JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

# Flatness Techniques of Control Application for Quad-rotor and Cloud Computing

[1]Swarnalatha Srinivas, [2]C P Harshavardhan

[1]Associate Professor of Electrical & Electronics Engineering, Bangalore Institute of Technology, Bengaluru-560004, India
[2]Assistant Professor of Electrical & Electronics Engineering, Bangalore Institute of Technology, Bengaluru-560004, India

***Abstract:*** Automation, with its mathematical sub-domain Control Theory, studies the properties and control of dynamic systems in engineering. Each controlled system is made up of commands (we also say input or control) u, states x and outputs y. The goal of control theory is to design a control so that states and outputs achieve a defined goal. The controls act on states which usually represent the internal dynamics of the system. While the outputs representing measurable components, are related to states. When the control depends only on the output measurement and only the error between the reference and the measured output is used in the design, the control is called a feedback (also known as feedback). Feedback control or the one-degree-of-freedom framework often fails for nonlinear systems, where there is no asymptotically stable global solution that satisfies the constraints of the systems. To deal with system disturbances, in this thesis, we use Modeless Control. The Command without Model (CSM), presented by Michel Fliess and Ceric Join, has already proven its power through a wide range of successful applications, and even, with experimental results where the model of the system and its perturbations are unknown. Successful attempts for nonlinear modeless control and for modeless control for delay systems are presented respectively.

***Keywords-*** ***Actuators, computing techniques, disturbances, nonlinear, perturbations, trajectory;***

## I. Introduction

Control System Engineering, together with its mathematical sub-field Control Theory, studies the properties of dynamical systems in engineering. Each control system is composed of inputs $u$, states $x$ and outputs $y$. Control theory's aim is to design a control input such that the states/outputs reach a defined goal. The inputs act on the states which usually represent the internal dynamics of the system. While the outputs representing the measurable components, are related to the states [1],[2],[3]. When the control input solely depends on the output measure, and only the error between the reference goal and the measured output is used in the design, the control is called *a feedback control*. The feedback controller or the *one-degree-of-freedom* framework often fails for nonlinear systems, where a global asymptotically stable solution that satisfies the *system constraints* does not exist [4],[5].

The *Model-Free Control* (MFC), introduced by Michel Fliess and Ceric Join [6], already proved its power through a wide range of successful applications [7] and even, with experimental results [8], where the system model and disturbances are unknown. Successful attempts for nonlinear MFC and for delay systems are presented in [9] and in [10] respectively. The first detailed proof of stability of the MFC that provides insights to the tuning of the control parameters was given in [11].

In the last few decades, Model Predictive Control (MPC) [12,[13] has achieved a big success in dealing with constrained control systems. Model predictive control is a form of control in which the current control law is obtained by solving, at each sampling instant, a finite horizon open-loop optimal control problem, using the current state of the system as the initial state; the optimization yields an optimal control sequence and the first control in this sequence is applied to the system. It has been widely applied in petro-chemical and related industries where satisfaction of constraints is particularly important because efficiency demands operating points on or close to the boundary of the set of admissible states and controls [14],[15].

The optimal control or MPC maximize or minimize a defined performance criterion chosen by the user. The optimal control techniques, even in the case without constraints are usually discontinuous, which makes them less robust and more dependent of the initial conditions. In practice, this means that the delay formulation renders the numerical computation of the optimal solutions difficult. A large part of the literature working on constrained control problems is focused on optimal trajectory generation [16]. These studies are trying to find feasible trajectories that optimize the performance following a specified criterion. Defining the right criterion to optimize may be a difficult problem in practice. Usually, in such cases, the feasible and the optimal trajectory are not too much different. For example, in the case of autonomous vehicles [17], due to the dynamics, limited curvature, and under-actuation, a vehicle often has few options for how it changes lines on highways or how it travels over the space immediately in front of it. Regarding the complexity of the problem, searching for a feasible trajectory is *easier*, especially in the case where we need *real-time re-planning* [18]. The goal of the QE procedure is to compute an equivalent quantifier free formula Â($U$) for a given first-order formula. It finds the feasible regions of free variables $U$ represented as semi-algebraic set where $G(X, U)$ is true. If the set $U$ is non-empty, there exists a point $u \in$ □ $^{m}$ which simultaneously satisfies all of the equations/inequalities. Such a point is called a feasible point and the set $U$ is then called feasible. If the set $U$ is empty, it is called unfeasible. In the case when $m = 0$, *i.e.* when all variables are quantified, the QE procedure decides whether the givenformula is true or false (decision problem).

Where $Q_i$ is one of the quantifiers $\forall$ (for all) and $\exists$ (there exists). Following the Tarski Seidenberg theorem (see [30]), for every prenex formula $G(X, U)$ there exists an equivalent quantifier-free formula Â($U$) defined by the free variables.

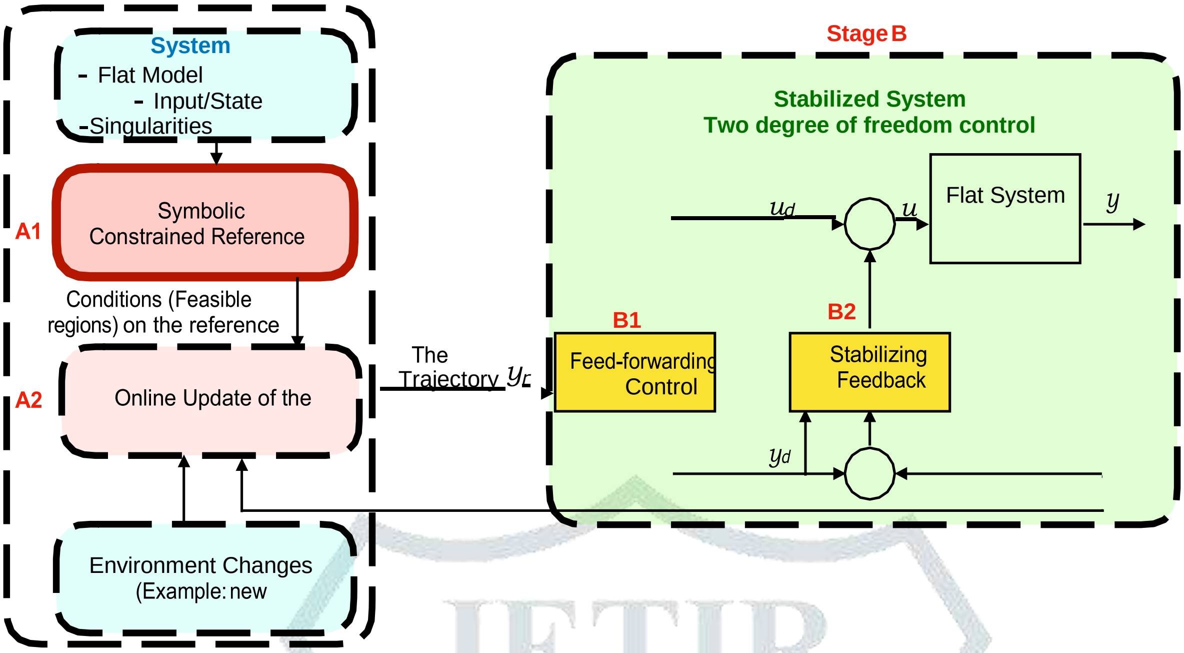

**Figure.1:** Two degrees of freedom control scheme overview

Considering that the evolution of transistor technologies is reaching its limits, low-complexity controllers that can take the constraints into account are of considerable interest. The same remark is valid when the system has sensors with limited performance [19].

The interpolating trajectory that passes through the points is prone to oscillatory effects (more unstable), while the approximating trajectory like the Bezier curve or B-Spline curve is more convenient since it only approaches defined so-called *control points* [20] and have simple geometric interpretations. The Bezier/B-spline curve can be handled by conveniently handling the curve's control points. The main reason in choosing the Bezier curves over the B-Splines curves is the simplicity of their arithmetic operators presented further in this Section. Despite the nice local properties of the B-spline curve, the direct symbolic multiplication of B-splines lacks clarity and has partly known practical implementation [21].

## II. LITERATURE REVIEW

To construct Stage A we first take advantage of the *differential flatness property* which serves as a base to construct our method. The differential flatness property yields exact expressions for the state and input trajectories of the system through trajectories of a flat output and its derivatives without integrating any differential equation. The latter property allows us to map the state/input constraints into the flat output trajectory space [22].

Then, in our symbolic approach (stage A1), we assign a Bezier curve to each flat output where the parameter to be chosen are the so-called *control points* (yielding a finite number of variables on a finite time horizon) given in a symbolic form. This kind of representation naturally offers several algebraic operations like the sum, the difference and multiplication, and affords us to preserve the explicit functions structure without employing discrete numerical methods. The advantage to deal with the constraints *symbolically*, rather than numerically, lies in that the symbolic solution explicitly depends on the control points of the reference trajectory. This allows studying how the input or state trajectories are influenced by the reference trajectory [23].

For that purpose, we divide the control problem in two stages (see Figure.1). Our objective will be to elaborate a *constrained reference trajectory management* (Stage A) which is meant to be applied to already pre-stabilized systems (Stage B). Unlike other receding horizon approaches which attempt to solve stabilization, tracking, and constraint fulfillment at the same time, we assume that in *Stage B*, a primal controller has already been designed to stabilize the system which provide nice tracking properties in the absence of constraints. In stage B, we employ the two-degree of freedom design consisting of a constrained trajectory design (constrained feed-forwarding) and a feedback control [24],[25].

- When a system should track a trajectory in a *static known environment*, then the exact set of feasible trajectories is found and the trajectory is fixed by our choice. If the system's environment changes, we only need to re-evaluate the exact symbolic solution with new numerical values.
- When a system should track a trajectory in an *unknown environment with moving objects*, then, whenever necessary, the reference design modifies the reference supplied to a primal control system so as to enforce the fulfilments of the constraints. This second problem is not addressed in the thesis.

Our approach is *not based on any kind of optimization* nor does it need computations for a given numerical value at each sampling step. We determine a set of feasible trajectories through the system constrained environment that enable a controller to make quick real-time decisions. For systems with singularities, we can isolate the singularities of the system by considering them as additional constraints [26].

## Existing Methods

- Considering actuator constraints based on the derivatives of the flat output (for instance, the jerk [27], and snap [28]) can be too conservative for some systems. The fact that a feasible reference trajectory is designed following the system model structure allows choosing a quite aggressive reference trajectory.
- In contrast to [29], we characterize the whole set of viable reference trajectory- rise which take the constraints into account.
- In [30], the problem of constrained trajectory planning of differentially flat systems is cast into a simple quadratic programming problem ensuing computational advantages by using the flatness property and the B-splines curve's properties. They simplify the computation complexity by taking advantage of the B-spline minimal (resp. maximal) control point. The simplicity comes at the price of having only minimal (resp. maximal) constant constraints that eliminate the possible feasible trajectories and renders this approach conservative.

- In [31], an inversion-based design is presented, in which the transition task between two stationary set-points is solved as a two-point boundary value problem. In this approach, the trajectory is defined as polynomial where only the initial and final states can be fixed.
- The thesis of Bak [32] compared existing methods to constrained controller design (anti-windup, predictive control, nonlinear methods), and introduced a nonlinear gain scheduling approach to handle actuator constraints.

## III. METHODOLOGY

The concept of *differential flatness* was introduced in [33] for non-linear finite dimensional systems. By the means of differential flatness, a non-linear system can be seen as a controllable linear system through a dynamical feedback. A model shall be described by a differential system as:

*Given the scalar function* $Z \in C^k(\square,\square)$ *and the number* $\alpha \in N$ *we denote by* $z^{(\alpha)}$ *the tuple of derivatives of Z up to the order to them* $\alpha \le k; z^{(a)} = z,, \dot{z},,, \ddot{z},,,,,, z^{(\alpha)}$. *Given the vector function* $v = (v_1,,,,,,,, v_q), v_i \in C^k(\square,\square)$ *and the tuple* $\alpha = (\alpha_1,,,,,,,,, \alpha_q), \alpha_i \in N_i$, *we denote by* $v^{(\alpha)}$ *the tuple of derivatives if each components* $v_i$ *of* $v$ *to its respective order* $\alpha_i \le k; v^{(\alpha)} = v_1,,,, v_1^{(\alpha_1)}, v_2.....v_2^{(\alpha_2)},,,,,, v_q,,,,, v_q^{(\alpha q)}$.

### 3.1 General problem formulation

Consider the nonlinear system

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \tag{3.1}$$

With state vector $\boldsymbol{x} = (x_1,. .., x_n)$ and control input $\boldsymbol{u} = (u_1,. .., u_m)$, $x_i$, $u_j$ æ $C^{\ddot{y}}([0,+\infty), \mathrm{R})$ for a suitable $k \in$ N. We assume the state, the input and their derivatives to be subject to both inequality and equality constraints of the form.

$$C_i(\boldsymbol{x}^{(\alpha_i^x)}(t), \boldsymbol{u}^{(\alpha_i^u)}(t)) \leqslant 0 \qquad \forall t \in [0, T], \ \forall i \in \{1, \dots, \nu^{\text{in}}\} \tag{3.2}$$

$$D_j(\boldsymbol{x}^{(\beta_j^x)}(t), \boldsymbol{u}^{(\beta_j^u)}(t)) = 0 \qquad \forall t \in I_j, \ \forall j \in \{1, \dots, \nu^{\text{eq}}\} \tag{3.3}$$

$$\{t_1, \dots, t_\gamma\}, \ 0 \le t_1 \leqslant \cdots \leqslant t_\gamma \leqslant T < +\infty$$

$$\alpha_i^x, \beta_j^x \in \mathbb{N}^n, \ \alpha_i^u, \beta_j^u \in \mathbb{N}^m. \tag{3.4}$$

With each $I_j$ being either $[0,T]$ (continuous equality constraint) or a discrete set $\{t_1,. .., t_g\}$, $0 \le t_1 \le \cdots \le t_i \le T \le +\infty$ (discrete equality constraint), and $\alpha^x$, $\beta^x \in \mathrm{N}^n$, $\alpha^u_i$, $\beta^u_j \in \mathrm{N}^m$. We stress that the relations (3.1) specify objectives (and Constraints) on the finite interval $[0,T]$. Objectives can be also formulated 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. Here we focus on just one of such intervals [34].

Our aim is to characterize the set of input and state trajectories ($\boldsymbol{x}, \boldsymbol{u}$) satisfying the system's equations (3.2) and the constraints (3.3). More formally we state the following problem.

Let C be a subspace of C$^k$([0,+ ∞),R). Constructively characterized the set C $^{cons}$ $\subseteq$C $^{n+m}$ of all extended trajectories (x, u) satisfying the system (2.4) and the constraints (2.5).

Problem 3.1 can be considered as a generalization of a constrained reachability problem (see for instance [35]). In such a reachability problem the stress is usually made 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 fulfilling the constraints. Here, we wish to give a functional characterization of the overall set of extended trajectories ($\boldsymbol{x}, \boldsymbol{u}$) satisfying some given differential constraints. A classical constrained reachability problem can be cast in the present formalism by limiting the constraints $C_i$ and $D_j$ to $\boldsymbol{x}$ and $\boldsymbol{u}$ (and not their derivatives) and by forcing two of the equality constraints to coincide with the initial and final set-points. Problem 2.1 is difficult to be addressed in its general setting. To simplify the problem, in the following we make some restrictions to the class of systems and to the functional space $C$. As a first assumption we limit the analysis to differentially flat systems [36].

### 3.2 Constraints in the flat output space

Let us assume that system (3.4) is differentially flat with flat output[1]

$$\boldsymbol{y} = (y_1, \ldots, y_m) = h(\boldsymbol{x}, \boldsymbol{u}^{\langle \rho^u \rangle}), \qquad (3.5)$$

With $\rho^u \in$ N$^m$. Following Equation (3.3), the parameterization or the feed forwarding trajectories associated to the reference trajectory $\boldsymbol{y_r}$ is

$$\boldsymbol{x_r} = \psi(\boldsymbol{y_r}^{\langle \eta^x \rangle}) \qquad (3.6)$$

$$\boldsymbol{u_r} = \zeta(\boldsymbol{y_r}^{\langle \eta^u \rangle}), \qquad (3.7)$$

with $\eta^x$ $\in$N$^n$ and $\eta^u$ $\in$ N$^m$ Through the first step of the dynamical extension algorithm [37], we get the flat output dynamics.

$$\begin{cases} y_1^{(k_1)} = \phi_1(\boldsymbol{y}^{\langle \mu_1^y \rangle}, \boldsymbol{u}^{\langle \mu_1^u \rangle}) \\ \quad \vdots \\ y_m^{(k_m)} = \phi_m(\boldsymbol{y}^{\langle \mu_m^y \rangle}, \boldsymbol{u}^{\langle \mu_m^u \rangle}), \end{cases} \qquad (3.8)$$

$$\mu_i^y = \left(\mu_{i,n}^y,,,,,,\mu_{i,m}^y\right) \in N^m, \mu_i^u = \left(\mu_{i,m}^u,,,,,,,\mu_{i,n}^u\right) \in N^m, K_i \succ \max{}_j\, \mu^2{}_{j,i}$$

The original $n$-dimensional dynamics (23.4) and the $K$-dimensional flat output dynamics (3.8) (K=$\sum_i^n K_{i\ i}$) are in one-to-one correspondence through (3.6) and (3.7). Therefore, the constraints (3.5) can be re-written as

[1] We recall that the flat output $\boldsymbol{y}$ has the same dimension $m$ as the input vector $\boldsymbol{u}$.

$$\Gamma_i(\boldsymbol{y_r}^{\langle \omega_i^{\text{in}} \rangle}) \leqslant 0 \qquad \forall t \in [0,T],\ \forall i \in \{1,\ldots,\nu^{\text{in}}\} \tag{3.9a}$$

$$\Delta_j(\boldsymbol{y_r}^{\langle \omega_j^{\text{eq}} \rangle}) = 0 \qquad \forall t \in I_j,\ \forall j \in \{1,\ldots,\nu^{\text{eq}}\} \tag{3.9ab}$$

With connection equations

$$\begin{aligned} \Gamma_i(\boldsymbol{y_r}^{\langle \omega_i^{\text{in}} \rangle}) &= C_i((\psi(\boldsymbol{y_r}^{\langle \eta x \rangle}))^{\langle \alpha_i^x \rangle}, \zeta(\boldsymbol{y_r}^{\langle \eta^u \rangle})^{\langle \alpha_i^u \rangle}), \\ \Delta_j(\boldsymbol{y_r}^{\langle \omega_j^{\text{eq}} \rangle}) &= D_j((\psi(\boldsymbol{y_r}^{\langle \eta^x \rangle})^{\langle \beta_j^x \rangle}, \zeta(\boldsymbol{y_r}^{\langle \eta^u \rangle})^{\langle \beta_j^u \rangle}) \end{aligned} \tag{3.10}$$

$$\omega_i^{\text{in}}, \omega_j^{\text{eq}} \in \mathbb{N}^m. \tag{3.11}$$

### 3.3 Problem specialization

For any practical purpose, one has to choose the functional space $\varsigma_y$ to which all components of the flat output belong. Instead of making reference to the space $\zeta^{gen} = C^p(0,+\infty), R$ mentioned in the statement of Problem 3.1, we focus on the space $\zeta^{gen} = C^p(0,T), R$. The constraints (2.9) specify finite-time objectives (and constraints) on the interval $[0,T]$. Still, the problem exhibits an infinite dimensional complexity, whose reduction leads to choose an approximation space $\zeta^{app}$ that is dense in $\zeta^{gen}$.A possible choice is to work with parametric functions expressed in terms of basis functions like, for instance, Bernstein-Bezier,Chebychev or Spline polynomials [38].

A scalar Bezier curve of degree $N \in$ R in the Euclidean space R is defined as

$$P(s) = \sum_{j=0}^{N} \alpha_j B_{jN}(s), \qquad s \in [0,1] \tag{3.12}$$

Where the $\alpha_j$ æ R are the control points and $B_{jN}(s) = \binom{N}{j}(1-s)^{N-j}s^j$ are Bernstein polynomials [34]. For sake of simplicity, we set here $T = 1$ and we choose as functional.

$$\mathscr{C}^{\text{app}} = \left\{ \sum_{0}^{N} \alpha_j B_j N | N \in \mathbb{N}, (\alpha_j)_0^N \in \mathbb{R}^{N+1}, B_j \in \mathcal{C}^0([0,1],\mathbb{R}) \right\} \tag{3.13}$$

The set of Bezier functions of generic degree has the very useful property of being closed with respect to addition, multiplication, degree elevation, derivation and integration operations (see section 3.4). As a consequence, any polynomial integral-differential operator applied to a Bezier curve, still produces a Bezier curve (in general of different degree). Therefore, if the flat outputs **y** are chosen in $C^{\text{app}}$ and the operators $T_i(\cdot)$ and $\Delta_j(\cdot)$ in (3.9) are integral-differential polynomials, thensuch constraints can still be expressed in terms of Bezier curves in $C^{\text{app}}$. We stress that, if some constraints do not admit such a description, we can still approximatethem up to a prefixed precision Á as function in $C^{\text{app}}$ by virtue of the densenessof $C^{\text{app}}$ in $C^{\text{gen}}$. Hence we assume the following [39].

### 3.4 Problem specialization

For the sake of completeness, we here list some important Bezier-Bernstein properties.

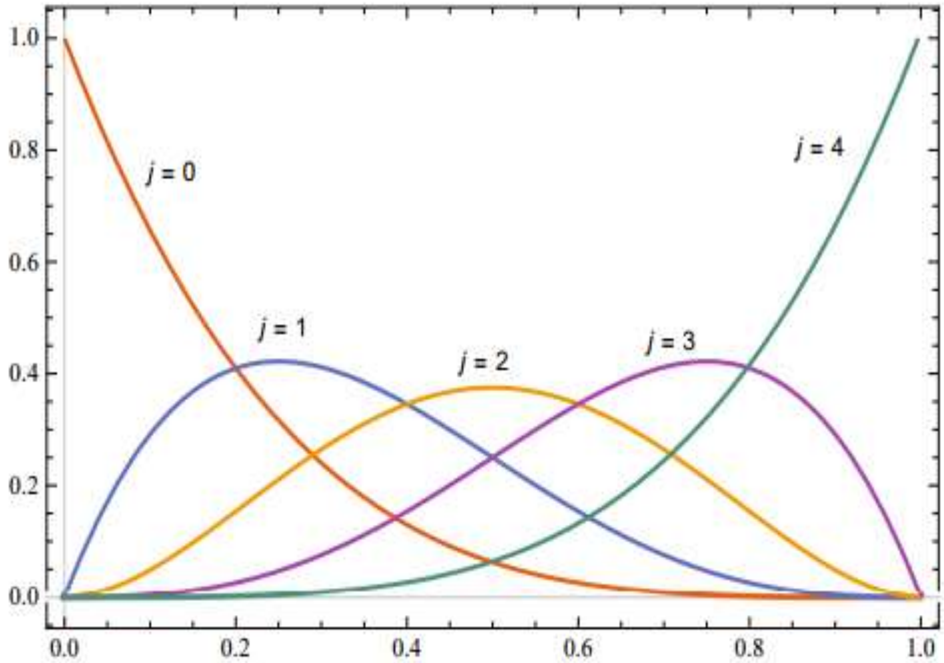

**Figure.2:** Bernstein Basis for degree $N = 4$.

A Bezier curve is a parametric one that uses the Bernstein polynomials as a basis. An $n$th degree Bezier curve is defined by

$$f(t) = \sum_{j=0}^{N} c_j B_{j,N}(t), \quad 0 \leqslant t \leqslant 1 \tag{3.13}$$

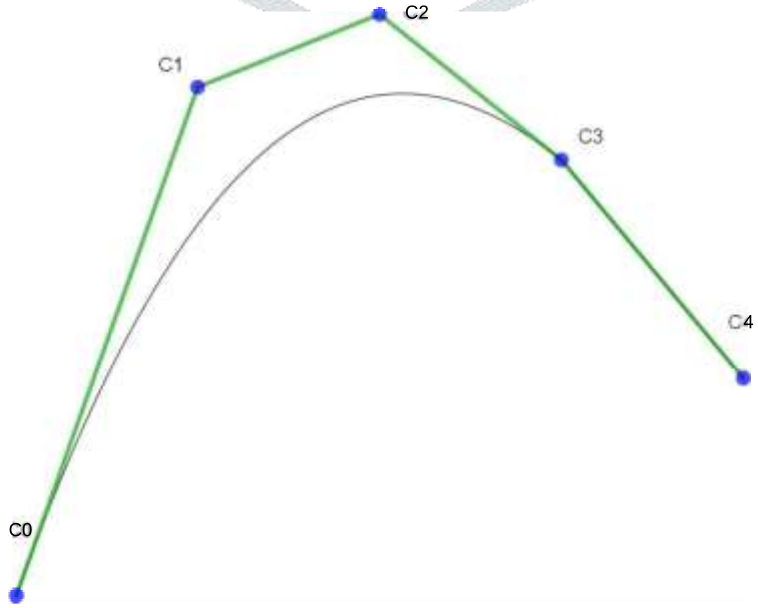

**Figure. 3:** The convex hull property for Bezier curve ($N$ =4) with control points $c_j$($j$ = 0,.., 4).

### 3.5 Quantitative envelopes for the Bezier curve

Working with the Bezier curve control points in place of the curve itself allows a simpler explicit representation. However, since our framework is not based on the Bezier curve itself, we are interested in the localization of the Bezier curve with respect to its control points, *i.e.* the control polygon. In this part, we review a result on *sharp quantitative bounds* between the Bezier curve and its control polygon [40]. For instance, in the case of a quad-rotor (discussed in Section 2), oncewe have selected the control points for the reference trajectory, these envelopes describe the exact localization of the quad-rotor trajectory and its distance from the obstacles. These quantitative envelopes may be of particular interest when avoiding corners of obstacles which traditionally in the literature [41] are modeledas additional constraints or introducing safety margin around the obstacle. We start by giving the definition for the control polygon.

### 3.6 Symbolic Bezier operations

In this section, we present the Bezier operators needed to find the Bezier control points of the states and the inputs. Let the two polynomials $f(t)$ (of degree $m$) and $g(t)$ (of degree $n$) with *control points* $f_j$ and $g_j$ be defined as follows:

$$f(t) = \sum_{j=0}^{m} f_j B_{j,m}(t), \quad 0 \leqslant t \leqslant 1$$

$$g(t) = \sum_{j=0}^{n} g_j B_{j,n}(t), \quad 0 \leqslant t \leqslant 1 \qquad (3.14)$$

We now show how to determine the control points for the degree elevation and for the arithmetic operations (the sum, difference, and product of these polynomials).For further information on Bezier operations, see [42].

To increase the degree from $n$ to $n + r$ and the number of control points from $n$ +1 to $n + r$ +1 without changing the shape, the new control points $b_j$ of the $(n + r)$th Bezier curve are given by:

$$b_j = \sum_{i=\max(0,j-r)}^{\min(n,j)} \frac{\binom{n}{i}\binom{r}{j-i}}{\binom{n+r}{j}} g_i \qquad j = 0, 1, \dots, n+r \qquad (3.15)$$

The latter constitutes the so-called *augmented control polygon*. The new control points are obtained as convex combinations of the original control points. This isan important operation exploited in addition/subtraction of two control polygonsof different lengths and in approaching the curve to a new control polygon by refining the original one [43].

If $m = n$ we simply add or subtract the coefficients

$$f(t) \pm g(t) = \sum_{j=0}^{m} (f_j \pm g_j) B_{j,m}(t) \qquad (3.16)$$

If $m > n$, we need to first elevate the degree of $g(t)$ $m - n$ times using (2.19) and then add or subtract the coefficients.

Multiplication of two polynomials of degree *m* and *n* yields a degree $m + n$ polynomial.

$$f(t)g(t) = \sum_{j=0}^{m+n} \underbrace{\left( \sum_{i=\max(0,j-n)}^{\min(m,j)} \frac{\binom{m}{i}\binom{n}{j-i}}{\binom{m+n}{j}} f_i g_{j-i} \right)}_{\text{Control points of the product}} B_{j,m+n}(t) \tag{3.17}$$

### 3.7 Bezier- time- derivatives

We give the derivative property of the Bezier curve in Proposition 2.1 which iscrucial in establishing the constrained trajectory procedure [44].

*The derivative of the jth Bernstein function of degree* $n > 1$ *is given by*

$$DB_{j,N}(t) = N\left(B_{j-1,N-1}(t) - B_{j,N-1}(t)\right) \ \textit{for}\ j = 0, \ldots, N. \tag{3.18}$$

*For any real number t and where* $B_{-1,N-1} = B_{N,N-1} = 0$.
*If the flat output or the reference trajectory y is a Bezier curve, its derivative is still a Bezier curve and we have an explicit expression for its controlpoints.*

*Let* $y^{(q)}(t)$ *denote the qth derivative of the flat output* $y(t)$*. We use the fixed time interval* $T = t_f - t_0$ *to define the time as* $t = T\,\tau$, $0 \le \tau \le 1$. *We can obtain* $y^{(q)}(\tau)$ *by computing the qth derivatives of the*

$$y^{(q)}(\tau) = \frac{1}{T^q} \sum_{j=0}^{N} c_j B_{j,N}^{(q)}(\tau) \tag{3.19}$$

Letting $c_j^{(0)} = c_j$ ,we write

$$y(\tau) = y^{(0)}(\tau) = \sum_{j=0}^{N} c_j^{(0)} B_{j,N}(\tau) \tag{3.20}$$

Then,

$$y^{(q)}(\tau) = \sum_{j=0}^{N-q} c_j^{(q)} B_{j,N-q}(\tau) \tag{3.21}$$

With derivative control points such that

$$c_j^{(q)} = \begin{cases} c_j, & q = 0 \\ \dfrac{(N-q+1)}{T^q}\left(c_{j+1}^{(q-1)} - c_j^{(q-1)}\right), & q > 0. \end{cases}$$

(3.22)

We can deduce the explicit expressions for all lower order derivatives up to order *N*-1. This means that if the reference trajectory $y_r(t)$ is a Bezier curveof degree $N > q$ ($q$ is the derivation order of the flat output $y$), by differentiating it, all states and inputs are given in straightforward Bezier form [45].

## IV. RESULTS AND DISCUSSION

### 4.1 Effect of time derivatives

We can deduce the explicit expressions for all lower order derivatives up to order *N*-1. This means that if the reference trajectory $y_r(t)$ is a Bezier curveof degree $N > q$ ($q$ is the derivation order of the flat output $y$), by differentiating it, all states and inputs are given in straightforward Bezier form.

This two-step design is better suited than a classical feedback controller (stabilization scheme) *i.e. one-degree of-freedom* framework. The first step obtains a first-order solution to the tracking problem while following the model instead of forcing it (like in a usual pure stabilization scheme). The second step is a refinement one, where the error between the actual measured values and the tracked references will be much smaller than in the pure stabilization case.

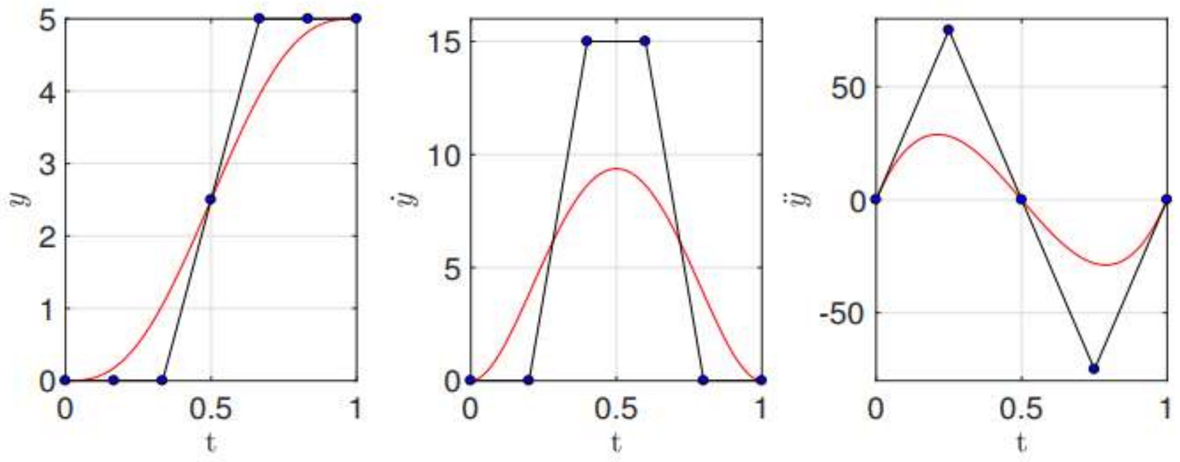

**Figure.4:** The time derivatives when $T$ = 1

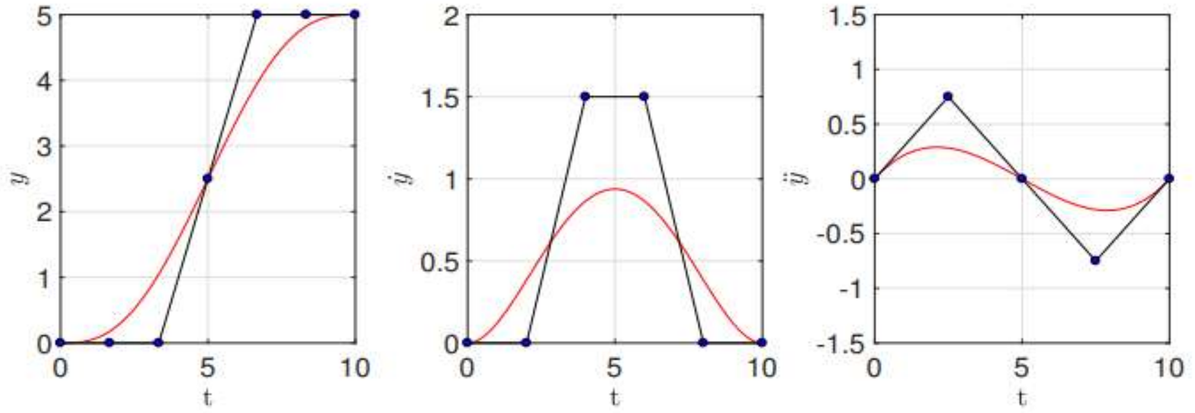

**Figure.5:** The time derivatives when $T$ = 10

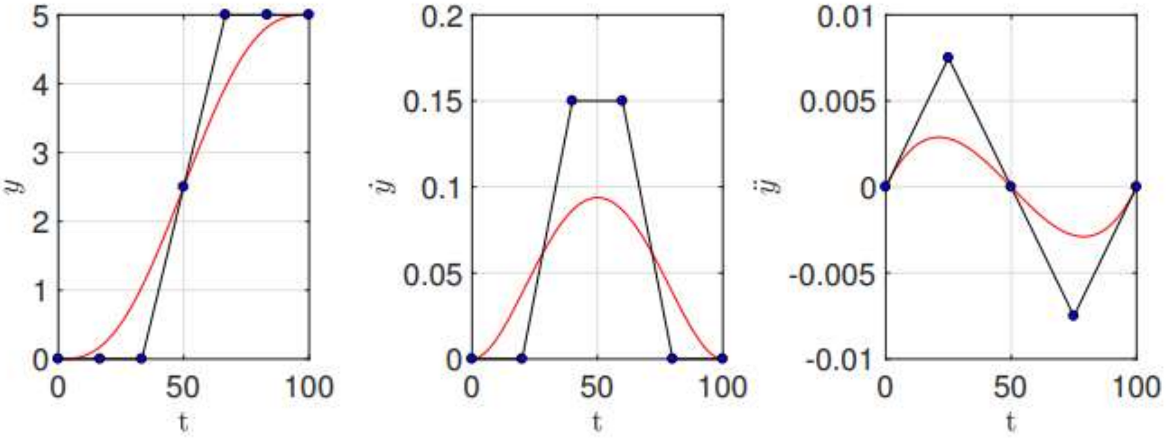

**Figure.6:** The time derivatives when $T = 100$

Through a simple example of a double integrator, we want to rep- resent the link between the time interval and the time derivatives. For a changingposition y, its time derivative $\dot{y}$ is its velocity, and its second derivative with respectto time $\ddot{y}$, is its acceleration [46]. Even higher derivatives are sometimes also used: the third derivative of position with respect to time is known as the jerk.

We here want to show the effect of the fixed time period T on the velocity, acceleration, etc. We remark the connection between the times scaling parameter appearing in the trajectory parameterization. We have a simple double integratordefined as:

$$\ddot{y} = u \qquad (4.1)$$

As a reference trajectory, we choose a Bezier curve $y = \sum_{i=0}^{N} a_i B_{i\ N}$ of order N = 4. Due to the Bezier derivative property, we can explicitly provide the link between the time interval T and control points of the Bezier curve's derivatives.

$$\dot{y} = \sum_{i=0}^{N-1} a_i^{(1)} B_{i,N-1} \qquad (4.2a)$$

$$\ddot{y} = \sum_{i=0}^{N-2} a_i^{(2)} B_{i,N-2} \qquad (4.2b)$$

Where $a_i^{(1)}$ and $a_j^{(2)}$ are the control points of the first and the second derivative of the B-spline curve respectively. We have the expressions of the $a_i^{(1)}$ and $a_j^{(2)}$ in terms of the $a_i$. This fact allow us to survey when the desired reference trajectory will respect the input constraints i.e. $a_i^{(2)} = f_1(a_i^{(1)}) = f_2(a_i)$. That means that if $\Delta a_j^{(2)} < K$ then $u < K$.

If we take a Bezier curve as reference trajectory $y_r(t) = \sum_{j=0}^{N} c_j B_{j,N}(t)$ for a flat system such that the input is a polynomial function of the flat output and its derivatives, then the open loop input is also a Bezier curve $u_r = B(y_r, ..., y_r^{(q)}) = \sum_{i=0}^{m} U_i B_{i,m}(t).$

We should take a Bezier curve of degree N > q to avoid introducing discontinuities in the control input [47].

### 4.2 Effect of longitudinal constraints

The constraints are essentials in the design of vehicle longitudinal control whichaims to ensure the passenger comfort, safety and fuel/energy reduction. The longitudinal control can be designed for a highway scenario or a city scenario. In thefirst scenario, the vehicle velocity keeps a constant form where the main objectiveis the vehicle inter-distance whiles the second one, deals with frequent stops and accelerations, the so-called Stop-and-Go scenario [48]. The inter-distance dynamics can be represented as an single integrator driven by the difference between the leader vehicle velocity $V_l$ and the follower vehicle velocity $V_x$ , i.e., $\dot{d} = V_i - V_z$.

In this example, suppose we want to follow the leader vehicle, and stay within a fixed distance from it (measuring the distance through a camera/radar system). Additionally, suppose we enter a desired destination through a GPS system, and sup-pose our GPS map contains all the speed information limits. Our goal is the follower longitudinal speed $V_x$ to follow a reference speed $V_{x,r}(t) \in [0, \min(V_i, V_{\max})]$, $V_{\max} \in$ □ $> 0$ given by the minimum between the leader vehicle speed and the speed limit. The longitudinal dynamics of a follower vehicle is given by the following model [49]:

$$M\dot{V}_x(t) = \frac{u(t)}{r} - C_a V_x^2(t) \tag{4.3}$$

Where $V_x$ is the longitudinal speed of the vehicle, $u$ is the motor torque, taken as *control input* and the physical constants: $M$ the vehicle's mass, $r$ the mean wheel radius, and $C_a$ the aerodynamic coefficient [50]. The model is differentially flat, with $V_x$ as a flat output. An open loop control yielding the tracking of the reference trajectory $V_{xr}$ by $V_x$, assuming the model to be perfect, is

$$u_r(t) = r\left(M\dot{V}_{xr}(t) + C_a V_{xr}^2(t)\right) \tag{4.4}$$

If we desire an open-loop trajectory $u_r \in C^0$, then for the flat output, we should assign a Bezier curve of degree $d > 1$. We take $V_{xr}$ as reference trajectory, a Bezier curve of degree 4 *i.e.* $C^4$-function.

$$\begin{gathered} V_{xr}(t) = \sum_{i=0}^{4} a_i B_{i,4}(t), \\ V_{xr}(t_0) = V_i, \quad V_{xr}(t_f) = V_f \end{gathered} \tag{4.5}$$

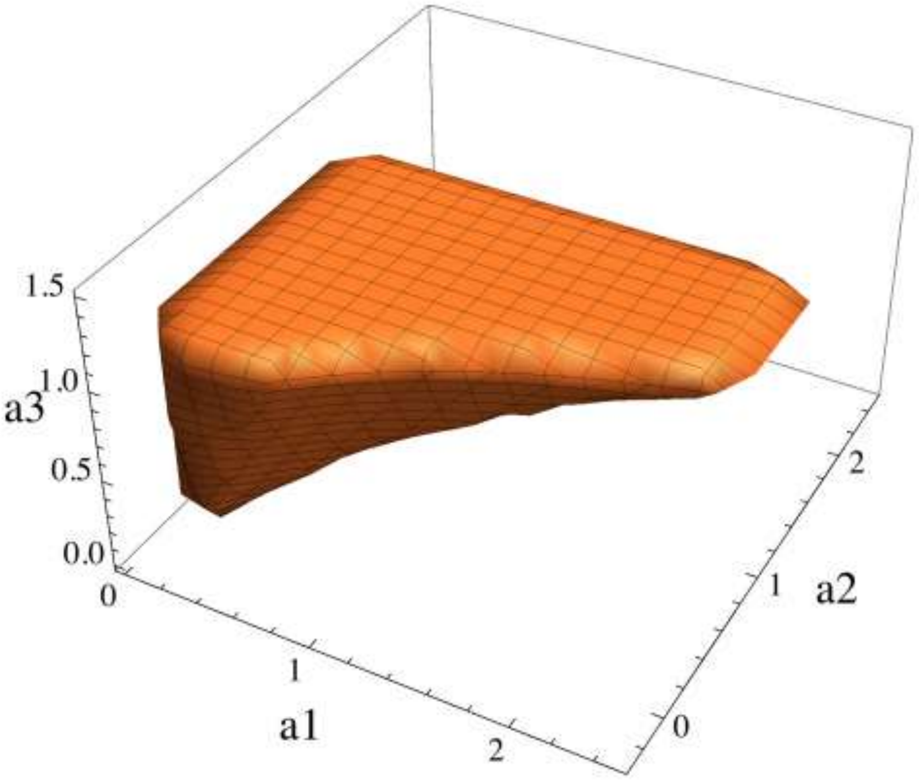

**Figure.7:** Feasible region for the control points of $V_{xr}$ when $U_{\min} = 0$ and $U_{\max} = 10$.

Where the $a_i$'s are the control points and the $B_{i,4}$ the Bernstein polynomials. Using the Bezier curve properties, we can find the control points of the open-loop control $u_r$ in terms of the $a_i$'s by the following steps [51]:

We want the input control points $U_i$ to be

$$U_{\min} < U_i < U_{\max} \quad i = 0, \ldots, 8 \tag{4.6}$$

Where $U_{\min} = 0$ is the lower input constraint and $U_{\max} = 10$ is the high input constraint. By limiting the control input, we indirectly constraint the fuel consumption. The initial and final trajectory control points are defined as $V_x(t_0) = a_0 = 0$and $V_x(t_1) = a_4 = 1$ respectively.

The constraint (4.3) directly corresponds to the semi-algebraic set: The constraint (4.3) corresponds to the *semi-algebraic set i.e.* the following system of nonlinear inequalities [52]:

$$\begin{cases} 0 < U_0 = 4\,a_1 < 10 \\ 0 < U_1 = a_1 + \frac{3\,a_2}{2} < 10 \\ 0 < U_2 = \frac{4\,a_1^2}{7} - \frac{5\,a_1}{7} + \frac{12\,a_2}{7} + \frac{3\,a_3}{7} < 10 \\ 0 < U_3 = \frac{15\,a_2}{14} - \frac{10\,a_1}{7} + a_3 + \frac{6\,a_1\,a_2}{7} + \frac{1}{14} < 10 \\ 0 < U_4 = \frac{18\,a_2^2}{35} - \frac{10\,a_1}{7} + \frac{10\,a_3}{7} + \frac{16\,a_1\,a_3}{35} + \frac{2}{7} < 10 \\ 0 < U_5 = \frac{10\,a_3}{7} - \frac{15\,a_2}{14} - \frac{6\,a_1}{7} + \frac{6\,a_2\,a_3}{7} + \frac{5}{7} < 10 \\ 0 < U_6 = \frac{4\,a_3^2}{7} + \frac{5\,a_3}{7} - \frac{3\,a_1}{7} - \frac{9\,a_2}{7} + \frac{10}{7} < 10 \\ 0 < U_7 = \frac{5}{2} - \frac{3\,a_2}{2} < 10 \\ 0 < U_8 = 5 - 4\,a_3 < 10 \end{cases} \tag{4.7}$$

In order to solve symbolically the system of inequalities *i.e.* to find the regions of the intermediate control points $a_i$, we use the Mathematical function *Cylindrical Decomposition*. The complete symbolic solution with three intermediate control points ($a_1$, $a_2$, $a_3$) is too long to be included. Since the latter is too long to be included, we illustrate the symbolic solution in the case of two intermediate control points ($a_1$, $a_2$):

$$\begin{aligned} &(0 < a_1 \leqslant 0.115563 \wedge -a_1 < a_2 < 1.33333) \\ &\vee \left(0.115563 < a_1 \leqslant 0.376808 \wedge 0.142857\left(-3.a_1^2 + 2.a_1 - 1\right) < a_2 < 1.33333\right) \\ &\vee \left(0.376808 < a_1 \leqslant 1.52983 \wedge \frac{4a_1 - 2}{3.a_1 + 4} < a_2 < 1.33333\right) \\ &\vee \left(1.52983 < a_1 < 2 \wedge 0.333333\sqrt{15.a_1 - 17} - 0.333333 < a_2 < 1.33333\right) \end{aligned} \tag{4.8}$$

The latter solution describing the feasible set of trajectories can be used to make a choice for the Bezier control points: "First choose $a_1$ in the interval (0, 0.115563] and then you may choose $a_2$ bigger than the chosen $-a_1$ and smaller than 1.33333. Or otherwise choose $a_1$ in the interval (0.115563, 0.376808] and, then choose $a_2$ such that $0.142857\,(-3a_1^2 + 2.a_1 - 1) < a_2 < 1.33333$, etc."

We can observe how the flat outputs influences the control input *i.e.* which part of the reference trajectory influences which part of the control input. For instance in (3.3), we observe that the second control point $a_1$ influences more than $a_2$ and $a_3$ the beginning of the control input (the control points $U_0$, $U_1$, $U_2$). The previous inequalities can be used as a prior study to the sensibility of the control inputs with respect to the flat outputs [53],[54],[55].

It should be stressed that the goal here is quite different than the traditional one in optimization problems. We do not search for the best trajectory according to a certain criterion under the some constraints, but we wish to obtain the set of all trajectories fulfilling the constraints; this for an end user to be able to pick one or another trajectory in the set and to switch from one to another in the same set [56],[57],[58]. The picking and switching operations aim to be really fast[59], [60].

### 4.3 Effect of quad-rotor constraints

However, during aggressive trajectory design, it is difficult to ensure trajectory feasibility while trying to exploit the entire range of feasible motor inputs [61],[62]. Moreover,in many applications, their role is to fly in complex cluttered environments; hencethere is a necessity of output constraints. Therefore, the constraints on the inputs and states are one of the crucial issues in the control of quad-rotors. Fortunately, with the hardware progress, today the quad-rotors have speed limitsof forty meters per second and more comparing to few meters per second in the past [63]. Therefore, it is important to conceive control laws for quad-rotors to a level where they can exploit their full potential especially in terms of agility.

In the famous paper [64], is proposed an algorithm that generates optimal trajectories such that they minimize cost functional that are derived from the square of the norm of the snap (the fourth derivative of position). There is a limited research investigating the quad-rotor constraints (see [65] and the paperstherein) without employing an online optimization.

The following application on quad-rotor is devoted to unify the dynamics constraints or demands constraints with the environmental constraints (*e.g.* , fixed obstacles). Fliess and his coworkers research work [54–56] on differentially flat systems and their properties led to a deeper understanding of trajectory tracking through the system trajectories parameterization. For a differentially flat system [55], all the states and the inputs can be parameterized through a so-called flat *output* and the trajectory planning can be obtained immediately without solving differential equations. Moreover, with the flatness property, the behavior of each system variable can be easily analyzed. From this perspective, the control design can thus be decomposed in two steps:

- Design of flat outputs reference trajectory; off-line computation of the open- loop controls *(feed forward part)*.
- On-line computation of the complementary closed-loop controls in order to stabilize the system around the reference trajectories *(feedback part)*.

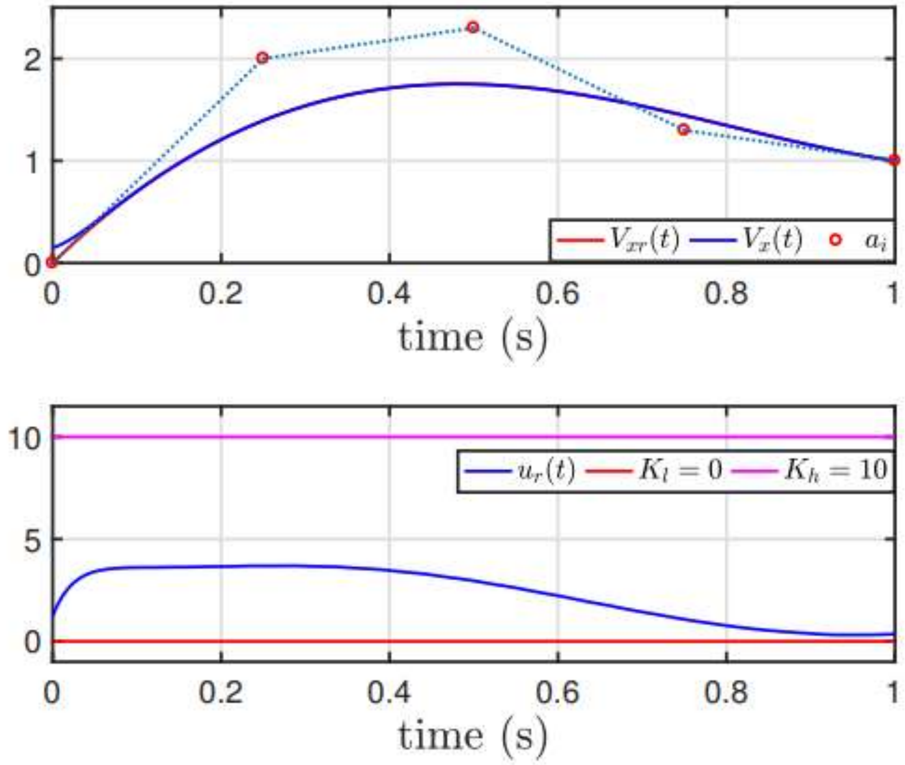

**Figure.8:** Closed-loop performance of trajectory tracking

## V.CONCLUSION

The state/input constraints are translated into a *system of inequalities and equalities* where the variables are the Bezier control points. This enables the input/state/output constraints to be considered into the trajectory design in a unified fashion. This allows us to develop a compact methodology to deal both with control limitations and space constraints as those arising in obstacle avoidance problems.

The core value of this work lies in two important advantages:

- The low complexity of the controller; fast real-time algorithms.
- The choice *i.e.* the user can select the desired feasible trajectory.
- Thesub-optimality may be seen as a drawback.

In the context of trajectory design, we find a successful simpler or approximated semi-algebraic set defined off-line. The closed form solution of the CAD establishes an explicit relationship between the desired constraints and the trajectory parameters. This gives us a rapid insight into how the reference trajectory influences the system behavior and the constraints fulfillment. Therefore, this method may serve as sensitivity analysis that reflects how the change in the reference trajectory influences the input reference trajectory.

## IV. ACKNOWLEDGMENT

The authors are grateful to Mr. Paul George M N, Onsite Coordinator, HCL Technologies Ltd, Uttar Pradesh, India. For his support to study the flatness of quad-rotor dynamics of cloud computing techniques.

## REFERENCES

[1] Hassane Abouaïssa, Michel Fliess, and Cédric Join. “On ramp metering: towards a better understanding of ALINEA via model-free control”. In: *International Journal of Control* 90.5 (2017), pp. 1018–1026.

[2] Kostas Alexis, George Nikolakopoulos, and Anthony Tzes. “Model predictive quadrotor control : attitude , altitude”. In: *IET Control Theory and Applications* 6.June 2011 (2012), pp. 1812–1827.

[3] Hirokazu Anai. “Effctive quantifier elimination for industrial applications.” In: *ISSAC*. 2014, pp. 18–19.

[4] Hamid Arabnejad et al. “A fuzzy load balancer for adaptive fault tolerance management in cloud platforms”. In: *European Conference on Service-Oriented and Cloud Computing*. Springer, 2017, pp. 109–124.

[5] Michael Armbrust et al. “Above the clouds: A Berkeley view of cloud computing”. In: *University of California, Berkeley, Tech. Rep. UCB* (2009), pp. 07–013. arXiv: 0521865719780521865715.

[6] Michael Armbrust et al. “A view of cloud computing”. In: *Communications of the ACM* 53.4 (2010), p. 50. arXiv: 0521865719780521865715.

[7] Martin Bak. “Control of Systems with Constraints”. PhD thesis. 2000, pp. 1–17.

[8] Luciano Baresi et al. “A discrete-time feedback controller for containerized cloud applications”. In: *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering*. ACM, 2016, pp. 217–228.

[9] Cornel Barna et al. “Cloud adaptation with control theory in industrial clouds”. In: *Proceedings - 2016 IEEE International Conference on Cloud Engineering Workshops, IC2EW 2016* (2016), pp. 231–238.

[10] Saugata Basu. “Algorithms in real algebraic geometry: a survey”. In: *arXiv preprint arXiv:1409.1534* (2014).

[11] Maria Bekcheva, Cédric Join, and Hugues Mounier. “Cascaded Model-Free Control for trajectory tracking of quadrotors.” In: *International Conference on Unmanned Aircraft Systems (ICUAS)*. 2018, pp. 1359–1368.

[12] Maria Bekcheva, Hugues Mounier, and Luca Greco. “Control of differentially flat linear delay systems with constraints. IFAC-PapersOnLine, 50(1), 13348-13353.” In: *IFAC World Congress*. 2017, pp. 13348–13353.

[13] Maria Bekcheva et al. “Meilleure élasticité « nuagique » par commande sans modèle”. In: *ISTE OpenScience Automatique* 2.1 (2018).

[14] Mihaly Berekmeri et al. “Feedback Autonomic Provisioning for Guaranteeing Performance in Map Reduce Systems”. In: *IEEE Transactions on Cloud Computing* PP.99 (2016), p. 1.

[15] David Bernstein. “Containers and cloud: From lxc to docker to kubernetes”. In: *IEEE Cloud Computing* 1.3 (2014), pp. 81–84.

[16] Mogens Blanke et al. *Diagnosis and fault-tolerant control*. Vol. 2. Springer, 2006.

[17] [18] Samir Bouabdallah, Andre Noth, and Roland Siegwart. “PID vs LQ control techniques applied to an indoor micro quadrotor”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Vol. 3. 2004,pp. 2451–2456.

[18] Samir Bouabdallah and Roland Siegwart. “Backstepping and sliding-mode techniques applied to an indoor micro Quadrotor”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2005, pp. 2247–2252.

[19] Christopher W Brown. “An overview of QEPCAD B: a tool for real quantifier elimination and formula simplification”. In: *Journal of Japan Society for Symbolic and Algebraic Computation* 10.1 (2003), pp. 13–22.

[20] Ning Cao and Alan F. Lynch. “Inner-Outer Loop Control for Quadrotor UAVs with Input and State Constraints”. In: *IEEE Transactions on Control Systems Technology* 24.5 (2016), pp. 1797–1804.

[21] Garcia Carlos E., Prett David M., and Morari Manfred. “Model Predictive Control : Theory and Practice a Survey”. In: *Automatica* 25.3 (1989), pp. 335–338.

[22] Sophie Cerf et al. “Cost function based event triggered Model Predictive Controllers application to Big Data Cloud services”. In: *2016 IEEE 55th Conference on Decision and Control (CDC)*. Las Vegas, 2016, pp. 1657–1662.

[23] Abbas Chamseddine et al. “Flatness-based trajectory planning/replanning for a quadrotor unmanned aerial vehicle”. In: *IEEE Transactions on Aerospace and Electronic Systems* 48.4 (2012), pp. 2832–2847.

[24] Abbas Chamseddine et al. “Trajectory Planning and Replanning Strategies Applied to a Quadrotor Unmanned Aerial Vehicle”. In: *Journal of Guidance, Control, and Dynamics* 35.5 (2012), pp. 1667–1671.

[25] Aneesh N Chand, Michihiro Kawanishi, and Tatsuo Narikiyo. “Non-linear model-free control of flapping wing flying robot using iPID”. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 2930–2937.

[26] George E Collins. “Quantifier elimination for real closed fields by cylindrical algebraic decompostion”. In: *Automata Theory and Formal Languages 2nd GI Conference Kaiserslautern, May 20–23, 1975*. Springer, 1975, pp. 134–183.

[27] Giuseppe Conte and Anna Maria Perdon. “Modeling Time-delay Systems by means of Systems with Coefficients in a Ring”. In: *In Proceedings Workshop on Modeling and Control of Complex Systems*. 2005.

[28] John Cortés-romero et al. “Algebraic estimation and active disturbance rejection in the control of flat systems”. In: *Control Engineering Practice* 61.45 (2017), pp. 173–182.

[29] Michel Coste. *An introduction to semialgebraic geometry*. Université de Rennes, 2002.

[30] Fabrizio Dabbene, Didier Henrion, and Constantino M. Lagoa. “Simple approximations of semi-algebraic sets and their applications to control”. In: *Automatica* 78 (2017), pp. 110–118. arXiv: 1509.04200.

[31]Michel Dambrine, Jean-Pierre Richard, and Pierre Borne. “Feedback Control of Time-Delay Systems with Bounded Control and State”. In: *Mathematical Problems in Engineering* 1.1 (1995), pp. 77–87.

[32] Brigitte D’Andréa-Novel et al. “A mathematical explanation via intelligent PID controllers of the strange ubiquity of PIDs”. In: *18th Mediterranean Conference on Control and Automation (MED)*. 2010, pp. 395–400. arXiv: 1005.0440.

[33] Carl de Boor. *A Practical Guide to Splines*. Vol. 27. New York: Springer, 2001.

[34] Emmanuel Delaleau. “A proof of stability of model-free control”. In: *2014 IEEE Conference on Norbert Wiener in the 21st Century (21CW).* Boston, MA, 2014, pp. 1–7.

[35] Sette Diop. “Observers for sampled data nonlinear systems via numerical differentiation”. In: *European Control Conference*. Kos, Greece, 2007,pp. 1179–1184.

[36] Maxime Doublet, Cédric Join, and Frédéric Hamelin. “Model-free control for unknown delayed systems”. In: *Conference on Control and Fault-Tolerant Systems, SysTol*. 2016.

[37] Nadeem Faiz, Sunil Agrawal, and Richard M. Murray. “Differentially Flat Systems with Inequality Constraints: An Approach to Real-Time Feasible Trajectory Generation”. In: *AIAA Journal of Guidance, Control, and Dynamics* 24.2 (2001), pp. 219–227.

[38] Guillermo P. Falconi et al. “Admissible thrust control laws for quadrotor position tracking”. In: *2013 American Control Conference* July 2016 (201302),pp. 4844–4849.

[39] Rida T Farouki and V. T. Rajan. “Algorithms for polynomials in Bernstein form”. In: *Computer Aided Geometric Design* 5.1 (1988), pp. 1–26.

[40] Matthias Fassler. “Quadrotor Control for Accurate Agile Flight”. PhD thesis. University of Zurich, 2018.

[41] Timm Faulwasser, Veit Hagenmeyer, and Rolf Findeisen. “Optimal exact path-following for constrained differentially flat systems”. In: *IFAC Proceedings Volumes (IFAC-PapersOnline)* 18.PART 1 (2011), pp. 9875–9880.

[42] Timm Faulwasser, Veit Hagenmeyer, and Rolf Findeisen. “Constrained reachability and trajectory generation for flat systems”. In: *Automatica* 50.4 (2014), pp. 1151–1159.

[43] Michel Fliess. “Generalized controller canonical form for linear and nonlinear dynamics”. In: *IEEE Transactions on Automatic Control* 35.9 (1990),pp. 994–1001.

[44] Michel Fliess. “Some basic structural properties of generalized linear systems”. In:*Systems & Control Letters* 15 (1990), pp. 391–396.

[45] Michel Fliess. “Analyse non standard du bruit”. In: *Comptes Rendus Mathematique* 342.10 (2006), pp. 797–802. arXiv: 0603003 [cs.CE].

[46] Michel Fliess and Cédric Join. “Model-free control and intelligent PID controllers: Towards a possible trivialization of nonlinear control?” In: *IFAC Proceedings Volumes (IFAC-Papers Online)* 15.PART 1 (2009), pp. 1531–1550.

[47] Michel Fliess and Cédric Join. “Model-free control”. In: *International Journal of Control* 86.12 (2013), pp. 2228–2252. arXiv: 1305.7085.

[48] Michel Fliess and Cédric Join. “Deux améliorations concurrentes des PID Two competing improvements of PID controllers : A comparison”. In: *ISTE OpenScience Automatique* 2 (2018), pp. 1–23.

[49] Michel Fliess and Hugues Mounier. “Tracking control and fi-freeness of infinite dimensional linear systems”. In: *Dynamical Systems, Control, Coding, Computer Vision*. Springer, 1999, pp. 45–68.

[50] Michel Fliess and Hugues Mounier. “On a class of linear delay systems often arising in practice”. In: *Kybernetika* 37.37 (2001), pp. 295–308.

[51] Michel Fliess and Hebertt Sira-Ramirez. “Closed-loop parametric identification for continuous-time linear systems via new algebraic techniques”. In: *Identification of Continuous-time Models from Sampled Data*. Springer London, 2008, pp. 362–391.

[52] Michel Fliess et al. “Easily implementable time series forecasting techniques for resource provisioning in cloud computing”. In: *6th International Conference on Control, Decision and Information Technologies (CoDIT)*. Paris.

[53] Michel Fliess et al. “On differentially flat nonlinear systems”. In: *Nonlinear Control Systems Design 1992*. Elsevier, 1993, pp. 159–163.

[54] Michel Fliess et al. “Flatness and defect of non-linear systems: introductory theory and examples”. In: *International Journal of Control* 61.6 (1995), pp. 1327–1361.

[55]Michel Fliess et al. “A lie-bäcklund approach to equivalence and flatness of nonlinear systems”. In: *IEEE Transactions on Automatic Control* 44.5 (1999), pp. 922–937.

[56] Melvin E. Flores and Mark B. Milam. “Trajectory generation for differentially flat systems via NURBS basis functions with obstacle avoidance”. In: *Proceedings of the American Control Conference* 2006 (2006), pp. 5769–5775.

[57] Alessandro Gasparetto and Vanni Zanotto. “A technique for time-jerk optimal planning of robot trajectories”. In: *Robotics and Computer-Integrated Manufacturing* 24.3 (2008), pp. 415–426.

[58] Pierre-Antoine. Gédouin et al. “Model-Free Control of Shape Memory Alloys Antagonistic Actuators”. In: *Preprints 17th IFAC World Congress*. Seoul, Korea, 2008.

[59] Pierre-Antoine Gédouin et al. “Experimental comparison of classical {PID} and model-free control: position control of a shape memory alloy active spring”. In: *Control Engineering Practice* 19.5 (2011), pp. 433–441.

[60] Nicole Gehring, Joachim Rudolph, and Frank Woittennek. *Controllability and prediction-free control of coupled transport processes viewed as linear systems with distributed delays*. Vol. 46. 26. IFAC, 2013, pp. 13–18.

[61] Knut Graichen and Michael Zeitz. “Feedforward control design for finite-time transition problems of non-linear MIMO systems under input constraints”. In: *International Journal of Control* 81.3 (2008), pp. 417–427.

[62] Knut Graichen and Michael Zeitz. “Feedforward Control Design for Finite-Time Transition Problems of Nonlinear Systems With Input and Output Constraints”. In: *IEEE Transactions on Automatic Control* 53.5 (2008), pp. 485–488.

[63]Domenico Grimaldi et al. “A feedback-control approach for resource management in public clouds”. In: *Global Communications Conference (GLOBECOM), IEEE*. 2015.

[64] Domenico Grimaldi et al. “A Fuzzy Approach based on Heterogeneous Metrics for Scaling Out Public Clouds”. In: *IEEE Transactions on Parallel and Distributed Systems* 9219.c (2017), pp. 1–1.

[65] Veit Hagenmeyer. “Robust nonlinear tracking control based on differential flatness”. In: *at-Automatisierungstechnik Methoden und Anwendungen der Steuerungs-, Regelungs-und Informationstechnik* 50.12/2002 (2002), p. 615.