pi_{\hat{k},\hat{\kappa},\hat{\nu}}$ with $\hat{k} \geq k$, and $\hat{\boldsymbol{\kappa}}$ and $\hat{\boldsymbol{\nu}}$ the refined break and continuity vectors such that $\boldsymbol{\kappa} \subseteq \hat{\boldsymbol{\kappa}}$ and $\nu_i \geq \hat{\nu}_j$, $i = 1, \ldots, m$ with j: $\kappa_i = \hat{\kappa}_j$. Then $\hat{s} \in \Pi_{\hat{k},\hat{\kappa},\hat{\nu}}$ with B-spline coefficients

$$\hat{\boldsymbol{\sigma}} = \mathbf{T}_{\hat{s}}^{s} \boldsymbol{\sigma},$$

where $\mathbf{T}_{\hat{s}}^{s}$ denotes the linear mapping from $\boldsymbol{b}_{\hat{s}}$ to $\boldsymbol{b}_{s}$, equals s and it can be shown that the control polygon of $\hat{s}$ will lie closer to the graph than that of s (de Boor, 2001).

A refinement of the break and/or continuity vectors acts locally on the spline and can target specific regions where conservatism is high. Order elevation is a global approach and changes the entire shape of the control polygon. This is illustrated in Fig. 2. In both cases it is clear that the new control polygon lies closer to the spline than the original one and hence conservatism is reduced. Note that order elevation increases the number of coefficients, and hence also the number of constraints, by $m + 1$. Inserting a single knot only amounts to one additional coefficient. Moreover, it can be shown that the convergence towards the spline for subsequent knot insertions is faster compared to order elevation (Prautzsch et al., 2002). For these reasons knot insertion is generally favored.

[1] Order elevation borrows from the idea of Polya's relaxation for polynomials.

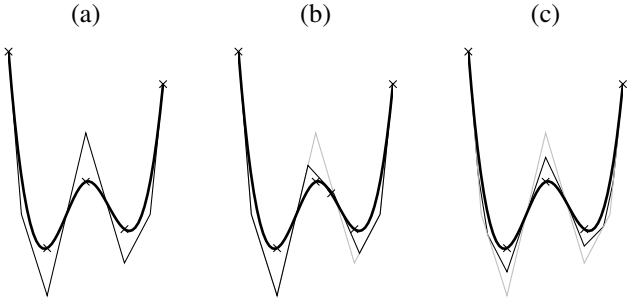

Figure 2. Refining the control polygon brings the control polygon closer to the spline: **(a)** the original spline, **(b)** one knot insertion and **(c)** elevate order by one. Note that knot insertion acts locally, while order elevation changes the control polygon globally.

4.2 Free end-time problems

For optimization problems where the final time t_f is a variable, a classical time scaling is applied to Eq. (3). The pseudo time $\tau = \frac{t}{t_\mathrm{f}}$ is used as free variable in the parameterization for the flat output instead of the time t. Consequently, the derivatives must be scaled by t_f and for free end-time problem Eq. (3) can be formulated as follows:

$$\begin{aligned} \underset{\boldsymbol{y}(\cdot),t_\mathrm{f}}{\text{minimize}} \quad & g\left(\boldsymbol{\psi}_x\left(\boldsymbol{y},\ldots,t_\mathrm{f}^{r-1}\boldsymbol{y}^{(r-1)}\right),\boldsymbol{\psi}_u\left(\boldsymbol{y},\ldots,t_\mathrm{f}^{r}\boldsymbol{y}^{(r)}\right),t_\mathrm{f}\right) \\ \text{subject to} \quad & \boldsymbol{y}^{(j)}(0) = t_\mathrm{f}^{j}\,\boldsymbol{y}_0^{(j)},\ j = 0,\ldots,r-1 \\ & \boldsymbol{y}^{(j)}(1) = t_\mathrm{f}^{j}\,\boldsymbol{y}_{t_\mathrm{f}}^{(j)},\ j = 0,\ldots,r-1 \\ & \boldsymbol{h}\left(\boldsymbol{\psi}_x\left(\boldsymbol{y},\ldots,t_\mathrm{f}^{r-1}\boldsymbol{y}^{(r-1)}\right),\boldsymbol{\psi}_u\left(\boldsymbol{y},\ldots,t_\mathrm{f}^{r}\boldsymbol{y}^{(r)}\right)\right) \geq 0,\ \forall\tau \in [0,1] \end{aligned} \quad . \tag{5}$$

Therefore, the proposed approach remains applicable to free end-time problems as well.

4.3 Software

To facilitate computations with splines and the translation of problem Eq. (3) into Eq. (4), a Matlab toolbox is made available at http://gitlab.mech.kuleuven.be/meco/splines-m. The aim is to be able to model optimization problems as easily as Yalmip (Löfberg, 2004), but with the variables being polynomial spline functions. An example listing is discussed in the following section.

5 Numerical validation

This section validates the proposed approach on the motion planning problem of a flexible link manipulator, introduced by Faiz (1999) and subsequently treated by Louembet et al. (2010). Numerical values and the constraints are taken from Louembet et al. (2010). Both a convex and nonconvex problem formulation Eq. (4) are compared to a classical sampling based and a sum-of-squares approach.

The dynamics of the manipulator are described by the equations

$$I_1\ddot{q}_1 + MgL\sin q_1 + k(q_1 - q_2) = 0,$$
$$I_2\ddot{q}_2 - k(q_1 - q_2) = u.$$

The system's state is given by $\boldsymbol{x} = (q_1,\ \dot{q}_1,\ q_2,\ \dot{q}_2)^\intercal$. The goal is to steer the system from the initial state $\boldsymbol{x}_0 = (0.8\,\mathrm{rad},\ 0\,\mathrm{rad\,s^{-1}},\ 0.67\,\mathrm{rad},\ 0\,\mathrm{rad\,s^{-1}})^\intercal$ to the final state $\boldsymbol{x}_{t_\mathrm{f}} = (-0.8\,\mathrm{rad},\ 0\,\mathrm{rad\,s^{-1}},\ -0.67\,\mathrm{rad},\ 0\,\mathrm{rad\,s^{-1}})^\intercal$ at $t_\mathrm{f} = 5.35\,\mathrm{s}$ with minimal deflection of the link, i.e. $g = \int_0^{t_\mathrm{f}} q_1^2(t)\mathrm{d}t$, while obeying the constraint on the joint positions

$$-\frac{\pi}{3} \le q_1 \le \frac{\pi}{3}, -\frac{\pi}{4} \le q_2 \le \frac{\pi}{4}, -\frac{\pi}{16} \le q_2 - q_1 \le \frac{\pi}{16}.$$

Note that the constraints on q_2 and $q_2 - q_1$ already imply the constraint on q_1. The state vector is described by the flat output, $y = q_1$, as:

$$\boldsymbol{x} = \boldsymbol{\psi}_x(y, \dot{y}, \ddot{y}, \dddot{y}) = \left(y, \dot{y}, \frac{I_1}{k}\ddot{y} + \frac{MgL}{k}\sin y + y, \frac{I_1}{k}\dddot{y} + \frac{MgL}{k}\dot{y}\cos y + \dot{y}\right)^\intercal.$$

Obviously, the mapping $\boldsymbol{\psi}_x$ is not polynomial. Therefore, in order to use the proposed approach, the constraints must be approximated or manipulated into polynomial expressions. To this end, we search for polynomial lower and upper bounds $\underline{p}(y)$ and $\overline{p}(y)$ such that

$$\underline{p}(y) \le \sin y \le \overline{p}(y), \forall y \in \left[-\frac{\pi}{3}, \frac{\pi}{3}\right].$$

The feasible set can then be replaced by the semi-algebraic inner approximation

$$\begin{array}{ll} \frac{I_1}{k}\ddot{y} + \frac{MgL}{k}\overline{p}(y) + y \le \frac{\pi}{4}, & \frac{I_1}{k}\ddot{y} + \frac{MgL}{k}\overline{p}(y) \le \frac{\pi}{16} \\ \frac{I_1}{k}\ddot{y} + \frac{MgL}{k}\underline{p}(y) + y \ge -\frac{\pi}{4}, & \frac{I_1}{k}\ddot{y} + \frac{MgL}{k}\underline{p}(y) \ge -\frac{\pi}{16} \end{array}.$$

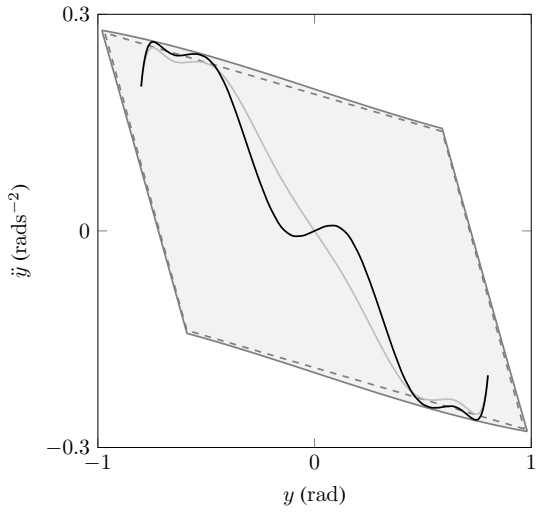

Figure 3. Solutions for the flexible manipulator problem with a polytopic approximation (gray) and semi-algebraic approximation (black).

The following polynomial bounds are used in this example:

$$\overline{p}(y), \underline{p}(y) = \pm c_0 + \sum_{i=1}^{l} c_i y^{2i-1},$$

where c_i, $i > 1$ are determined from a least-squares fit on $[-\frac{\pi}{3}, \frac{\pi}{3}]$ and the offset c_0 is determined such that $\underline{p}(y) \le \sin y \le \overline{p}(y)$. Note that for linear $\underline{p}(y)(\cdot)$, $\overline{p}(y)(\cdot)$, i.e. $l = 1$, we would get at a polytopic (convex) feasible set similar to the one suggested in Louembet et al. (2010). Note that in this example a semi-algebraic formulation is easily found. For more involved systems such as a six-dof robot, determining such a semi-algebraic approximation is a crucial step. In order to keep computation time low, it is key to limit the degree of the approximating polynomials.

We can now write the optimization problem as

$$\begin{array}{ll} \underset{y(\cdot)}{\text{minimize}} & \int_0^{t_\mathrm{f}} y(t)^2\mathrm{d}t \\ \text{subject to} & y(0) = 0.8,\ y(t_\mathrm{f}) = -0.8 \\ & \dot{y}(0) = 0,\ \dot{y}(t_\mathrm{f}) = 0 \\ & \ddot{y}(0) = -0.2,\ \ddot{y}(t_\mathrm{f}) = 0.2 \\ & \dddot{y}(0) = 0,\ \dddot{y}(t_\mathrm{f}) = 0 \\ & \frac{I_1}{k}\ddot{y}(t) + \frac{MgL}{k}\overline{p}(y(t)) + y(t) \le \frac{\pi}{4},\ \forall t \in [0, t_\mathrm{f}] \\ & \frac{I_1}{k}\ddot{y}(t) + \frac{MgL}{k}\underline{p}(y(t)) + y(t) \ge -\frac{\pi}{4},\ \forall t \in [0, t_\mathrm{f}] \\ & \frac{I_1}{k}\ddot{y}(t) + \frac{MgL}{k}\overline{p}(y(t)) \le \frac{\pi}{16},\ \forall t \in [0, t_\mathrm{f}] \\ & \frac{I_1}{k}\ddot{y}(t) + \frac{MgL}{k}\underline{p}(y(t)) \ge -\frac{\pi}{16},\ \forall t \in [0, t_\mathrm{f}] \end{array}. \quad (6)$$

The flat output is parameterized by a polynomial spline with 11 equidistant knots. Using the proposed approach the above optimization problem is cast in terms of its spline coefficients as in Eq. (4) using the accompanying software tool from Sect. 4.3. Listing 1 shows the code used for solving the problem with $l = 1$.

Figure 3 illustrates the solution for polynomial bounds of degree one ($l = 1$) (gray) and three ($l = 2$) (black). Clearly,

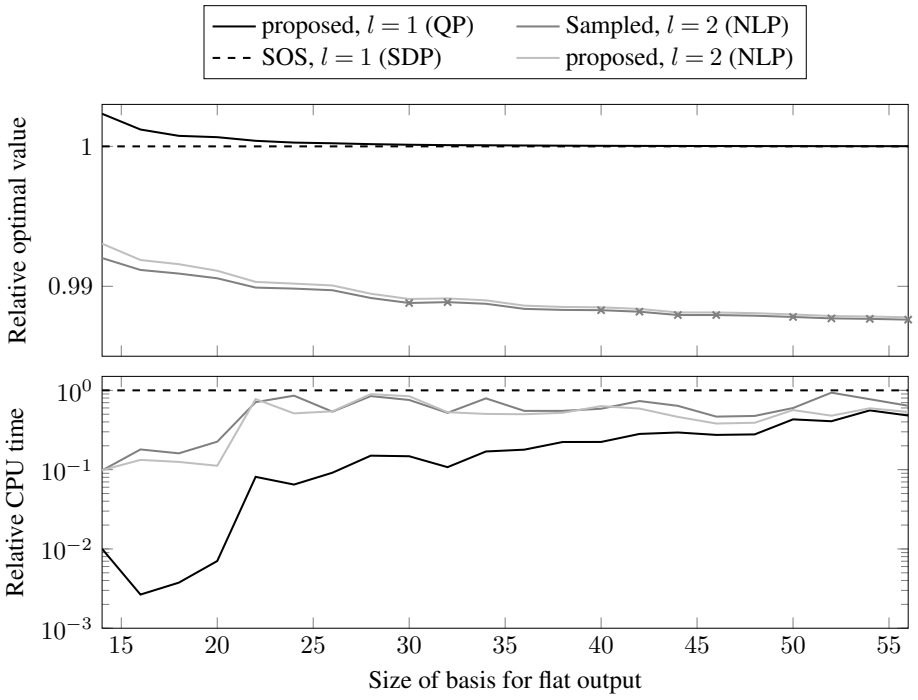

Figure 4. The optimal value and cpu time relative to the SDP for increasing size of the basis for the flat output. The crosses indicate where the sampled approach is feasible.

the latter solution is less conservative. Note, however, that the former problem is a convex quadratic program (QP) whereas the latter is nonconvex. Being able to use nonpolytopic sets is a clear advantage of our method over previous results (Louembet et al., 2010; Suryawan et al., 2012).

In a following numerical experiment, we compute the solution for increasing number of knots and compare the following cases:

1. Our proposed approach for $l = 1$, which is comparable to that of Suryawan et al. (2012). The resulting optimization problem is a convex QP and is solved using qpOASES (Ferreau et al., 2014).

2. Our proposed approach for $l = 2$. To the best of our knowledge, our method is unique in that it can guarantee these nonpolytopic constraints. The resulting optimization problen is a nonlinear progam (NLP). It is solved with Ipopt (Wächter and Biegler, 2006) with exact Hessians obtained using CasADi (Andersson, 2013).

3. A sampled solution for $l = 2$ as in Milam et al. (2000). The number of equidistant time samples is taken equal to the number of constraints in our approach. The resulting optimization problem is a NLP and similarly solved with Ipopt.

4. A globally optimal sum-of-squares (SOS) approach on each segment for $l = 1$ as in Henrion and Lasserre (2006) and Louembet et al. (2010). MOSEK ApS (2015) is used for solving the resulting convex semidefinite program (SDP). Note that this approach for $l > 1$ would require solving a nonconvex optimization problem with polynomial matrix inequalities for which to date no reliable solver exists.

Figure 4 shows the optimal value of the objective function and the CPU-time relative to that of the SDP as a function of n, the size of the basis for the flat output. The SDP is chosen as reference as it is the current state-of-the-art with respect to guaranteed constraint satisfaction. For the sampled approach, solutions that do not violate the constraints are indicated by crosses. A number of observations can be made.

1. For growing n, the optimal value of the QP converges to that of the SDP. This also illustrates that basis refinements by knot insertion or order elevation effectively reduce conservatism.

2. Solving a QP is significantly cheaper compared to the other approaches, especially for small n.

3. It is somewhat surprising to see that solving the convex SDP is more expensive than solving the NLP, illustrating great potential for NLP solvers. However, a global minimum cannot be guaranteed for the nonconvex programs.

4. The difference in optimal value between our approach and the sampled approach with $l = 2$ is small while the

former is guaranteed to be feasible. This is especially important in critical cases where constraint violation is not tolerated.

5. For $l = 2$ our approach is slightly cheaper to solve for larger bases compared to the sampled approach, offering a numerical advantage over traditional methods.

6 Conclusions

This paper focuses on optimal motion planning for systems that admit a polynomial description through differential flatness. The optimization problem is cast in terms of the flat output and a polynomial spline parameterization is proposed that allows us to guarantee state and input constraints by means of simple constraints on the B-spline coefficients. An intuitive relaxation of the constraints is achieved by representing the spline in a higher dimensional basis. Furthermore, a supporting software package is released. Numerical experiments show superior performance to existing approaches in the literature both in terms of computational time and optimality. For systems that do not admit a polynomial representation through differential flatness, an approximation is sought for as illustrated in the numerical validation.

Appendix A: Spline properties

In this Appendix we detail the sum and product properties of splines.

Property 2 (Summation) *Let* $p \in \Pi_{k,\kappa,\mu}$ *and* $r \in \Pi_{l,\lambda,\nu}$. *Then* $s = p + r \in \Pi_{\max(k,l),\xi,\omega}$, *where* $\boldsymbol{\xi} = \boldsymbol{\kappa} \underset{<}{\cup} \boldsymbol{\lambda}$ *the sorted, strictly increasing union of* $\boldsymbol{\kappa}$ *and* $\boldsymbol{\lambda}$, *and*

$$\omega_i = \begin{cases} \min(\mu_m, \nu_n) & \text{if } \xi_i = \kappa_m = \nu_n \text{ for some } m, n \\ \mu_m & \text{if } \xi_i = \kappa_m \text{ for some } m \\ \nu_n & \text{if } \xi_i = \lambda_n \text{ for some } n \end{cases}.$$

The spline coefficients, $\boldsymbol{\sigma}$, *of* s *are determined through a linear transformation of* $\boldsymbol{\pi}$ *and* $\boldsymbol{\rho}$:

$$\boldsymbol{\sigma} = \mathbf{T}_s^p \boldsymbol{\pi} + \mathbf{T}_s^r \boldsymbol{\rho},$$

where $\mathbf{T}_s^p$ *denotes the linear mapping from the B-spline basis* $\boldsymbol{b}_p$ *to* $\boldsymbol{b}_s$:

$$\boldsymbol{b}_p = \left(\mathbf{T}_s^p\right)^\intercal \boldsymbol{b}_s, \tag{A1}$$

and similarly for $\mathbf{T}_s^r$.

Property 3 (Multiplication) *Let* $p \in \Pi_{k,\kappa,\mu}$ and $r \in \Pi_{l,\lambda,\nu}$. Then $s = p\,r \in \Pi_{k+l,\xi,\omega}$, *where* $\boldsymbol{\xi}$ *and* $\boldsymbol{\omega}$ *are determined as in Property 2. Note, however, that the continuity over a given knot could also be higher. The spline coefficients,* $\boldsymbol{\sigma}$, *of* s *are determined through*:

$$\boldsymbol{\sigma} = \mathbf{T}_s^{p\otimes r}(\boldsymbol{\pi} \otimes \boldsymbol{\rho}),$$

where $\otimes$ *denotes the Kronecker product and* $\mathbf{T}_s^{p\otimes r}$ *is the linear mapping from* $\boldsymbol{b}_p \otimes \boldsymbol{b}_r$ *to* $\boldsymbol{b}_s$:

$$\boldsymbol{b}_p \otimes \boldsymbol{b}_r = \left(\mathbf{T}_s^{p\otimes r}\right)^\intercal \boldsymbol{b}_s. \tag{A2}$$

These linear mappings are easily found by solving a set of linear equations given by Eqs. (A1) or (A2), or by using dedicated algorithms that are readily available in the literature (e.g. Piegl and Tiller, 1997).

Acknowledgements. This research was supported by IWT ICON project Sitcontrol: Control with Situational Information, IWT SBO project MBSE4Mechatronics: Model-based Systems Engineering for Mechatronics, FWO project G0C4515N: Optimal control of mechatronic systems: a differential flatness based approach. This work also benefits from KU Leuven-BOF PFV/10/002 Center-of-Excellence Optimization in Engineering (OPTEC), from the Belgian Programme on Interuniversity Attraction Poles (DYSCO), initiated by the Belgian Federal Science Policy Office. Goele Pipeleers is partially supported by the Research Foundation Flanders (FWO Vlaanderen).

Edited by: A. Müller
Reviewed by: two anonymous referees

References

Andersson, J.: A General-Purpose Software Framework for Dynamic Optimization, PhD thesis, Arenberg Doctoral School, KU Leuven, Department of Electrical Engineering (ESAT/SCD) and Optimization in Engineering Center, Heverlee, Belgium, 2013.

de Boor, C.: A practical guide to splines, revised Edn.,Springer-Verlag, New York, 2001.

de Boor, C. and Daniel, J. W.: Splines with nonnegative *b*-spline coefficients, Math. Comput., 28, 565–568, 1974.

Faiz, T. N.: Real time and optimal trajectory generation for nonlinear systems, PhD thesis, University of Delaware, Newark, DE, 1999.

Ferreau, H., Kirches, C., Potschka, A., Bock, H., and Diehl, M.: qpOASES: A parametric active-set algorithm for quadratic programming, Math. Program. Comput., 6, 327–363, 2014.

Fliess, M., Lévine, J., Martin, P., and Rouchon, P.: Flatness and defect of nonlinear systems: Introductory theory and examples, Int. J. Control, 61, 1327–1361, doi:10.1020/00207179508921959, 1995.

Henrion, D. and Lasserre, J. B.: LMIs for constrained polynomial interpolation with application in trajectory planning, Syst. Control Lett., 55, 473–477, doi:10.1016/j.sysconle.2005.09.011, 2006.

Lévine, J.: Analysis and control of nonlinear systems: a flatness-based approach, in: Mathematical Engineering, Springer-Verlag, Berlin, Heidelberg, 2010.

Löfberg, J.: YALMIP : A Toolbox for Modeling and Optimization in MATLAB, in: Proceedings of the CACSD Conference, Taipei, Taiwan, http://users.isy.liu.se/johanl/yalmip (last access: 31 July 2015, 2004.

Louembet, C., Cazaurang, F., Zolghadri, A., Charbonnel, C., and Pittet, C.: Path planning for satellite slew manoeuvres: a combined flatness and collocation-based approach, Control Theory Appl., 3, 481–491, doi:10.1049/iet-cta.2008.0054, 2009.

Louembet, C., Cazaurang, F., and Zolghadri, A.: Motion planning for flat systems using positive *b*-splines: An LMI approach, Automatica, 46, 1305–1309, 2010.

Martin, P., Murray, R. M., and Rouchon, P.: Flat systems, equivalence and trajectory generation, unpublished technical report from Caltech, CDS Tech. Rep., CDS 2003-008, 2003.

Milam, M. B., Mushambi, K., and Murray, R. M.: A new computational approach to real-time trajectory generation for constrained mechanical systems, in: vol. 1, Proceedings of the IEEE Conference on Decision and Control, Sydney, Australia, 845–851, doi:10.1109/CDC.2000.912875, 2000.

MOSEK ApS: The MOSEK optimization toolbox for MATLAB manual, Version 7.1 (Revision 28), http://docs.mosek.com/7.1/toolbox/index.html, last access: 31 July 2015.

Piegl, L. and Tiller, W.: Symbolic operators for {NURBS}, Comput.-Aided Design, 29, 361–368, doi:10.1016/S0010-4485(96)00074-7, 1997.

Prautzsch, H., Boehm, W., and Paluszny, M.: Bezier and B-Spline Techniques, Springer-Verlag, New York, Inc., Secaucus, NJ, USA, 2002.

Suryawan, F., De Doná, J., and Seron, M.: Splines and polynomial tools for flatness-based constrained motion planning, Int. J. Syst. Sci., 43, 1396–1411, doi:10.1080/00207721.2010.549592, 2012.

Wächter, A. and Biegler, L. T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, Math. Program., 106, 25–57, doi:10.1007/s10107-004-0559-y, 2006.