Reachable Bézier Polytopes: A Primitive for Motion Planning with Layered Architectures

Noel Csomay-Shanklin, and Aaron D. Ames

I. Introduction

This is along the vein of Phi-related systems, but that property is too strong for many control systems. Instead, we want approximate relations, of the form of tracking error bounds.

And bisimulation

Authors are with the Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA 91125, USA, Corresponding author: noelcs@caltech.edu.

This research is supported by Technology Innovation Institute (TII).

II. BACKGROUND

A. Reduced-Order Models and Problem Statement

Consider the following nonlinear control system:

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

where $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^n$ is the state, $\mathbf{u} \in \mathcal{U} \subseteq \mathbb{R}^m$ is the input, and the dynamics $\mathbf{f} : \mathcal{X} \times \mathcal{U} \to \mathbb{R}^n$ is assumed to be continuously differentiable in its arguments. The system (1) will be represented as the tuple $\mathcal{S} = \{\mathcal{X}, \mathcal{U}, \mathbf{f}\}$. Due to the potential complexity of the dynamics $\mathbf{f}$, directly synthesizing control actions for challenging tasks may be intractable. To address this, control engineers often rely on *reduced-order models*, which serve as template systems that enable desired behaviors to be constructed in a computationally tractable way. These are defined as:

Definition 1. A system $\mathcal{S}_d = \{\mathcal{X}_d, \mathcal{U}_d, \mathbf{f}_d\}$ is said to be a *reduced-order model* for a system $\mathcal{S}$ if there exists a surjective mapping $\mathbf{\Pi} : \mathcal{X} \to \mathcal{X}_d$ and a right inverse $\mathbf{\Psi} : \mathcal{X}_d \hookrightarrow \mathcal{X}$ such that $\mathbf{\Pi} \circ \mathbf{\Psi} = \mathrm{id}_{\mathcal{X}_d}$.

As the dimensionality of $\mathcal{X}_d$ is typically much smaller than $\mathcal{X}$, there are many possible inverse mappings $\mathbf{\Psi}$, each of which induce an embedding of the reduced state space $\mathcal{X}_d$ into the full state space $\mathcal{X}$. The specific details of this embedding are coupled to the controller design process. To link a full-order system with a reduced-order model and establish this mapping, we must define a feedback controller $\mathbf{k} : \mathcal{X} \times \mathcal{X}_d \to \mathcal{U}$ which aims to track the states of the reduced-order model. This controller results in the following closed-loop system:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{k}(\mathbf{x}, \mathbf{x}_d)) \triangleq \mathbf{f}_{\text{cl}}(\mathbf{x}, \mathbf{x}_d), \tag{2}$$

which, given any initial condition $\mathbf{x}_0 \in \mathcal{X}$, has continuously differentiable solution $\mathbf{x} : I \to \mathcal{X}$ over some interval $I \subset \mathbb{R}_{\geq 0}$ defined as:

$$\mathbf{x}(t) \triangleq \mathbf{x}_0 + \int_0^t \mathbf{f}_{\text{cl}}(\mathbf{x}(\tau), \mathbf{x}_d(\tau)) d\tau.$$

A key property of this controller is its ability to maintain bounded tracking error:

Definition 2. Let $\mathcal{S}_d$ with states $\mathbf{x}_d \in \mathcal{X}_d$ and inputs $\mathbf{u}_d \in \mathcal{U}_d$ be a reduced order model for system $\mathcal{S}$. A set-valued function $\mathcal{E} : \mathcal{U}_d \to \mathcal{P}(\mathcal{X})$ is a *tracking invariant* for the system $\mathcal{S}$ if:

$$\mathbf{x}_{\text{cl}}(t) \in \mathbf{\Psi}(\mathbf{x}_d(t)) \oplus \mathcal{E}(\mathbf{u}_d(t)),$$

where $\oplus$ denotes the Minkowski sum.

For a tracking invariant set, we also define an upper bound $e : \mathcal{U}_d \to \mathbb{R}_{\geq 0}$ as:

$$e(\mathbf{u}_d) \triangleq \sup_{\mathbf{e} \in \mathcal{E}(\mathbf{u}_d)} \|\mathbf{e}\|_2,$$

which, if we assume $\mathcal{E}$ is described as the zero sublevel set of a function that is differentiable with respect to $\mathbf{u}_d$, is locally Lipschitz with respect to $\mathbf{u}_d$.

Example 1. Let $\mathcal{S}$ represent the closed-loop system of a quadruped tracking a center of mass velocity command with whole-body MPC. In this scenario, the reduced order system $\mathcal{S}_d$ is that of a single integrator and the mapping $\mathbf{\Pi}$ projects the full state space of the quadruped into the center of mass planar positions. The function $\mathbf{k}$ and mapping $\mathbf{\Psi}$ define the process of MPC, which takes in desired velocity trajectories and produces joint-space trajectories which can be tracked with bounded error as a function of how much input the reduced-order system applies.

Given a reduced order model $\mathcal{S}_d$, we will be interested in characterizing the space of all trajectories for the system $\mathcal{S}$ which satisfy the following problem:

Problem 1. Consider a compact state constraint set $\mathcal{C}_{\mathcal{X}} \subset \mathcal{X}_d$ and compact input constraint set $\mathcal{C}_{\mathcal{U}} \subset \mathcal{U}$. Produce trajectories $\mathbf{x}_d$, which when tracked achieve the following:

- $\mathbf{\Pi}(\mathbf{x}_{\text{cl}}(t)) \in \mathcal{C}_{\mathcal{X}}$ for all $t \in I$,
- $\mathbf{k}(\mathbf{x}_{\text{cl}}(t), \mathbf{x}_d(t)) \in \mathcal{C}_{\mathcal{U}}$ for all $t \in I$.

We will go about solving this problem by appropriately constraining the space of trajectories $\mathbf{x}_d(\cdot)$. As such, we must reason about the structure of integral curves of the reduced-order model dynamics Although reduced-order models can have any system structure (and are useful as long as there exist an appropriate mapping $\mathbf{\Psi}$ and controller $\mathbf{k}$), in order to make constructive guarantees we make further assumptions about the structure of the dynamics of $\mathbf{f}_d$. Specifically, consider a nonlinear reduced-order model system with coordinates $\mathbf{q}_d \in \mathbb{R}^m$, state $\mathbf{x}_d = [\mathbf{q}_d^\top, \dot{\mathbf{q}}_d^\top, \ldots, \mathbf{q}_d^{(\gamma-1)\top}]^\top \in \mathbb{R}^n$ for some $\gamma \in \mathbb{N}$ and control-affine dynamics of the form:

$$\dot{\mathbf{x}}_d = \begin{bmatrix} \mathbf{0} & \mathbf{I}_{n-m} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{x}_d + \begin{bmatrix} \mathbf{0} \\ \mathbf{f}_d(\mathbf{x}_d) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{g}_d(\mathbf{x}_d) \end{bmatrix} \mathbf{u}_d, \tag{3}$$

where $\mathbf{I}_{n-m}$ is an identity matrix of size $n-m$, the $\mathbf{0}$ matrices are appropriately sized, $\mathbf{u} \in \mathbb{R}^m$ is the input, and the drift vector $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ and actuation matrix $\mathbf{g} : \mathbb{R}^n \to \mathbb{R}^{m \times m}$ are assumed to be locally Lipschitz continuous on $\mathbb{R}^n$. Additionally, we make the following assumptions:

Assumption 1. We have that $\mathbf{f}_d(\mathbf{0}) = \mathbf{0}$ and the matrix $\mathbf{g}_d(\mathbf{x}_d)$ is invertible for all $\mathbf{x}_d \in \mathbb{R}^n$.

This structure is imposed as it allows us to effectively parameterize solution curves of (3) via Bézier curves.

B. Bézier Curves

A curve $\mathbf{b} : I \triangleq [0, \tau] \to \mathbb{R}^m$ for $\tau \in \mathbb{R}_{>0}$ is said to be a Bézier curve [1] of order $p \in \mathbb{N}$ if it is of the form:

$$\mathbf{b}(t) = \mathbf{p}\mathbf{z}(t),$$

where $\mathbf{z} : I \to \mathbb{R}^{p+1}$ is a Bernstein basis polynomial of degree p and $\mathbf{p} \in \mathbb{R}^{m \times p+1}$ are a collection of $p+1$ *control points* of dimension m. There exists a matrix $\mathbf{H} \in \mathbb{R}^{p+1 \times p+1}$ which defines a linear relationship between control points of a curve $\mathbf{b}$ and its derivative via:

$$\dot{\mathbf{b}}(t) = \mathbf{p}\mathbf{H}\mathbf{z}(t).$$

For a complete construction of the matrices discussed in this section, see the Appendix. This enables us to define a state space curve $\mathbf{B} : I \to \mathbb{R}^n$:

$$\mathbf{B}(t) \triangleq \begin{bmatrix} \mathbf{b}^{(0)}(t) \\ \vdots \\ \mathbf{b}^{(\gamma-1)}(t) \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{p}\mathbf{H}^0 \\ \vdots \\ \mathbf{p}\mathbf{H}^{\gamma-1} \end{bmatrix}}_{\triangleq \mathbf{P}} \mathbf{z}(t). \tag{4}$$

The columns of the matrix $\mathbf{P} \in \mathbb{R}^{n \times p+1}$, denoted as $\mathbf{P}_j$ for $j = 0, \dots, p$, represent the collection of n dimensional control points of the Bézier curve $\mathbf{B}$ in the state space. Furthermore, if we take $\mathbf{x}_d = \mathbf{B}$ to represent a desired trajectory of Bézier curves, eobserve that:

$$\dot{\mathbf{x}}_d(t) = \begin{bmatrix} \mathbf{0} & \mathbf{I}_{n-m} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{x}_d(t) + \begin{bmatrix} \mathbf{0} \\ \mathbf{f}_d(\mathbf{x}_d(t)) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{g}_d(\mathbf{x}_d(t)) \end{bmatrix} \mathbf{u}_d(t),$$

for the continuous input signal:

$$\mathbf{u}_d(t) = \mathbf{g}_d(\mathbf{x}_d(t))^{-1}(-\mathbf{f}_d(\mathbf{x}_d(t)) + \mathbf{q}_d^{(\gamma)}(t)).$$

Thus $\mathbf{B}$ is a dynamically feasible trajectory.

Property 1 (Convex Hull [1])**.**

$$\mathbf{B}(t) \in \operatorname{conv}(\{\mathbf{P}_j\}), \quad j = 0, \dots, p, \quad \forall t \in I.$$

Herein lies the beauty of Bézier curves – this property allows us to make conclusions about the behavior of the continuous time curve while only reasoning about the discrete collection of control points. Furthermore, we have that if the control points satisfy linear inequalities, then so does the continuous-time curve:

Property 2 (Linear Bounding)**.** For a vector $\mathbf{d} \in \mathbb{R}^k$, matrix $\mathbf{C} \in \mathbb{R}^{k \times n}$, and an integer $j \in \mathbb{Z}_{>0}$, we have:

$$\mathbf{C}\mathbf{P}_j \le \mathbf{d}, \quad j = 0, \dots, p \implies \mathbf{C}\mathbf{B}(t) \le \mathbf{d}, \quad \forall t \in I.$$

Proof. The convex hull property of Bézier curves implies that for any $\tau \in [0, T]$, we may write:

$$\mathbf{C}\mathbf{B}(t) = \mathbf{C} \sum_{j=0}^{p} \lambda_j(\tau) \mathbf{P}_j$$

for some $\lambda_j(\tau) \ge 0$ and $\sum_{j=0}^{p} \lambda_j(\tau) = 1$. Therefore,

$$\begin{aligned} \mathbf{C}\mathbf{B}(t) &\le \mathbf{C} \sum_{j=0}^{p} \lambda_j(\tau) \sup_j \mathbf{P}_j \\ &= \mathbf{C} \sup_j \mathbf{P}_j \le \mathbf{d}, \end{aligned}$$

as each $\mathbf{P}_j$ term satisfies the inequality $\mathbf{C}\mathbf{P}_j \le \mathbf{d}$ from the assumption of the Lemma. $\square$

This property will allow us to express convex constraints along the entire curve as a finite collection of convex constraints on the control points. We will specifically be interested in producing Bézier curves that connect initial conditions and terminal conditions in a fixed time τ. For a given initial condition $\mathbf{x}_d(0) \in \mathbb{R}^n$ and terminal condition $\mathbf{x}_d(\tau) \in \mathbb{R}^n$, the Bézier curve $\mathbf{B}$ must satisfy the following set of equality constraints:

$$\begin{aligned} \mathbf{b}^{(k)}(0) &= \mathbf{p}\mathbf{H}^k \mathbf{z}(0) = \mathbf{q}_d^{(k)}(0), \quad k = 0, \dots, \gamma - 1, \\ \mathbf{b}^{(k)}(\tau) &= \mathbf{p}\mathbf{H}^k \mathbf{z}(\tau) = \mathbf{q}_d^{(k)}(\tau), \quad k = 0, \dots, \gamma - 1. \end{aligned}$$

These equality constraints can be rewritten in the form:

$$\mathbf{P}\boldsymbol{\Delta} = \begin{bmatrix} \mathbf{x}_d(0) & \mathbf{x}_d(\tau) \end{bmatrix} \tag{5}$$

for $\boldsymbol{\Delta} \triangleq [\boldsymbol{\Delta}_0 \ \ \boldsymbol{\Delta}_\tau]$ with $\boldsymbol{\Delta}_0 \triangleq [1 \ \ \mathbf{0}_{1\times p}]^\top$ and $\boldsymbol{\Delta}_\tau \triangleq [\mathbf{0}_{1\times p} \ \ 1]^\top$.

Lemma 1. *There exists a matrix $\mathbf{P}$ satisfying* (5) *if the the order of the curve satisfies $p \ge 2\gamma - 1$.*

Proof. this does not go anymore because D is defined with respect to $\mathbf{P}$. To fix this, define a new D with respect to $\mathbf{p}$, and make the following argument Another way to represent the matrix $\mathbf{D}$ is via the basis polynomials $\mathbf{z}$ and their derivatives:

$$\mathbf{D}_\tau = \begin{bmatrix} \mathbf{z}^{(0)}(0) \ \cdots \ \mathbf{z}^{(\gamma-1)}(0) \ \mathbf{z}^{(0)}(T) \ \cdots \ \mathbf{z}^{(\gamma-1)}(T) \end{bmatrix}^\top,$$

wherein for $j = 0, \dots, \gamma - 1$ we have:

$$\begin{aligned} \mathbf{z}^{(j)}(0) &= [\underbrace{\star \ \cdots \ \star}_{j+1} \ \ \underbrace{0 \ \cdots \ 0}_{p-j}], \\ \mathbf{z}^{(j)}(T) &= [\underbrace{0 \ \cdots \ 0}_{p-j} \ \ \underbrace{\star \ \cdots \ \star}_{j+1}], \end{aligned}$$

with nonzero entries $\star$. Therefore, in the case that $p \ge 2\gamma - 1$ the columns are linearly independent and thus the matrix $\mathbf{D}$ has full column rank, implying that a solution $\mathbf{p}$ exists (but is not unique unless $p = 2\gamma - 1$). $\square$

Remark 1. Lemma 1 can be viewed as a requirement on the number of control points in order to full determine systems of a specific order. If more control points than the minimum are added, then the solution is simply non-unique, allowing for additional cost terms to be optimized in a least squares problem.

III. State and Input Constraint Satisfaction

For this section, we assume that a kinematic decomposition of the state space has been completed and therefore will focus on a compact polytopic state constraint set of the form $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{C}\mathbf{x} \le \mathbf{d}\}$ with $\mathbf{C} \in \mathbb{R}^{k\times n}$ and $\mathbf{d} \in \mathbb{R}^k$ – the next section will be devoted to discussing how to generate this kinematic decomposition. Additionally, consider a Bézier curve of order $p \ge 2n - 1$, and make the following assumption about $\mathcal{U}$:

Assumption 2. The state constraint set is described by $\mathcal{C}_\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{C}\mathbf{x} \le \mathbf{d}\}$ with $\mathbf{C} \in \mathbb{R}^{k\times n}$ and $\mathbf{d} \in \mathbb{R}^k$. Furthermore, we have that the input constraint set $\mathcal{U} \triangleq \{\mathbf{u} \in \mathbb{R}^m \mid \|\mathbf{u}\|_\infty \le u_{\max}\}$ for $u_{\max} \in \mathbb{R}_{>0}$, i.e., we have a box input constraint.

Remark 2. The following constructions can also be performed with a positive diagonal weighting matrix $\mathbf{W} \in \mathbb{S}^m_{\succ 0}$ to scale the box constraint on $\mathbf{u}$, such that $\|\mathbf{W}\mathbf{u}\|_\infty \le u_{\max}$.

The remainder of the section will be devoted to proving the following statement:

Theorem 1. *Let system $\mathcal{S}_d$ be a reduced order model for system $\mathcal{S}$. There exist matrices $\mathbf{F}$ and $\mathbf{G}$ such that any trackable trajectory parameterized as a Bézier curve $\mathbf{B} : I \to \mathcal{X}_d$ with control points $\mathbf{p}$ satisfying:*

$$\mathbf{F}\mathbf{p} \leq \mathbf{G}$$

results in the closed loop system satisfying $\mathbf{\Pi}(\mathbf{x}_{\text{cl}}(t)) \in \mathcal{C}_{\mathcal{X}}$ and $\mathbf{k}(\mathbf{x}_{\text{cl}}(t)) \in \mathcal{C}_{\mathcal{U}}$ for all $t \in I$.

Towards proving this, define the decoupling matrix $\mathcal{G} : \mathbb{R}^n \to \mathbb{R}^{m\times m}$ and decoupling vector $\mathcal{F} : \mathbb{R}^n \times I \to \mathbb{R}^m$ as:

$$\mathcal{G}(\mathbf{x}) \triangleq \mathbf{g}(\mathbf{x})^{-1}, \quad \mathcal{F}(\mathbf{x}, t) \triangleq \mathbf{f}(\mathbf{x}) - \mathbf{q}_d^{(\gamma)}(t).$$

The following steps will emulate the steps taken in [2], wherein more details can be found about the exact constructions. The main difference in this work as that the $\infty-$norm is used instead of the $2-$norm, which will allow us to represent the sets of constraint satisfaction as polytopes instead of second order cone constraints. We can bound the feedback linearizing input in (6) via:

$$\|\mathbf{u}_d\|_\infty \leq \|\mathcal{G}(\mathbf{x})\|_\infty \left(\|\mathcal{F}(\mathbf{x},t)\|_\infty + \|\mathbf{K}\mathbf{e}\|_\infty\right). \tag{6}$$

for all $\mathbf{x} \in \mathbb{R}^n$ and $t \in I$. Next, taking $\bar{\mathbf{x}} \in \mathbb{R}^n$ to be a reference point in the state space we have:

$$\|\mathcal{G}(\mathbf{x})\|_\infty \leq \|\mathcal{G}(\bar{\mathbf{x}})\|_\infty + L_{\mathcal{G}}\|\mathbf{x}_d(t) - \bar{\mathbf{x}}\|_\infty + L_{\mathcal{G}}\|\mathbf{e}\|_\infty,$$

where $L_{\mathcal{G}}$ is a Lipschitz constant of $\mathcal{G}$ with respect to the ∞-norm on $\mathcal{X}$, which is well defined by the local Lipschitz continuity and nonzero assumptions on $\mathbf{g}$ and the compactness of $\mathcal{X}$.

Similarly:

$$\|\mathcal{F}(\mathbf{x},t)\|_\infty \leq \|\mathcal{F}(\bar{\mathbf{x}},t)\|_\infty + L_{\mathbf{f}}\|\mathbf{x}_d(t) - \bar{\mathbf{x}}\|_\infty + L_{\mathbf{f}}\|\mathbf{e}\|_\infty.$$

As in [2], the bound in (6) can be rewritten as:

$$\|\mathbf{k}(\mathbf{x}, \mathbf{x}_d)\|_\infty \leq \frac{1}{2}\boldsymbol{\sigma}^\top\mathbf{M}\boldsymbol{\sigma} + \mathbf{N}^\top\boldsymbol{\sigma} + \Gamma,$$

by organizing terms, where:

$$\boldsymbol{\sigma} \triangleq \begin{bmatrix} \|\mathbf{x}_d(t) - \bar{\mathbf{x}}\|_\infty \\ \|\mathbf{q}_d^{(\gamma)}(t) - \mathbf{f}(\bar{\mathbf{x}})\|_\infty \end{bmatrix}, \tag{7}$$

and the details of this reorganization, along with the exact form of $\mathbf{M} \in \mathbb{S}^2_{\succeq}$, $\mathbf{N} \in \mathbb{R}^2_{\geq 0}$, and $\Gamma \in \mathbb{R}_{\geq 0}$ can be found in the appendix. Ideally, we would like to enforce the following constraint, as that would bound the possible control inputs that we could take:

$$\|\mathbf{k}(\mathbf{x}, \mathbf{x}_d)\|_\infty \leq \frac{1}{2}\boldsymbol{\sigma}^\top\mathbf{M}\boldsymbol{\sigma} + \mathbf{N}^\top\boldsymbol{\sigma} + \Gamma \leq u_{\max}. \tag{8}$$

However, in order to efficiently enforce this constraint, a few reformulation steps are needed. We begin by defining the set of $\boldsymbol{\sigma}$ satisfying the constraint (8) as:

$$\boldsymbol{\Sigma} \triangleq \{\boldsymbol{\sigma} \in \mathbb{R}^2_{\geq 0} \mid \frac{1}{2}\boldsymbol{\sigma}^\top\mathbf{M}\boldsymbol{\sigma} + \mathbf{N}^\top\boldsymbol{\sigma} + \Gamma \leq u_{\max}\}$$

where we only consider points in the nonnegative orthant as $\boldsymbol{\sigma}$ is elementwise positive because as its entries are norms. We make the following feasibility assumption:

Assumption 3. The set $\boldsymbol{\Sigma} \neq \emptyset$.

The above assumption is one of feasibility of the low level controller – if $\boldsymbol{\Sigma}$ is empty, then the error bound $\mathcal{E}$ is larger than the set $\mathcal{U}$ meaning regardless of desired trajectory there could exist a perturbed state which would violate the input constraints. If this is the case, then the error tracking bound of the low-level controller needs to be improved before proceeding. Towards reformulating the bound in (8), we present the following Lemma:

Lemma 2. *There exists a vector $\mathbf{c} \in \mathbb{R}^2$ such that:*

$$\mathbf{c}^\top\boldsymbol{\sigma} \leq 1 \implies \frac{1}{2}\boldsymbol{\sigma}^\top\mathbf{M}\boldsymbol{\sigma} + \mathbf{N}^\top\boldsymbol{\sigma} + \Gamma \leq u_{\max}. \tag{9}$$

Proof. The set $\boldsymbol{\Sigma}$ is convex due to $\mathbf{M}$ being positive semi-definite; therefore, given any collection of points $\{\boldsymbol{\sigma}_i\}$ with $\boldsymbol{\sigma}_i \in \boldsymbol{\Sigma}$, we have that $\text{conv}(\{\boldsymbol{\sigma}_i\}) \subseteq \boldsymbol{\Sigma}$. Under the assumption 3, we know that the point $\boldsymbol{\sigma}_0 = \mathbf{0} \in \boldsymbol{\Sigma}$. Next, plugging in $\boldsymbol{\sigma}_1 = \begin{bmatrix}\sigma_1 & 0\end{bmatrix}$ and $\boldsymbol{\sigma}_2 = \begin{bmatrix}0 & \sigma_2\end{bmatrix}$ into (9) for unknown $\sigma_1, \sigma_2 \in \mathbb{R}_{\geq 0}$ and solving with equality yields the the x and y intercept of 1-level set of the quadratic form. We can now evaluate $\text{conv}(\{\boldsymbol{\sigma}_i\})$ for $i = 0, 1, 2$, and choosing the halfspace representation of the convex polytope $\boldsymbol{\Sigma}$ provides a constructive way to generate a vector $\mathbf{c}$. As $\boldsymbol{\sigma}$ is positive valued, this is given by $\mathbf{c} = \begin{bmatrix}\frac{1}{\sigma_1} & \frac{1}{\sigma_2}\end{bmatrix}$. □

Remark 3. In general, this reformulation would represent a polytopic approximation of the convex set $\boldsymbol{\Sigma}$ and therefore would be a relaxation of the problem. However, because $\mathbf{M}$ has one zero eigenvalue, this reformulation is in fact exact.

Beyond this, we can further reformulate the form in (9) from a linear constraint on the $\infty-$norm into a linear constraint on the desired trajectory itself:

Lemma 3. *There exists a matrix $\mathbf{\Pi} \in \mathbb{R}^{4nm\times n+m}$ and a vector $\mathbf{h} \in \mathbb{R}^{4nm}$ such that:*

$$\mathbf{\Pi}\begin{bmatrix}\mathbf{x}_d(t) \\ \mathbf{q}_d^{(\gamma)}(t)\end{bmatrix} \leq \mathbf{h} \iff \mathbf{c}^\top\boldsymbol{\sigma} \leq 1$$

Proof. First observe that based on the definition of $\boldsymbol{\sigma}$ in (7), the constraint $\mathbf{c}^\top\boldsymbol{\sigma} \leq 1$ is given by:

$$\mathbf{c}^\top\begin{bmatrix}\max_i |\mathbf{x}_d(t) - \bar{\mathbf{x}}|_i \\ \max_i \left|\mathbf{q}_d^{(\gamma)}(t) - \mathbf{f}(\bar{\mathbf{x}})\right|_i\end{bmatrix} \leq 1,$$

which, taking $\mathbf{c}^\top = [c_1, c_2]$, is equivalent to:

$$\underbrace{\begin{bmatrix} c_1 & c_1 & -c_1 & -c_1 \\ c_2 & -c_2 & c_2 & -c_2 \end{bmatrix}^\top}_{\triangleq\mathbf{C}^\top}\begin{bmatrix}(\mathbf{x}_d(t) - \bar{\mathbf{x}})_i \\ \left(\mathbf{q}_d^{(\gamma)}(t) - \mathbf{f}(\bar{\mathbf{x}})\right)_j\end{bmatrix} \leq \mathbf{1},$$

for all row pairs $i \leq n$ and $j \leq m$.

Letting $\mathbf{\Pi} \in \{0,1\}^{4nm\times n+m}$ be a matrix capturing the i,j permutations with scaling of $\mathbf{C}^\top$ above, we can reformulate this as:

$$\mathbf{\Pi}\begin{bmatrix}\mathbf{x}_d(t)-\bar{\mathbf{x}}\\ \mathbf{q}_d^{(\gamma)}(t)-\mathbf{f}(\bar{\mathbf{x}})\end{bmatrix}\leq\mathbf{1},$$

which can be further rearranged as:

$$\mathbf{\Pi}\begin{bmatrix}\mathbf{x}_d(t)\\ \mathbf{q}_d^{(\gamma)}(t)\end{bmatrix}\leq\underbrace{\mathbf{1}+\mathbf{\Pi}\begin{bmatrix}\bar{\mathbf{x}}\\ \mathbf{f}(\bar{\mathbf{x}})\end{bmatrix}}_{\triangleq\mathbf{h}}, \tag{10}$$

as desired. □

With the above lemma, we have a linear inequality on the desired trajectory, which if satisfied implies input constraint satisfaction for the closed loop control system. In order to make a similar claim for state constraints, we state the following Lemma:

Lemma 4 ([3] Lemma 4)**.** *For $j=1,\dots,k$, we have:*

$$\mathbf{x}_d(t)\in\mathcal{X}\ominus\mathcal{E}\Leftrightarrow\mathbf{C}\mathbf{x}_d(t)\leq\mathbf{k}. \tag{11}$$

where $\mathbf{k}\triangleq\mathbf{d}-\sqrt{\beta\overline{\mathbf{w}}^2}\sqrt{\mathrm{diag}(\mathbf{C}^\top\mathbf{P}^{-1}\mathbf{C})}$.

The above inequalities on the desired trajectory for state and input constraint satisfaction imposes constraints on the design of possible trajectories.

As curves are infinite dimensional objects, these constraints would only be able to be enforced apprriximately in a traditional trajectory optimizer. This is precisely where we see the usefulness of Bézier curves – we can exactly enforce these constraints on the continuous-time curve by reasoning about a discrete, low-dimensional collection of Bézier control points (as captured by Property 2).

With this in mind, we are now equipped to prove the main statement of the section:

(Proof of Theorem 1). From Lemmas 2 and 3, we know that enforcing (10) will result in $\|\mathbf{k}(\mathbf{x},\mathbf{x}_d))\|_\infty\leq u_{\max}$, and subsequently that $\|\mathbf{k}_{\text{clf}}(\mathbf{x}(t),t)\|_\infty\leq u_{\max}$ (as in [3], Corollary 1). Furthermore, from Lemma 4, we know that enforcing (11) results in $\mathbf{x}(t)\in\mathcal{X}$. Therefore, all that remains is reformulating these inequalities from constraints on $\mathbf{x}_d$ to constraints on $\boldsymbol{\xi}$. Combining these state and input constraints and decomposing $\mathbf{x}_d$ as in (4) results in:

$$\begin{bmatrix}\mathbf{\Pi}\left(\begin{bmatrix}\mathbf{P}\\ \mathbf{p}\mathbf{H}^\gamma\end{bmatrix}\mathbf{z}(t)\right)\\ \mathbf{C}(\mathbf{P}\mathbf{z}(t))\end{bmatrix}\leq\begin{bmatrix}\mathbf{h}\\ \mathbf{k}\end{bmatrix}.$$

Based on Property 2, we know that if we enforce this constraint on the control points, it will be enforced for the continuous time curve. Therefore, instead must enforce:

$$\begin{bmatrix}\mathbf{\Pi}\begin{bmatrix}(\mathbf{P})_j\\ (\mathbf{p}\mathbf{H}^\gamma)_j\end{bmatrix}\\ \mathbf{C}\mathbf{P}_j\end{bmatrix}\leq\begin{bmatrix}\mathbf{h}\\ \mathbf{k}\end{bmatrix},$$

for $j=0,\dots,p$. As this imposes linear constraints on the columns of $\mathbf{p}$, this expression can be vectorized (the details of which can be found in the Appendix) and written as:

$$\mathbf{F}\leq\mathbf{G}$$

whereby enforcing this constraint results in state and input constraint satisfaction as desired. □

Remark 4. The matrix $\mathbf{F}$ and vector $\mathbf{G}$ represent an efficient oracle to check whether bezier curves connecting initial and terminal points satifsy state and input constraints. As such, we have completed the goal of Supbroblem 2.

A. Reducing Conservatism

In the previous subsection, we re-framed the problem of bounding $\|\mathcal{G}(\mathbf{x}_d(t))\|$ and $\|\mathcal{F}(\mathbf{x}_d(t),t)\|$ as bounding $\|\mathbf{x}_d(t)-\bar{\mathbf{x}}\|$ and $\|\mathbf{q}_d^{(\gamma)}(t)-\mathbf{f}(\bar{\mathbf{x}})\|$, respectively, for a reference point $\bar{\mathbf{x}}$. While this enables tractability, it creates conservatism in the bound as the same reference point was used over the entire trajectory $\mathbf{x}_d(t)$. To resolve this conservatism, we would like to instead bound the trajectory with a collection of reference points $\{\bar{\mathbf{x}}_k\}$ spread out over the time interval $[0,T]$. Towards this goal, we define the notion of a refinement of an interval:

Definition 3. A k-refinement of an interval $[0,T]$ is a collection of times $\{t_i\}$ for $i=0,\dots,k$ and associated intervals $\{[t_{i-1},t_i]\}$ with $t_{i-1}<t_i$, $t_0=0$, and $t_k=T$.

Given a k-refinement of the interval $[0,T]$ as well as reference points $\{\bar{\mathbf{x}}_i\}$ for $i=1,\dots k$, we can construct a piecewise constant reference trajectory $\bar{\mathbf{x}}(t)=\bar{\mathbf{x}}_i$ for $t\in[t_{i-1},t_i)$ with $i=1,\dots,k$. We can then define a time-varying error signal as:

$$\boldsymbol{\sigma}(t)\triangleq\begin{bmatrix}\mathbf{x}_d(t)-\bar{\mathbf{x}}(t)\\ \mathbf{q}_d^{(\gamma)}-\mathbf{f}(\bar{\mathbf{x}}(t))\end{bmatrix}$$

With this, we can state the following result:

Corollary 1. *Consider an order p Bézier curve $\mathbf{B}(t)$ defined for $t\in[0,T]$, and a piecewise-constant trajectory $\bar{\mathbf{x}}(t)$ defined with respect to a $k-$refinement of the interval $[0,T]$ with associated reference points $\{\bar{\mathbf{x}}_i\}$ for $i=1,\dots,k$. Then, there exists a matrix $\tilde{\mathbf{F}}$ and a vector $\tilde{\mathbf{G}}$ such that:*

$$\tilde{\mathbf{F}}\leq\tilde{\mathbf{G}}\implies\mathbf{c}^\top\boldsymbol{\sigma}(t)\leq 1. \tag{12}$$

Proof. This proof is similar to that of Theorem 1 and can be found in the Appendix. □

Remark 5. By enforcing the constraint in (12), we are able to ensure that the desired trajectory stays close to the piecewise constant reference trajectory, as opposed to a single reference point. This will reduce the conservatism of the bound, but requires increasing the number of constraints that we need to enforce in the opimization program, demonstrating an obvious tradeoff.

Another source of conservatism arises from the fact that the matrix $\mathbf{F}$ and vector $\mathbf{G}$ are functions of the Lipschitz constants of the systemm, whioch may be difficult or impossible to find in practice. The reachable set description is a function of a number of parameters, including a reference point $\bar{\mathbf{x}}$ and Lipschitz constants $L_{\mathbf{f}}$ and $L_{\mathcal{G}}$. We will discuss how to reduce conservatism in each of these, starting with the reference point $\bar{\mathbf{x}}$. Instead of bounding the curve away from a single point $\bar{\mathbf{x}}$, we can instead bound the curve from a sequence of intermediate points $\{\bar{\mathbf{x}}_k\}$ for $k = 1, \dots, K$ for some $k \in \mathbb{N}$. To do this, recall that the process of refining a curve is achieved by simple matrix multiplication of the Bézier coefficients and a matrix $\mathbf{Q} \in \mathbb{R}^{Kp \times p}$. Then, instead of enforcing the constraint $\mathbf{F}(\bar{\mathbf{x}})\mathbf{\Xi}_i \leq \mathbf{G}(\bar{\mathbf{x}})$ for $i = 1, \dots p$, we instead enforce $\mathbf{F}(\bar{\mathbf{x}}_k)\mathbf{Q}_k\mathbf{\Xi}_i \leq \mathbf{G}(\bar{\mathbf{x}}_k)$ for $i = 1, \dots p$ and $k = 1, \dots, K$. As was discussed in [2], Lipschitz constants for nonlinear systems may be difficult to find in practice.

Therefore, we can instead replace them with tuning parameters. This means that instead of enforcing the constraint $\mathbf{F}(L_{\mathbf{f}}, L_{\mathcal{G}})\mathbf{\Xi}_i \leq \mathbf{G}(L_{\mathbf{f}}, L_{\mathcal{G}})$ for $i = 1, \dots p$, we instead enforce $\mathbf{F}(\alpha, \beta)\mathbf{Q}_k\mathbf{\Xi}_i \leq \mathbf{G}(\alpha, \beta)$ for $i = 1, \dots p$ and $\alpha, \beta > 0$. As was shown in [2], these tuning knobs satisfy a monotinoicity property, meaning that if they are tuned to be sufficiently large then guarantees of input constraint satisfaction will be maintained. This means that in practice, one may begin with small values of α and β and increase them until the closed-loop nonlinear system meets input constraints.

Proof.

$$\begin{aligned}\|\mathbf{k}(\mathbf{x},\mathbf{x}_d,\mathbf{u}_d)\| &= \|\mathbf{k}(\mathbf{x},\mathbf{x}_d,\mathbf{u}_d) - \mathbf{k}(\mathbf{\Psi}(\mathbf{x}_d),\mathbf{x}_d,\mathbf{u}_d) + \mathbf{k}(\mathbf{\Psi}(\mathbf{x}_d),\mathbf{x}_d,\mathbf{u}_d) - \mathbf{k}(\mathbf{\Psi}(\bar{\mathbf{x}}_d),\bar{\mathbf{x}}_d,\bar{\mathbf{u}}_d) + \mathbf{k}(\mathbf{\Psi}(\bar{\mathbf{x}}_d),\bar{\mathbf{x}}_d,\bar{\mathbf{u}}_d)\| \\ &\le L_k(\|\mathbf{x}-\mathbf{\Psi}(\mathbf{x}_d)\| + \|\mathbf{\Psi}(\mathbf{x}_d)-\mathbf{\Psi}(\bar{\mathbf{x}}_d)\| + \|\mathbf{x}_d-\bar{\mathbf{x}}_d\| + \|\mathbf{u}_d-\bar{\mathbf{u}}_d\|) + \|\mathbf{k}(\mathbf{\Psi}(\bar{\mathbf{x}}_d),\bar{\mathbf{x}}_d,\bar{\mathbf{u}}_d)\| \\ &\le L_k(e(\mathbf{u}_d)-e(\bar{\mathbf{u}}_d) + (1+L_{\mathbf{\Psi}})\|\mathbf{x}_d-\bar{\mathbf{x}}_d\| + \|\mathbf{u}_d-\bar{\mathbf{u}}_d\|) + \|\mathbf{k}(\mathbf{\Psi}(\bar{\mathbf{x}}_d),\bar{\mathbf{x}}_d,\bar{\mathbf{u}}_d)\| + L_k e(\bar{\mathbf{u}}_d) \\ &\le L_k((1+L_{\mathbf{\Psi}})\|\mathbf{x}_d-\bar{\mathbf{x}}_d\| + (1+L_e)\|\mathbf{u}_d-\bar{\mathbf{u}}_d\|) + \|\mathbf{k}(\mathbf{\Psi}(\bar{\mathbf{x}}_d),\bar{\mathbf{x}}_d,\bar{\mathbf{u}}_d)\| + L_k e(\bar{\mathbf{u}}_d)\end{aligned}$$

$$\begin{aligned}\|\mathbf{k}(\mathbf{x},\mathbf{x}_d)\| &= \|\mathbf{k}(\mathbf{x},\mathbf{x}_d) - \mathbf{k}(\mathbf{\Psi}(\mathbf{x}_d),\mathbf{x}_d) + \mathbf{k}(\mathbf{\Psi}(\mathbf{x}_d),\mathbf{x}_d) - \mathbf{k}(\mathbf{\Psi}(\bar{\mathbf{x}}_d),\bar{\mathbf{x}}_d) + \mathbf{k}(\mathbf{\Psi}(\bar{\mathbf{x}}_d),\bar{\mathbf{x}}_d)\| \\ &\le L_k(\|\mathbf{x}-\mathbf{\Psi}(\mathbf{x}_d)\| + \|\mathbf{\Psi}(\mathbf{x}_d)-\mathbf{\Psi}(\bar{\mathbf{x}}_d)\| + \|\mathbf{x}_d-\bar{\mathbf{x}}_d\|) + \|\mathbf{k}(\mathbf{\Psi}(\bar{\mathbf{x}}_d),\bar{\mathbf{x}}_d)\| \\ &\le L_k(e(\mathbf{u}_d)-e(\bar{\mathbf{u}}_d) + (1+L_{\mathbf{\Psi}})\|\mathbf{x}_d-\bar{\mathbf{x}}_d\|) + \|\mathbf{k}(\mathbf{\Psi}(\bar{\mathbf{x}}_d),\bar{\mathbf{x}}_d)\| + L_k e(\bar{\mathbf{u}}_d) \\ &\le L_k((1+L_{\mathbf{\Psi}})\|\mathbf{x}_d-\bar{\mathbf{x}}_d\| + L_e\|\mathbf{u}_d-\bar{\mathbf{u}}_d\|) + \|\mathbf{k}(\mathbf{\Psi}(\bar{\mathbf{x}}_d),\bar{\mathbf{x}}_d)\| + L_k e(\bar{\mathbf{u}}_d)\end{aligned}$$

$$\begin{aligned}\|\mathbf{k}(\mathbf{x},\mathbf{x}_d)\| &= \|\mathbf{k}(\mathbf{x},\mathbf{x}_d) - \mathbf{k}(\mathbf{\Psi}(\mathbf{x}_d),\mathbf{x}_d) + \hat{\mathbf{k}}(\mathbf{x}_d) - \hat{\mathbf{k}}(\bar{\mathbf{x}}_d) + \hat{\mathbf{k}}(\bar{\mathbf{x}}_d)\| \\ &\le L_k(\|\mathbf{x}-\mathbf{\Psi}(\mathbf{x}_d)\| + \|\mathbf{\Psi}(\mathbf{x}_d)-\mathbf{\Psi}(\bar{\mathbf{x}}_d)\| + \|\mathbf{x}_d-\bar{\mathbf{x}}_d\|) + \|\hat{\mathbf{k}}(\bar{\mathbf{x}}_d)\| \\ &\le L_k(e(\mathbf{u}_d) + (1+L_{\mathbf{\Psi}})\|\mathbf{x}_d-\bar{\mathbf{x}}_d\|) + \|\hat{\mathbf{k}}(\bar{\mathbf{x}}_d)\| \\ &\le L_k((1+L_{\mathbf{\Psi}})\|\mathbf{x}_d-\bar{\mathbf{x}}_d\| + L_e\|\mathbf{u}_d\|) + \|\hat{\mathbf{k}}(\bar{\mathbf{x}}_d)\|\end{aligned}$$

where $\hat{\mathbf{k}}(\mathbf{x}_d) \triangleq \mathbf{k}(\mathbf{\Psi}(\mathbf{x}_d),\mathbf{x}_d)$ □

Towards proving this, define the decoupling matrix $\mathcal{G} : \mathbb{R}^n \to \mathbb{R}^{m\times m}$ and decoupling vector $\mathcal{F} : \mathbb{R}^n \times I \to \mathbb{R}^m$ as:

$$\mathcal{G}(\mathbf{x}_d(t)) \triangleq \mathbf{g}(\mathbf{x}_d(t))^{-1}, \quad \mathcal{F}(\mathbf{x}_d(t)) \triangleq \mathbf{q}_d^{(\gamma)}(t) - \mathbf{f}(\mathbf{x}_d(t)).$$

$$\mathbf{u}_d(t) = \mathbf{g}_d(\mathbf{x}_d(t))^{-1}(-\mathbf{f}_d(\mathbf{x}_d(t)) + \mathbf{q}_d^{(\gamma)}(t)).$$

Proof.

$$\begin{aligned}\|\mathbf{u}_d(t)\| &= \|\mathcal{G}(\mathbf{x}_d(t))\mathcal{F}(\mathbf{x}_d(t))\| \\ \|\mathcal{G}(\mathbf{x}_d(t))\| &\le \|\mathcal{G}(\bar{\mathbf{x}}_d)\| + L_{\mathcal{G}}\|\mathbf{x}_d(t)-\bar{\mathbf{x}}_d\| \\ \|\mathcal{F}(\mathbf{x}_d(t))\| &\le \|\mathbf{q}_d^{(\gamma)}(t) - \mathbf{f}_d(\bar{\mathbf{x}}_d)\| + L_{\mathbf{f}_d}\|\mathbf{x}_d(t)-\bar{\mathbf{x}}_d\|\end{aligned}$$

Therefore,

$$\|\mathbf{k}(\mathbf{x},\mathbf{x}_d)\| \le \begin{bmatrix}\|\mathbf{x}_d(t)-\bar{\mathbf{x}}_d\| \\ \|\mathbf{q}_d^{(\gamma)}(t)-\mathbf{f}_d(\bar{\mathbf{x}}_d)\|\end{bmatrix}^\top \begin{bmatrix} a & b \\ c & d\end{bmatrix} \begin{bmatrix}\|\mathbf{x}_d(t)-\bar{\mathbf{x}}_d\| \\ \|\mathbf{q}_d^{(\gamma)}(t)-\mathbf{f}_d(\bar{\mathbf{x}}_d)\|\end{bmatrix} + \begin{bmatrix} a \\ b\end{bmatrix}\begin{bmatrix}\|\mathbf{x}_d(t)-\bar{\mathbf{x}}_d\| \\ \|\mathbf{q}_d^{(\gamma)}(t)-\mathbf{f}_d(\bar{\mathbf{x}}_d)\|\end{bmatrix} + \Gamma$$

□

IV. Reachable Bézier Polytopes

Corollary 2. *Given an initial condition $\mathbf{x}_d(0)$, the forward reachable set is characterized by:*

$$\mathcal{F}(\mathbf{x}_d(0)) = \{\mathbf{x}_d \in \mathcal{X}_d \mid \mathbf{FP}\begin{bmatrix}\mathbf{x}_d(0)^\top & \mathbf{x}_d^\top\end{bmatrix}^\top \leq \mathbf{G}\}.$$

Similarly, given a terminal condition $\mathbf{x}_d(T)$, the backward reachable set is characterized by:

$$\mathcal{B}(\mathbf{x}_d(T)) = \{\mathbf{x}_d \in \mathcal{X}_d \mid \mathbf{FP}\begin{bmatrix}\mathbf{x}_d^\top & \mathbf{x}_d(T)^\top\end{bmatrix}^\top \leq \mathbf{G}\}.$$

References

[1] M. Kamermans, “A primer on bézier curves,” *(online book)*, 2020. [Online]. Available: https://pomax.github.io/bezierinfo/

[2] N. Csomay-Shanklin, A. J. Taylor, U. Rosolia, and A. D. Ames, “Multi-rate planning and control of uncertain nonlinear systems: Model predictive control and control lyapunov functions,” *arXiv:2204.00152*, 2022.

[3] ——, “Multi-Rate Planning and Control of Uncertain Nonlinear Systems: Model Predictive Control and Control Lyapunov Functions,” in *2022 IEEE 61st Conference on Decision and Control (CDC)*, Dec. 2022, pp. 3732–3739, iSSN: 2576-2370.

V. Appendix

A. Bézier Curves

At the cornerstone of the constructive methods discussed in this work is the notion of Bézier curves, a polynomial curve which will serve as a useful parameterization for producing kinodynamically admissible trajectories. A curve $\mathbf{b} : I \triangleq [0, \tau] \to \mathbb{R}^m$ for $\tau \in \mathbb{R}_{>0}$ is said to be a Bézier curve of order $p \in \mathbb{N}$ if it is of the form:

$$\mathbf{b}(t) = \mathbf{z}(t)^\top \boldsymbol{\xi},$$

where $\mathbf{z} : I \to \mathbb{R}^{p+1}$ is a Bernstein basis polynomial of degree p defined elementwise as:

$$z_k(t) = \binom{p}{k} \left(\frac{t}{\tau}\right)^k \left(1 - \frac{t}{\tau}\right)^{p-k}, \quad k = 0, \dots, p,$$

and $\boldsymbol{\xi} \in \mathbb{R}^{p+1 \times m}$ are a collection of $p+1$ *control points*. Defining a phasing variable over the interval I as $\tau : t \mapsto \frac{t}{\tau}$, Bézier curves can be rewritten in matrix form as follows:

$$\mathbf{b}(t) = \underbrace{\begin{bmatrix} 1 & \tau(t) & \cdots & \tau^p(t) \end{bmatrix}}_{\mathbf{T}(t)} \underbrace{\mathbf{P}\mathbf{L}^{-1}}_{\mathbf{M}} \boldsymbol{\xi},$$

where $\mathbf{T} : I \to \mathbb{R}^{p+1}$ parameterizes time, $\mathbf{P} \in \mathbb{R}^{p+1 \times p+1}$ is a diagonal matrix with entries equal to the binomial expansion of dimension $p+1$ and $\mathbf{L} \in \mathbb{R}^{p+1 \times p+1}$ is the lower triangular Pascal matrix [**?**].

Differentiating a Bézier curve of order p results in a Bézier curve of order $p-1$ defined via:

$$\dot{\mathbf{b}}(t) = \mathbf{z}(t)^\top \mathbf{S}_\tau \boldsymbol{\xi}$$

where the matrix $\mathbf{S}_\tau \in \mathbb{R}^{p \times p+1}$ as:

$$\mathbf{S}_{\tau,ii} = -\frac{p}{\tau}, \quad \mathbf{S}_{\tau,i,i+1}(\tau) = \frac{p}{\tau}, \quad i = 1, \dots, p,$$

with zeros everywhere else. Given a Bézier curve of order p, we can raise it to a curve of order $p+1$ with control points $\mathbf{R}^p \boldsymbol{\xi}$ where $\mathbf{R}^p \in \mathbb{R}^{p+1 \times p}$ contains all zeros except:

$$\mathbf{R}^p_{ii} = \frac{p+1-i}{p}, \quad \mathbf{R}^p_{i+1,i} = \frac{i}{p}, \quad i = 1, \dots, p.$$

The matrix $\mathbf{R}$ is taking a convex combination of the existing control points by evaluating the Bernstein basis polynomials at intermediate points along the curve. The process of curve refinement can be iterated to refine a curve arbitrarily to order $o > p$ with control points $\mathbf{R}^o \cdots \mathbf{R}^{p+1} \mathbf{R}^p \boldsymbol{\Xi}$.

With these tools, we can construct a matrix $\mathbf{H} \in \mathbb{R}^{p+1 \times p+1}$ relating a Bézier curve to its derivatives via a Bézier curve of the same order as $\mathbf{b}$ defined as:

$$\dot{\mathbf{b}}(t) = \mathbf{z}(t)^\top \underbrace{\mathbf{R}^p \mathbf{S}_\tau}_{\mathbf{H}} \boldsymbol{\xi}$$

Next, we introduce the notion of B-splines as a piecewise continuous collection of Bézier curves. A $j-$segmented B-spline of order p is simply a collection of order p Bézier curves connected with C^0 continuity. We can split an order p Beźier curve into a k segmented B-spline of order p for any $k \in \mathbb{N}$. This is accomplished via a matrix $\mathbf{Q}_i$, $i = 1, \dots, k$, defined as:

$$\sigma_i \triangleq \begin{cases} \frac{k-i}{k-i+1} & i < k \\ 1 & i = k \end{cases}$$

$$\mathbf{Z}(\sigma_i) \triangleq \begin{bmatrix} 1 & & & \\ & \sigma_i & & \\ & & \ddots & \\ & & & \sigma_i^p \end{bmatrix}, \quad \mathbf{Z}'(\sigma_i) \triangleq \begin{bmatrix} 1 & & & \\ & (1-\sigma_i) & & \\ & & \ddots & \\ & & & (1-\sigma_i)^p \end{bmatrix}$$

$$\mathbf{Q}^u_i \triangleq \mathbf{M}^{-1} \mathbf{Z}(\sigma_i) \mathbf{M},$$

$$\mathbf{Q}^l_i \triangleq \begin{bmatrix} \mathbf{0} & & 1 \\ & \iddots & \\ 1 & & \mathbf{0} \end{bmatrix} \mathbf{M}^{-1} \mathbf{Z}'(\sigma_i) \mathbf{M} \begin{bmatrix} \mathbf{0} & & 1 \\ & \iddots & \\ 1 & & \mathbf{0} \end{bmatrix}$$

$$\mathbf{Q}_i \triangleq \mathbf{Q}^u_i \mathbf{Q}^l_{i-1} \cdots \mathbf{Q}^l_0,$$

This notion will be useful when comparing Bézier curves with reference B-splines.

B. Vectorization

In this section, we will review matrix vectorization, its properties, and will explicitly define the vectorized matrices discussed in the text. Cosider matices $\mathbf{A} \in \mathbb{R}^{n \times m}$ and $\mathbf{B} \in \mathbb{R}^{k \times l}$. The vectorized versions of these matrices $\mathbf{a} \triangleq \mathrm{vec}(\mathbf{A}) \in \mathbb{R}^{nm}$ and $\mathbf{b}\mathrm{vec}(\mathbf{B}) \in \mathbb{R}^{kl}$ are defined as:

$$\begin{aligned} \mathbf{a}_{i+nj} &= \mathbf{A}_{i,j}, \quad i = 0, \dots, n, \quad j = 0, \dots m \\ \mathbf{b}_{i+lj} &= \mathbf{B}_{i,j}, \quad i = 0, \dots, k, \quad j = 0, \dots l \end{aligned}$$

i.e., they take the columns of the matrices and stack them vertically. There are a number of useful properties related to vectorization, summarized here:

$$\mathbf{a} = \mathbf{K}^{(n,m)} \mathrm{vec}(\mathbf{A}^\top)$$

$$\mathrm{vec}(\mathbf{AB}) = (\mathbf{I}_l \otimes \mathbf{A}) \mathrm{vec}(\mathbf{B}) = (\mathbf{B}^\top \otimes \mathbf{I}_n) \mathrm{vec}(\mathbf{A})$$

where $\mathbf{K}^{(n,m)} \in \mathbb{R}^{mn \times nm}$ is the commutation matrix, and $\otimes$ is the Kronecker product.

Using these properties, we can transform equation (14) into $\boldsymbol{\xi} = \mathbf{H}_{\text{vec}}$ via:

$$\boldsymbol{\xi} = \mathrm{vec}(\boldsymbol{\Xi}) = \mathrm{vec}\left(\begin{bmatrix} \mathbf{pH}^0 \\ \vdots \\ \mathbf{pH}^{\gamma-1} \end{bmatrix}\right)$$

$$= (\mathbf{K}^{(p+1,\gamma)} \otimes \mathbf{I}_m) \begin{bmatrix} \mathrm{vec}(\mathbf{pH}^0) \\ \vdots \\ \mathrm{vec}(\mathbf{pH}^{\gamma-1}) \end{bmatrix}$$

$$=(\mathrm{K}^{(p+1,\gamma)} \otimes \mathbf{I}_m) \begin{bmatrix} (\mathbf{H}^{0\top} \otimes \mathbf{I}_m) \\ \vdots \\ (\mathbf{H}^{\gamma-1\top} \otimes \mathbf{I}_m) \end{bmatrix}$$

$$= \underbrace{(\mathbf{K}^{(p+1,\gamma)} \otimes \mathbf{I}_m) \begin{bmatrix} (\mathbf{H}^{0\top} \otimes \mathbf{I}_m) \\ \vdots \\ (\mathbf{H}^{\gamma-1\top} \otimes \mathbf{I}_m) \end{bmatrix}}_{\mathbf{H}_{\text{vec}}}$$

Next, we can consider equation (15) as:

$$\begin{aligned}\mathrm{vec}(\boldsymbol{\Xi}\boldsymbol{\Delta}) &= \mathrm{vec}(\begin{bmatrix}\mathbf{x}_d(0) & \mathbf{x}_d(\tau)\end{bmatrix}) \\ \underbrace{(\boldsymbol{\Delta}^\top \otimes \mathbf{I}_n)}_{\triangleq \boldsymbol{\Delta}_{\text{vec}}}\boldsymbol{\xi} &= \begin{bmatrix}\mathbf{x}_d(0) \\ \mathbf{x}_d(\tau)\end{bmatrix}\end{aligned}$$

whereby we can reformulate this to a constraint on via:

$$\underbrace{\boldsymbol{\Delta}_{\text{vec}}\mathbf{H}_{\text{vec}}}_{\mathbf{D}_{\text{vec}}} = \begin{bmatrix}\mathbf{x}_d(0) \\ \mathbf{x}_d(\tau)\end{bmatrix}$$