HAL open science

# An approximate characterisation of the set of feasible trajectories for constrained flat systems

Luca Greco, Hugues Mounier, Maria Bekcheva


## ▶ To cite this version:

Luca Greco, Hugues Mounier, Maria Bekcheva. An approximate characterisation of the set of feasible trajectories for constrained flat systems. Automatica, 2022, 144, pp.110484. 10.1016/j.automatica.2022.110484 . hal-03692611

## HAL Id: hal-03692611
## https://centralesupelec.hal.science/hal-03692611

Submitted on 9 Jun 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# An approximate characterisation of the set of feasible trajectories for constrained flat systems [1]


Luca Greco [a,2], Hugues Mounier [a], Maria Bekcheva [b]

[a] *Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des signaux et systémes, 91190, Gif-sur-Yvette, France*

[b] *INAGRAL, 75008, Paris, France*



**Abstract**

We study differentially flat systems, subject to constraints on the input, the state and their derivatives. We are interested in providing an approximate characterisation of the set of trajectories satisfying both the equations of the system and the constraints. The flatness property and the choice of a polynomial parametrisation in terms of Bézier functions enable a finite dimension formulation of the problem in the parameter space of the Bézier control points. Such formulation provides a unified method to express a large variety of relevant constraint types such as, for instance, physical obstacles, actuator limitations, tubes around nominal trajectories, performance requirements and energy restrictions. We present a simple and effective inner approximation of the aforementioned set of trajectories obtained by simple algebraic manipulations of the original constraints formulation. The symbolic nature of the proposed characterisation allows a certain degree of adaptation to changes in the constraints without the need for any re-computation. The entire method (formulation an characterisation) is illustrated via a classical benchmark mechanical system.



*Key words:* Constraints, Differentially flat systems, Bézier curves, Trajectory generation, Semialgebraic sets


## 1 Introduction

Steering the state of a system to a desired position or through a defined path is a fundamental control problem. It becomes notoriously difficult if the system is nonlinear and the tracking has to be performed while fulfilling physical constraints on the actuators and/or the state. To alleviate such complexity, a 2-dof scheme, separating the feedforward and feedback parts (see, e.g. Hagenmeyer and Delaleau [2010]), is often adopted. The feedback stage has to take place entirely online, while the feedforward part can be pre-computed offline, thus enabling the generation of trajectories satisfying the constraints.

A particularly skilful means to simplify the trajectory generation problem is to exploit some kind of system inversion or parametrisation property such as the differential flatness (Fliess et al. [1995]), the orbital flatness (Fliess et al. [1999]) or the Liouvillian character (Chelouah [1997]). Under these structural properties, all system variables come as a direct (i.e. functional) expression of a particular (possibly fictitious) output, named a flat (or Liouvillian) output, and of a finite number of its derivatives. This parametrisation has a number of advantages, among which we mention: 1) An automatic satisfaction of the system's equations by the state and input induced by the parametrisation. Their trajectories will be named *feasible* in the sequel. 2) A direct mapping of the constraints in the flat output space. 3) A natural fulfilment of such constraints that is built in the flat output trajectory design. 4) An intrinsically unified formulation of any kind of constraints (input, output and state).

Most of the literature on steering for both differentially flat and general systems with constraints makes use of optimisation procedures, sometimes with collocation and/or discretisation. Moreover, the envisioned constraints are quite often in the state and input, but not in their successive derivatives. With the exception of Suryawan et al. [2012] in the case of polynomial systems, when such collocation or discretisation procedures are involved, usually no guarantee of constraint fulfilment between the collocation or discretisation points can be given (see, e.g. Faiz et al. [2001], Petit et al. [2001],


*Email addresses:* luca.greco@centralesupelec.fr (Luca Greco), hugues.mounier@universite-paris-saclay.fr (Hugues Mounier), maria@inagral.com (Maria Bekcheva).


[1] This paper was not presented at any IFAC meeting.


[2] Corresponding author, Tel. +33 1 69 85 17 33.

Oldenburg and Marquardt [2002], Joubert et al. [2015], Roberts and Hanrahan [2016]).

Several authors sought to ensure the satisfaction of constraints over the entire time horizon without any post elaboration. A simple way to avoid any recourse to collocation or discretisation is to specialise the flat outputs to a linear combination of known functions. While the initial problem is infinite dimensional by nature (search in a general functional space), the one after specialisation is a finite dimensional one (e.g. in the weights of the linear combination).

For linear systems, Henrion and Lasserre [2006] parametrise the flat output with polynomial functions, which are naturally closed under differentiation; the corresponding constraint problem then amounts to a polynomial non-negativity one, which is coped with a sum-of-squares decomposition and solved with semidefinite programming. Louembet et al. [2010] also make use of piecewise polynomial representations (B-splines) and a sum-of-squares formulation, while Suryawan et al. [2012] resort to the convex hull property of B-splines, which introduces some conservatism. This convex hull property is also used by Flores and Milam [2006] together with NURBS, the optimal problem being solved through sequential quadratic programming. For nonlinear systems, some restrictions or simplifications are usually adopted. Louembet et al. [2010], for instance, consider an inner polytopic approximation of the constraints (actually, only linear constraints can be solved); Suryawan et al. [2012] focus on a special class of polynomial systems (or polynomial approximations) where only linear and cubic monomial terms are present; Van Loock et al. [2015] use a similar approximation to Louembet et al. [2010], but extend to polynomial upper and lower bounds for the constraints.

Another noticeable contribution (Cichella et al. [2018]) uses a Bernstein polynomial approximation (Bézier curves) of an optimal solution for a constraint minimisation problem of a differentially flat system. This approximation also minimises the original cost function in the Bernstein space of order $N$; the approximant sequence then converges to the original optimal solution. Other interesting optimisation approaches, although more specifically focused on particular steering problems, are represented by Mellinger and Kumar [2011], Roberts and Hanrahan [2016], Stoican et al. [2016].

Particularly relevant for the present paper is the work of Faulwasser et al. [2014]. Instead of looking for a specific trajectory, solution of an optimisation problem, the authors formulate a constrained reachability problem. More in depth, they provide some formal conditions for the existence of a suitable control function capable of steering the state from an initial set-point to a final set-point, while satisfying constraints on both the state and the input.

In this paper, we study differentially flat systems, subject to constraints on the input, the state and a finite (arbitrary large) number of their derivatives, without the need to explicitly specify starting and ending point conditions. We do not focus on the generation of one particular trajectory resulting from an optimisation procedure. Our aim is to provide an approximate characterisation of the whole set of trajectories satisfying both the equations of the system and the constraints. The states and the inputs are written in terms of the flat outputs (and their derivatives) and the constraints are, therefore, directly remapped on the desired outputs. We assume those mappings to be expressed (natively or through approximation) in polynomial form, but some special non polynomial forms are also discussed. In order to reduce the problem to a finite dimensional one, we choose a polynomial parametrisation of the desired outputs as Bézier functions. The convex hull property of Bézier functions allows the satisfaction of the constraints on the entire time horizon without resorting to discretisation methods and post-elaborations. The characterisation of the set of feasible and constraint-compliant trajectories, is then performed in the parameter space of the control points defining the Bézier curves.

The main contribution of the paper is twofold: 1) a unified formulation of the feasible constrained trajectories encompassing a large variety of constraint types; 2) a simple and effective inner approximation of the aforementioned set of trajectories.

The present formulation allows to specify, in a unified fashion, many relevant constraints in the states (e.g. initial and final points or regions, banned regions for obstacle avoidance, corridor constraints, tubes around nominal trajectories), in the inputs (e.g. saturations, upper and lower bounds due to actuator limitations, time-varying bounds) and in arbitrary polynomial functions of states, inputs and their derivatives (e.g. limitations on the snap, jerk, curvature, energy or power consumption).

The characterisation of the set of trajectories can be performed through standard formal (like the Cylindrical Algebraic Decomposition) or numerical (Sum of Squares formulations) methods. We discuss merits and limitations of both approaches and we present a simple characterisation which, despite some restrictions, offers many interesting advantages. More in depth, our characterisation:

- provides a family of approximating sets of increasing precision converging to the whole set of feasible constrained trajectories;
- can be produced by simple algebraic manipulation of the general formulation;
- retains the constraint structure. If some of the bounds in the constraints change, the approximation keeps

its form and just adapts some coefficients without the need for any re-computation;
- allows for a simple robustness analysis w.r.t. some parameters of the constraints;
- allows the extraction of a specific trajectory from the set and a fast on-line re-computation of a new solution in case of changes in the constraints.

The entire method (formulation an characterisation) is thoroughly illustrated via two interesting test cases based on a classical benchmark mechanical system.

Some preliminary results have been elaborated in the PhD thesis of the third author, whose relevant chapter is available at the following arXiv preprint Bekcheva [2020].

*Notation*

The set of non-negative integers is denoted by $\mathbb{N}$. Given a scalar function $z \in C^\kappa(\mathbb{R}, \mathbb{R})$ and a constant $\alpha \in \mathbb{N}$, we denote by $z^{\langle\alpha\rangle}$ the derivative tuple of $z$ up to the order $\alpha \leqslant \kappa$: $z^{\langle\alpha\rangle} = z, \dot{z}, \ddot{z}, \ldots, z^{(\alpha)}$. Given the vector function $v = (v_1, \ldots, v_q)$, $v_i \in C^\kappa(\mathbb{R}, \mathbb{R})$ and the tuple $\alpha = (\alpha_1, \ldots, \alpha_q)$, $\alpha_i \in \mathbb{N}$, we denote by $v^{\langle\alpha\rangle}$ the tuple of derivatives of each component $v_i$ of $v$ up to its respective order $\alpha_i \leqslant \kappa$: $v^{\langle\alpha\rangle} = v_1, \ldots, v_1^{(\alpha_1)}, v_2, \ldots, v_2^{(\alpha_2)}, \ldots, v_q, \ldots, v_q^{(\alpha_q)}$. We denote by $\mathbb{R}[a_1, \ldots, a_n]$ (or by $\mathbb{R}[a]$ with $a = (a_1, \ldots, a_n) \in \mathbb{R}^n$) the ring of polynomials in the variables $a_1, \ldots, a_n$ and with coefficients in $\mathbb{R}$.

## 2 Problem statement

### 2.1 General problem formulation

Consider the nonlinear system

$$\dot{x}(\tau) = f(x(\tau), u(\tau)) \tag{1}$$

with state vector $x = (x_1, \ldots, x_n)$ and control input $u = (u_1, \ldots, u_m)$, $x_i, u_j \in C^\kappa([0, +\infty), \mathbb{R})$ for a suitable $\kappa \in \mathbb{N}$. Systems with discontinuous or non differentiable functions are beyond the scope of this paper. One can apply techniques involving Filippov solutions or the functions algebra of Kaldmäe et al. [2017] to extend our results to these cases. We assume the state, the input and their derivatives to be subject to both inequality and equality scalar constraints of the form

$$G_i(x^{\langle\alpha_i^x\rangle}(\tau), u^{\langle\alpha_i^u\rangle}(\tau)) \leqslant 0 \qquad \forall\tau \in I_i,\ \forall i \in \{1, \ldots, \nu^{\mathrm{in}}\} \tag{2a}$$

$$D_j(x^{\langle\beta_j^x\rangle}(\tau), u^{\langle\beta_j^u\rangle}(\tau)) = 0 \qquad \forall\tau \in I_j,\ \forall j \in \{1, \ldots, \nu^{\mathrm{eq}}\} \tag{2b}$$

with each $I_j$ being either $[0, T]$ (continuous constraint) or a discrete set $\{t_1, \ldots, t_\gamma\}$, $0 \leqslant t_1 \leqslant \cdots \leqslant t_\gamma \leqslant T < +\infty$ (discrete constraint), and $\alpha_i^x, \beta_j^x \in \mathbb{N}^n$, $\alpha_i^u, \beta_j^u \in \mathbb{N}^m$. We stress that the relations (2) specify objectives (set-points, constraints, etc.) on the finite interval $[0, T]$. Objectives can be also formulated on larger intervals as a concatenation of sub-objectives on a union of sub-intervals, provided that some continuity and/or regularity constraints are imposed on the boundaries of each sub-interval. We here focus on just one of such intervals.

Our aim is to provide a characterisation of the set of input and state trajectories $(x, u)$ satisfying the system dynamics (1) (usually referred to as *feasible trajectories*) and the constraints (2). More formally we state the following problem.

**Problem 1 (Constrained trajectory set)** *Let $\mathscr{C}$ be a subalgebra of $C^\kappa([0, T], \mathbb{R})$. Constructively characterise the set $\mathscr{C}^{\mathrm{cons}} \subseteq \mathscr{C}^{n+m}$ of all extended trajectories $(x, u)$ satisfying the system equations (1) and the constraints (2).*

Problem 1 can be considered as a generalisation of a constrained reachability problem (see for instance Faulwasser et al. [2014]). In a reachability problem the stress is usually on initial and final set-points and the goal is to find a suitable input to steer the state from the initial to the final point while, possibly, respecting the constraints. Here we wish to give a functional characterisation of the overall set of extended trajectories $(x, u)$ satisfying some given differential constraints. A classical constrained reachability problem can be cast in the present formalism by limiting the constraints $G_i$ and $D_j$ to $x$ and $u$ (not their derivatives) and by forcing two of the equality constraints to coincide with the initial and final set-points.

Problem 1 is difficult to be addressed in its general setting, thus, in the following, we make some restrictions to the class of systems and to the functional space $\mathscr{C}$. As a first assumption we limit the analysis to differentially flat systems (Fliess et al. [1995]).

### 2.2 Constraints in the flat output space

Let us assume that system (1) is differentially flat with flat output [3]

$$y = (y_1, \ldots, y_m) = h(x, u^{\langle\rho^u\rangle}), \tag{3}$$

with $\rho^u \in \mathbb{N}^m$. The parameterisation associated to $y$ is

[3] We recall that the flat output $y$ has the same dimension $m$ as the input vector $u$.

$$x = \psi(y^{\langle \eta^x \rangle}) \quad (4a)$$
$$u = \zeta(y^{\langle \eta^u \rangle}), \quad (4b)$$

with $\eta^x \in \mathbb{N}^n$ and $\eta^u \in \mathbb{N}^m$.

Through the dynamical extension algorithm (Li and Feng [1987]), we get the flat output dynamics

$$\begin{cases} y_1^{(\theta_1)} = \phi_1(y^{\langle \mu_1^y \rangle}, u^{\langle \mu_1^u \rangle}) \\ \quad \vdots \\ y_m^{(\theta_m)} = \phi_m(y^{\langle \mu_m^y \rangle}, u^{\langle \mu_m^u \rangle}), \end{cases} \quad (5)$$

with $\mu_i^y = (\mu_{i1}^y, \dots, \mu_{im}^y) \in \mathbb{N}^m$, $\mu_i^u = (\mu_{i1}^u, \dots, \mu_{im}^u) \in \mathbb{N}^m$ and $\theta_i > \max_j \mu_{ji}^y$. The original $n$-dimensional dynamics (1) and the $\Theta$-dimensional flat output dynamics (5) ($\Theta = \sum_i \theta_i$) are in one-to-one correspondence through (3) and (4). Therefore, the constraints (2) can be re-written as

$$\Gamma_i(y^{\langle \omega_i^{\text{in}} \rangle}(\tau)) \leqslant 0 \qquad \forall \tau \in I_i,\ \forall i \in \{1, \dots, \nu^{\text{in}}\} \quad (6a)$$
$$\Delta_j(y^{\langle \omega_j^{\text{eq}} \rangle}(\tau)) = 0 \qquad \forall \tau \in I_j,\ \forall j \in \{1, \dots, \nu^{\text{eq}}\} \quad (6b)$$

with $\Gamma_i(y^{\langle \omega_i^{\text{in}} \rangle}) = G_i((\psi(y^{\langle \eta_x \rangle}))^{\langle \alpha_i^x \rangle}, \zeta(y^{\langle \eta^u \rangle})^{\langle \alpha_i^u \rangle})$, $\Delta_j(y^{\langle \omega_j^{\text{eq}} \rangle}) = D_j((\psi(y^{\langle \eta^x \rangle})^{\langle \beta_j^x \rangle}, \zeta(y^{\langle \eta^u \rangle})^{\langle \beta_j^u \rangle})$ and $\omega_i^{\text{in}}, \omega_j^{\text{eq}} \in \mathbb{N}^m$.

Thus, the Problem 1 can be transformed in terms of the flat output dynamics (5) and the constraints (6) as follows.

**Problem 2 (Constrained flat output set)** [4] *Let $\mathscr{C}_y$ be a subspace of $C^\sigma([0,T],\mathbb{R})$ with $\sigma = \max((\theta_1, \dots, \theta_m), \omega_1^{\text{in}}, \dots, \omega_{\nu^{\text{in}}}^{\text{in}}, \omega_1^{\text{eq}}, \dots, \omega_{\nu^{\text{eq}}}^{\text{eq}})$. Constructively characterise the set $\mathscr{C}_y^{\text{cons}} \subseteq \mathscr{C}_y^m$ of all flat outputs satisfying the dynamics (5) and the constraints (6).*

Working with differentially flat systems allows us to translate, in a unified fashion, all the state and input constraints as constraints in the flat outputs and their derivatives (see (6)). We remark that $\psi$ and $\zeta$ in (4) are such that $\psi(y^{\langle \eta_x \rangle})$ and $\zeta(y^{\langle \eta_u \rangle})$ satisfy the dynamics of system (1) by construction and lead to intrinsically feasible trajectories. In other words, the extended trajectories $(x, u)$ of (1) are in one-to-one correspondence with $y \in \mathscr{C}_y^m$ given by (3). Hence, choosing $y$ as a solution of Problem 2 ensures that $x$ and $u$ given by (4) are solutions of Problem 1.

[4] Here the max operator is applied elementwise on each vector.

*2.3 Problem specialisation - Bézier curves*

For any practical purpose, one has to choose the functional space $\mathscr{C}_y$ to which all components of the flat output belong. In order to reduce the infinite dimensional complexity, one is led to choose an approximation space $\mathscr{C}^{\text{app}}$ that is dense in $C^\sigma([0,T],\mathbb{R})$. A possible choice is to work with parametric functions expressed in terms of basis functions like, for instance, Bernstein-Bézier, Chebychev or Spline polynomials (Prautzsch et al. [2002]).

**Definition 1** *A Bézier curve of degree $N \in \mathbb{N}$ in the Euclidean space $\mathbb{R}^r$ is defined as*

$$P(\tau) = \sum_{j=0}^{N} c_j B_{jN}(\tau), \qquad \tau \in [0,1]$$

*where $c_j \in \mathbb{R}^r$ are control points and $B_{jN}(\tau) = \binom{N}{j}(1-\tau)^{N-j}\tau^j$ are the Bernstein polynomials* [5] *.*

We stress that a generic Bézier curve $P(t)$ with $t \in [a,b]$ and $a \neq b$ can be easily written in terms of the previously defined $P(\tau)$, $\tau \in [0,1]$ by means of the affine transformation $\tau = (t-a)/(b-a)$. For the sake of simplicity, we here set $T = 1$ and we chose as functional space

$$\mathscr{C}_N^{\text{app}} = \left\{ \begin{array}{r} g \in C^\sigma([0,1],\mathbb{R}) \mid g(\tau) = \sum_{j=0}^{N} c_j B_{jN}(\tau), \\ N \in \mathbb{N}, c_j \in \mathbb{R}^r \end{array} \right\}. \quad (7)$$

The set of Bézier functions of generic degree has the very useful property of being closed with respect to addition, multiplication, degree elevation, derivation and integration operations (Farouki and Rajan [1988]). As a consequence, any integro-differential polynomial operator applied to a Bézier curve, still produces a Bézier curve (in general of different degree). Therefore, if the flat outputs $y$ are chosen in $\mathscr{C}_N^{\text{app}}$ and the operators $\Gamma_i(\cdot)$ and $\Delta_j(\cdot)$ in (6) are integro-differential polynomials, then such constraints can be expressed, after a reordering of the coefficients and without any added conservatism, in terms of Bézier curves in $\mathscr{C}_{\bar{N}}^{\text{app}}$ for some $\bar{N} \in \mathbb{N}$ which is, in general, different from $N$.

The main difficulty in ensuring a polynomial form for the constraints (6) is represented by the mappings $\psi(\cdot)$ and $\zeta(\cdot)$ in the flat parametrisation (4). Indeed, while the functions $G_i(\cdot)$ and $D_j(\cdot)$ in (2) are defined by the designer, the mappings $\psi(\cdot)$ and $\zeta(\cdot)$ are imposed by the

[5] For all $j \neq 0$ and $j \neq N$, and by definition of the Bernstein polynomials, $B_{jN}(0) = B_{jN}(1) = 0$. For $j = 0$, by convention $\tau^j = \tau^0 = 1$, whatever the value of $\tau$ is; likewise, for $j = N$, $\tau^{N-j} = 1$. Thus $B_{0N}(0) = \binom{N}{0}$, $B_{NN}(0) = \binom{N}{N}$.

system. However, not the entire system needs to be expressed in polynomial form w.r.t. the flat outputs, but only the states and inputs that actually enter the constraints. Moreover, in some cases this is even not necessary: the trigonometric function in the example in Section 4 is managed without resorting to a polynomial representation. Even functions expressed as ratio of polynomials can be directly tackled without any change, provided that the polynomial at the denominator is always positive (see again the example in Section 4 for the variable $\omega$).

In the other cases, polynomial constraints can be enforced by adding some conservativeness. A common technique consists in providing a polynomial upper and lower bound for the non polynomial function (see for instance Van Loock et al. [2015]). If an estimate of the flat outputs is available (or can be imposed) in the interval $[0, 1]$ in the form of a tube around a nominal trajectory or of maximum and minimum values, then the mappings $\psi(\cdot)$ and $\zeta(\cdot)$ can be approximated up to a desired precision. There exists a wide literature on approximation techniques (Powell [1981]) and many efficient methods based on Chebyshev polynomials (Rivlin [1974], Trefethen [2019]).

In the light of the previous analysis, we assume the following.

**Assumption 1** *Considering $y \in \mathscr{C}_N^{\text{app}}$*

$$y(\tau) = \sum_{j=0}^{N} a_j B_{jN}(\tau) \tag{8}$$

*with $a_j := (a_{j1}, \ldots, a_{jm}) \in \mathbb{R}^m$, the constraint in (6) can be written as*

$$\Gamma_i(y^{\langle \omega_i^{\text{in}} \rangle}(\tau)) = \sum_{k=0}^{N_i^{\text{in}}} r_{ik}^{\text{in}}(a) B_{kN_i^{\text{in}}}(\tau) \leqslant 0 \qquad \forall \tau \in I_i, \; \forall i \in \{1, \ldots, \nu^{\text{in}}\} \tag{9a}$$

$$\Delta_j(y^{\langle \omega_j^{\text{eq}} \rangle}(\tau)) = \sum_{k=0}^{N_j^{\text{eq}}} r_{jk}^{\text{eq}}(a) B_{kN_j^{\text{eq}}}(\tau) = 0 \qquad \forall \tau \in I_j, \; \forall j \in \{1, \ldots, \nu^{\text{eq}}\} \tag{9b}$$

*where $N_i^{\text{in}}, N_j^{\text{eq}} \in \mathbb{N}$ and $r_{ik}^{\text{in}}, r_{jk}^{\text{eq}} \in \mathbb{R}[a]$ polynomials in $a := (a_1, \ldots, a_N) \in \mathbb{R}^{m(N+1)}$.*

**Remark 1 (Time-varying and tube constraints)** *The use of Bézier functions, or more in general of basis functions, allows an interesting extension to the constraints formulation. Instead of the constant constraint $\Gamma_i(y^{\langle \omega_i^{\text{in}} \rangle}(\tau)) \leqslant 0$ one can impose the more general time-varying constraint $\Gamma_i(y^{\langle \omega_i^{\text{in}} \rangle}(\tau)) \leqslant \sum_{k=0}^{N_i^{\text{in}}} c_{ik} B_{kN}(\tau)$ for some $c_{ik} \in \mathbb{R}$. Indeed, by defining $\tilde{\Gamma}_i(y^{\langle \omega_i^{\text{in}} \rangle}(\tau)) = \sum_{k=0}^{N_i^{\text{in}}} \left( r_{ik}^{\text{in}}(a) - c_{ik} \right) B_{kN}(\tau)$, the new constraint can be cast in the form (9a) with respect to the polynomials $\tilde{r}_{ik}^{\text{in}}(a) = r_{ik}^{\text{in}}(a) - c_{ik}$.*

*Tube constraints can be imposed on states and inputs in a similar fashion. Such constraints are particularly relevant in obstacle avoidance tasks, where a non feasible nominal trajectory avoiding the obstacles can be easily available via geometric planning. The goal, in this case, is to compute a feasible trajectory, still avoiding obstacles, which is not too far from the nominal one. Let us consider, for instance, a vector $v = (v_1, \ldots, v_p)$ of state variables (or inputs) expressed as polynomials (see previous discussion) of the flat outputs and their derivatives according to (4). If a nominal trajectory $\bar{v}(\tau) = (\bar{v}_1(\tau), \ldots, \bar{v}_p(\tau))$ is given in terms of Bézier functions as $\bar{v}_i(\tau) = \sum_{k=0}^{N_i^{\text{in}}} \bar{v}_{ik} B_{kN}(\tau)$, then a tube constraint can be imposed as follows $\sum_{i=1}^{p} \left(v_i(\tau) - \bar{v}_i(\tau)\right)^2 \leqslant c^2$, where $c$ is the radius of the tube. It is easy to verify that, after rearranging all the terms w.r.t. the Bernstein polynomials, the previous expression is in the form (9a).*

**Remark 2 (Polynomial ratio constraints)** *The adoption of Bézier functions enables another useful extension, namely the definition of constraints described in terms of polynomial ratios. Suppose to have the following constraints:*

$$\underline{b} \leqslant \frac{\sum_{k=0}^{N_n} n_k(a) B_{kN}(\tau)}{\sum_{k=0}^{N_d} d_k(a) B_{kN}(\tau)} \leqslant \overline{b}$$

*where $\underline{b} < \overline{b} \in \mathbb{R}$ and $n_k, d_k \in \mathbb{R}[a_{01}, \ldots, a_{0m}, \ldots, a_{N1}, \ldots, a_{Nm}]$. Let us also assume $\sum_{k=0}^{N_d} d_k(a) B_{kN}(\tau) > 0$ for every $\tau \in [0, 1]$. For the sake of simplicity we consider here $N_n = N_d = N$, the general case can be put in this form by adding identically zero polynomials. We can split the previous constraints as follows:*

$$\underline{b} \sum_{k=0}^{N} d_k(a) B_{kN}(\tau) - \sum_{k=0}^{N} n_k(a) B_{kN}(\tau) \leqslant 0$$
$$\sum_{k=0}^{N} n_k(a) B_{kN}(\tau) - \overline{b} \sum_{k=0}^{N} d_k(a) B_{kN}(\tau) \leqslant 0\,.$$

*These two constraints can then be cast in the form (9a) with respect to the polynomials $\tilde{r}_{ik}^{\text{in}}(a) = \underline{b} d_k(a) - n_k(a)$ and $\tilde{r}_{jk}^{\text{in}}(a) = n_k(a) - \overline{b} d_k(a)$ for some index $i, j$.*

In order to reduce the infinite dimensional Problem 2 to a finite dimensional one, we exploit the well known fact (see for instance Prautzsch et al. [2002]) that any Bézier curve lies within the convex hull of the points $(t_j^*, c_j)$, where $t_j^* = \frac{j}{N}$ are the Greville abscissae and $c_j$

the control points. By adding some conservatism, we can replace the constraints (9), which have to hold for any $\tau \in [0,1]$, with a set of constraints expressed in terms of the (finitely many) control points of (9).

**Definition 2** *Let*

$$r^{\text{in}} = (r^{\text{in}}_{10}, \dots, r^{\text{in}}_{1N^{\text{in}}_1}, \dots, r^{\text{in}}_{\nu^{\text{in}}0}, \dots, r^{\text{in}}_{\nu^{\text{in}}N^{\text{in}}_{\nu^{\text{in}}}})$$
$$r^{\text{eq}} = (r^{\text{eq}}_{10}, \dots, r^{\text{eq}}_{1N^{\text{eq}}_1}, \dots, r^{\text{eq}}_{\nu^{\text{eq}}0}, \dots, r^{\text{eq}}_{\nu^{\text{eq}}N^{\text{eq}}_{\nu^{\text{eq}}}})$$

*and $r = (r^{\text{in}}, r^{\text{eq}})$, where $r^{\text{in}}_{ij}$ and $r^{\text{eq}}_{kh}$ are the polynomials occurring in (9). We define the semi-algebraic set associated to the constraints (9) as*

$$\mathscr{S}(r, \mathbb{A}) = \{a \in \mathbb{A} \mid r^{\text{in}}(a) \leqslant 0,\, r^{\text{eq}}(a) = 0\} \tag{10}$$

*for any parallelotope*

$$\mathbb{A} = [\underline{a}_0, \overline{a}_0] \times \cdots \times [\underline{a}_N, \overline{a}_N] \tag{11}$$

*with $\underline{a}_i, \overline{a}_i \in (\mathbb{R} \cup \{-\infty, +\infty\})^m$. The sets $[\underline{a}_i, \overline{a}_i] = [\underline{a}_{i1}, \overline{a}_{i1}] \times \cdots \times [\underline{a}_{im}, \overline{a}_{im}]$, with $\underline{a}_{ij} < \overline{a}_{ij}$, are themselves parallelotopes bounding the elements of the vectors $a_i \in \mathbb{R}^m$.*

The parallelotope $\mathbb{A}$, endowed with the usual $\mathbb{R}^{N+1}$ topology, is the topological space associated to the sheaf of available trajectories among which the user is allowed to choose a reference, through the representation (8). $\mathbb{A}$ is defined by the designer in terms of the upper and lower bounds of the control points of the Bézier functions defining the flat outputs. The flat outputs have almost always a physical meaning and it is not difficult to find bounds for them. The semi-algebraic set $\mathscr{S}(r, \mathbb{A})$ represents the subset of $\mathbb{A}$ of those trajectories fulfilling the constraints (6). Picking $a$ in $\mathscr{S}(r, \mathbb{A})$ automatically ensures that the output $y$ in (8) satisfies the constraints (9). Unfortunately, however, there is no way to ensure a priori that such a solution exists even in the whole parallelotope $\mathbb{A}$. The infinite dimensional Problem 2 has become a finite dimensional one in the vector $a$, and can be reformulated as follows.

**Problem 3 (Semi-algebraic set)** *For any fixed parallelotope $\mathbb{A}$ as in (11), constructively characterise the semi-algebraic set $\mathscr{S}(r, \mathbb{A})$ in (10).*

**Remark 3 (Reducing the conservatism)** *The conservatism introduced by the latter formulation of the problem is due to the, potentially large, distance of a Bézier curve from its control points. A sharp estimate of the maximum distance of a Bézier curve from its defining control polygon, i.e. the piecewise linear function connecting the points $(t^*_j, c_j)$, is provided in Nairn et al. [1998]. A way to reduce such a conservatism, without increasing the number of the parameters, is to apply a suitable degree elevation of the Bézier curve. It has been proved, see e.g. Prautzsch et al. [2002], that repeated degree elevations produce a sequence of control polygons converging to the underlying Bézier curve. An estimate of the reduction of the maximum distance is again given in Nairn et al. [1998].*

## 3 Characterisation of the semi-algebraic set

### *3.1 Existing approaches*

A full characterisation of the semi-algebraic set $\mathscr{S}(r, \mathbb{A})$ mentioned in Problem 3 would consist of a symbolic representation of each connected component defining the set and of, at least, one internal, sample point for each component. This goal can be achieved with exact, symbolic methods like the Cylindrical Algebraic Decomposition (CAD) (Arnon et al. [1984]). The CAD is a powerful method, originally developed for the existential quantifiers elimination problem, that provides a partition of the parameter space in cells described as simpler semi-algebraic sets. On each cell the polynomials defining equalities and inequalities are sign-invariant (negative, positive or zero). Among all the cells, those where all the conditions are verified are returned and for each cell a sample point is also provided.

The CAD algorithm has two main drawbacks. The first is related to its computational complexity: it is doubly exponential in the number of variables. More precisely, for $s$ polynomials of maximum degree $d$ in $n$ variables, the complexity is bounded by $(sd)^{2^{O(n)}}$ (Davenport and Heintz [1988], England and Daven [2016]). There exist many improvements and optimisations of the basic algorithm (see for instance Wilson et al. [2014] for sub-CAD and Basu et al. [2006] for a general overview of the field), which increase the practical usability of the CAD, but none of them changes its theoretical complexity. There also exists a family of algorithms based on the critical points method (see Grigoriev and Vorobjov [1988], Basu et al. [2006]) whose complexity is singly exponential.

The second drawback is the huge number of cells often produced by the partition. Even if their expression is potentially much simpler than that of the original semi-algebraic set, their number negatively affect the CAD usability.

To overcome these limitations, many numerical, not exact, approaches have been proposed in the literature. They aim to solve a large variety of problems including ascertaining the non-negativity of a polynomial on a basic semi-algebraic set [6], testing emptiness of semi-algebraic sets and computing inner or outer approxima-

[6] A basic semi-algebraic set is built in terms of only (weak) polynomial inequalities.

tions of basic semi-algebraic sets. They all consist in converting the exact problem in an approximated one, which can be efficiently solved by exploiting the semidefinite programming theory and Sum of Squares (SoS) polynomials (Laurent [2009], Lasserre [2006], Parrilo [2003]).

In this vein, an algorithm that can be exploited to solve, at least partially, the characterisation of Problem 3 has been proposed in Dabbene et al. [2017]. It is able to provide a numerical approximation (inner or outer) of a basic semi-algebraic set. Unfortunately, despite its theoretically soundness and simplicity of implementation, it often suffers from numerical problems. A way to limit such shortcomings is to add further ad-hoc constraints forcing the correct solution, but there is no systematic procedure to guide in their choice.

Moreover, all the aforementioned numerical methods have the tendency to loose the link with the symbolic constraints. That is, if a bound on a variable or a parameter on a constraint is changed, a new numerical problem has to be solved and the previous solution is of scarce (or null) use.

### *3.2 A simple partial characterisation*

We propose here a simple numerical method to provide an approximate characterisation of the semi-algebraic set in Problem 3, which has the advantage of retaining its link with the symbolic formulation and to require few computations.

#### *3.2.1 The p-norm ellipsoids*

As in Dabbene et al. [2017], we look for an inner approximation of a basic semi-algebraic set instead of the general set $\mathscr{S}(r, \mathbb{A})$ in Problem 3. More precisely, we assume that the equality constraints can be solved via symbolic variable substitutions or can be converted, up to a desired precision, to a couple of inequalities. In many practical problems the use of equality constraints is limited to the specification of some waypoints (such as the departure and arrival points). As a consequence, we do not deem their potential replacement by couples of inequalities a significant limitation. We also assume that the inequalities come naturally, or can be put, in the form $\underline{r}_i \leqslant r_i(a) \leqslant \overline{r}_i$ for some $\underline{r}_i, \overline{r}_i \in \mathbb{R}$ and $r_i \in \mathbb{R}[a]$. We stress that, in practical cases, the inequalities often come in that form and, in general, the polynomials $r_i$ always admit a finite upper and lower bound when the parallelotope $\mathbb{A}$ is bounded, namely when $\underline{a}_i, \overline{a}_i \in \mathbb{R}^m$. Thus we make the following assumption.

**Assumption 2** *There exist $M$ polynomials $r_i \in \mathbb{R}[a]$ such that the semi-algebraic set $\mathscr{S}(r, \mathbb{A})$ in (10) can be transformed in the following basic semi-algebraic set defined in terms of $2M$ inequalities*

$$\mathscr{S}(\mathbb{A}) = \{a \in \mathbb{A} \mid \underline{r}_i \leqslant r_i(a) \leqslant \overline{r}_i,\, \underline{r}_i, \overline{r}_i \in \mathbb{R},\, i = 1, \dots, M\}\,. \tag{12}$$

We want to compute an inner approximation $\mathscr{A} \subset \mathscr{S}(\mathbb{A})$. To this aim, let us define the parallelotope

$$\mathbb{P} = [\underline{r}_1, \overline{r}_1] \times \cdots \times [\underline{r}_M, \overline{r}_M] \tag{13}$$

with $\underline{r}_i, \overline{r}_i \in \mathbb{R}$ as in (12), and the vector polynomial function $R : a \mapsto (r_1(a), \dots, r_M(a))$. The inverse of $R$ is a multivalued function defined as $R^{-1} : \rho \mapsto \{a \in \mathbb{R}^{m(N+1)} \mid R(a) = \rho\}$. We can easily verify that $\mathscr{S}(\mathbb{A}) = R^{-1}(\mathbb{P}) \cap \mathbb{A}$, so, for any $\mathscr{M} \subset \mathbb{P}$ we have $\mathscr{S}(\mathbb{A}) \supset R^{-1}(\mathscr{M}) \cap \mathbb{A}$. The idea is to build a suitable set $\mathscr{M} \subset \mathbb{P}$ (or a family of them) and then to define the sought inner approximation as $\mathscr{A} = R^{-1}(\mathscr{M}) \cap \mathbb{A}$, which automatically satisfies $\mathscr{A} \subset \mathscr{S}(\mathbb{A})$. In particular, if $\mathscr{M}$ is a semi-algebraic set defined by only one inequality, then also $R^{-1}(\mathscr{M})$ is a semi-algebraic set defined by one inequality.

**Definition 3** *Let us define the unit $M$-dimensional ellipsoid in $p$-norm centred in $c = (c_1, \dots, c_M) \in \mathbb{R}^m$ as*

$$\mathscr{E}^{M,p} = \left\{ q \in \mathbb{R}^M \;\Big|\; \sum_{i=1}^{M} \frac{|q_i - c_i|^p}{b_i^p} - 1 \leqslant 0 \right\} \tag{14}$$

*for some constants $b_i > 0$ representing the semi-axes. If $p = 2d$, $d \in \mathbb{N}$ we can drop the absolute value and the ellipsoid $\mathscr{E}^{M,p}$ becomes a semi-algebraic set.*

In the following we impose $p = 2d$, $d \in \mathbb{N}$. Moreover, if we set

$$\begin{aligned} 2b_i &= \overline{r}_i - \underline{r}_i \\ 2c_i &= \overline{r}_i + \underline{r}_i \end{aligned} \tag{15}$$

we have $\mathscr{E}^{M,p} \subset \mathbb{P}$. In the light of the previous discussion, we can state the following proposition.

**Proposition 1** *Given the semi-algebraic set*

$$R^{-1}(\mathscr{E}^{M,p}) = \left\{ a \in \mathbb{R}^{m(N+1)} \;\Big|\; \sum_{i=1}^{M} \left( \frac{r_i(a) - c_i}{b_i} \right)^p - 1 \leqslant 0 \right\} \tag{16}$$

*where $r_i$ are the polynomials defining $\mathscr{S}(\mathbb{A})$ in (12), the set $\mathscr{A}^{M,p} = R^{-1}(\mathscr{E}^{M,p}) \cap \mathbb{A}$ satisfies $\mathscr{A}^{M,p} \subset \mathscr{S}(\mathbb{A})$. It is thus an approximation of $\mathscr{S}(\mathbb{A})$ parametrised by $M$ and $p$.*

The present approximation has the advantage of being directly generated with few symbolic manipulations, once the order $p$ has been fixed. Moreover, the positions (15) show clearly the link with the symbolic constraints.

If the bounds $\underline{r}_i, \overline{r}_i$ in (15) change, the approximation keeps its form and just adapts some coefficients without the need for any re-computation.

**Remark 4** *The use of the ellipsoidal approximation provides an alternative formulation to the inequality constraints expressed as ratio of polynomials. In the Remark 2 such ratios where converted in two sets of inequalities in terms of the polynomials $\tilde{r}^{\text{in}}_{ik}(a) = \underline{b}d_k(a) - n_k(a)$ and $\tilde{r}^{\text{in}}_{jk}(a) = n_k(a) - \overline{b}d_k(a)$. Working with the control points, and accepting the ensuing conservatism, leads to the inequalities $\underline{b}d_k(a) - n_k(a) \leqslant 0$ and $n_k(a) - \overline{b}d_k(a) \leqslant 0$. These inequalities can be put back in rational form as follows:*

$$\underline{b} \leqslant \frac{n_k(a)}{d_k(a)} \leqslant \overline{b}\,, \tag{17}$$

*with the further constraints $0 \leqslant d_k(a)$ to ensure the positivity of $d_k$. We notice that such positivity constraints can be dropped whenever the polynomials $d_k$ are squares or sum of squares. This way we have two-boundary inequalities that can be directly included in the semi-algebraic set (16) by setting (with a slight abuse of notation)*

$$\begin{aligned} r_k(a) &= \frac{n_k(a)}{d_k(a)} \\ 2b_i &= \overline{b} - \underline{b} \\ 2c_i &= \overline{b} + \underline{b}\,. \end{aligned} \tag{18}$$

*If a purely polynomial representation of the set $R^{-1}(\mathscr{E}^{M,p})$ is desired, the inequality in (16) can be put in rational form through the computation of the LCD and the denominator discarded due to its positivity.*

### 3.2.2 Volume coverage

**Proposition 2** *Given $M \in \mathbb{N}$, the sets $\mathscr{A}^{M,p} = R^{-1}(\mathscr{E}^{M,p}) \cap \mathbb{A}$ constitute a family of approximating sets for $\mathscr{S}(\mathbb{A})$ parametrised in $p$. In other words, $\mathscr{A}^{M,p_1} \subseteq \mathscr{A}^{M,p_2}$ for any $2 \leqslant p_1 \leqslant p_2 \in \mathbb{N}$ and*

$$\lim_{p\to\infty} \mathscr{A}^{M,p} = \mathscr{S}(\mathbb{A})\,.$$

**PROOF.** By definition, the ellipsoids $\mathscr{E}^{M,p}$ for fixed $M$ are such that $\mathscr{E}^{M,p_1} \subseteq \mathscr{E}^{M,p_2}$ for any $2 \leqslant p_1 \leqslant p_2 \in \mathbb{N}$. Moreover $\lim_{p\to\infty} \mathscr{E}^{M,p} = \mathbb{P}$. By continuity of $R^{-1}(\cdot)$ we have $\mathscr{A}^{M,p_1} \subseteq \mathscr{A}^{M,p_2}$ for any $2 \leqslant p_1 \leqslant p_2 \in \mathbb{N}$ and

$$\lim_{p\to\infty} \mathscr{A}^{M,p} = \lim_{p\to\infty} R^{-1}(\mathscr{E}^{M,p}) \cap \mathbb{A} = R^{-1}(\mathbb{P}) \cap \mathbb{A} = \mathscr{S}(\mathbb{A})\,.$$

The previous proposition ensures that, if $\mathscr{S}(\mathbb{A})$ is not empty, then there exists $\bar{p} \geqslant 2$ such that for any $p \geqslant \bar{p}$, the set $\mathscr{A}^{M,p}$ is not empty too, hence the approximation $\mathscr{A}^{M,p}$ provides a set of feasible solutions. The previous assertions can be made more precise by considering firstly the volume of the ellipsoid $\mathscr{E}^{M,p}$ (see for instance Wang [2005])

$$V(\mathscr{E}^{M,p}) = \frac{2^M \mathbf{\Gamma}^M\left(\frac{1}{p}+1\right)}{\mathbf{\Gamma}\left(\frac{M}{p}+1\right)} \prod_{i=1}^{M} c_i$$

where $\mathbf{\Gamma}(\cdot)$ is the gamma function and secondly the one of the parallelotope $\mathbb{P}$

$$V(\mathbb{P}) = 2^M \prod_{i=1}^{M} c_i\,.$$

The ratio between these two volumes is given by

$$\varrho(M,p) = \frac{\mathbf{\Gamma}^M\left(\frac{1}{p}+1\right)}{\mathbf{\Gamma}\left(\frac{M}{p}+1\right)}$$

and, for fixed $M$, clearly $\lim_{p\to\infty} \varrho(M,p) = 1$. Unfortunately, if we fix $p$ and we compute $\lim_{M\to\infty} \varrho(M,p) = 0$ we verify that, for increasing dimension, the volume of a parallelotope is concentrated at the corners. Hence, the higher the dimension $M$, the higher $p$ must be to capture a relevant part of the volume of $\mathbb{P}$. To give an idea of how much $p$ must increase when $M$ increases, let us as consider a linear relation between the two: $p = kM$ for some $k \in \mathbb{N}$ and compute the limit

$$\varrho(k) = \lim_{M\to\infty} \varrho(M,kM) = \frac{\mathrm{e}^{-\frac{\gamma}{k}}}{\mathbf{\Gamma}\left(\frac{1}{k}+1\right)}$$

where $\gamma = -\mathbf{\Gamma}'(1)$ is the Euler-Mascheroni constant. Being $\varrho(M,kM) \geqslant \varrho(k)$ for any $M$, $\varrho(k)$ provides a lower bound on the percentage of coverage of the volume of $\mathbb{P}$. For instance, $\varrho(1) \approx 0.56$, $\varrho(2) \approx 0.84$ and $\varrho(3) \approx 0.92$.

### 3.2.3 Internal sample points

The problem with this approximation is that, even if there exists a $\bar{p} \geqslant 2$ such that for any $p \geqslant \bar{p}$, the set $\mathscr{A}^{M,p}$ is not empty, there is no general rule to compute such $\bar{p}$. Generally speaking, $\mathscr{A}^{M,p}$ is an approximation of $\mathscr{S}(\mathbb{A})$, but there is no assurance of $\mathscr{A}^{M,p}$ being not empty.

To complete the characterisation of the semi-algebraic set, hence to solve the Problem 3, we need to establish whether the set is not empty and, in this case, to provide at least an internal sample point. The CAD method is of no use now as the number of variables is not changed, but the fact of having the set $R^{-1}(\mathscr{E}^{M,p})$ defined by only one polynomial allows us to use simpler numerical methods. More precisely, we can pose the following minimisation problem $\min_{a\in\mathbb{A}} \sum_{i=1}^{M} \left(\frac{r_i(a)-c_i}{b_i}\right)^p - 1$ and look for a local minimiser $a^*$. If we can find a negative minimum value, then the set is not empty and the corresponding $a^*$ is the required sample point. The aforementioned minimisation problem has constraints of a very simple form, namely only upper and lower bounds on the variables. Therefore, a simple algorithm can be employed, whose core is a standard line search method based, for instance, on the *gradient descent* (Nocedal and Wright [Chapt. 3, 2006]). The constraints can be accounted for by simply modifying the search direction according to the gradient projection method (Nocedal and Wright [Chapt. 16, 2006]). It is important to remark that such a minimum point can be computed quite efficiently since the gradient (and Hessian if needed) can be symbolically provided to the minimisation algorithm. Unfortunately, if the minimum value is positive, there is no assurance of the emptiness of the set as the defining polynomial has in general many minima. Multi-start approaches are beneficial in this case to improve the chance of finding the global minimum. If a negative minimum cannot be found, the order $p$ of the polynomial can be increased and the search repeated.

**Remark 5 (On-line trajectory re-computation)** *The advantage of retaining the link with the symbolic constraints becomes particularly relevant when an on-line re-computation of a trajectory is necessary to cope with a change in the constraints. If some of the bounds* $\underline{r}_i, \overline{r}_i$ *in (15) change, the previously computed trajectory could be no longer valid; but the formal expression of the set* $\mathscr{A}^{M,p}$ *is not changed and it can still provide feasible trajectories. A quick on-line execution of the gradient descent algorithm can provide a new internal sample point, thus a valid trajectory.*

#### *3.2.4 Local approximations*

As mentioned before, with the generic ellipsoidal approximation there is no assurance that for a fixed $p$ the set $\mathscr{A}^{M,p}$ is not empty. This lack is due to the fact that the centre $c$ of the ellipsoid defined in (15) or (18) is in the middle of the parallelotope $\mathbb{P}$, but there is no reason for this point to be the image of a valid solution. In fact, quite often the opposite is true: $R^{-1}(c) \notin \mathscr{S}(\mathbb{A})$. On the contrary, if an internal sample point $s \in \mathscr{S}(\mathbb{A})$ is known, it is possible to build a non-empty set $\mathscr{A}^{M,p}$ for any order $p$ (even just 2). The trick is to centre the ellipsoidal approximation in the image of the point $s$ along the map $R$. More precisely, we define the new centre and the semi-axes as

$$\begin{aligned} c &= (c_1, \ldots, c_M) = R(s) \\ b_i &= \min(c_i - \underline{r}_i, \overline{r}_i - c_i) \end{aligned}$$

(or $b_i = \min(c_i - \underline{b}, \overline{b} - c_i)$ if we consider inequalities of the type (17)). The semi-axes are defined so as to ensure that $\mathscr{E}^{M,p} \subset \mathbb{P}$. Actually $\mathscr{E}^{M,p}$ is now a local ellipsoidal approximation of a portion of $\mathbb{P}$. Notice that, the point $s$ belongs to the ensuing set $R^{-1}(\mathscr{E}^{M,p})$ by construction, hence $\mathscr{A}^{M,p}$ is not empty whatever $p$ is.

#### *3.2.5 Robustness of solutions*

An interesting aspect of the present approximation is related to the possibility of performing a robustness analysis of the solutions with respect to individual constraints. Assume that $a^* \in \mathscr{A}^{M,p}$ is a solution to Problem 3 and $\tilde{r}$ is one of the constraint bounds $\underline{r}_k, \overline{r}_k$ for any fixed $k$. The constants $b_k$ and $c_k$ in (15) are the only ones to be functions of $\tilde{r}$. The defining polynomial in (16) computed in $a^*$ is now a mono-variate function (actually a ratio of polynomials or a polynomial) of the variable $\tilde{r}$:

$$l_{a^*}(\tilde{r}) = \sum_{i=1}^{M} \left( \frac{r_i(a^*) - c_i(\tilde{r})}{b_i(\tilde{r})} \right)^p - 1 \tag{19}$$

The function $l_{a^*}(\cdot)$ can be easily exploited to evaluate the validity of the solution $a^*$ with respect to variations of $\tilde{r}$. It is important to remark, however, the conservativeness of such a robustness analysis. The solution $a^*$ can still be valid for a certain $\tilde{r}$ even if it fails the robustness check for that $\tilde{r}$. The robustness analysis is indeed related to the set $\mathscr{A}^{M,p}$, namely the set produced by the chosen ellipsoidal approximation, not to the set $\mathscr{S}(\mathbb{A})$. If the test fails for some fixed $M$ and $p$, then $a^* \notin \mathscr{A}^{M,p}$, but it is still possible that $a^* \in \mathscr{S}(\mathbb{A})$.

## 4 Numerical test cases

In the present section we will illustrate the steps involved in the solution of Problem 3 via two different mechanical systems.

### *4.1 The unicycle*

A kinematic model of the unicycle mobile robot is given by

$$\begin{aligned} \dot{x} &= v \cos\theta \\ \dot{y} &= v \sin\theta \\ \dot{\theta} &= \omega \end{aligned} \tag{20}$$

where $(x, y)$ represents the position of the cart with respect to an inertial frame, $\theta$ is the orientation, that is the angle between the $x$-axis and the heading direction measured counterclockwise. The control inputs are the linear velocity $v > 0$ and the angular velocity $\omega$. The model is flat with respect to the outputs $x$ and $y$. The flat parameterisation (4) is given by

$$\theta = \psi(x, \dot{x}, y, \dot{y}) = \arctan \frac{\dot{y}}{\dot{x}} \tag{21a}$$

$$v = \zeta_1(x, \dot{x}, y, \dot{y}) = \sqrt{\dot{x}^2 + \dot{y}^2} \tag{21b}$$

$$\omega = \zeta_2(x, \dot{x}, \ddot{x}, y, \dot{y}, \ddot{y}) = \frac{\ddot{y}\dot{x} - \ddot{x}\dot{y}}{\dot{x}^2 + \dot{y}^2}\,. \tag{21c}$$

We remark that, being $v > 0$, both $\theta$ and $\omega$ are well defined.

#### *4.1.1 Test case 1*

In the first case we consider a unicycle moving from a given initial position and orientation at time $t = 0$ to a given final position and orientation at time $t = 1$. We assume limited control authorities both in $v$ and $\omega$. The equality and inequality constraints (2) thus read

$$\begin{aligned} x(0) &= x_0 & x(1) &= x_{\mathrm{f}} \\ y(0) &= y_0 & y(1) &= y_{\mathrm{f}} \\ \theta(0) &= \theta_0 & \theta(1) &= \theta_{\mathrm{f}} \end{aligned}$$

$$\begin{aligned} \underline{v} &\leqslant v \leqslant \overline{v} \\ \underline{\omega} &\leqslant \omega \leqslant \overline{\omega}\,. \end{aligned}$$

For the flat outputs we choose cubic Bézier curves

$$x(t) = \sum_{j=0}^{3} X_j B_{j3}(t), \qquad y(t) = \sum_{j=0}^{3} Y_j B_{j3}(t)\,.$$

Bézier curves are often represented in terms of the vectors of their control points: $X = [X_0, X_1, X_2, X_3]$ and $Y = [Y_0, Y_1, Y_2, Y_3]$. Using this notation, we can give an idea of the transformation induced by the derivative operations on the control points:

$$\begin{aligned} \dot{X} &= 3[X_1 - X_0,\, X_2 - X_1,\, X_3 - X_2] \\ \ddot{X} &= 6[X_0 - 2X_1 + X_2,\, X_1 - 2X_2 + X_3] \end{aligned}$$

where with $\dot{X}$ and $\ddot{X}$ we denoted the vectors of control points of $\dot{x}$ and $\ddot{x}$ respectively.

The equality constraints on the initial and final positions directly fix some control points as: $X_0 = x_0$, $X_3 = x_{\mathrm{f}}$, $Y_0 = y_0$ and $Y_3 = y_{\mathrm{f}}$. Imposing the constraints on the orientation is slightly more involved since the parameterisation (21a) is not polynomial. Let us consider the case $\theta_0$ ($\theta_{\mathrm{f}}$ is similar). We translate the original equality constraint in an equality and an inequality constraints in terms of the control points of the flat outputs. If $\theta_0 = \frac{\pi}{2}$ we pose $\dot{x}(0) = 0$ and $\dot{y}(0) > 0$. If $\theta_0 = \frac{3\pi}{2}$ we pose $\dot{x}(0) = 0$ and $\dot{y}(0) < 0$. If $\theta_0 \in \left(\frac{\pi}{2}, \frac{3\pi}{2}\right)$ we pose $\dot{y}(0) = \dot{x}(0)\tan\theta_0$ and $\dot{x}(0) < 0$. Finally, if $\theta_0 \in \left[0, \frac{\pi}{2}\right) \cup \left(\frac{3\pi}{2}, 2\pi\right)$ we pose $\dot{y}(0) = \dot{x}(0)\tan\theta_0$ and $\dot{x}(0) > 0$. As an example let us consider $\theta_0 = 0$. In this case we have $\dot{y}(0) = 0$ and $\dot{x}(0) > 0$, which directly translates in terms of control points to: $3(Y_1 - Y_0) = 0$ and $3(X_1 - X_0) > 0$. If a two-boundary inequality of the kind considered in (12) is required, an upper bound on $3(X_1 - X_0)$ can be easily added either assuming the parallelotope $\mathbb{A}$ in (11) is bounded or by recalling that $|\dot{x}(0)| < |v|$.

The inequality constraints for $v$ can be easily written as (recall $\underline{v}$ has to be positive by hypothesis) $\underline{v}^2 \leqslant \dot{x}^2 + \dot{y}^2 \leqslant \overline{v}^2$, which has to be satisfied for each control points of $v$. As a result we have a set of five quadratic two-boundary inequalities. For example, the one for the first control point of $v^2$ is given by: $\underline{v}^2 \leqslant 9(X_0 - X_1)^2 + 9(Y_0 - Y_1)^2 \leqslant \overline{v}^2$.

The inequality constraints for $\omega$ are expressed as a ratio of polynomials, so we make reference to Remark 4 to produce a set of five two-boundary inequalities of the form $\underline{\omega} \leqslant \frac{N_i}{D_i} \leqslant \overline{\omega}$, where $N_i$ and $D_i$ are the control points associated to the Bézier functions at the numerator and denominator of the ratio (21c). As an example we report here the first control points $N_0 = 18(Y_0 - Y_1)(X_0 - 2X_1 + X_2) - 18(X_0 - X_1)(Y_0 - 2Y_1 + Y_2)$ and $D_0 = 9(X_0 - X_1)^2 + 9(Y_0 - Y_1)^2$.

In order to make more concrete the present test case, let us fix the following values for the constraints:

$$\begin{aligned} x_0 &= 0 & x_{\mathrm{f}} &= 1 \\ y_0 &= 0 & y_{\mathrm{f}} &= 1/2 \\ \theta_0 &= 0 & \theta_{\mathrm{f}} &= 0 \\ \underline{v} &= 0 & \overline{v} &= \sqrt{10} \\ \underline{\omega} &= -10 & \overline{\omega} &= 10 \end{aligned}$$

and the following parallelotope for the control points: $\mathbb{A} = [-1, 1]^8$. Once the equality constraints have been removed by symbolic variable substitutions, the (basic) semi-algebraic set $\mathscr{S}(r, \mathbb{A})$ in (10) is made up of 22 polynomial inequalities of maximal degree 2 in the two variables $X_1$ and $X_2$.

In this simple case the CAD is able to provide an elegant solution (see the blue triangle in figure 1) to Problem 3 given by:

$$\frac{1}{10} < X_1 < \frac{17}{20}, \quad \frac{1}{20} + X_1 < X_2 < \frac{9}{10}\,.$$

Concerning the ellipsoidal approximation described in Section 3.2, the set $\mathscr{S}(\mathbb{A})$ in (12) is made up of $M = 12$ two-boundary inequalities (i.e. 24 single inequalities), 5 of which expressed as ratios of polynomials and 7 expressed as polynomials, all in the two variables $X_1$ and

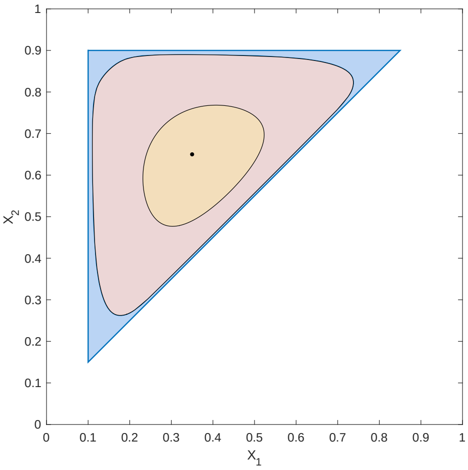

Figure 1. Solution domains for the test case 1. The basic semi-algebraic set $\mathscr{S}(\mathbb{A})$ is in blue, the generic ellipsoidal approximation of order $p = 20$ is in red and that of order $p = 8$ is in orange. The black dot represents the sample point $s \approx [0.34,\ 0.65]$ computed by the gradient descent algorithm.

$X_2$. Two inner approximations $\mathscr{A}^{M,p} = R^{-1}(\mathscr{E}^{M,p}) \cap \mathbb{A}$ of $\mathscr{S}(\mathbb{A})$ are shown in figure 1, one for $p = 8$ ($\varrho(M,p) \approx 0.37$) in orange, the other for $p = 20$ ($\varrho(M,p) \approx 0.81$) in red. The gradient descent algorithm initialised with starting point $s_0 = [1,\ 10^{-5}]$ produces a valid internal sample point: $s \approx [0.34,\ 0.65]$ (see the black dot in figure 1). It is worth noting that the initial point $s_0 \in \mathbb{A}$ but is not a solution: $s_0 \notin \mathscr{S}(\mathbb{A})$.

In order to improve the approximation of $\mathscr{S}(\mathbb{A})$ without increasing the order of the ellipsoid, one can look for a local approximation as described in Section 3.2.4. Defining the sample point $s$ as the new centre and the semi-axes accordingly (see again Section 3.2.4), a suitable approximation of order $p = 4$ is found (see the green set in figure 2). It is worth noting that, despite its low order, such local approximation is even better than the general one of order $p = 8$.

We can perform a robustness analysis of the solution $s \approx [0.34,\ 0.65]$ with respect to variations of the bounds $\overline{v}$, $\underline{\omega}$ and $\overline{\omega}$ according to the method described in section 3.2.5. We use the ellipsoidal approximation of order $p = 8$ and we consider two independent variations in $\overline{v}$ and in $\tilde{\omega} > 0$ such that $\underline{\omega} = -\tilde{\omega}$ and $\overline{\omega} = \tilde{\omega}$. The functions $l_s(\overline{v})$ and $l_s(\tilde{\omega})$ in (19) turn out to be a ratio of polynomials with an always positive denominator. We focus now only on the numerators $n_{l_s}(\overline{v})$ and $n_{l_s}(\tilde{\omega})$ of the two functions. $n_{l_s}(\tilde{\omega})$ takes on the very simple form $n_{l_s}(\tilde{\omega}) \approx -0.2987\tilde{\omega}^8 + 6255.66$, which leads to the validity interval $\tilde{\omega} > 3.4684$. $n_{l_s}(\overline{v})$ takes on a more complicated form of a polynomial of order 8 in the variable $\overline{v}^2$. By numerically computing the roots, one finds the following validity interval $1.5269 < \overline{v} < 3.5055$. The robustness test ensures that the solution $s \approx [0.34,\ 0.65]$ can be used, without any re-computation, even if one of the bounds $\overline{v}$, $\underline{\omega}$ and $\overline{\omega}$ varies according to the computed validity margins.

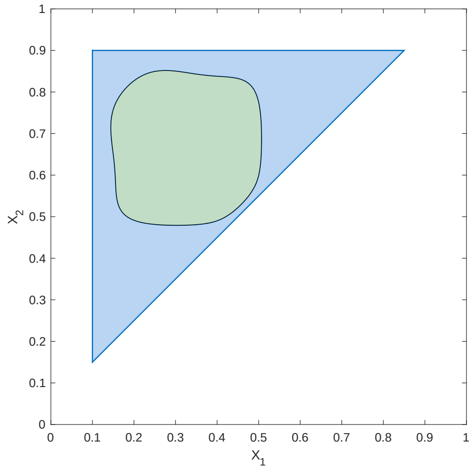

Figure 2. Solution domains for the test case 1. The basic semi-algebraic set $\mathscr{S}(\mathbb{A})$ is in blue, the local approximation of order $p = 4$ computed with respect to the sample point $s = [0.34,\ 0.65]$ is in green.

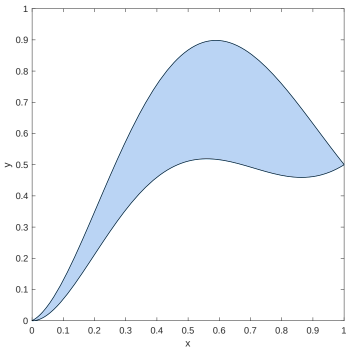

Figure 3. Bounds for the $x$ and $y$ states in test case 2.

### *4.1.2 Test case 2*

For the second test case we show how the present approach allows to impose constraints to the state variables and control inputs in a unified fashion. This is particularly relevant when performing, for instance, obstacle avoidance.

We choose quintic Bézier curves for the $x$ and $y$ variables and we impose them to stay in the interior of the region bounded by two Bézier curves as depicted in figure 3. Such constraints directly translate in terms of control points for the $x$ and $y$ variables as follows:

$$
\begin{aligned}
X &= \left[0,\ \frac{1}{10},\ \frac{1}{5},\ \frac{4}{5},\ \frac{9}{10},\ 1\right] \\
\underline{Y} &\leqslant Y \leqslant \overline{Y} \\
\underline{Y} &= \left[0,\ -\frac{1}{10^3},\ \frac{9}{10},\ \frac{1}{2},\ \frac{3}{8},\ \frac{1}{2}\right] \\
\overline{Y} &= \left[0,\ \frac{1}{10},\ \frac{6}{5},\ \frac{11}{10},\ \frac{3}{4},\ \frac{1}{2}\right].
\end{aligned}
$$

Notice that, with the present constraints, the initial and final positions of the vehicle are the same of test case 1. Concerning the initial and final orientation we strictly fix the former and we provide a bound for the latter:

$$
\begin{aligned}
\theta_0 &= 0 \\
-\frac{\pi}{3} &\leqslant \theta_\mathrm{f} \leqslant \frac{\pi}{3}.
\end{aligned}
$$

The constraints on the control inputs are:

$$
\begin{aligned}
\underline{v} &= 0 & \overline{v} &= 10 \\
\underline{\omega} &= -50 & \overline{\omega} &= 50.
\end{aligned}
$$

Since the control points for $x$ are completely fixed, the parallelotope $\mathbb{A}$ has the form: $\mathbb{A} = \prod_{i=0}^{5}[\underline{Y}_i,\ \overline{Y}_i]$.

Once the equality constraints have been removed by symbolic variable substitutions, the (basic) semi-algebraic set $\mathscr{S}(r,\mathbb{A})$ in (10) is made up of 47 polynomial inequalities (7 of degree 0, 12 of degree 1 and 28 of degree 2) in the three variables $Y_2$, $Y_3$ and $Y_4$. In this case we did not look for a solution with the CAD algorithm, but we directly computed an ellipsoidal approximation. The set $\mathscr{S}(\mathbb{A})$ in (12) is made up of $M = 20$ two-boundary inequalities (once those of degree zero have been removed being automatically satisfied), 7 of which expressed as ratios of polynomials and 13 expressed as polynomials. All the polynomials are of maximal degree 2 in the three variables $Y_2$, $Y_3$ and $Y_4$. A numerical representation of $\mathscr{S}(\mathbb{A})$, obtained by fine gridding the parameter space, is shown in blue in figure 4.

In order to compute the approximation $\mathscr{A}^{M,p} = R^{-1}(\mathscr{E}^{M,p}) \cap \mathbb{A}$ we chose here $p = 80$ ($\varrho(M,p) \approx 0.96$) obtaining the set represented in red in figure 4. The gradient descent algorithm initialised with starting point $s_0 = 1/2[\underline{Y}_2 + \overline{Y}_2,\ \underline{Y}_3 + \overline{Y}_3,\ \underline{Y}_4 + \overline{Y}_4]$ produces a valid internal sample point: $s \approx [0.92,\ 0.93,\ 0.66]$. It is worth noting that the initial point $s_0 \in \mathbb{A}$ but is not a solution: $s_0 \notin \mathscr{S}(\mathbb{A})$.

Despite the high order ($p = 80$) and the high coverage index of the parallelotope $\mathbb{P}$, the approximation $\mathscr{A}^{M,p}$ is appreciably smaller than $\mathscr{S}(\mathbb{A})$. The reason is that the solution set is manly located in a corner of $\mathbb{P}$. In order to significantly lower the degree of the approximation, one

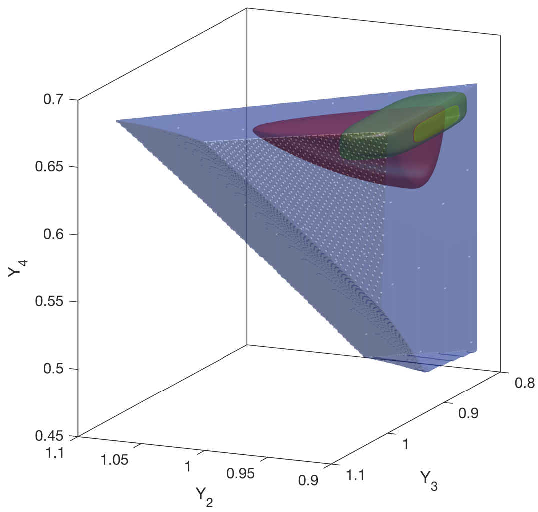


Figure 4. Solution domains for the test case 2. The basic semi-algebraic set $\mathscr{S}(\mathbb{A})$ is in blue, the generic ellipsoidal approximation of order $p = 80$ is in red, the local approximation of order $p = 4$ computed with respect to the sample point $s = [0.92,\ 0.93,\ 0.66]$ is in green.

can look for a local approximation as discussed in Section 3.2.4. Defining $s$ as the new centre and the semi-axes accordingly (see again Section 3.2.4), a suitable approximation of order $p = 4$ is found (see the green surface in figure 4). An even better approximation is found by choosing a slightly different sample point $s = [0.95,\ 0.95,\ 0.6]$ and $p = 8$ as shown in figure 5. More than one local approximation can be computed, if other sample points are known, and a partial covering of the set $\mathscr{S}(\mathbb{A})$ can thus be obtained.

### *4.2 The planar manipulator*

The dynamics of a planar manipulator can be written as (see Murray et al. [1994])

$$
D(q)\ddot{q} + C(q,\dot{q})\dot{q} = \tau
$$

where $\tau = [\tau_1, \tau_2]$ are the couples at the joints and $q = [\theta_1, \theta_2]$ with $\theta_1$ the angle of the first link w.r.t. the horizontal axis measured CCW and $\theta_2$ is the angle of the second link w.r.t. the axis of the first link measured CCW. The matrices are

$$
\begin{aligned}
D(q) &= \begin{pmatrix} \alpha + 2\beta c_2 & \delta + \beta c_2 \\ \delta + \beta c_2 & \delta \end{pmatrix} \\
C(q,\dot{q}) &= \begin{pmatrix} -\beta s_2 \dot{\theta}_2 & -\beta s_2(\dot{\theta}_1 + \dot{\theta}_2) \\ \beta s_2 \dot{\theta}_1 & 0 \end{pmatrix}
\end{aligned}
$$

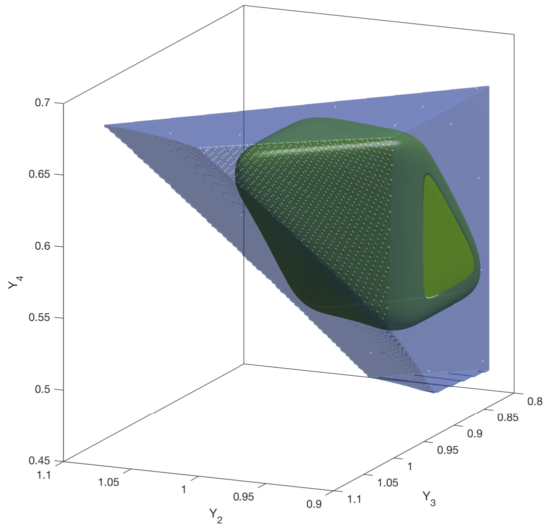


Figure 5. Solution domains for the test case 2. The basic semi-algebraic set $\mathscr{S}(\mathbb{A})$ is in blue, the local approximation of order $p = 8$ computed with respect to the sample point $s = [0.95,\ 0.95,\ 0.6]$ is in green.

where $c_2 = \cos\theta_2$ and $s_2 = \sin\theta_2$. The position of the end–effector is given by

$$\begin{aligned} x_2 &= \ell_1 \cos\theta_1 + \ell_2 \cos(\theta_1 + \theta_2) \\ y_2 &= \ell_1 \sin\theta_1 + \ell_2 \sin(\theta_1 + \theta_2)\,. \end{aligned}$$

For the simulations we choose: $\ell_1 = 0.5\,\mathrm{m}$, $\ell_2 = 0.3\,\mathrm{m}$, $\alpha \approx 0.18428\,\mathrm{kg\,m^2}$, $\beta = 0.027\,\mathrm{kg\,m^2}$, $\delta \approx 0.01082\,\mathrm{kg\,m^2}$. The system is flat w.r.t. the outputs $\theta_1$, $\theta_2$.

This second example show how the present approach allows to impose constraints to the state variables and control inputs in a unified fashion. More specifically, we want the manipulator to move from the initial position $\theta_1(0) = \dot\theta_1(0) = \ddot\theta_1(0) = \dot\theta_2(0) = \ddot\theta_2(0) = 0$, $\theta_2(0) = \frac{\pi}{6}$ to the final position $\theta_1(1) = \frac{\pi}{2}$, $\theta_2(1) = \frac{11\pi}{6}$, $\dot\theta_1(1) = \ddot\theta_1(1) = \dot\theta_2(1) = \ddot\theta_2(1) = 0$, with the end–effector avoiding a circular obstacle of radius $\bar R = 0.15\,\mathrm{m}$ centred in $[x_o, y_o] = (\ell_1 + \ell_2)[\cos\frac{\pi}{4}, \sin\frac{\pi}{4}]$. We also impose upper and lower bounds on $\theta_i$ and $\tau_i$, $i = 1, 2$. The inequality constraints are hence:

$$\begin{aligned} 0 \le \theta_1 \le \frac{\pi}{2} \qquad & \qquad 0 \le \theta_2 \le 2\pi \\ |\tau_1| \le 30 \qquad & \qquad |\tau_2| \le 10 \end{aligned}$$

and

$$\bar R^2 \le (x_2 - x_o)^2 + (y_2 - y_o)^2 \le (\ell_1 + \ell_2)^2\,.$$

The expressions for $x_2$, $y_2$, $\tau_1$ and $\tau_2$ are clearly not polynomials, so we approximate the trigonometric functions with polynomials whose coefficients are found with a least square method. More precisely, we found for $\sin\theta_1$, $\cos\theta_1$, $\sin\theta_2$, $\cos\theta_2$ an approximation in terms of polynomials of order 3, 4, 5, 6 respectively, with maximum absolute error 0.00236, 0.00019, 0.01436, 0.00302 respectively. We remark that the smaller the interval of the angle, the smaller the order of the approximating polynomials.
We choose a Bézier of order 5 for $\theta_1$ and of order 8 for $\theta_2$. The equality constraints associated to the initial and final points produce the following vectors of control points:

$$\begin{aligned} \Theta_1 &= \left[0,\ 0,\ 0,\ \frac{\pi}{2},\ \frac{\pi}{2},\ \frac{\pi}{2}\right] \\ \Theta_2 &= \left[\frac{\pi}{6},\ \frac{\pi}{6},\ \frac{\pi}{6},\ X_3,\ X_4,\ X_5,\ \frac{11\pi}{6},\ \frac{11\pi}{6},\ \frac{11\pi}{6}\right] \end{aligned}$$

where $X_3$, $X_4$ and $X_5$ are free parameters. As suggested in Remark 3, we reduce the conservatism of the Bézier for $\theta_2$ by a degree elevation from 8 to 10, which produces the control vector

$$\begin{aligned} \tilde\Theta_2 = \Big[&\frac{\pi}{6},\ \frac{\pi}{6},\ \frac{\pi}{6},\ \frac{7X_3}{15} + \frac{4\pi}{45},\ \frac{8X_3}{15} + \frac{X_4}{3} + \frac{\pi}{45}, \\ &\frac{2X_3}{9} + \frac{5X_4}{9} + \frac{2X_5}{9},\ \frac{X_4}{3} + \frac{8X_5}{15} + \frac{11\pi}{45}, \\ &\frac{7X_5}{15} + \frac{44\pi}{45},\ \frac{11\pi}{6},\ \frac{11\pi}{6},\ \frac{11\pi}{6}\Big] \end{aligned}$$

By the constraints on $\theta_2$ and the analysis of the vector $\tilde\Theta_2$, the parallelotope for the free control points is chosen as $\mathbb{A} = \left[-\frac{4\pi}{21}, \frac{86\pi}{21}\right] \times \left[-\frac{688\pi}{105}, \frac{131\pi}{21}\right] \times \left[-\frac{44\pi}{21}, \frac{46\pi}{21}\right]$.

The (basic) semi-algebraic set $\mathscr{S}(r, \mathbb{A})$ in (10) is made up of 500 polynomial inequalities (22 of degree 0, the other 478 ranging from degree 1 to 12) in the three variables $X_3$, $X_4$ and $X_5$. In this case we did not look for a solution with the CAD algorithm, but we directly computed an ellipsoidal approximation. The set $\mathscr{S}(\mathbb{A})$ in (12) is made up of $M = 239$ two-boundary inequalities, once those of degree zero have been removed being automatically satisfied. A numerical representation of $\mathscr{S}(\mathbb{A})$, obtained by fine gridding the parameter space, is shown in blue in figure 6. In order to compute the approximation $\mathscr{A}^{M,p} = R^{-1}(\mathscr{E}^{M,p}) \cap \mathbb{A}$ we chose here $p = 100$ ($\varrho(M,p) = 0.09$) obtaining the set represented in red in figure 6. An approximation of order $p = 300$ ($\varrho(M,p) = 0.68$, not shown here) produces an almost perfect coverage of $\mathscr{S}(\mathbb{A})$. The gradient descent algorithm initialised with starting point $s_0 = [1,\ 1,\ 1]$ produces a valid internal sample point: $s \approx [7.16,\ 3.05,\ -0.90]$. It is worth noting that the initial point $s_0 \in \mathbb{A}$ but is not a solution: $s_0 \notin \mathscr{S}(\mathbb{A})$.

In order to significantly lower the degree of the approximation, one can look for a local approximation as discussed in Section 3.2.4. Defining $s$ as the new centre and the semi-axes accordingly (see again Section 3.2.4), a

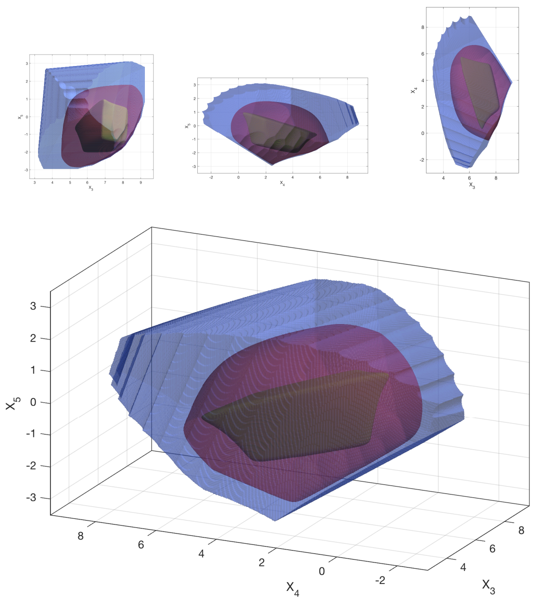


Figure 6. Solution domains for the planar manipulator. The basic semi-algebraic set $\mathscr{S}(\mathbb{A})$ is in blue, the generic ellipsoidal approximation of order $p = 100$ is in red, the local approximation of order $p = 10$ computed with respect to the sample point $s \approx [7.16,\ 3.05,\ -0.90]$ is in green.

suitable approximation of order $p = 10$ is found (see the green surface in figure 6).

## 5 Conclusions

n this paper we investigate the problem of characterising the set of feasible trajectories satisfying constraints on the input, the state and their derivatives. We exploit the flatness property and a parametrisation based on Bézier functions to reduce the complexity of the problem to a finite dimension one. Other than providing a unified formulation encompassing a wide variety of different types of constraints, we present a simple symbolic approximation of the set of feasible constrained trajectories.

Future works will investigate how the present characterisation can be exploited to build other relevant sets, such as for instance the actuated energy set, and to analyse some system properties (e.g. sensitivity to parameters and constraint bounds, cross-influence of physical quantities like curvature or snap on energy consumption).

## References

D. S. Arnon, G. E. Collins, and S. McCallum. Cylindrical algebraic decomposition I: The basic algorithm. *SIAM Journal on Computing*, 13(4):865–877, 1984.

S. Basu, R. Pollack, and M.-F. Roy. *Algorithms in Real Algebraic Geometry*, volume 10 of *Algorithms and Computation in Mathematics*. Springer-Verlag, 2006.

M. Bekcheva. Constraints on nonlinear finite dimensional flat systems. ArXiv, 2020. URL `https://arxiv.org/pdf/2011.05775.pdf`.

A. Chelouah. Extensions of differential flat fields and Liouvillian systems. In *Proceedings of the 36th IEEE Conference on Decision and Control*, 1997.

V. Cichella, I. Kaminer, C. Walton, and N. Hovakimyan. Optimal motion planning for differentially flat systems using Bernstein approximation. *IEEE Control Systems Letters*, 2(1):181–186, 2018.

F. Dabbene, D. Henrion, and C. M. Lagoa. Simple approximations of semialgebraic sets and their applications to control. *Automatica*, 78:110–118, 2017.

J. H. Davenport and J. Heintz. Real quantifier elimination is doubly exponential. *Journal of Symbolic Computation*, 5(1-2):29–35, 1988.

M. England and J. H. Daven. The complexity of cylindrical algebraic decomposition with respect to polynomial degree. In *Int. Workshop on Computer Algebra in Scientific Computing*, pages 172–192. Springer, 2016.

N. Faiz, S. K. Agrawal, and R. M. Murray. Trajectory planning of differentially flat systems with dynamics and inequalities. *AIAA Journal of Guidance, Control, and Dynamics*, 24(2):219–227, 2001.

R. T. Farouki and V. T. Rajan. Algorithms for polynomials in Bernstein form. *Computer Aided Geometric Design*, 5(1):1–26, 1988.

T. Faulwasser, V. Hagenmeyer, and R. Findeisen. Constrained reachability and trajectory generation for flat systems. *Automatica*, 50(4):1151–1159, 2014.

M. Fliess, J. Lévine, P. Martin, and P. Rouchon. Flatness and defect of non-linear systems: introductory theory and examples. *International journal of control*, 61(6):1327–1361, 1995.

M. Fliess, J. Lévine, P. Martin, and P. Rouchon. A Lie-Backlund approach to equivalence and flatness of nonlinear systems. *IEEE Transactions on Automatic Control*, 44(5):922–937, 1999.

M. E. Flores and M. B. Milam. Trajectory generation for differentially flat systems via NURBS basis functions with obstacle avoidance. In *American Control Conference*. IEEE, 2006.

D. Grigoriev and N. Vorobjov. Solving systems of polynomial inequalities in subexponential time. *Journal of Symbolic Computation*, 5(1/2):37–64, 1988.

V. Hagenmeyer and E. Delaleau. Robustness analysis with respect to exogenous perturbations for flatness-based exact feedforward linearization. *IEEE Transactions on Automatic Control*, 55(3):727–731, 2010.

D. Henrion and J.-B. Lasserre. LMIs for constrained polynomial interpolation with application in trajectory planning. *Systems & Control Letters*, 55(6):473–477, 2006.

N. Joubert, M. Roberts, A. Truong, F. Berthouzoz, and P. Hanrahan. An interactive tool for designing quadrotor camera shots. *ACM Transactions on Graphics*, 34(6):1–11, 2015.

A. Kaldmäe, Ü. Kotta, A. Shumsky, and A. Zhirabok. Feedback linearization of possibly non-smooth systems. *Proceedings of the Estonian Academy of Sciences*, 66(2):109, 2017.

J. B. Lasserre. A sum of squares approximation of nonnegative polynomials. *SIAM Journal on Optimization*, 16(3):751–765, 2006.

M. Laurent. *Sums of Squares, Moment Matrices and Optimization Over Polynomials*, pages 157–270. Springer New York, 2009.

C.-W. Li and Y.-K. Feng. Decoupling theory of general multivariable analytic non-linear systems. *International Journal of Control*, 45(4):1147–1160, 1987.

C. Louembet, F. Cazaurang, and A. Zolghadri. Motion planning for flat systems using positive B-splines: An LMI approach. *Automatica*, 46(8):1305–1309, 2010.

D. Mellinger and V. Kumar. Minimum snap trajectory generation and control for quadrotors. In *IEEE International Conference on Robotics and Automation*, 2011.

R. M. Murray, L. Zexiang, and S. Shankar. *A mathematical introduction to robotic manipulation*. Boca Raton: CRC Press, 1994.

D. Nairn, J. Peters, and D. Lutterkort. Sharp, quantitative bounds on the distance between a Bezier curve and its control polygon. Technical Report 98-007, Purdue University, 1998.

J. Nocedal and S. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer New York, 2006.

J. Oldenburg and W. Marquardt. Flatness and higher order differential model representations in dynamic optimization. *Computers & Chemical Engineering*, 26 (3):385–400, 2002.

P. A. Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical Programming*, 96(2):293–320, 2003. ISSN 1436-4646.

N. Petit, M. B. Milam, and R. M. Murray. Inversion based constrained trajectory optimization. In *IFAC Proceedings Volumes*, volume 34, pages 1211–1216. Elsevier BV, 2001.

M. J. D. Powell. *Approximation Theory and Methods*. Cambridge University Press, 1981.

H. Prautzsch, W. Boehm, and M. Paluszny. *Bézier and B-Spline Techniques*. Springer-Verlag, 2002.

T. J. Rivlin. *The Chebyshev Polynomials: From Approximation Theory to Algebra and Number Theory*. Tracts in Pure & Applied Mathematics. John Wiley & Sons Inc, 1974.

M. Roberts and P. Hanrahan. Generating dynamically feasible trajectories for quadrotor cameras. *ACM Transactions on Graphics*, 35(4):1–11, 2016.

F. Stoican, V.-M. Ivanusca, I. Prodan, and D. Popescu. Obstacle avoidance via B-spline parametrizations of flat trajectories. In *Mediterranean Conference on Control and Automation*. IEEE, 2016.

F. Suryawan, J. De Doná, and M. Seron. Splines and polynomial tools for flatness-based constrained motion planning. *International Journal of Systems Science*, 43(8):1396–1411, 2012.

L. N. Trefethen. *Approximation Theory and Approximation Practice, Extended Edition*. Other Titles in Applied Mathematics. SIAM, 2019.

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

X. Wang. Volumes of generalized unit balls. *Mathematics Magazine*, 78(5):390–395, 2005.

D. J. Wilson, R. J. Bradford, J. H. Davenport, and M. England. Cylindrical algebraic sub-decompositions. *Mathematics in Computer Science*, 8 (2):263–288, 2014.