C 2.2 Trajectory planning

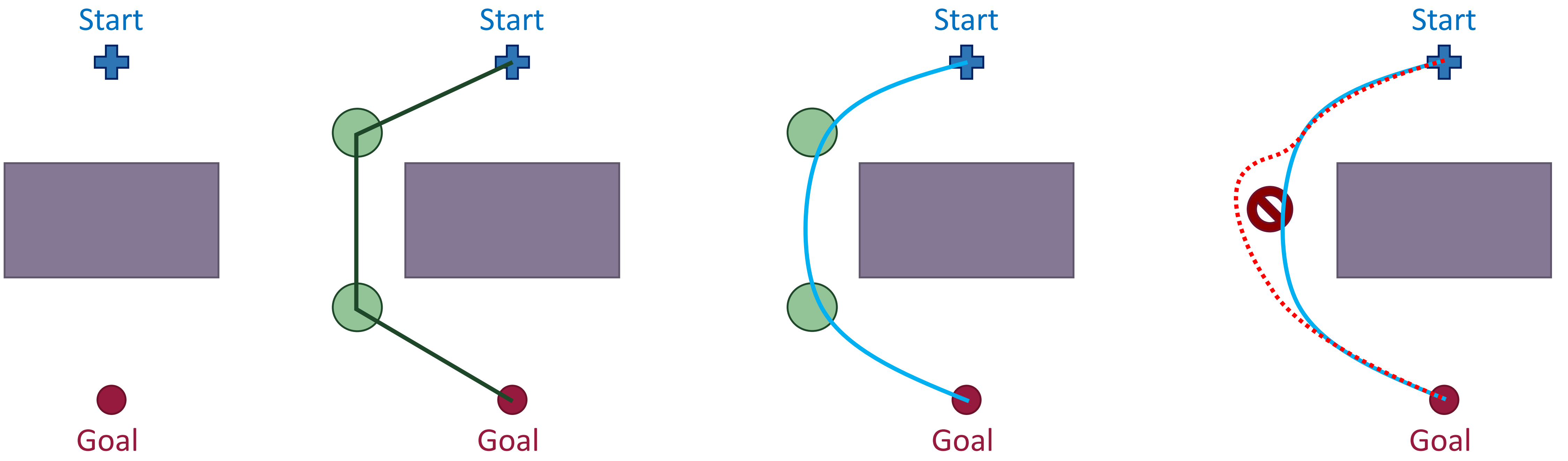
Gauthier ROUSSEAU

gauthier.rousseau.cs@gmail.com

Global to local planning

Motion planning hierarchization

Example of hierarchization Path finding Global planning Local (re)planning



Case study 1

Mission details

Problem

Considering a quadrotor initially at rest at $\mathbf{p}_{\mathrm{start}}$, generate trajectory of duration T, free, such that

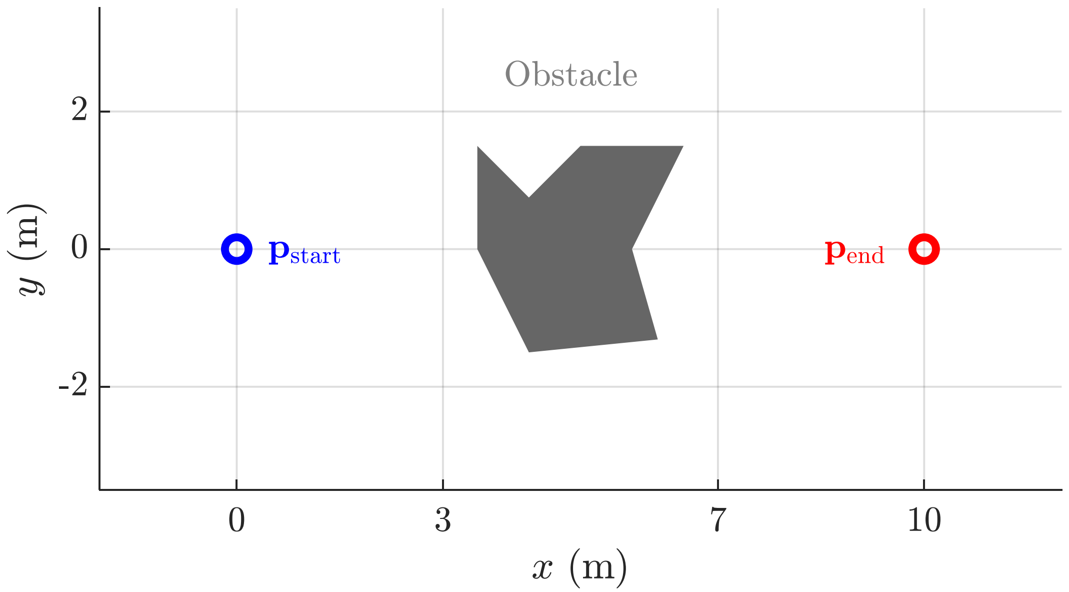
- The quadrotor ends resting at $\mathbf{p}_{\mathrm{end}}$
- The trajectory is collision-free
- The ground speed of the quadrotor v does not exceed $v_{\max} = 5\mathrm{m/s}$
- The angle of the quadrotor relatively to the ground α does not exceed $\alpha_{\max} = 20°$
- The rotation speed Ω of the quadrotor does not exceed $\Omega_{\max} = 100°/\mathrm{s}$

Hypothesis

- 2D problem (constant altitude)
- No movement on the yaw axis

Case study 1

Mission details



Bézier curves

Introduction

Introduced in the sixties for CAD

- 1958 **Paul de Casteljau** (ENS), engineer at Citröen
- 1962 **Pierre Bezier** (Arts&Metiers/Supelec), engineer at Renault

Intuitive way to parameterize polynomials - focused on the shape of the curve

Bézier curves

Definition

Set of $(n+1)$ control points

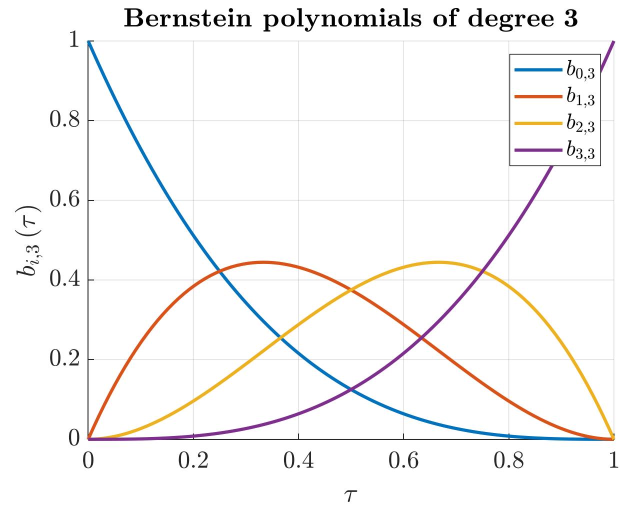
$$\mathbb{P} = \{\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_n\}$$

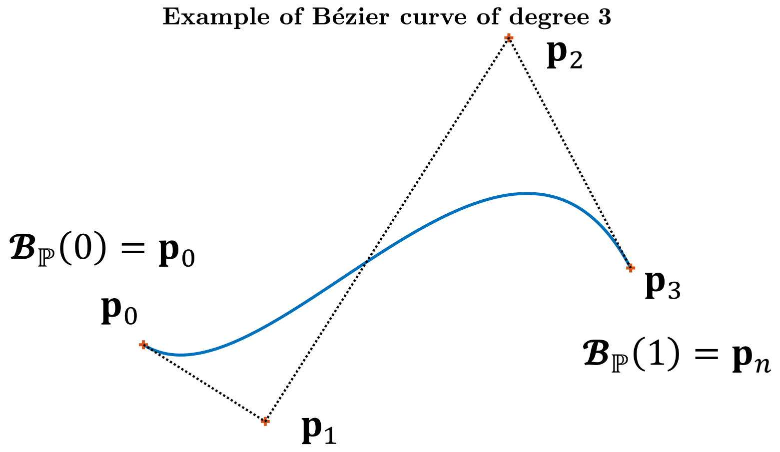
Bézier curve = polynomial curve of degree n

$$\boldsymbol{\mathcal{B}}_{\mathbb{P}} = \left(\begin{array}{ccc} [0,1] & \to & \mathbb{R}^d \\ \tau & \mapsto & \displaystyle\sum_{i=0}^{n} b_{i,n}(\tau)\, \mathbf{p}_i \end{array} \right)$$

Bernstein polynomials of degree n

$$b_{i,n}(\tau) = \binom{n}{i} \tau^i (1-\tau)^{n-i} = \frac{n!}{i!\,(n-i)!} \tau^i (1-\tau)^{n-i}$$





Bézier curves

Affine parameter transformation

Affine parameter transformation to change the interval of definition

$$\tau = \begin{pmatrix} [t_0, t_1] & \rightarrow & [0,1] \\ t & \mapsto & \dfrac{t - t_0}{t_1 - t_0} \end{pmatrix} \qquad \boldsymbol{\zeta}(t) = \boldsymbol{\mathcal{B}}_{\mathbb{P}}\big(\tau(t)\big)$$

Remember to include it in the derivative/integrals!

$$\frac{\mathrm{d}^l \boldsymbol{\zeta}}{\mathrm{d}t^l}(t) = \frac{1}{(t_1 - t_0)^l} \frac{\mathrm{d}^l \boldsymbol{\mathcal{B}}_{\mathbb{P}}}{\mathrm{d}\tau^l}\big(\tau(t)\big) \qquad \int_a^b \boldsymbol{\zeta}(t)\mathrm{d}t = (t_0 - t_1) \int_{\tau(a)}^{\tau(b)} \boldsymbol{\mathcal{B}}_{\mathbb{P}}(\nu)\mathrm{d}\nu$$

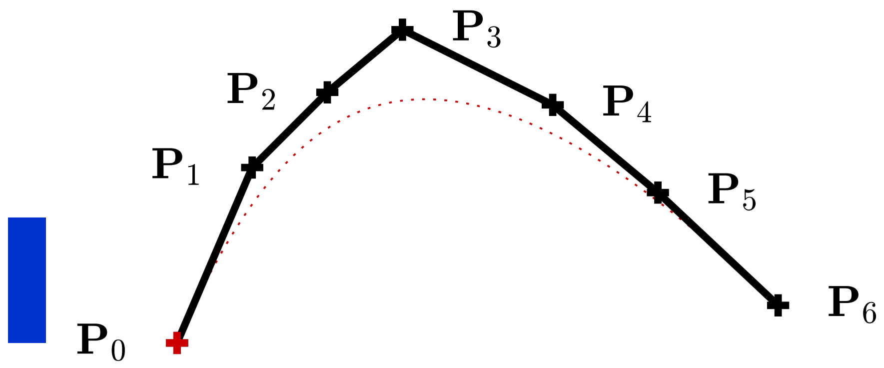
Bézier curves

Barycenter interpretation

Partition of unity

$$\forall \tau \in [0,1] : \begin{cases} \forall i \in [\![0,n]\!]\ b_{i,n}(\tau) \geq 0 \\ \displaystyle\sum_{i=0}^{n} b_{i,n}(\tau) = 1 \end{cases}$$

Bézier curve = Convex combination of control points weighted by Bernstein polynomials



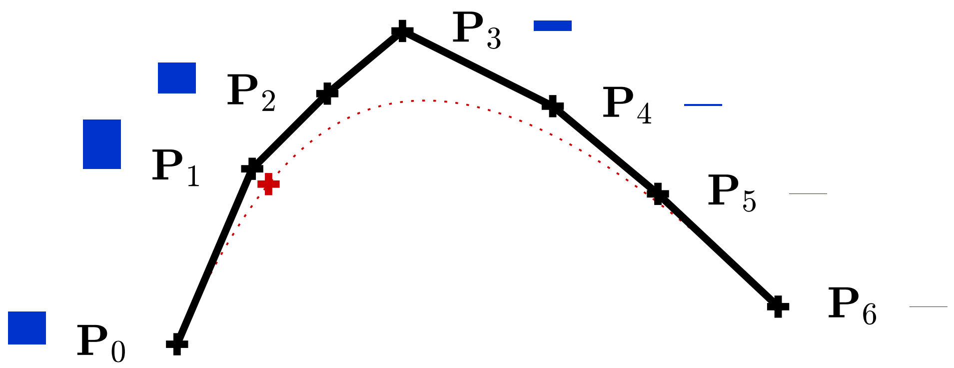
Bézier curves

Barycenter interpretation

Partition of unity

$$\forall \tau \in [0,1] : \begin{cases} \forall i \in [\![0, n]\!] \; b_{i,n}(\tau) \geq 0 \\ \displaystyle\sum_{i=0}^{n} b_{i,n}(\tau) = 1 \end{cases}$$

Bézier curve = Convex combination of control points weighted by Bernstein polynomials



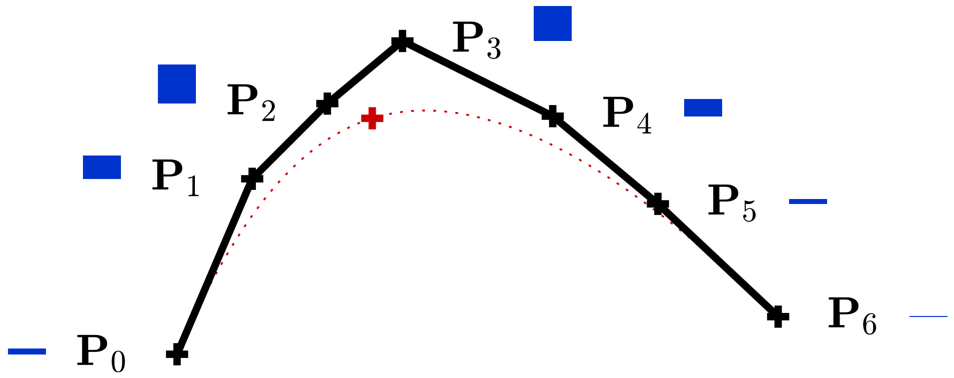
Bézier curves

Barycenter interpretation

Partition of unity

$$\forall \tau \in [0,1] : \begin{cases} \forall i \in [\![0,n]\!]\ b_{i,n}(\tau) \geq 0 \\ \displaystyle\sum_{i=0}^{n} b_{i,n}(\tau) = 1 \end{cases}$$

Bézier curve = Convex combination of control points weighted by Bernstein polynomials



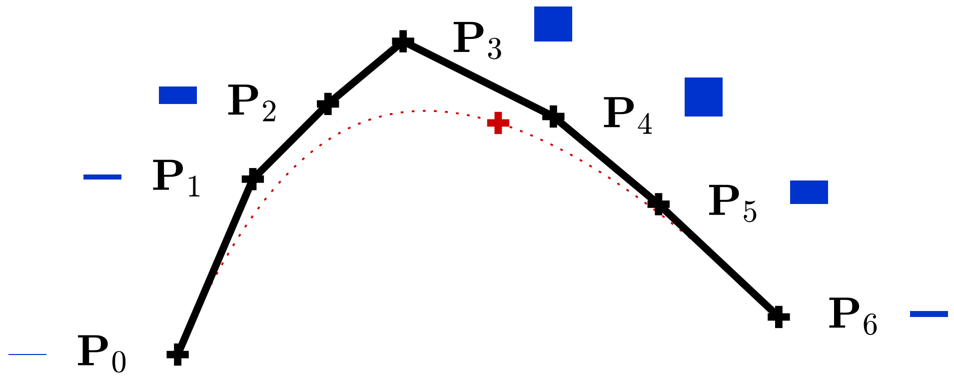
Bézier curves

Barycenter interpretation

Partition of unity

$$\forall \tau \in [0,1] : \begin{cases} \forall i \in [\![0, n]\!]\ b_{i,n}(\tau) \geq 0 \\ \displaystyle\sum_{i=0}^{n} b_{i,n}(\tau) = 1 \end{cases}$$

Bézier curve = Convex combination of control points weighted by Bernstein polynomials



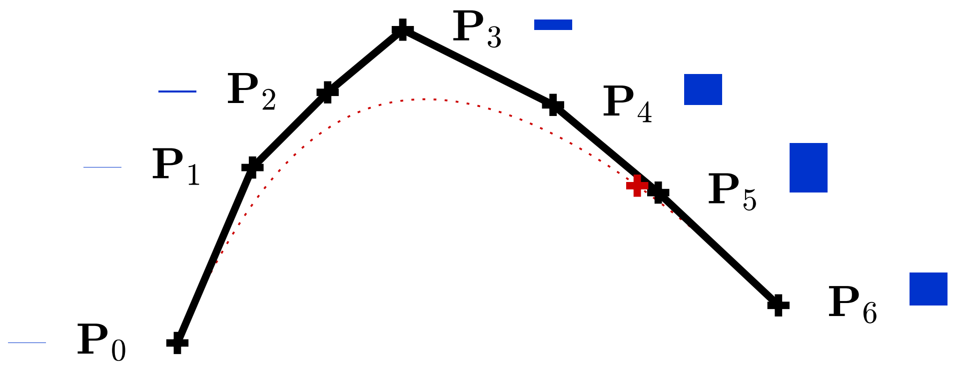
Bézier curves

Barycenter interpretation

Partition of unity

$$\forall \tau \in [0,1] : \begin{cases} \forall i \in [\![0,n]\!] \; b_{i,n}(\tau) \geq 0 \\ \sum_{i=0}^{n} b_{i,n}(\tau) = 1 \end{cases}$$

Bézier curve = Convex combination of control points weighted by Bernstein polynomials



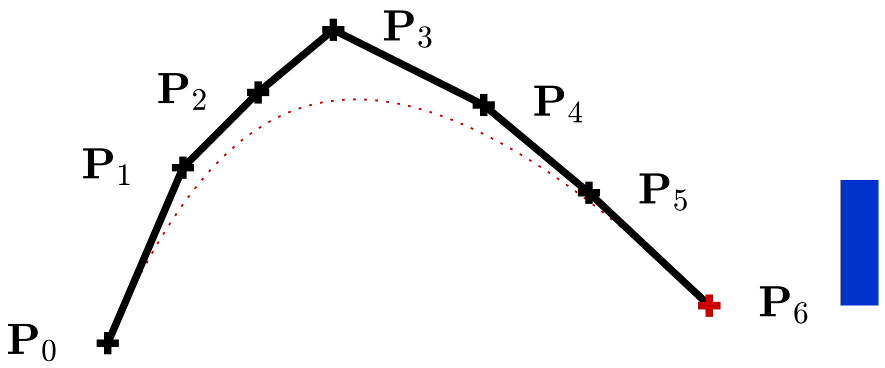
Bézier curves

Barycenter interpretation

Partition of unity

$$\forall \tau \in [0{,}1] : \begin{cases} \forall i \in [\![0, n]\!]\ b_{i,n}(\tau) \geq 0 \\ \displaystyle\sum_{i=0}^{n} b_{i,n}(\tau) = 1 \end{cases}$$

Bézier curve = Convex combination of control points weighted by Bernstein polynomials



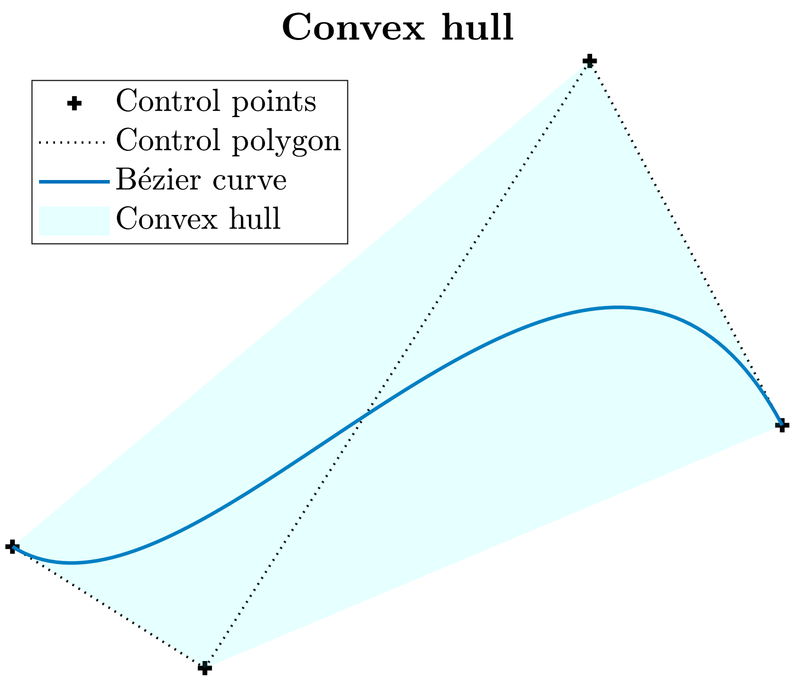
Bézier curves

Convex hull property

A Bézier curve lies **within the convex hull of its control points**

$$\forall \tau \in [0,1] \quad \boldsymbol{\mathcal{B}}_{\mathbb{P}}(\tau) \in \mathrm{Conv}(\mathbb{P})$$

- Direct consequence of convex combination result
- Can be used to constrain curve into a convex region (only a **sufficient condition**)

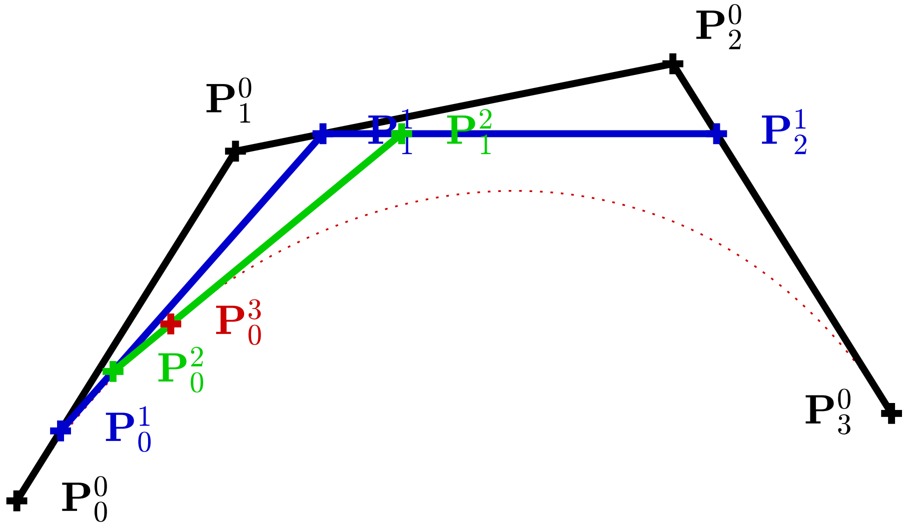


Bézier curves

De Casteljau algorithm

Numerically stable evaluation of a Bézier curve through successive convex combinations

$$\boldsymbol{\mathcal{B}}_{\mathbb{P}}(\tau) = \sum_{i=0}^{n} b_{i,n}(\tau)\,\mathbf{p}_i^0 = \sum_{i=0}^{n-1} b_{i,n-1}(\tau)\,\mathbf{p}_i^1 = \cdots = \sum_{i=0}^{0} b_{i,n}(\tau)\,\mathbf{p}_i^n = \mathbf{p}_i^n$$



with $\mathbf{p}_i^0 = \mathbf{p}_i$ and $\mathbf{p}_i^j = (1-\tau)\mathbf{p}_i^{j-1} + \tau\mathbf{p}_{i+1}^{j-1}$

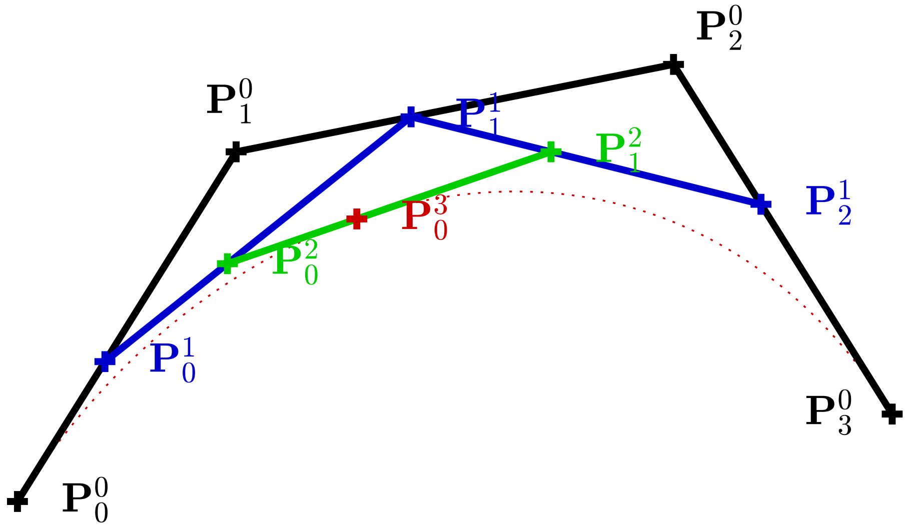
Bézier curves

De Casteljau algorithm

Numerically stable evaluation of a Bézier curve through successive convex combinations

$$\boldsymbol{\mathcal{B}}_{\mathbb{P}}(\tau) = \sum_{i=0}^{n} b_{i,n}(\tau)\,\mathbf{p}_i^0 = \sum_{i=0}^{n-1} b_{i,n-1}(\tau)\,\mathbf{p}_i^1 = \cdots = \sum_{i=0}^{0} b_{i,n}(\tau)\,\mathbf{p}_i^n = \mathbf{p}_i^n$$

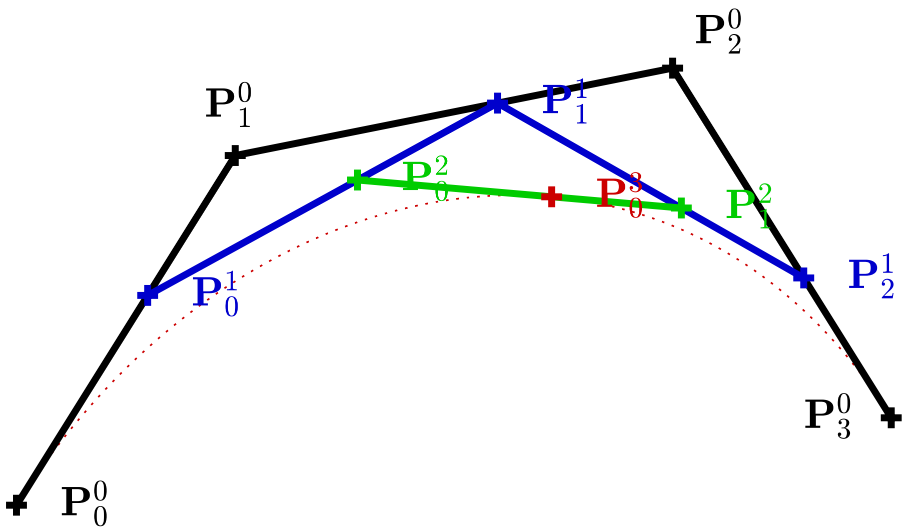
with $\mathbf{p}_i^0 = \mathbf{p}_i$ and $\mathbf{p}_i^j = (1-\tau)\mathbf{p}_i^{j-1} + \tau\mathbf{p}_{i+1}^{j-1}$



Bézier curves

De Casteljau algorithm

Numerically stable evaluation of a Bézier curve through successive convex combinations



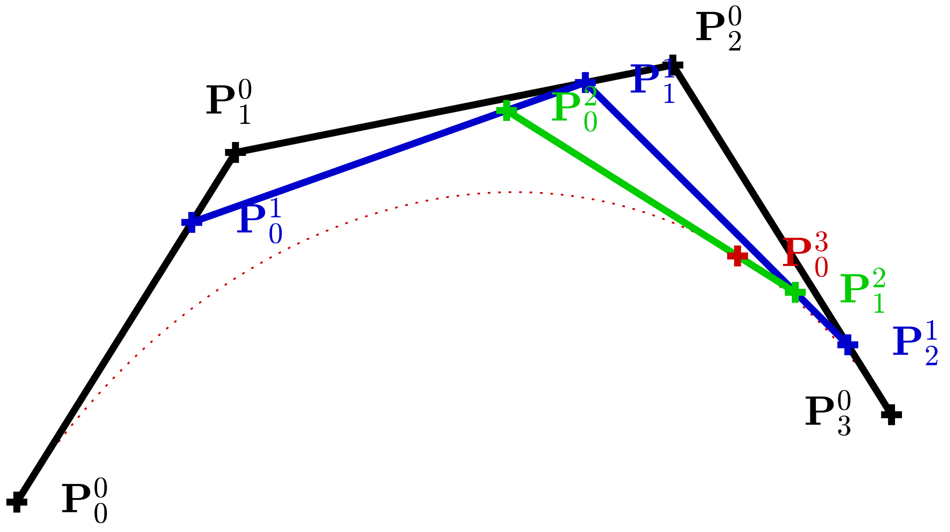
$$\boldsymbol{\mathcal{B}}_{\mathbb{P}}(\tau) = \sum_{i=0}^{n} b_{i,n}(\tau)\,\mathbf{p}_i^0 = \sum_{i=0}^{n-1} b_{i,n-1}(\tau)\,\mathbf{p}_i^1 = \cdots = \sum_{i=0}^{0} b_{i,n}(\tau)\,\mathbf{p}_i^n = \mathbf{p}_i^n$$

with $\mathbf{p}_i^0 = \mathbf{p}_i$ and $\mathbf{p}_i^j = (1-\tau)\mathbf{p}_i^{j-1} + \tau\mathbf{p}_{i+1}^{j-1}$

Bézier curves

De Casteljau algorithm

Numerically stable evaluation of a Bézier curve through successive convex combinations



$$\boldsymbol{\mathcal{B}}_{\mathbb{P}}(\tau) = \sum_{i=0}^{n} b_{i,n}(\tau)\,\mathbf{p}_i^0 = \sum_{i=0}^{n-1} b_{i,n-1}(\tau)\,\mathbf{p}_i^1 = \cdots = \sum_{i=0}^{0} b_{i,n}(\tau)\,\mathbf{p}_i^n = \mathbf{p}_i^n$$

with $\mathbf{p}_i^0 = \mathbf{p}_i$ and $\mathbf{p}_i^j = (1-\tau)\mathbf{p}_i^{j-1} + \tau\mathbf{p}_{i+1}^{j-1}$

Bézier curves

Polynomial interpretation

Bernstein polynomials of degree n are a basis of the vector space of polynomials of degree n and less

Bézier curve = Projection of a polynomial curve on the Bernstein basis (**control points** = corresponding **coordinates**)

Monomial basis

$$\boldsymbol{\mathcal{B}}_{\mathbb{P}}(\tau) = \sum_{i=0}^{n} \mathbf{c}_i\, \tau^i$$

$$\mathbf{p}_i = \binom{n}{i}^{-1} \sum_{j=0}^{i} \binom{n-j}{i-j} \mathbf{c}_i$$

$$\mathbf{c}_i = \binom{n}{i} \sum_{j=0}^{i} (-1)^{i-j} \binom{i}{j} \mathbf{p}_i$$

Bernstein basis

$$\boldsymbol{\mathcal{B}}_{\mathbb{P}}(\tau) = \sum_{i=0}^{n} b_{i,n}(\tau)\, \mathbf{p}_i$$

Bézier curves

Derivatives and integrals

Derivative

- Derivative is also a Bézier curve
- Degree $n-1$
- Control points obtained by linear combination

$$\boldsymbol{\mathcal{B}}'_{\mathbb{P}} = \boldsymbol{\mathcal{B}}_{\mathbb{P}^{(1)}}$$

$$\forall i \in [\![0, n-1]\!] \quad \mathbf{p}_i^{(1)} = n(\mathbf{p}_{i+1} - \mathbf{p}_i)$$

Integral

- Integral is also a Bézier curve
- Degree $n+1$
- Control points obtained by linear combination

$$\int \boldsymbol{\mathcal{B}}_{\mathbb{P}} = \boldsymbol{\mathcal{B}}_{\mathbb{P}^{(-1)}}$$

$$\forall i \in [\![0, n]\!] \quad \mathbf{p}_{i+1}^{(-1)} = \frac{1}{n+1}\mathbf{p}_i + \mathbf{p}_i^{(-1)}$$

with $\mathbf{p}_0^{(-1)}$ initial condition

Bézier curves

A powerful tool

Convex hull property

Derivatives and integral are Bézier curve, their control points are linear combinations of the original ones

➡ Powerful tool for constraining a trajectory and its derivatives/integrals in convex regions

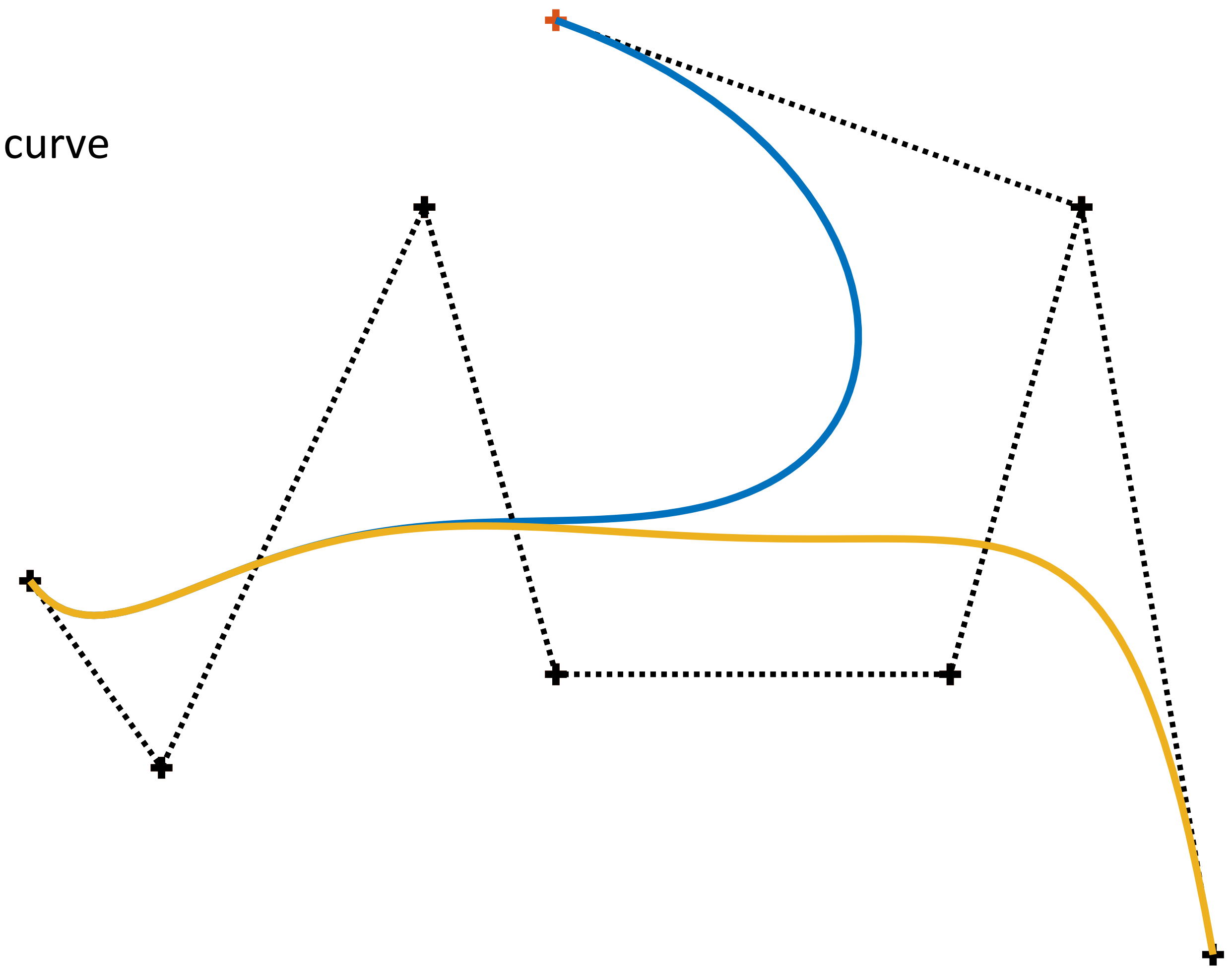
Examples: flight corridors, obstacle free convex regions, speed/acceleration limitations, …

Bézier curves

Limitations

Each control point impact the entire curve

+1 control ➡ point degree +1



Bézier curves

Limitations

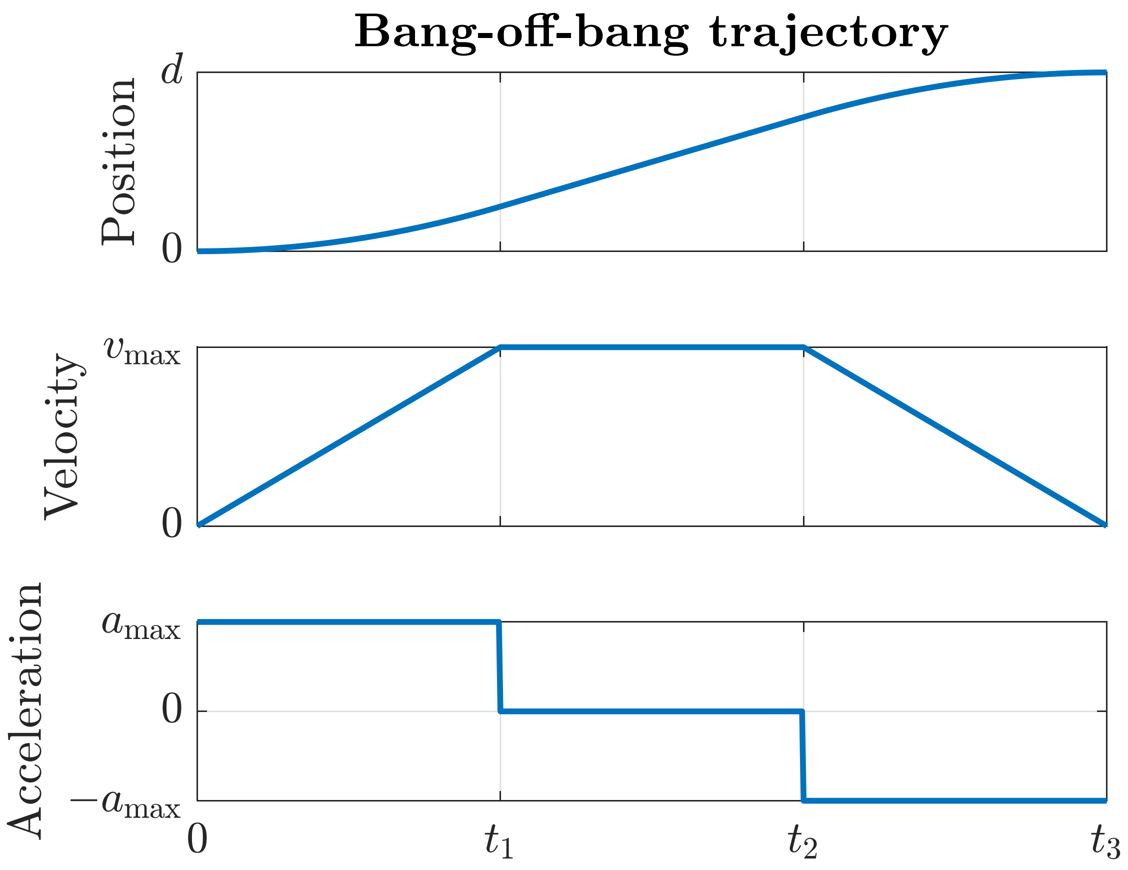
“Simple” polynomial

Generate a **minimum-time, rest-to-rest trajectory**

- Distance d
- Maximum speed $v_{\max}$
- Maximum acceleration $a_{\max}$

➡ Intuitive and optimal solution:
Bang-off-bang acceleration law

Simple piecewise degree 2 polynomial



Bézier curves

Limitations

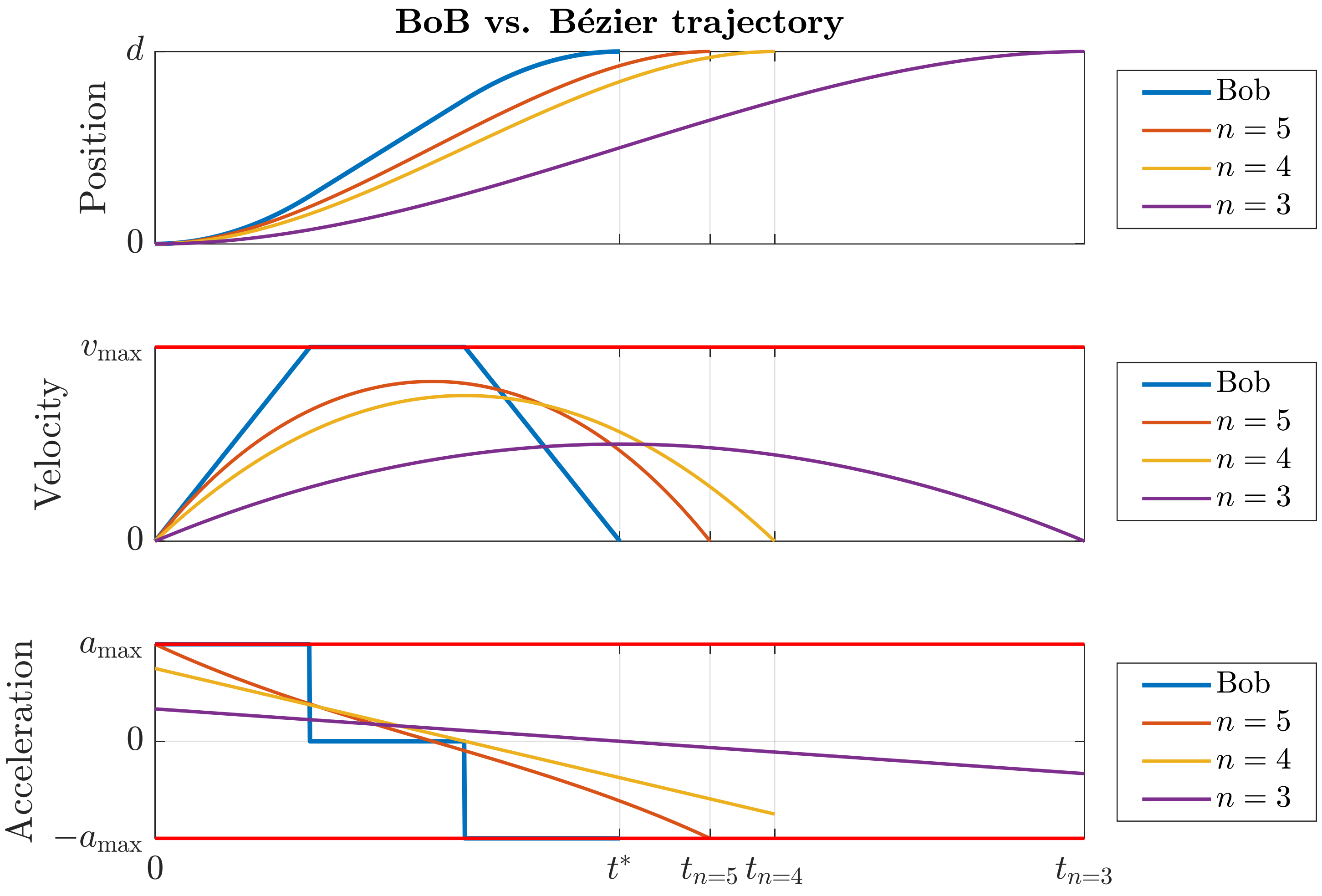
"Simple" polynomial

Generate a **minimum-time, rest-to-rest trajectory**

- Distance d
- Maximum speed $v_{\max}$
- Maximum acceleration $a_{\max}$

➡ Intuitive and optimal solution:
Bang-off-bang acceleration law

Simple piecewise degree 2 polynomial



Bézier curves

Limitations

“Simple” polynomial

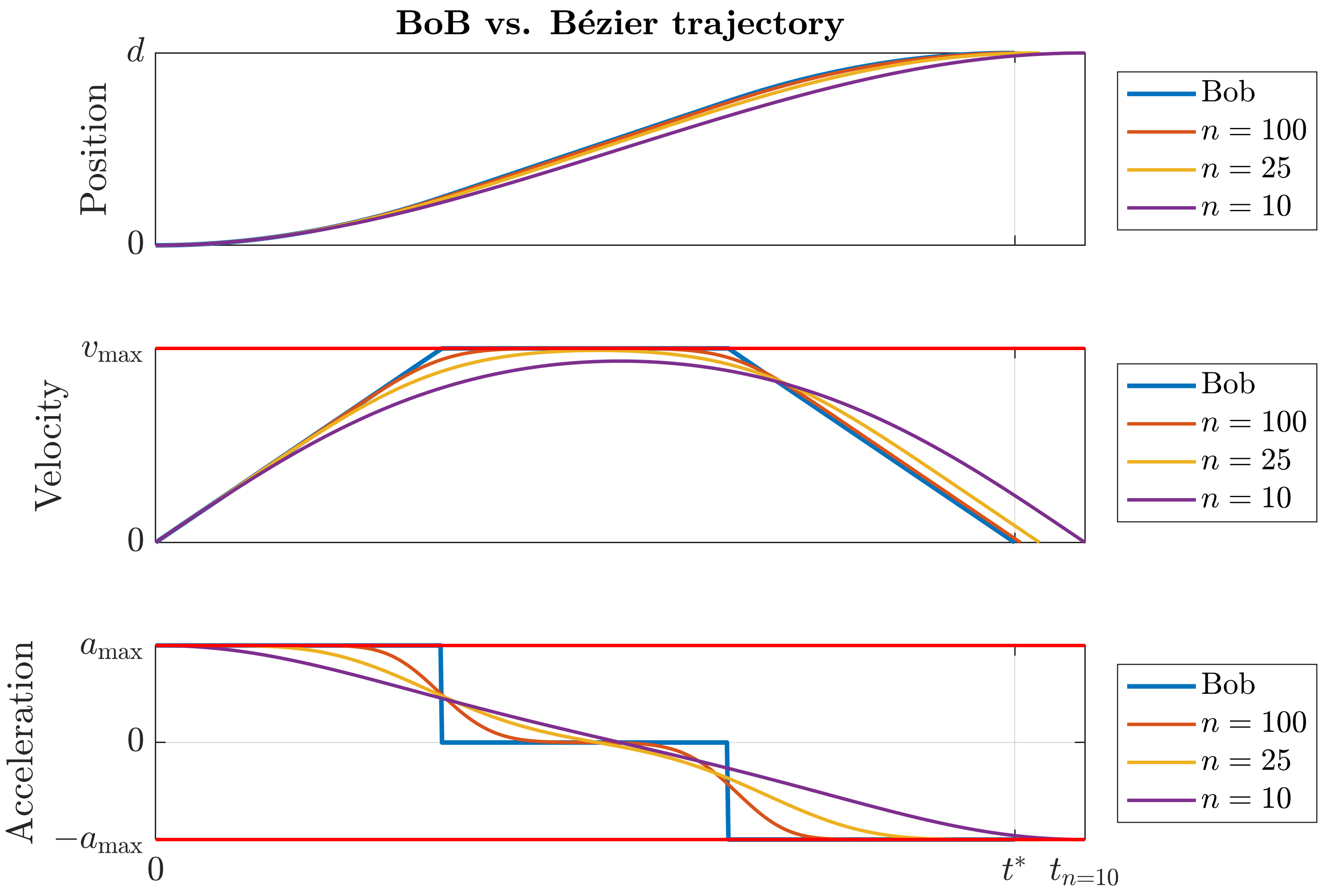
Generate a **minimum-time, rest-to-rest trajectory**

- Distance d
- Maximum speed $v_{\max}$
- Maximum acceleration $a_{\max}$

➡ Intuitive and optimal solution:
Bang-off-bang acceleration law

Simple piecewise degree 2 polynomial

Piecewise trajectory = more degrees of freedom



B-spline curves

Introduction

Introduced in the **forties** (Isaac J. Schoenberg)

Developed in the **seventies for CAD and computer graphics**
(Carl R. de Boor, Maurice Cox, Richard Riesenfeld, Wolfgang Boehm)

Extension of Bezier curves to **piecewise polynomials**

B-spline curves

Introduction

Bézier curve

- Degree given by number of control points
- Defined on $[0,1]$
- Each control point impacts the shape of the whole curve

B-spline curve

- Degree can be lower than number of control points
- Defined on arbitrary interval
- Piecewise, with adjustable continuity at the connections
- Local influence of control points

B-spline curves

Definition - Curve

- Set of $(n+1)$ control points $\mathbb{P} = \{\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_n\}$
- Polynomial degree $k \leq n$
- Vector of $(m+1) = n+k+2$ increasing knots $\boldsymbol{\tau} = (\tau_0, \tau_1, \dots, \tau_m)$ ➡ Instants where the polynomial representation changes

B-spline curve = piecewise polynomial curve of degree k

$$\boldsymbol{\mathcal{B}}_{\mathbb{P},\boldsymbol{\tau}} = \begin{pmatrix} [\tau_k, \tau_{n+1}] & \rightarrow & \mathbb{R}^d \\ t & \mapsto & \displaystyle\sum_{i=0}^{n} N_{i,k}^{\boldsymbol{\tau}}(t)\ \mathbf{p}_i \end{pmatrix}$$

$N_{i,k}^{\boldsymbol{\tau}}(t)$ Basis-spline (B-spline) functions of degree k and knots $\boldsymbol{\tau}$

B-spline curves

Definition – Basis functions

$N_{i,k}^{\boldsymbol{\tau}}(t)$ Basis-spline (B-spline) functions of degree k and knots $\boldsymbol{\tau}$

$\forall i \in [\![0, n]\!] \quad \forall t \in \mathbb{R}$

If $k = 0$

$$N_{i,k}^{\boldsymbol{\tau}}(t) = f(x) = \begin{cases} 1 & \text{if } \tau_i \leq t < \tau_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

Else

$$N_{i,k}^{\boldsymbol{\tau}}(t) = \omega_{i,k}^{\boldsymbol{\tau}}(t) N_{i,k-1}^{\boldsymbol{\tau}}(t) + \left(1 - \omega_{i,k}^{\boldsymbol{\tau}}(t)\right) N_{i+1,k-1}^{\boldsymbol{\tau}}(t)$$

with

$$\omega_{i,k}^{\boldsymbol{\tau}}(t) = \begin{cases} \dfrac{t - \tau_i}{\tau_{i+k} - \tau_i} & \text{if } \tau_{i+k} > \tau_i \\ 0 & \text{otherwise} \end{cases}$$

Support

$N_{i,k}^{\boldsymbol{\tau}}(t) = 0$ outside of $[\tau_i, \tau_{i+k+1}[$

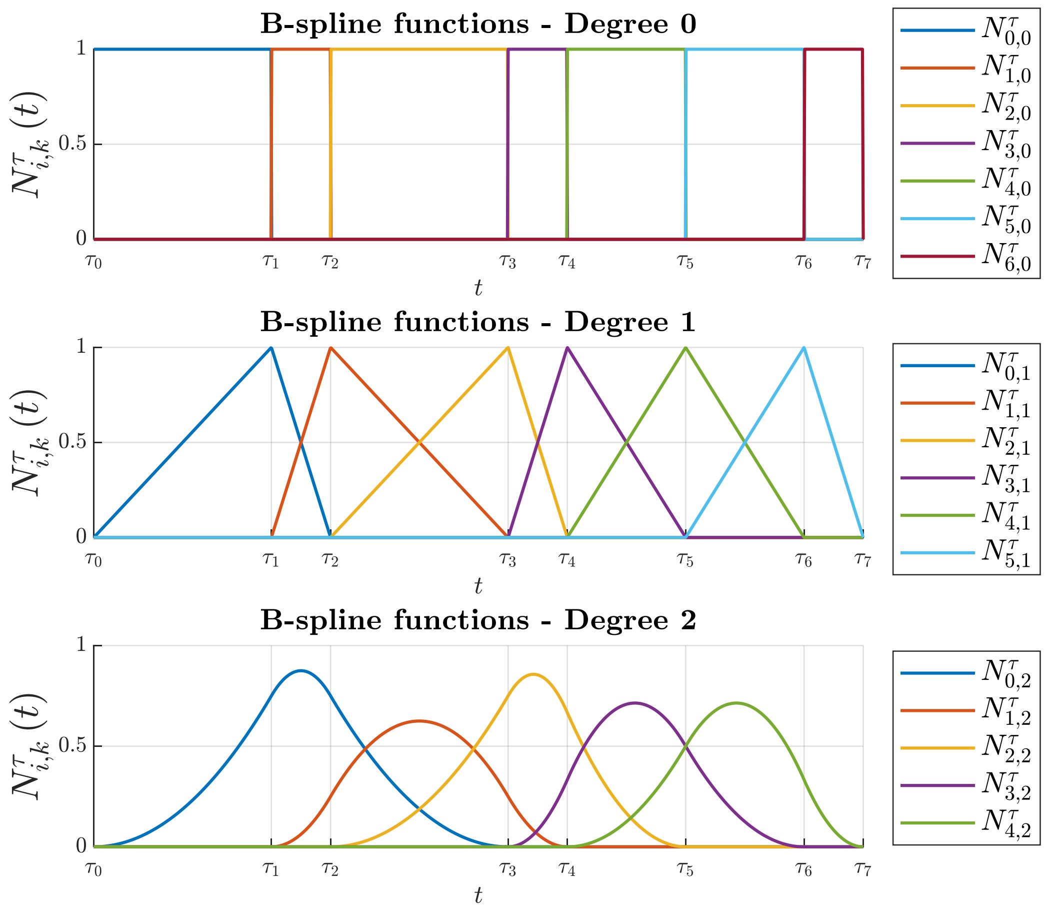
Special case

If $\tau = (\underbrace{0, \ \ldots \ , \ 0,}_{k+1 \text{ knots}} \ \underbrace{1, \ \ldots \ , \ 1}_{k+1 \text{ knots}})$

then $N_{i,k}^{\boldsymbol{\tau}}(t) = b_{i,k}$

B-spline curves

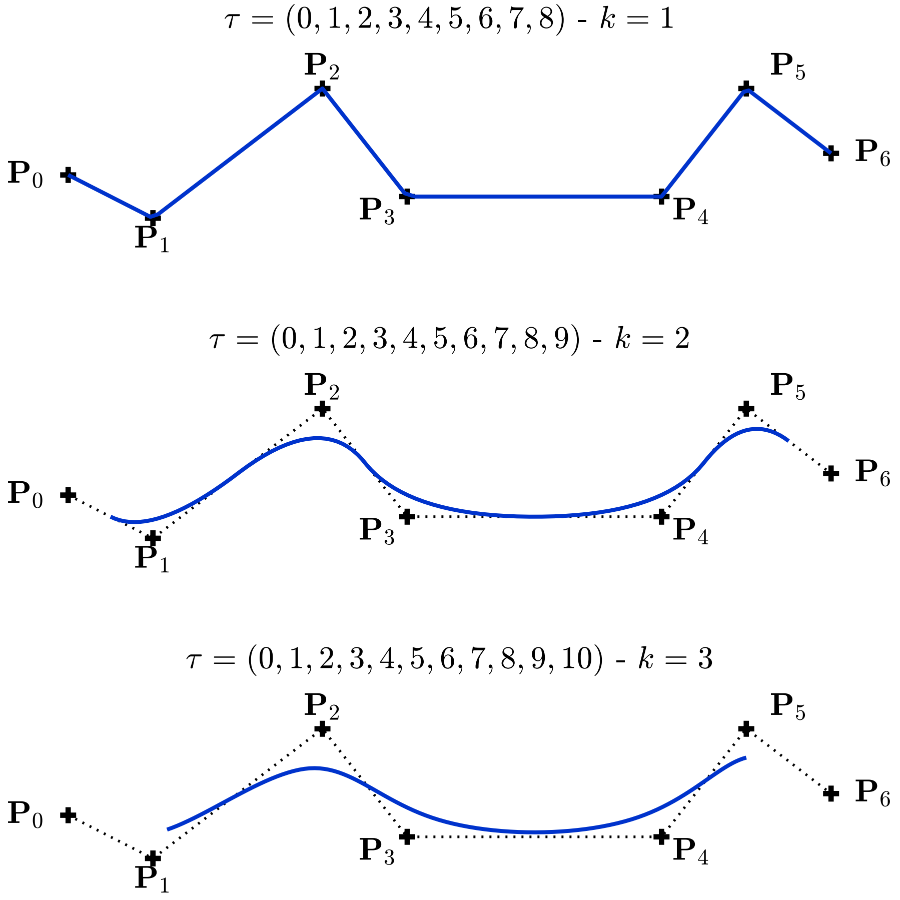
Examples



$$\tau = (0, 3, 4, 7, 8, 10, 12, 13)$$

B-spline curves

Examples

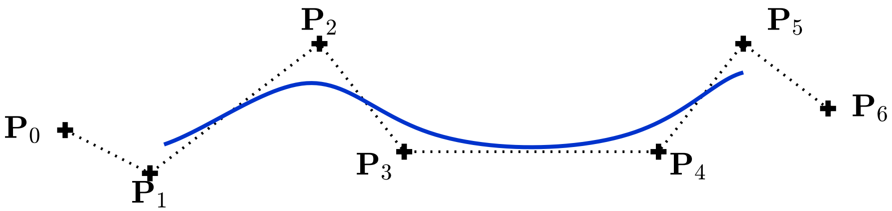


$$\mathbb{P} = \left\{ \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 3 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 0.5 \end{pmatrix}, \begin{pmatrix} 7 \\ 0.5 \end{pmatrix}, \begin{pmatrix} 8 \\ 3 \end{pmatrix}, \begin{pmatrix} 9 \\ 1.5 \end{pmatrix} \right\}$$

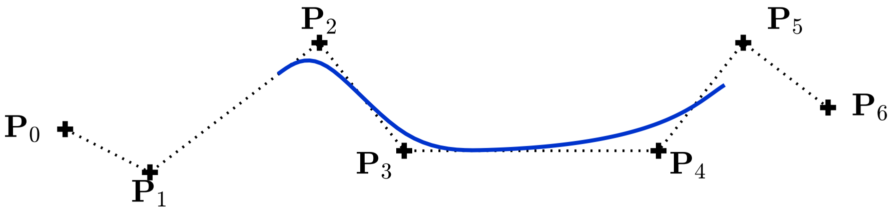
B-spline curves

Examples

$\tau = (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$ - $k = 3$

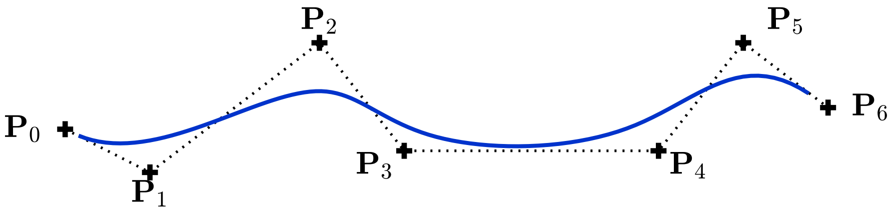


$\tau = (0, 10, 20, 30, 31, 32, 33, 40, 50, 60, 70)$ - $k = 3$



$$\mathbb{P} = \left\{ \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 3 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 0.5 \end{pmatrix}, \begin{pmatrix} 7 \\ 0.5 \end{pmatrix}, \begin{pmatrix} 8 \\ 3 \end{pmatrix}, \begin{pmatrix} 9 \\ 1.5 \end{pmatrix} \right\}$$

$\tau = (0, 1, 2, 3, 20, 30, 40, 51, 52, 53, 54)$ - $k = 3$

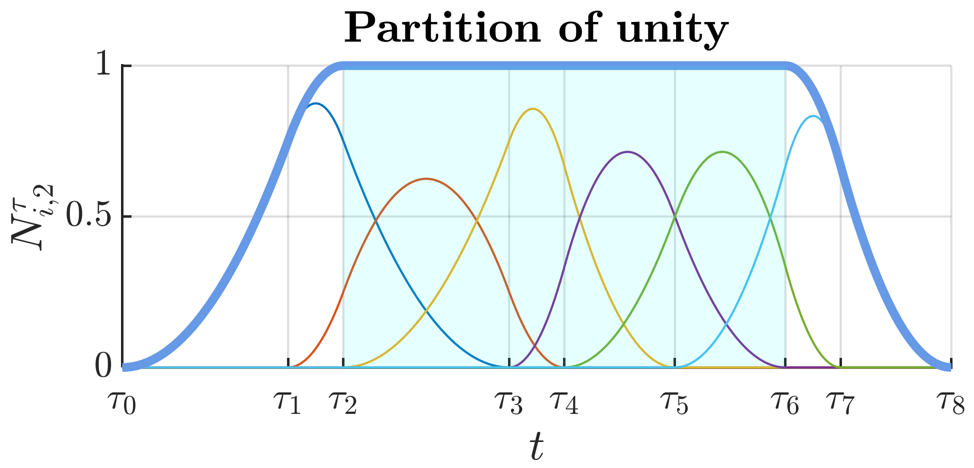


B-spline curves

Convex hull property

Partition of unity

$$\begin{cases} \forall i \in [\![0, n]\!] \quad N^{\boldsymbol{\tau}}_{i,k} \geq 0 \\ \forall j \in [\![k, n]\!] \quad \forall t \in [\tau_j, \tau_{j+1}[\quad \sum_{i=0}^{n} N^{\boldsymbol{\tau}}_{i,k}(t) = \sum_{i=j}^{j+k+1} N^{\boldsymbol{\tau}}_{i,k}(t) = 1 \end{cases}$$



Convex hull property

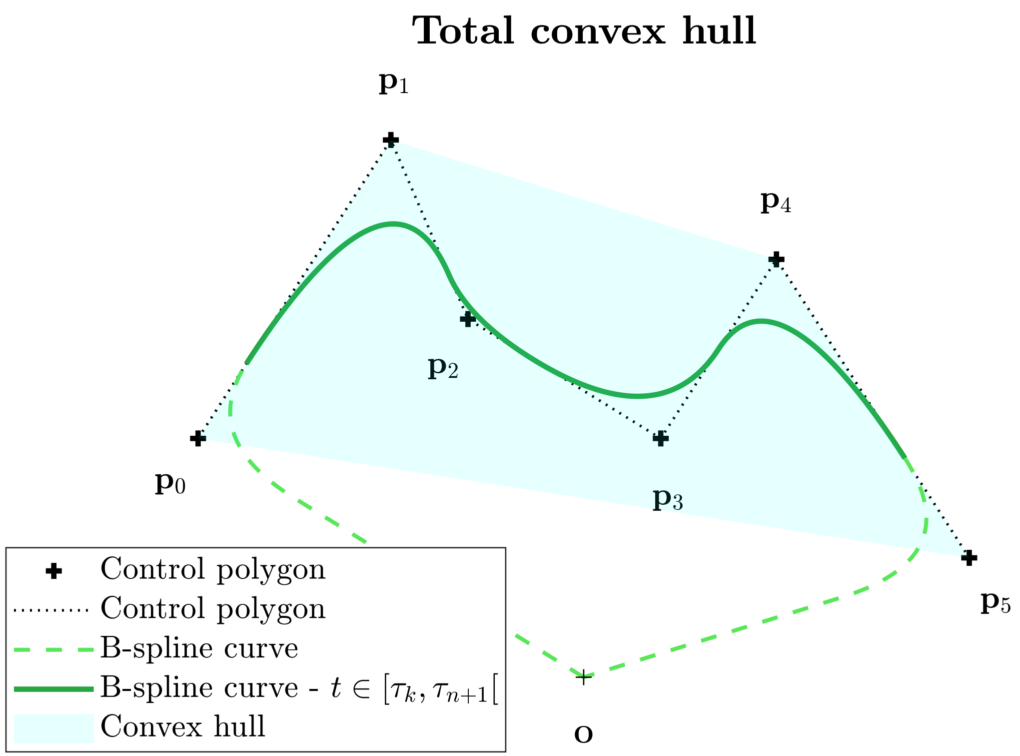
$$\forall j \in [\![k, n]\!] \quad \forall t \in [\tau_j, \tau_{j+1}[\quad \boldsymbol{\mathcal{B}}_{\mathbb{P},\boldsymbol{\tau}}(\boldsymbol{t}) \in \mathrm{Conv}(\{\mathbf{p}_i \mid i \in [\![j-k, j]\!]\})$$

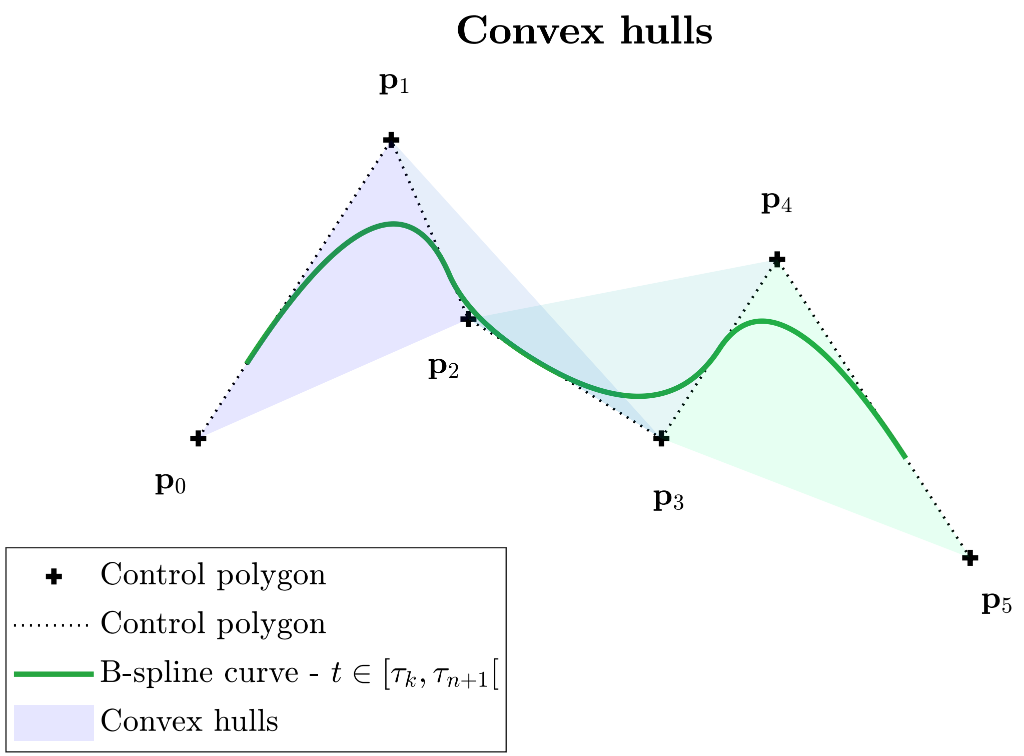
Internal knots

$$(\underbrace{\tau_0, \ldots, \tau_{k-1}}_{k \text{ knots}}, \underset{\text{start knot}}{\underset{\downarrow}{\tau_k}}, \underbrace{\tau_{k+1}, \ldots, \tau_n}_{\substack{n-k \\ \text{internal knots}}}, \underset{\text{end knot}}{\underset{\downarrow}{\tau_{n+1}}}, \underbrace{\tau_{n+2}, \ldots, \tau_m}_{k \text{ knots}})$$

B-spline curves

Convex hull property





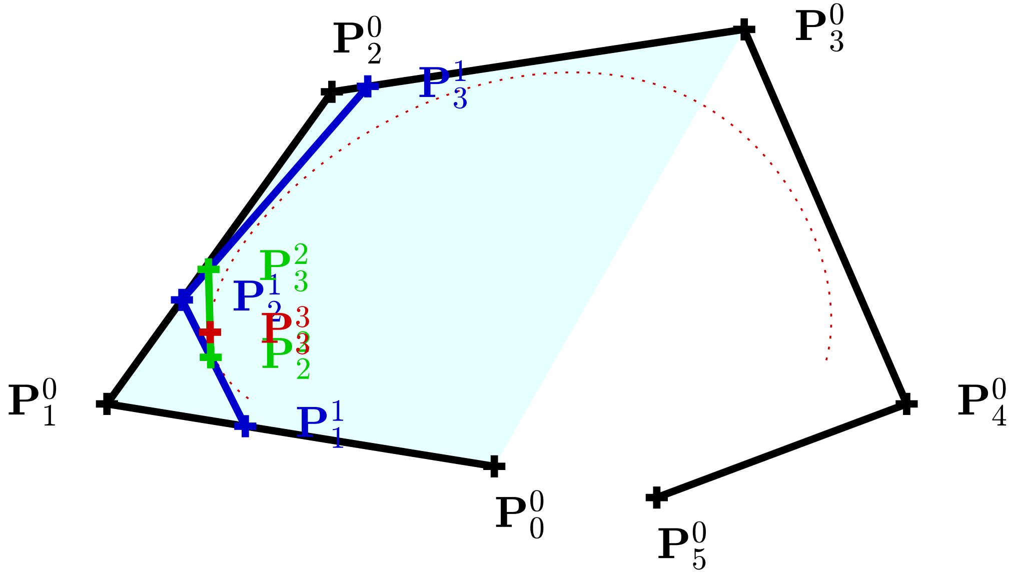
B-spline curves

de Boor – Cox algorithm

Extension of de Casteljau algorithm to B-splines

$\forall j \in [\![k, n]\!] \quad \forall t \in [\tau_j, \tau_{j+1}[$

$$
\begin{aligned}
\mathcal{B}_{\mathbb{P},\boldsymbol{\tau}}(t) &= \sum_{i=j-k}^{j} N_{i,k}^{\boldsymbol{\tau}}(t)\, \mathbf{p}_i^0 \\
&= \sum_{i=j-k+1}^{j} N_{i,k-1}^{\boldsymbol{\tau}}(t)\, \mathbf{p}_i^1 \\
&= \cdots \\
&= \sum_{i=j}^{j} N_{i,0}^{\boldsymbol{\tau}}(t)\, \mathbf{p}_i^k = \mathbf{p}_i^k
\end{aligned}
$$



with $\mathbf{p}_i^0 = \mathbf{p}_i$ and $\mathbf{p}_i^j = \omega_{i,k-1}^{\boldsymbol{\tau}}(t)\, \mathbf{p}_i^{j-1} + \left(1 - \omega_{i,k-1}^{\boldsymbol{\tau}}(t)\right) \mathbf{p}_{i-1}^{j-1}$

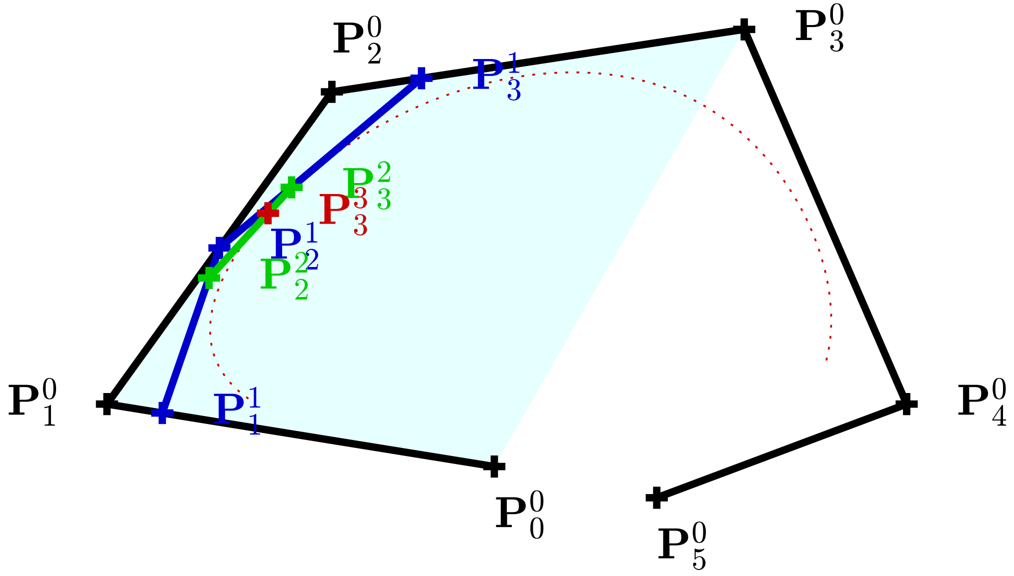
B-spline curves

de Boor – Cox algorithm

Extension of de Casteljau algorithm to B-splines

$\forall j \in [\![k, n]\!] \quad \forall t \in [\tau_j, \tau_{j+1}[$

$$\begin{aligned}\boldsymbol{\mathcal{B}}_{\mathbb{P},\boldsymbol{\tau}}(t) &= \sum_{i=j-k}^{j} N_{i,k}^{\boldsymbol{\tau}}(t)\,\mathbf{p}_i^0 \\ &= \sum_{i=j-k+1}^{j} N_{i,k-1}^{\boldsymbol{\tau}}(t)\,\mathbf{p}_i^1 \\ &= \cdots \\ &= \sum_{i=j}^{j} N_{i,0}^{\boldsymbol{\tau}}(t)\,\mathbf{p}_i^k = \mathbf{p}_i^k\end{aligned}$$



with $\mathbf{p}_i^0 = \mathbf{p}_i$ and $\mathbf{p}_i^j = \omega_{i,k-1}^{\boldsymbol{\tau}}(t)\,\mathbf{p}_i^{j-1} + \left(1 - \omega_{i,k-1}^{\boldsymbol{\tau}}(t)\right)\mathbf{p}_{i-1}^{j-1}$

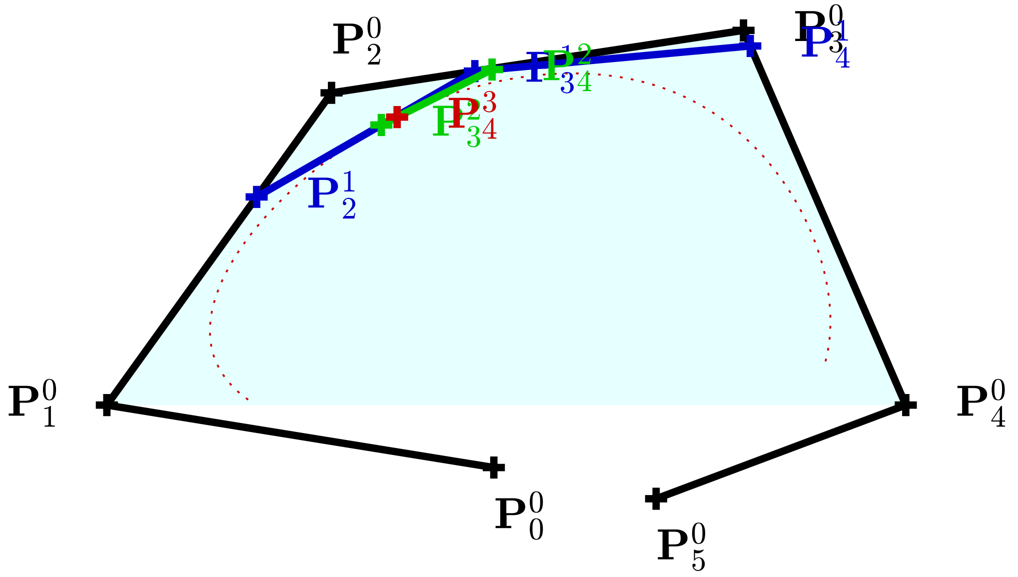
B-spline curves

de Boor – Cox algorithm

Extension of de Casteljau algorithm to B-splines

$\forall j \in [\![k, n]\!] \quad \forall t \in \left[\tau_j, \tau_{j+1}\right[$

$$
\begin{aligned}
\mathcal{B}_{\mathbb{P},\boldsymbol{\tau}}(t) &= \sum_{i=j-k}^{j} N_{i,k}^{\boldsymbol{\tau}}(t)\, \mathbf{p}_i^0 \\
&= \sum_{i=j-k+1}^{j} N_{i,k-1}^{\boldsymbol{\tau}}(t)\, \mathbf{p}_i^1 \\
&= \cdots \\
&= \sum_{i=j}^{j} N_{i,0}^{\boldsymbol{\tau}}(t)\, \mathbf{p}_i^k = \mathbf{p}_i^k
\end{aligned}
$$



with $\mathbf{p}_i^0 = \mathbf{p}_i$ and $\mathbf{p}_i^j = \omega_{i,k-1}^{\boldsymbol{\tau}}(t)\, \mathbf{p}_i^{j-1} + \left(1 - \omega_{i,k-1}^{\boldsymbol{\tau}}(t)\right) \mathbf{p}_{i-1}^{j-1}$

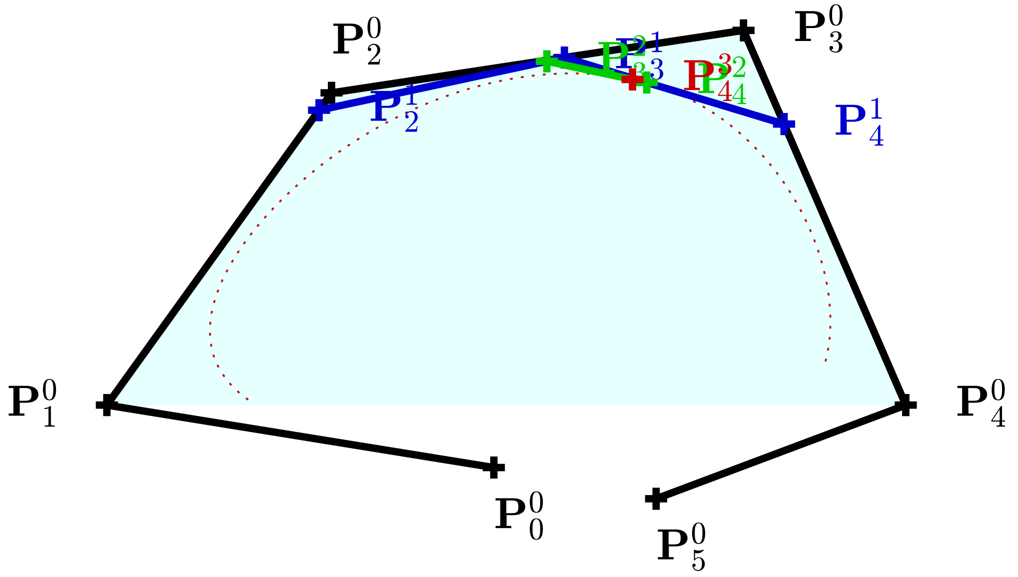
B-spline curves

de Boor – Cox algorithm

Extension of de Casteljau algorithm to B-splines

$\forall j \in [\![k, n]\!] \quad \forall t \in \left[\tau_j, \tau_{j+1}\right[$

$$\begin{aligned}
\boldsymbol{\mathcal{B}}_{\mathbb{P},\boldsymbol{\tau}}(t) &= \sum_{i=j-k}^{j} N_{i,k}^{\boldsymbol{\tau}}(t)\,\mathbf{p}_i^0 \\
&= \sum_{i=j-k+1}^{j} N_{i,k-1}^{\boldsymbol{\tau}}(t)\,\mathbf{p}_i^1 \\
&= \cdots \\
&= \sum_{i=j}^{j} N_{i,0}^{\boldsymbol{\tau}}(t)\,\mathbf{p}_i^k = \mathbf{p}_i^k
\end{aligned}$$



with $\mathbf{p}_i^0 = \mathbf{p}_i$ and $\mathbf{p}_i^j = \omega_{i,k-1}^{\boldsymbol{\tau}}(t)\,\mathbf{p}_i^{j-1} + \left(1 - \omega_{i,k-1}^{\boldsymbol{\tau}}(t)\right)\mathbf{p}_{i-1}^{j-1}$

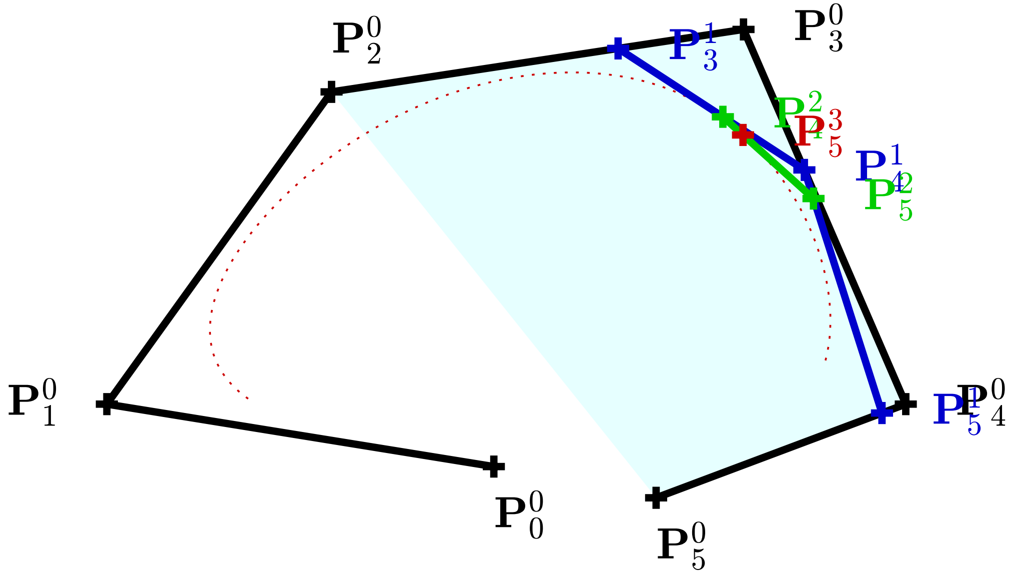
B-spline curves

de Boor – Cox algorithm

Extension of de Casteljau algorithm to B-splines

$$\forall j \in [\![k, n]\!] \quad \forall t \in \left[\tau_j, \tau_{j+1}\right[$$

$$\begin{aligned}
\boldsymbol{\mathcal{B}}_{\mathbb{P},\boldsymbol{\tau}}(t) &= \sum_{i=j-k}^{j} N_{i,k}^{\boldsymbol{\tau}}(t)\,\mathbf{p}_i^0 \\
&= \sum_{i=j-k+1}^{j} N_{i,k-1}^{\boldsymbol{\tau}}(t)\,\mathbf{p}_i^1 \\
&= \cdots \\
&= \sum_{i=j}^{j} N_{i,0}^{\boldsymbol{\tau}}(t)\,\mathbf{p}_i^k = \mathbf{p}_i^k
\end{aligned}$$



with $\mathbf{p}_i^0 = \mathbf{p}_i$ and $\mathbf{p}_i^j = \omega_{i,k-1}^{\boldsymbol{\tau}}(t)\,\mathbf{p}_i^{j-1} + \left(1 - \omega_{i,k-1}^{\boldsymbol{\tau}}(t)\right)\mathbf{p}_{i-1}^{j-1}$

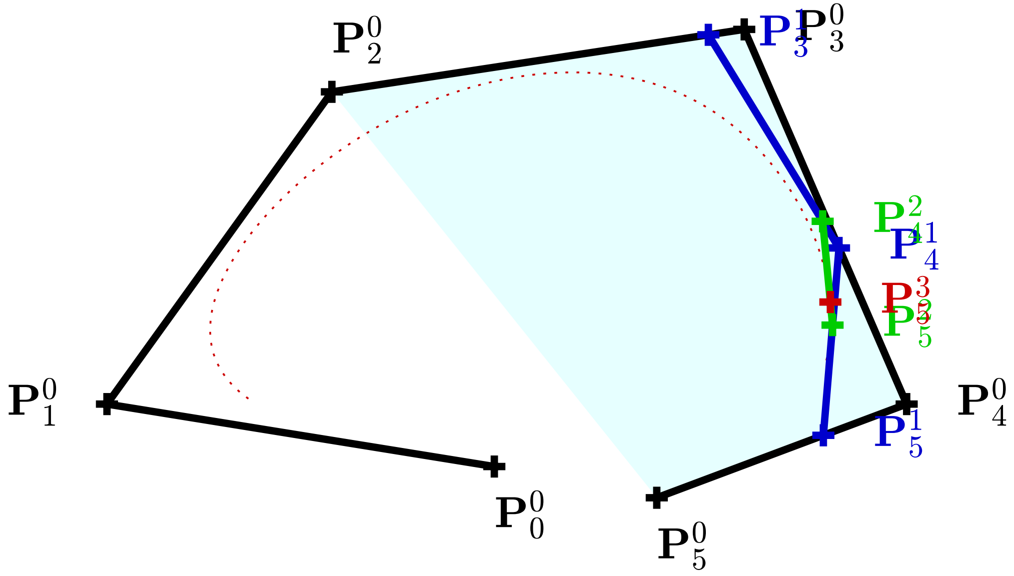
B-spline curves

de Boor – Cox algorithm

Extension of de Casteljau algorithm to B-splines

$\forall j \in [\![k,n]\!] \quad \forall t \in \left[\tau_j, \tau_{j+1}\right[$

$$
\begin{aligned}
\boldsymbol{\mathcal{B}}_{\mathbb{P},\boldsymbol{\tau}}(t) &= \sum_{i=j-k}^{j} N_{i,k}^{\boldsymbol{\tau}}(t)\,\mathbf{p}_i^0 \\
&= \sum_{i=j-k+1}^{j} N_{i,k-1}^{\boldsymbol{\tau}}(t)\,\mathbf{p}_i^1 \\
&= \cdots \\
&= \sum_{i=j}^{j} N_{i,0}^{\boldsymbol{\tau}}(t)\,\mathbf{p}_i^k = \mathbf{p}_i^k
\end{aligned}
$$



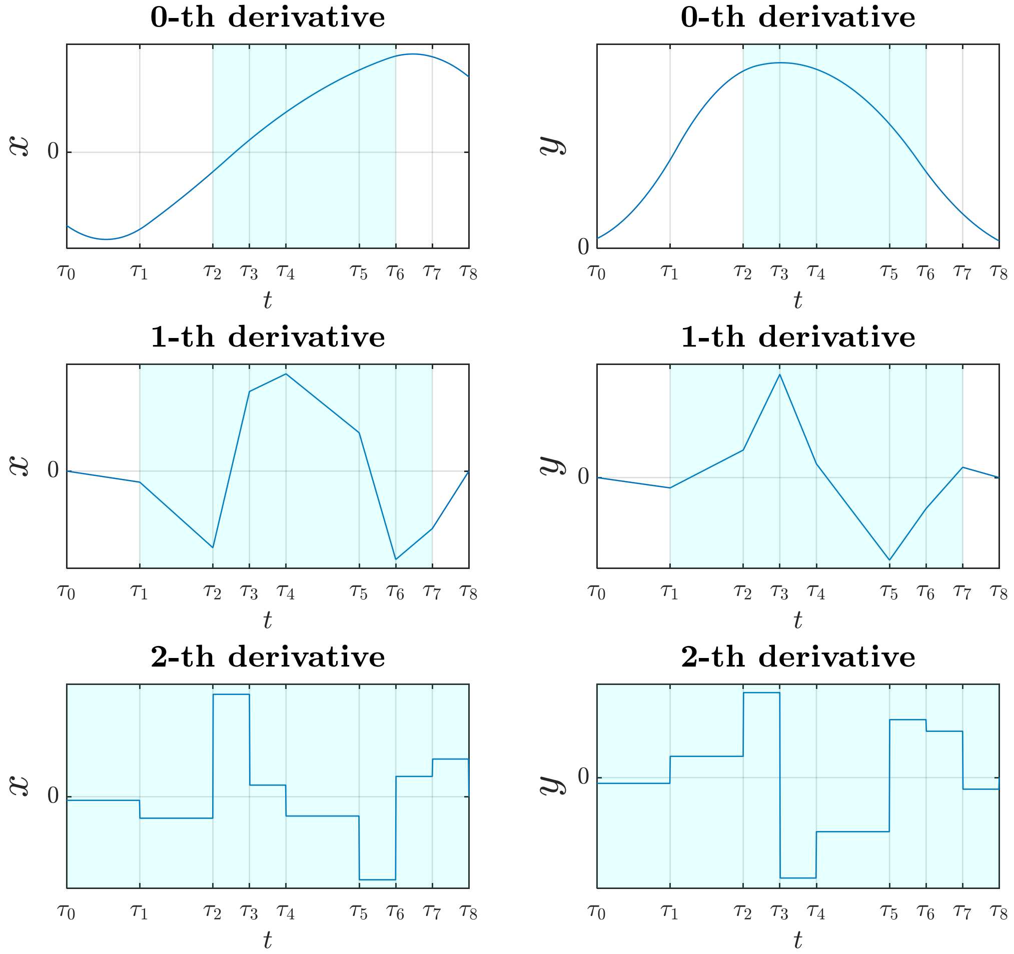
with $\mathbf{p}_i^0 = \mathbf{p}_i$ and $\mathbf{p}_i^j = \omega_{i,k-1}^{\boldsymbol{\tau}}(t)\,\mathbf{p}_i^{j-1} + \left(1 - \omega_{i,k-1}^{\boldsymbol{\tau}}(t)\right)\mathbf{p}_{i-1}^{j-1}$

B-spline curves

Derivatives

- Derivative is also a B-spline curve
- Degree k − 1
- Same knots
- $n + 1$ Control points given by linear combination

$$\mathcal{B}'_{\mathbb{P},\boldsymbol{\tau}} = \mathcal{B}_{\mathbb{P}^{(1)},\boldsymbol{\tau}} \quad \text{with} \quad \begin{cases} \mathbf{p}_0^{(1)} = \dfrac{k}{\tau_k - \tau_0}\mathbf{p}_0 \\ \forall i \in [\![1, n]\!] \quad \mathbf{p}_i^{(1)} = \dfrac{k}{\tau_{i+k} - \tau_i}(\mathbf{p}_i - \mathbf{p}_{i-1}) \\ \mathbf{p}_{n+1}^{(1)} = \dfrac{-k}{\tau_{n+k+1} - \tau_{n+1}}\mathbf{p}_n \end{cases}$$



B-spline curves

Knots multiplicity and continuity

Multiplicity

The number of repetition of a knot τ_i in the knot vector is called its **multiplicity** $\mu_i \in \mathbb{N}^*$

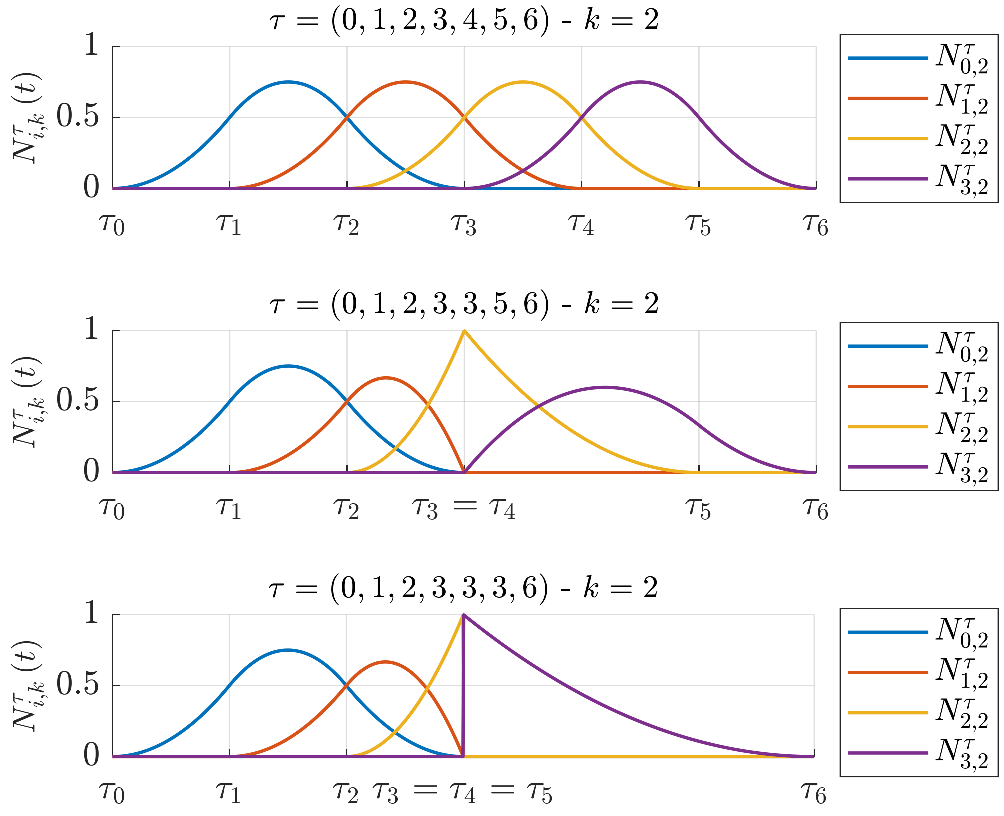
Continuity

For $i \in [\![0, m-k-1]\!]$ and $j \in [\![i, i+k]\!]$

$N_{i,k}^{\boldsymbol{\tau}}$ is $\mathcal{C}^{\infty}$ in $]\tau_j, \tau_{j+1}[$

$N_{i,k}^{\boldsymbol{\tau}}$ is $\mathcal{C}^{k-\mu_j}$ in a neighborhood of τ_j

$N_{i,k}^{\boldsymbol{\tau}}$ is $\mathcal{C}^{k-\mu_{j+1}}$ in a neighborhood of τ_{j+1}



B-spline curves

Clamped B-spline curve

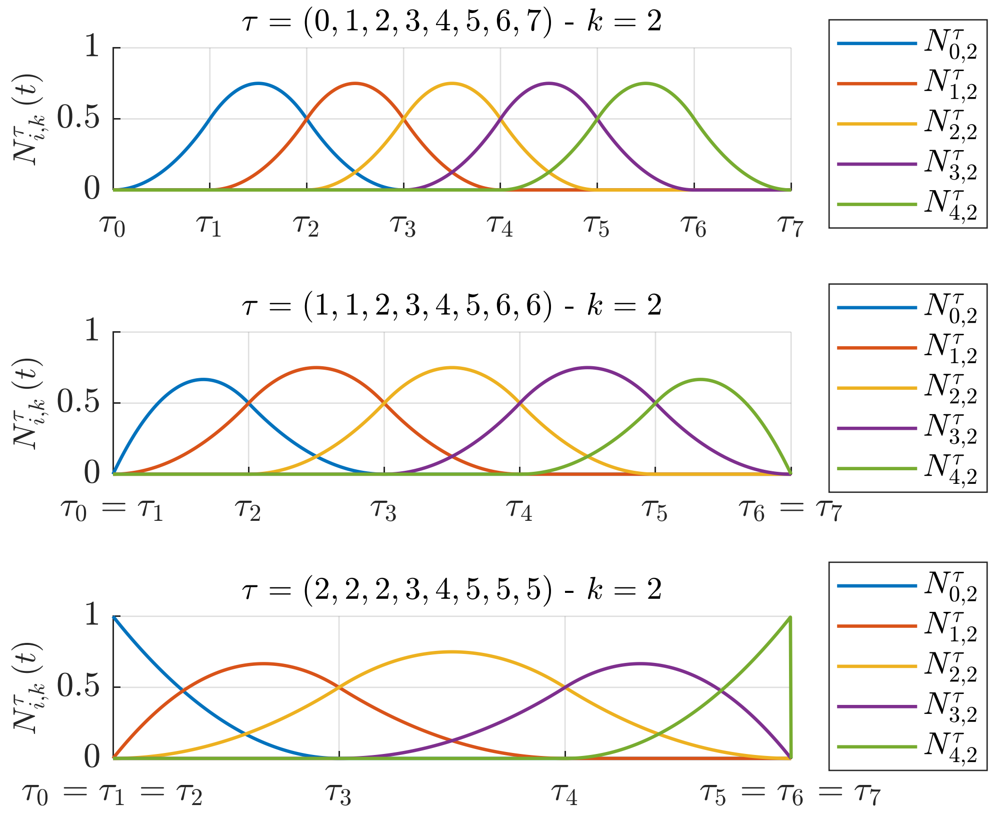
If

- first $k+1$ knots are equal
- last $k+1$ knots are equal

➡ B-spline curve "**clamped**" to first and last control points

$$(\underbrace{\tau_k, \dots, \tau_k}_{k+1 \text{ knots}}, \underbrace{\tau_{k+1}, \dots, \tau_n}_{n-k \text{ knots}}, \underbrace{\tau_{n+1}, \dots, \tau_{n+1}}_{k+1 \text{ knots}})$$

A B-spline can be converted into a clamped one by knot insertion (Boehm's algorithm)



B-spline curves

Clamped B-spline curve - Definition

If

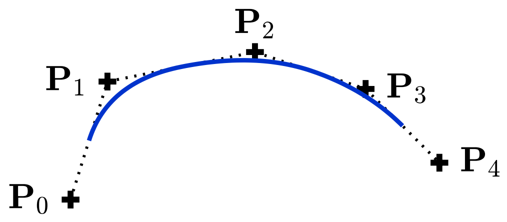
- first $k + 1$ knots are equal
- last $k + 1$ knots are equal

➡ B-spline curve "**clamped**" to first and last control points

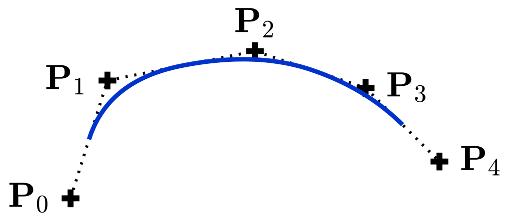
$$(\underbrace{\tau_k, \dots, \tau_k}_{k+1 \text{ knots}}, \underbrace{\tau_{k+1}, \dots, \tau_n}_{n-k \text{ knots}}, \underbrace{\tau_{n+1}, \dots, \tau_{n+1}}_{k+1 \text{ knots}})$$

A B-spline can be converted into a clamped one by knot insertion (Boehm's algorithm)

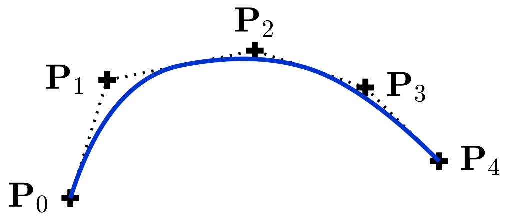
$\tau = (0, 1, 2, 3, 4, 5, 6, 7)$ - $k = 2$



$\tau = (1, 1, 2, 3, 4, 5, 6, 6)$ - $k = 2$



$\tau = (2, 2, 2, 3, 4, 5, 5, 5)$ - $k = 2$



B-spline curves

Clamped B-spline curve - Derivatives

- Derivative is also a B-spline curve
- Degree $\mathrm{k}-1$
- Same knots
- $n+1$ Control points given by linear combination

$$(\underbrace{\tau_k, \dots, \tau_k}_{k+1 \text{ knots}}, \underbrace{\tau_{k+1}, \dots, \tau_n}_{n-k \text{ knots}}, \underbrace{\tau_{n+1}, \dots, \tau_{n+1}}_{k+1 \text{ knots}})$$

$$\boldsymbol{\mathcal{B}}'_{\mathbb{P},\boldsymbol{\tau}} = \boldsymbol{\mathcal{B}}_{\mathbb{P}^{(1)},\boldsymbol{\tau}} \quad \text{with} \quad \begin{cases} \mathbf{p}_0^{(1)} = \dfrac{k}{\tau_k - \tau_0}\mathbf{p}_0 \\ \forall i \in [\![1,n]\!] \quad \mathbf{p}_i^{(1)} = \dfrac{k}{\tau_{i+k} - \tau_i}(\mathbf{p}_i - \mathbf{p}_{i-1}) \\ \mathbf{p}_{n+1}^{(1)} = \dfrac{-k}{\tau_{n+k+1} - \tau_{n+1}}\mathbf{p}_n \end{cases}$$

B-spline curves

Clamped B-spline curve - Derivatives

- Derivative is also a clamped B-spline curve
- Degree $\mathrm{k}-1$
- Multiplicity of first and last knots decremented by 1
- n Control points given by linear combination

$$\boldsymbol{\mathcal{B}}'_{\mathbb{P},\boldsymbol{\tau}} = \boldsymbol{\mathcal{B}}_{\mathbb{P}^{(1)},\boldsymbol{\tau}^{(1)}} \quad \text{with} \quad \begin{cases} \forall i \in [\![0, n-1]\!] \quad \mathbf{p}_i^{(1)} = \dfrac{k}{\tau_{i+k+1}-\tau_{i+1}}(\mathbf{p}_{i+1}-\mathbf{p}_i) \\ \boldsymbol{\tau}^{(1)} = (\underbrace{\tau_k, \dots, \tau_k}_{k \text{ knots}}, \underbrace{\tau_{k+1}, \dots, \tau_n}_{n-k \text{ knots}}, \underbrace{\tau_{n+1}, \dots, \tau_{n+1}}_{k \text{ knots}}) \end{cases}$$

Uniform B-spline curve

A B-spline is said **uniform** when its knots are equally distributed

Knot vector replaced by 2 parameter

- the step $\Delta\tau$ between 2 knots
- First knot τ_0

$$\boldsymbol{\tau} = (\tau_0, \tau_0 + \Delta\tau, \dots, \tau_0 + m\,\Delta\tau)$$

Only1 knot parameter if parameter if $\tau_0 = 0$

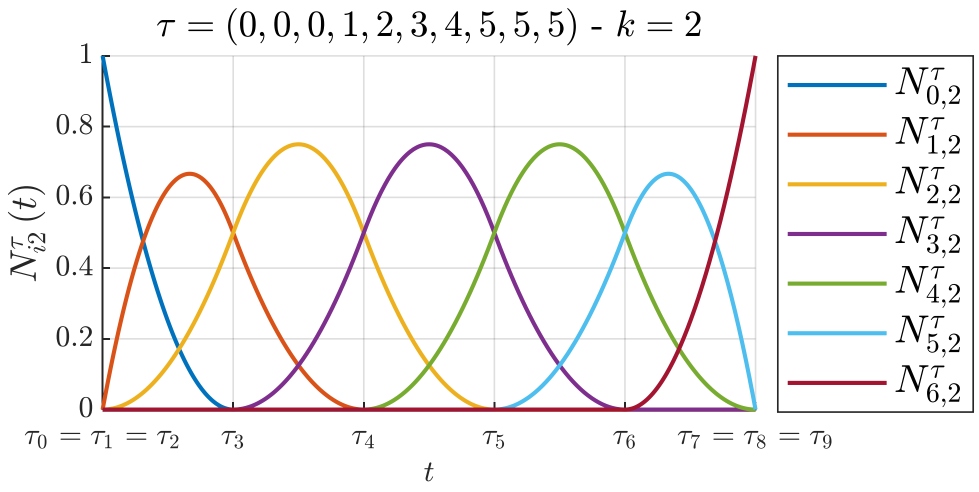
$$\boldsymbol{\tau} = (0, \Delta\tau, \dots, m\,\Delta\tau)$$

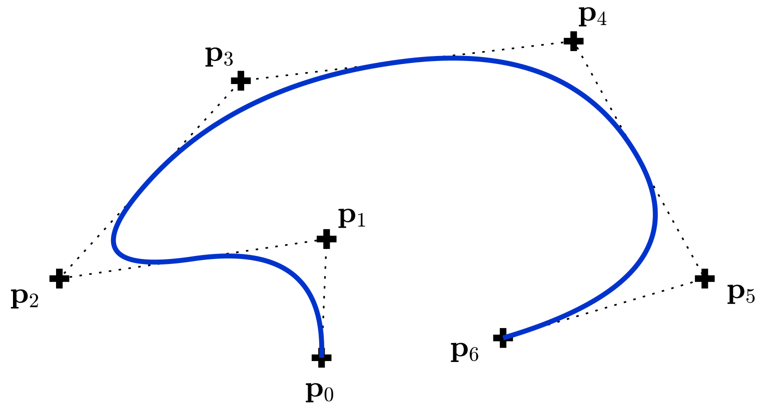
B-spline curves

Uniform clamped B-spline curve

Uniform + clamped = **uniform clamped B-spline curve**

$$\underbrace{(0,\ \dots,\ 0,}_{k+1 \text{ knots}}\ \underbrace{\Delta\tau,\ \dots,\ (n-k)\Delta\tau,}_{n-k \text{ knots}}\ \underbrace{(n-k+1)\Delta\tau,\ \dots,\ (n-k+1)\Delta\tau)}_{k+1 \text{ knots}}$$





B-spline curves

Uniform clamped B-spline curve - Derivatives

- Derivative is also a uniform clamped B-spline curve
- Degree $\mathrm{k}-1$
- Multiplicity of first and last knots decremented by 1
- n Control points given by linear combination

$$\begin{cases} \forall i \in [\![0, n-1]\!] \quad \mathbf{p}_i^{(1)} = \dfrac{k}{\tau_{i+k+1} - \tau_{i+1}} (\mathbf{p}_{i+1} - \mathbf{p}_i) \\ \boldsymbol{\tau}^{(1)} = (\underbrace{\tau_k, \dots, \tau_k}_{k \text{ knots}}, \underbrace{\tau_{k+1}, \dots, \tau_n}_{n-k \text{ knots}}, \underbrace{\tau_{n+1}, \dots, \tau_{n+1}}_{k \text{ knots}}) \end{cases}$$

$$\mathbf{p}_0^{(1)} = \frac{k}{\Delta\tau}(\mathbf{p}_1 - \mathbf{p}_0)$$

$$\mathbf{p}_1^{(1)} = \frac{k}{2\,\Delta\tau}(\mathbf{p}_2 - \mathbf{p}_1)$$

$$\vdots$$

$$\mathbf{p}_{k-2}^{(1)} = \frac{k}{(k-1)\Delta\tau}(\mathbf{p}_k - \mathbf{p}_{k-1})$$

$$\mathbf{p}_{k-1}^{(1)} = \frac{1}{\Delta\tau}(\mathbf{p}_k - \mathbf{p}_{k-1})$$

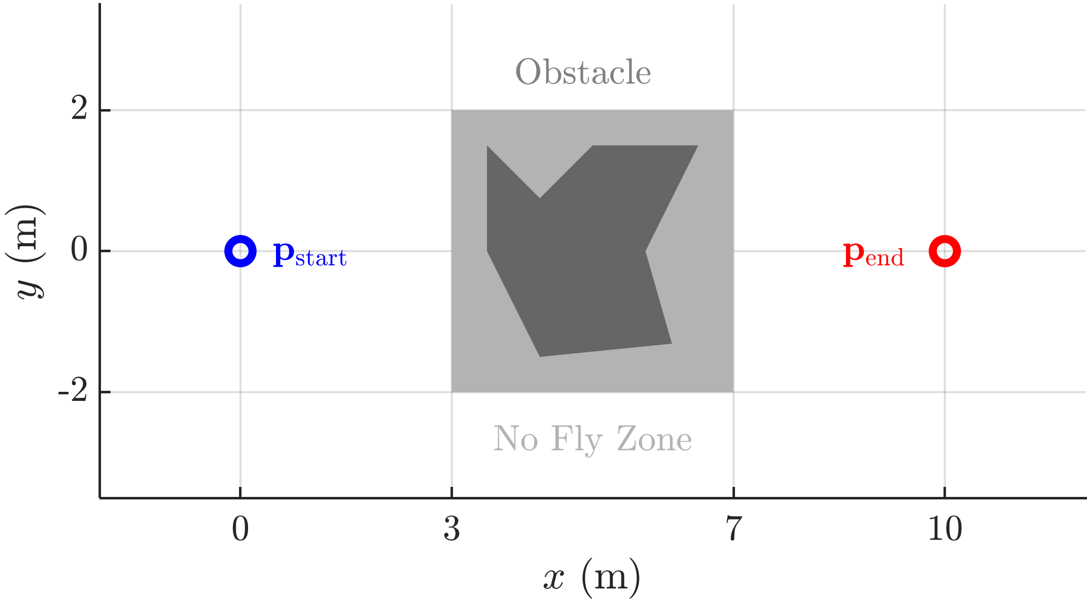
$$\mathbf{p}_k^{(1)} = \frac{1}{\Delta\tau}(\mathbf{p}_{k+1} - \mathbf{p}_k)$$

$$\vdots$$

Case study 1

Mission details

Obstacle bounded in a **convex no fly zone** (NFZ), with security margins



Case study 1

Strategy

B-spline trajectory generation in **2 steps**

- **Control points.** Choose the control points such that the **path** is smooth and collision-free, using the convex hull property
- **Knot vector.** Choose the duration of the **trajectory** so that it is feasible, by applying the convex hull property on the control points of its derivatives

Use **uniform clamped** B-spline curves as they are easy to work with

Path generation

Strategy

One solution:

Shortest (in terms of length), **collision-free** B-spline curve joining the starting and the ending positions

3 parameters:

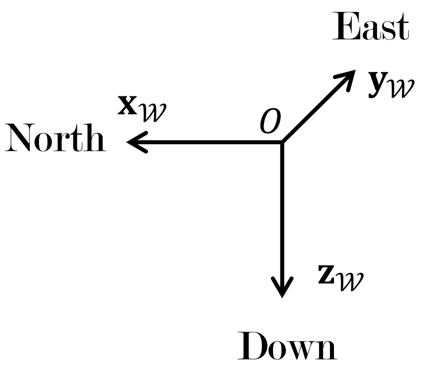
- Degree k ➡ differentiability class
- Control points $\mathbb{P}$ ➡ collision-free curve
- Knots $\boldsymbol{\tau}$ ➡ uniform clamped B-spline

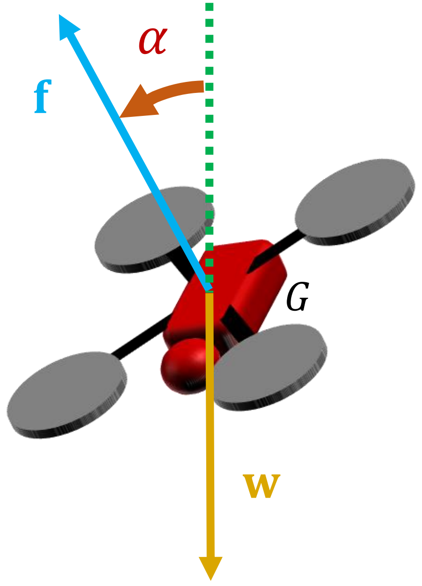
Path generation

Continuity

Simple quadrotor model

$$\begin{cases} \ddot{\zeta}_z = g - \dfrac{f}{m}\cos(\alpha) \\ \ddot{\zeta}_{xy} = \dfrac{f}{m}|\sin(\alpha)| \end{cases}$$





Lateral motion

$$\ddot{\zeta}_z = 0 \Rightarrow f = \frac{mg}{\cos(\alpha)} \Rightarrow \ddot{\zeta}_{xy} = g\tan(\alpha)$$

Rotation speed limited ➡ drone attitude continuous

$$\alpha = \arctan\left(\frac{\ddot{\zeta}_{xy}}{g}\right)$$

If the trajectory $\boldsymbol{\zeta}$ is $\mathcal{C}^2$ then α is continuous

Uniform B-spline ➡ multiplicity = 1 for all knots ➡ $\mathcal{C}^{k-1}$ curve

$$k = 3$$

Path generation

Continuity

- Clamped B-spline ➡ starts on first control point, ends on last control point
- Derivative of a clamped B-spline = clamped B-spline

$\mathcal{C}^2$ rest-to-rest trajectory

$$\mathbf{p}_i^{(1)} = \frac{k}{\tau_{i+k+1} - \tau_{i+1}} (\mathbf{p}_{i+1} - \mathbf{p}_i)$$

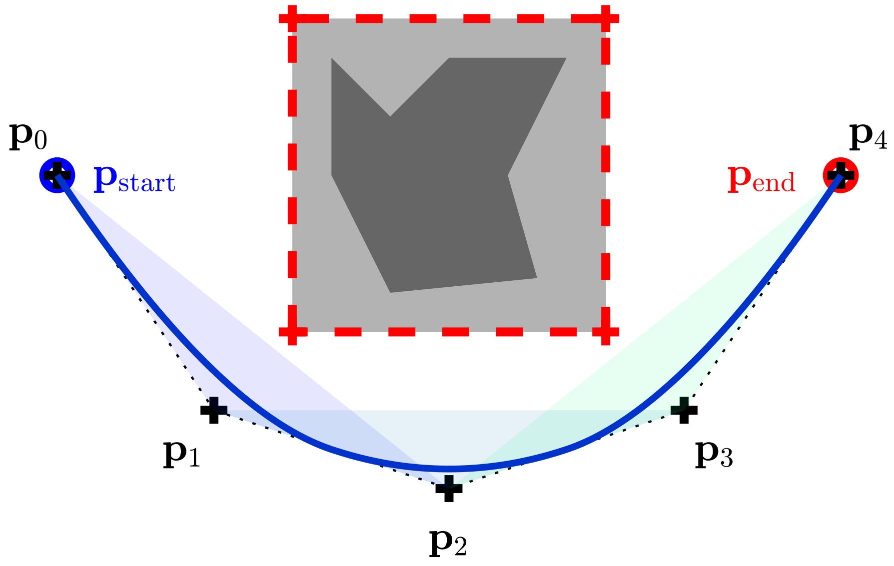
$$\begin{cases} \mathbf{p}_0 = \mathbf{p}_{\text{start}} \\ \mathbf{p}_n = \mathbf{p}_{\text{end}} \\ \mathbf{p}_0^{(1)} = \mathbf{p}_0^{(2)} = \mathbf{p}_n^{(1)} = \mathbf{p}_{n-2}^{(2)} = \mathbf{0} \end{cases} \Rightarrow \begin{cases} \mathbf{p}_0 = \mathbf{p}_1 = \mathbf{p}_2 = \mathbf{p}_{\text{start}} \\ \mathbf{p}_n = \mathbf{p}_{n-1} = \mathbf{p}_{n-2} = \mathbf{p}_{\text{end}} \end{cases}$$

Path generation

Obstacle management

Use convex hull property to garantee the absence of collision

➡ Forbid the convex hulls to contain any vertices of the convex NFZ

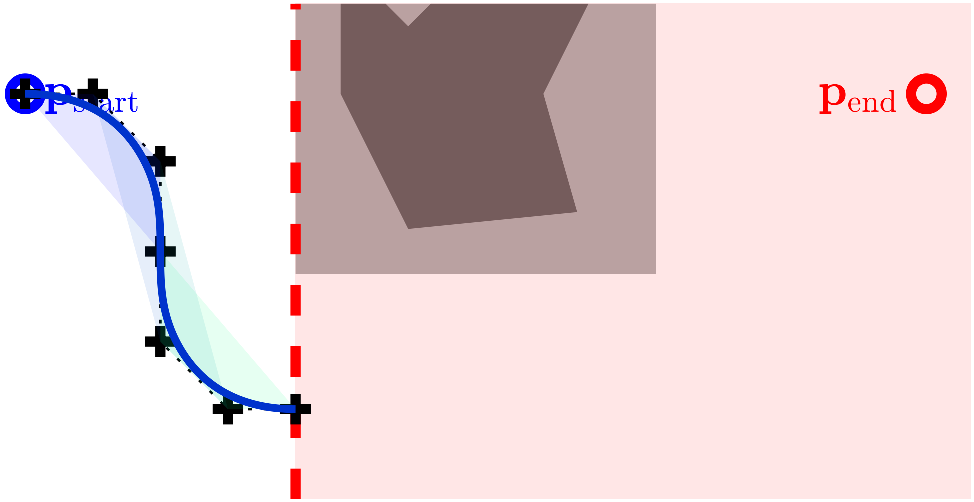


Path generation

Obstacle management

This constraint can be hard to check

➡ simpler formulation with convex obstacle-free regions (more conservative)



Region 1

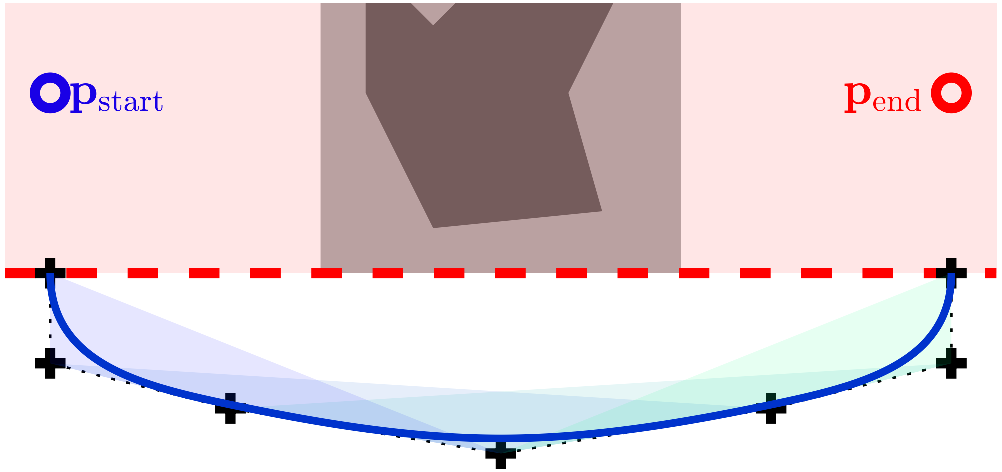
$$p_i^x \leq 3$$

Path generation

Obstacle management

This constraint can be hard to check

➡ simpler formulation with convex obstacle-free regions (more conservative)



Region 2

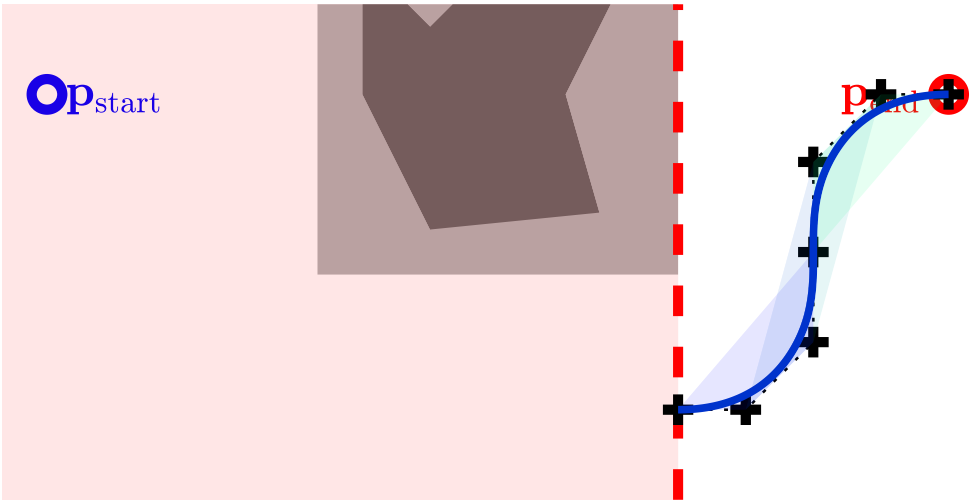
$$p_i^y \leq -2$$

Path generation

Obstacle management

This constraint can be hard to check

➡ simpler formulation with convex obstacle-free regions (more conservative)



Region 3

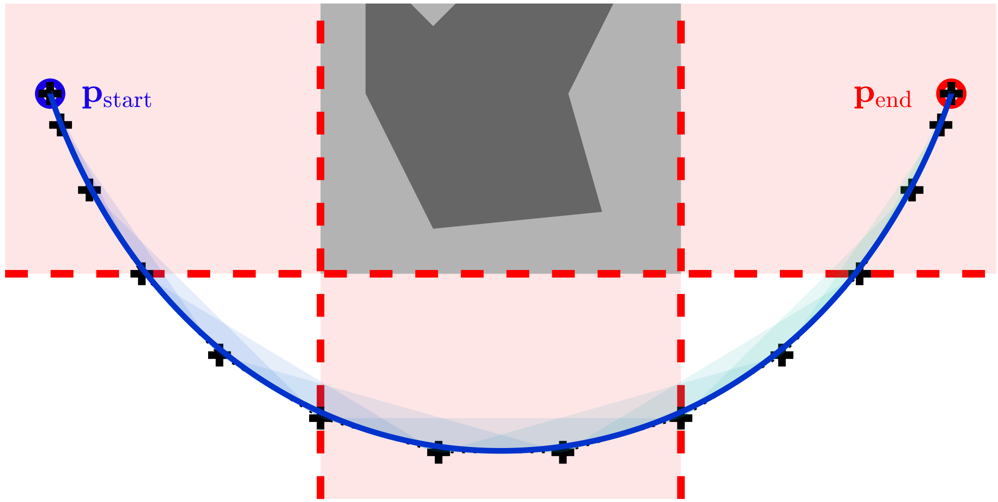
$$-p_i^x \leq -7$$

Path generation

Obstacle management

Some control points are in the convex hulls of other control points in different convex, obstacle-free regions

➡ these points are constrained in both regions



Region 1 & 2

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} p_i^x \\ p_i^y \end{pmatrix} \leq \begin{pmatrix} 3 \\ -2 \end{pmatrix}$$

Region 2 & 3

$$\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} p_i^x \\ p_i^y \end{pmatrix} \leq \begin{pmatrix} -2 \\ -7 \end{pmatrix}$$

Path generation

Control points

- 3 control points on start position
- n_1 control points in region 1
- $k = 3$ control points in region 1 & 2
- n_2 control points in region 2
- $k = 3$ control points in region 2 & 3
- n_2 control points in region 3
- 3 control points on end position

$$n + 1 = 6 + 2k + n_1 + n_2 + n_3$$

For $n_1 = n_2 = n_3 = 0$ $\qquad n = 11$

Path generation

Knots

Uniform clamped B-spline

$$\boldsymbol{\tau} = (\underbrace{0, \dots, 0}_{k+1 \text{ knots}}, \underbrace{\Delta\tau, \dots, (n-k)\Delta\tau}_{n-k \text{ knots}}, \underbrace{(n-k+1)\Delta\tau, \dots, (n-k+1)\Delta\tau}_{k+1 \text{ knots}})$$

$$k = 3, n = 11$$

Only looking for a **path** ➡ arbitrary $\Delta\tau = 1$

$$\boldsymbol{\tau} = (0,0,0,0,1,2,3,4,5,6,7,8,9,9,9,9)$$

Path generation

Parameters and constraints

For a $\mathcal{C}^L$ path

Parameters

- $k = L + 1$
- $n = 2(L+1) + 2k + n_1 + n_2 + n_3 - 1$
- $\boldsymbol{\tau} = (\underbrace{0, \dots, 0}_{k+1 \text{ knots}}, \underbrace{\Delta\tau, \dots, (n-k)\Delta\tau}_{n-k \text{ knots}}, \underbrace{(n-k+1)\Delta\tau, \dots, (n-k+1)\Delta\tau}_{k+1 \text{ knots}})$

Constraints

- $\forall i \in [\![0, L]\!] \quad \mathbf{p}_i = \mathbf{p}_{\text{start}}$
- $\forall i \in [\![0, L]\!] \quad \mathbf{p}_{n-i} = \mathbf{p}_{\text{end}}$
- $\forall i \in [\![L+1, L+n_1+k]\!] \quad p_i^x \leq 3$
- $\forall i \in [\![L+n_1+1, L+n_1+n_2+2k]\!] \quad p_i^y \leq -2$
- $\forall i \in [\![L+n_1+n_2+k+1, n-L-1]\!] \quad -p_i^x \leq 7$

Path generation

Parameters and constraints

For a $\mathcal{C}^2$ path with $n_1 = n_2 = n_3 = 0$

Parameters

- $k = 3$
- $n = 11$
- $\boldsymbol{\tau} = (0,0,0,0,1,2,3,4,5,6,7,8,9,9,9,9)$

12 free parameters

Constraints

- $\mathbf{p}_0 = \mathbf{p}_1 = \mathbf{p}_2 = \mathbf{p}_{\text{start}}$
- $\mathbf{p}_{11} = \mathbf{p}_{10} = \mathbf{p}_9 = \mathbf{p}_{\text{end}}$
- $\forall i \in [\![3,5]\!] \quad p_i^x \leq 3$
- $\forall i \in [\![3,8]\!] \quad p_i^y \leq -2$
- $\forall i \in [\![6,8]\!] \quad -p_i^x \leq 7$

Path generation

Criterion

Infinity of path verifying the constraints

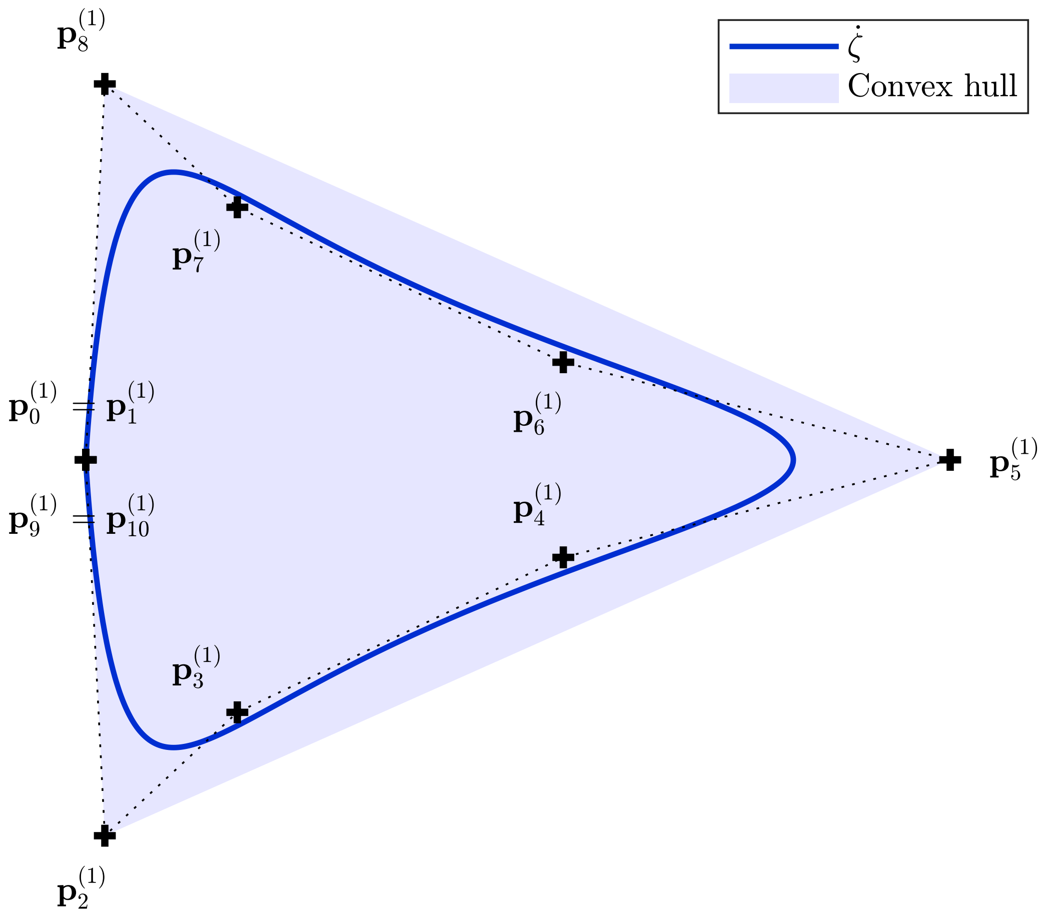
- Choose the best path according to a **criterion**
- Shortest path = **minimal length**

Length $\mathcal{L}$ of the path $\boldsymbol{\zeta}$

$$\mathcal{L} = \int_{\tau_k}^{\tau_{n+1}} \|\dot{\boldsymbol{\zeta}}(u)\| \mathrm{d}u$$

$\dot{\boldsymbol{\zeta}} = \mathcal{B}'_{\mathbb{P},\boldsymbol{\tau}}$ has convex hull property

$$J_1 = \sum_{i=0}^{n-1} \left\|\mathbf{p}_i^{(1)}\right\|^2$$



Path generation

Optimization problem

Control points obtained by solving an optimization problem

$$\mathbb{P}^* = \arg \min_{\mathbb{P} \in (\mathbb{R}^2)^{12}} \sum_{i=0}^{n-1} \left\| \mathbf{p}_i^{(1)} \right\|^2$$

$$\text{s.t.} \begin{cases} \mathbf{p}_0 = \mathbf{p}_1 = \mathbf{p}_2 = \mathbf{p}_{\text{start}} \\ \mathbf{p}_{11} = \mathbf{p}_{10} = \mathbf{p}_9 = \mathbf{p}_{\text{end}} \\ \forall i \in [\![3,5]\!] \quad p_i^x \leq 3 \\ \forall i \in [\![3,8]\!] \quad p_i^y \leq -2 \\ \forall i \in [\![6,8]\!] \quad -p_i^x \leq 7 \end{cases}$$

Path generation

Optimization problem

Optimization vector

$$\mathbf{x} = \begin{pmatrix} p_0^x & p_1^x & \dots & p_n^x & p_0^y & p_1^y & \dots & p_n^y \end{pmatrix}^\intercal$$

J_1 is a **quadratic cost**

$$J_1 = \sum_{i=0}^{n-1} \left\| \mathbf{p}_i^{(1)} \right\|^2$$

$$J_1 = \mathbf{x}^\intercal \mathbf{H}_1 \mathbf{x}$$

$$\mathbf{H}_1 = \begin{pmatrix} \frac{1}{\Delta\tau}\mathbf{Q}_1 & \\ & \frac{1}{\Delta\tau}\mathbf{Q}_1 \end{pmatrix}^\intercal \begin{pmatrix} \frac{1}{\Delta\tau}\mathbf{Q}_1 & \\ & \frac{1}{\Delta\tau}\mathbf{Q}_1 \end{pmatrix}$$

$$\mathbf{Q}_1 = \begin{pmatrix} -3 & 3 & & & & & & & & & & \\ & -1.5 & 1.5 & & & & & & & & & \\ & & -1 & 1 & & & & & & & & \\ & & & -1 & 1 & & & & & & & \\ & & & & -1 & 1 & & & & & & \\ & & & & & -1 & 1 & & & & & \\ & & & & & & -1 & 1 & & & & \\ & & & & & & & -1 & 1 & & & \\ & & & & & & & & -1 & 1 & & \\ & & & & & & & & & -1.5 & 1.5 & \\ & & & & & & & & & & -3 & 3 \end{pmatrix}$$

Path generation

Optimization problem

Optimization vector

$$\mathbf{x} = \begin{pmatrix} p_0^x & p_1^x & \dots & p_n^x & p_0^y & p_1^y & \dots & p_n^y \end{pmatrix}^\intercal$$

$$\mathbf{C}_{\mathrm{eq}} = \begin{pmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & 1 & & & & \\ & & & & 1 & & \\ & & & & & 1 & \\ & & & & & & 1 \end{pmatrix}$$

Linear equality constraints

- $\mathbf{p}_0 = \mathbf{p}_1 = \mathbf{p}_2 = \mathbf{p}_{\mathrm{start}}$
- $\mathbf{p}_{11} = \mathbf{p}_{10} = \mathbf{p}_9 = \mathbf{p}_{\mathrm{end}}$

$$\mathbf{A}_{\mathrm{eq}} = \begin{pmatrix} \mathbf{C}_{\mathrm{eq}} & \\ & \mathbf{C}_{\mathrm{eq}} \end{pmatrix}$$

$$\mathbf{b}_{\mathrm{eq}}^x = \begin{pmatrix} p_{\mathrm{start}}^x & p_{\mathrm{start}}^x & p_{\mathrm{start}}^x & p_{\mathrm{end}}^x & p_{\mathrm{end}}^x & p_{\mathrm{end}}^x \end{pmatrix}^\intercal$$

$$\mathbf{b}_{\mathrm{eq}}^y = \begin{pmatrix} p_{\mathrm{start}}^y & p_{\mathrm{start}}^y & p_{\mathrm{start}}^y & p_{\mathrm{end}}^y & p_{\mathrm{end}}^y & p_{\mathrm{end}}^y \end{pmatrix}^\intercal$$

$$\mathbf{A}_{\mathrm{eq}}\,\mathbf{x} = \mathbf{b}_{\mathrm{eq}}$$

$$\mathbf{b}_{\mathrm{eq}} = \begin{pmatrix} \mathbf{b}_{\mathrm{eq}}^x \\ \mathbf{b}_{\mathrm{eq}}^y \end{pmatrix}$$

Path generation

Optimization problem

Optimization vector

$$\mathbf{x} = \begin{pmatrix} p_0^x & p_1^x & \dots & p_n^x & p_0^y & p_1^y & \dots & p_n^y \end{pmatrix}^\intercal$$

Linear inequality constraints

- $\forall i \in [\![3,5]\!] \quad p_i^x \leq 3$
- $\forall i \in [\![3,8]\!] \quad p_i^y \leq -2$
- $\forall i \in [\![6,8]\!] \quad -p_i^x \leq 7$

$$\mathbf{A}_{\text{ineq}}\,\mathbf{x} \leq \mathbf{b}_{\text{ineq}}$$

Similar method as for $\mathbf{A}_{\text{eq}}$ and $\mathbf{b}_{\text{eq}}$

Path generation

Optimization problem

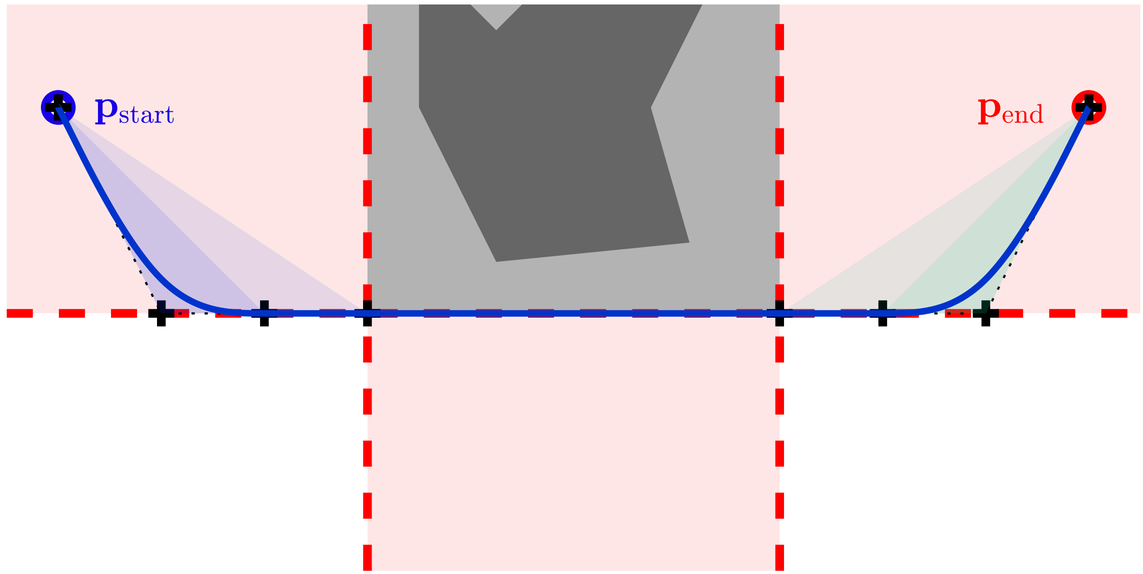
Small, convex QP problem

$$\mathbf{x}^* = \arg\min_{\mathbb{P}\in\mathbb{R}^{24}} \mathbf{x}^\intercal \mathbf{H}_1 \mathbf{x}$$

$$\text{s.t.} \begin{cases} \mathbf{A}_{\text{eq}}\,\mathbf{x} = \mathbf{b}_{\text{eq}} \\ \mathbf{A}_{\text{ineq}}\,\mathbf{x} \leq \mathbf{b}_{\text{ineq}} \end{cases}$$

➡ Fast and easy to solve

Collision free path!



Simplified formulation ➡ sub-optimal path

Path generation

Smooth path

Penalty on the jerk to smooth the path

$$\mathbf{H}_3 = \begin{pmatrix} \frac{1}{\Delta\tau}\mathbf{Q}_3 & \\ & \frac{1}{\Delta\tau}\mathbf{Q}_3 \end{pmatrix}^\intercal \begin{pmatrix} \frac{1}{\Delta\tau}\mathbf{Q}_3 & \\ & \frac{1}{\Delta\tau}\mathbf{Q}_3 \end{pmatrix}$$

$$J_3 = \sum_{i=0}^{n-1} \left\| \mathbf{p}_i^{(3)} \right\|^2$$

$$J_3 = \mathbf{x}^\intercal \mathbf{H}_3 \mathbf{x}$$

$$\mathbf{Q}_1 = \begin{pmatrix} -6 & 10.5 & -5.5 & 1 & & & & & & & & \\ & -1.5 & 3.5 & -3 & 1 & & & & & & & \\ & & -1 & 3 & -3 & 1 & & & & & & \\ & & & -1 & 3 & -3 & 1 & & & & & \\ & & & & -1 & 3 & -3 & 1 & & & & \\ & & & & & -1 & 3 & -3 & 1 & & & \\ & & & & & & -1 & 3 & -3 & 1 & & \\ & & & & & & & -1 & 3 & -3.5 & 1.5 & \\ & & & & & & & & -1 & 5.5 & -10.5 & 6 \end{pmatrix}$$

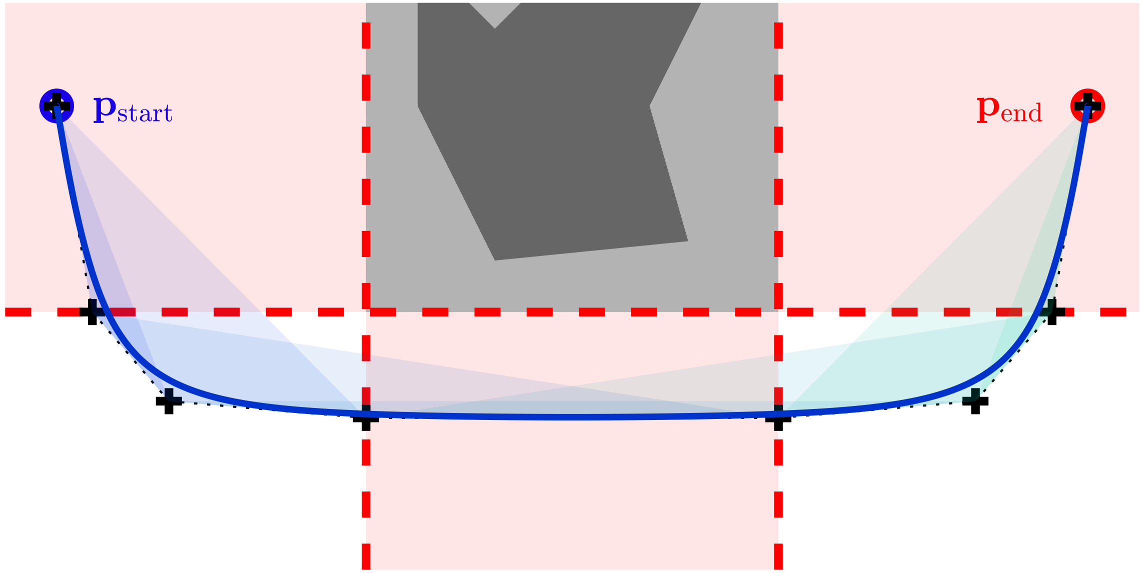
Path generation

Smooth path

Compromise length vs. smoothness $J = \mathbf{x}^\intercal \mathbf{H}\mathbf{x} = \sigma\, \mathbf{x}^\intercal \mathbf{H}_1 \mathbf{x} + (1-\sigma)\, \mathbf{x}^\intercal \mathbf{H}_3 \mathbf{x}$

$$\mathbf{x}^* = \arg\min_{\mathbb{P}\in\mathbb{R}^{24}} \mathbf{x}^\intercal \mathbf{H}\mathbf{x}$$

$$\text{s.t.} \begin{cases} \mathbf{A}_{\text{eq}}\,\mathbf{x} = \mathbf{b}_{\text{eq}} \\ \mathbf{A}_{\text{ineq}}\,\mathbf{x} \le \mathbf{b}_{\text{ineq}} \end{cases}$$



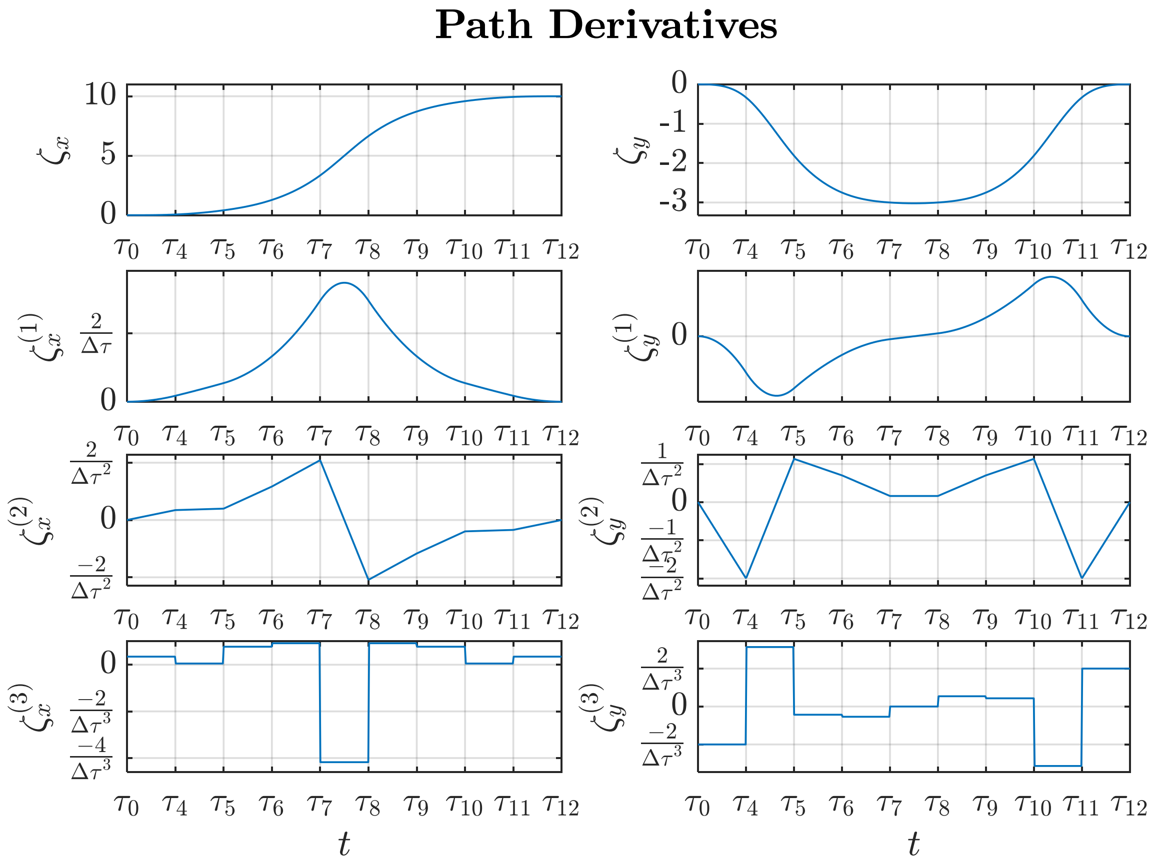
$\sigma = 0.8$

Trajectory duration

From path to trajectory

Path + duration = trajectory

➡ **Choice of $\Delta\tau$**



Trajectory duration

Strategy

Specification

$$v \leq v_{\max}$$

Translate into constraints on the derivatives

Feasibility

➡ use convex hull property

$$\alpha \leq \alpha_{\max}$$

$$\Omega \leq \Omega_{\max}$$

Trajectory duration

Dynamics constraints

Acceleration

$$a = g\ \tan(\alpha) \Rightarrow a \leq g\ \tan(\alpha_{\max}) \Rightarrow \alpha \leq \alpha_{\max}$$

Jerk

$$\begin{cases} \ddot{\zeta} = -\dfrac{f}{m}\mathbf{R}\,\mathbf{z}_W + g\,\mathbf{z}_W \\ \dot{\mathbf{R}} = \mathbf{R}\widehat{\mathbf{\Omega}} \end{cases} \xrightarrow{\frac{\mathrm{d}}{\mathrm{dt}}} \boldsymbol{\zeta}^{(3)} = -\frac{\dot{f}}{m}\mathbf{R}\,\mathbf{z}_W - \frac{f}{m}\mathbf{R}\widehat{\mathbf{\Omega}}\,\mathbf{z}_W$$

$$\mathbf{R}^{\intercal}\boldsymbol{\zeta}^{(3)} = -\frac{\dot{f}}{m}\,\mathbf{z}_W - \frac{f}{m}\widehat{\mathbf{\Omega}}\,\mathbf{z}_W$$

$$\mathbf{R}^{\intercal}\boldsymbol{\zeta}^{(3)} = -\frac{\dot{f}}{m}\,\mathbf{z}_W - \frac{f}{m}\widehat{\mathbf{\Omega}}\,\mathbf{z}_W$$

$$\Rightarrow \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}\mathbf{R}^{\intercal}\boldsymbol{\zeta}^{(3)} = -\frac{f}{m}\begin{pmatrix} \Omega_y \\ -\Omega_x \end{pmatrix}$$

$$+$$

$$f = \frac{mg}{\cos(\alpha)}$$

$$\Downarrow$$

$$\begin{pmatrix} \Omega_y \\ \Omega_x \end{pmatrix} = \frac{\cos(\alpha)}{g}\begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}\mathbf{R}^{\intercal}\boldsymbol{\zeta}^{(3)}$$

$$j \leq g\Omega_{\max} \Rightarrow \Omega \leq \Omega_{\max}$$

Trajectory duration

Dynamics constraints

Acceleration

$$a = g\ \tan(\alpha) \Rightarrow a \leq g\ \tan(\alpha_{\max}) \Rightarrow \alpha \leq \alpha_{\max}$$

Jerk

$$\begin{cases} \ddot{\zeta} = -\dfrac{f}{m}\mathbf{R}\,\mathbf{z}_{\mathcal{W}} + g\,\mathbf{z}_{\mathcal{W}} \\ \dot{\mathbf{R}} = \mathbf{R}\widehat{\mathbf{\Omega}} \end{cases} \xrightarrow{\frac{\mathrm{d}}{\mathrm{d}t}} \boldsymbol{\zeta}^{(3)} = -\frac{\dot{f}}{m}\mathbf{R}\,\mathbf{z}_{\mathcal{W}} - \frac{f}{m}\mathbf{R}\widehat{\mathbf{\Omega}}\,\mathbf{z}_{\mathcal{W}}$$

$$\mathbf{R}^{\intercal}\boldsymbol{\zeta}^{(3)} = -\frac{\dot{f}}{m}\mathbf{z}_{\mathcal{W}} - \frac{f}{m}\widehat{\mathbf{\Omega}}\,\mathbf{z}_{\mathcal{W}}$$

$$\mathbf{R}^{\intercal}\boldsymbol{\zeta}^{(3)} = -\frac{\dot{f}}{m}\mathbf{z}_{\mathcal{W}} - \frac{f}{m}\widehat{\mathbf{\Omega}}\,\mathbf{z}_{\mathcal{W}}$$

$$\Rightarrow \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}\mathbf{R}^{\intercal}\boldsymbol{\zeta}^{(3)} = -\frac{f}{m}\begin{pmatrix} \Omega_y \\ -\Omega_x \end{pmatrix}$$

$$+$$

$$f = \frac{mg}{\cos(\alpha)}$$

$$\Downarrow$$

$$\begin{pmatrix} \Omega_y \\ \Omega_x \end{pmatrix} = \frac{\cos(\alpha)}{g}\begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}\mathbf{R}^{\intercal}\boldsymbol{\zeta}^{(3)}$$

$$j \leq g\Omega_{\max} \Rightarrow \Omega \leq \Omega_{\max}$$

Trajectory duration

Optimization problem

Minimum-time trajectory

$$\Delta\tau^* = \arg\min_{\Delta\tau\in\mathbb{R}} \Delta\tau$$

$$\text{s.t.}\begin{cases} \Delta\tau > 0 \\ \forall i \in [\![0, n-1]\!] \quad \left\|\mathbf{p}_i^{(1)}\right\| \leq v_{\max} \\ \forall i \in [\![0, n-2]\!] \quad \left\|\mathbf{p}_i^{(2)}\right\| \leq g\tan(\alpha_{\max}) \\ \forall i \in [\![0, n-3]\!] \quad \left\|\mathbf{p}_i^{(3)}\right\| \leq g\Omega_{\max} \end{cases}$$

Trajectory duration

Optimization problem

$$\mathbf{P} = \begin{pmatrix} p_0^x & p_1^x & \dots & p_n^x \\ p_0^y & p_1^y & \dots & p_n^y \end{pmatrix}^{\intercal}$$

$$\mathbf{P}^{(1)} = \frac{1}{\Delta\tau}\mathbf{Q}_1\mathbf{P} = \frac{1}{\Delta\tau}\widetilde{\mathbf{P}}^{(1)}$$

$$\mathbf{P}^{(2)} = \frac{1}{\Delta\tau^2}\mathbf{Q}_2\mathbf{P} = \frac{1}{\Delta\tau^2}\widetilde{\mathbf{P}}^{(2)}$$

$$\mathbf{P}^{(3)} = \frac{1}{\Delta\tau^3}\mathbf{Q}_3\mathbf{P} = \frac{1}{\Delta\tau^3}\widetilde{\mathbf{P}}^{(3)}$$

$$\mathbf{Q}_2 = \begin{pmatrix} 6 & -9 & 3 & & & & & & & & & \\ & 1.5 & -2.5 & 1 & & & & & & & & \\ & & 1 & -2 & 1 & & & & & & & \\ & & & 1 & -2 & 1 & & & & & & \\ & & & & 1 & -2 & 1 & & & & & \\ & & & & & 1 & -2 & 1 & & & & \\ & & & & & & 1 & -2 & 1 & & & \\ & & & & & & & 1 & -2 & 1 & & \\ & & & & & & & & 1 & -2.5 & 1.5 & \\ & & & & & & & & & 3 & -9 & 6 \end{pmatrix}$$

$\widetilde{\mathbf{P}}^{(1)}, \widetilde{\mathbf{P}}^{(2)}, \widetilde{\mathbf{P}}^{(3)}$ fixed

Trajectory duration

Optimization problem

Minimum-time trajectory

$$\Delta\tau^* = \arg\min_{\Delta\tau\in\mathbb{R}} \Delta\tau$$

$$\text{s.t.}\begin{cases} \Delta\tau \geq \varepsilon \\ \forall i \in [\![0, n-1]\!] \quad \Delta\tau \geq \dfrac{\left\|\tilde{\mathbf{p}}_i^{(1)}\right\|}{v_{\max}} \\ \forall i \in [\![0, n-2]\!] \quad \Delta\tau \geq \sqrt{\dfrac{\left\|\tilde{\mathbf{p}}_i^{(2)}\right\|}{g\tan(\alpha_{\max})}} \\ \forall i \in [\![0, n-3]\!] \quad \Delta\tau \geq \sqrt[3]{\dfrac{\left\|\tilde{\mathbf{p}}_i^{(3)}\right\|}{g\Omega_{\max}}} \end{cases}$$

$\Rightarrow$

$$\Delta\tau^* = \arg\min_{\Delta\tau\in\mathbb{R}} \Delta\tau$$

$$\text{s.t. } \mathbf{1}_{3n-2}\Delta\tau \geq \boldsymbol{\Delta\tau}_{\min}$$

$$\mathbf{1}_{\mathrm{i}} = \underbrace{(1 \quad 1 \quad \dots \quad 1)}_{i \text{ elements}}^{\intercal}$$

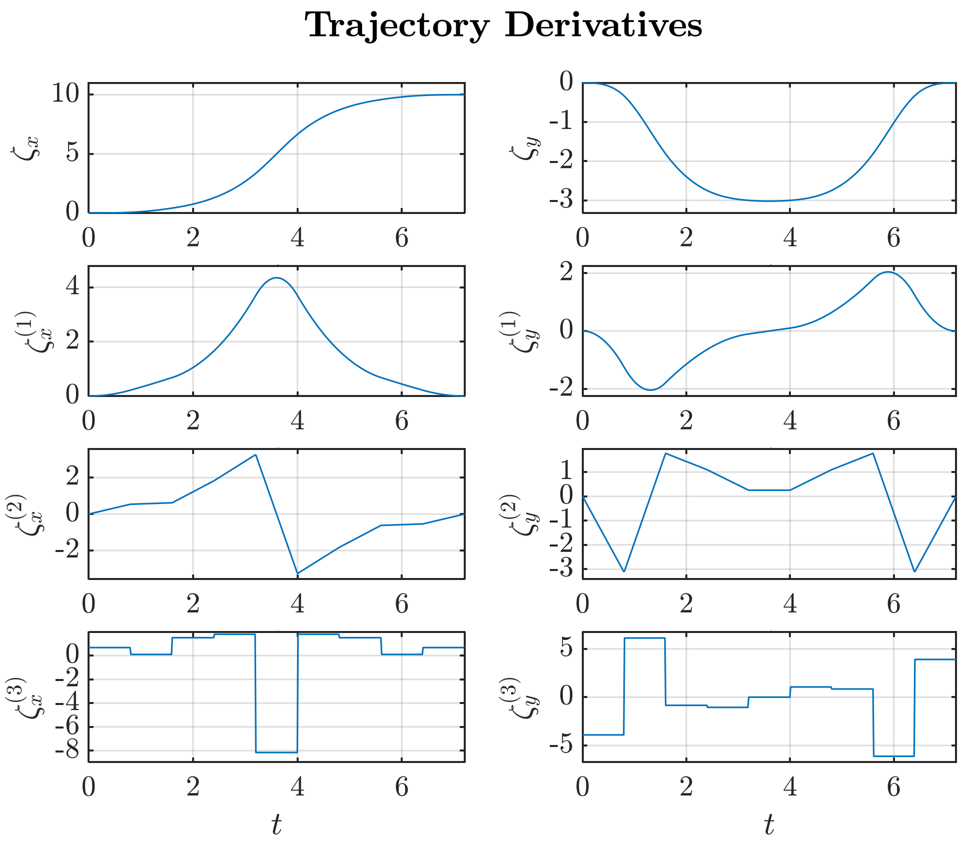
$\Rightarrow$

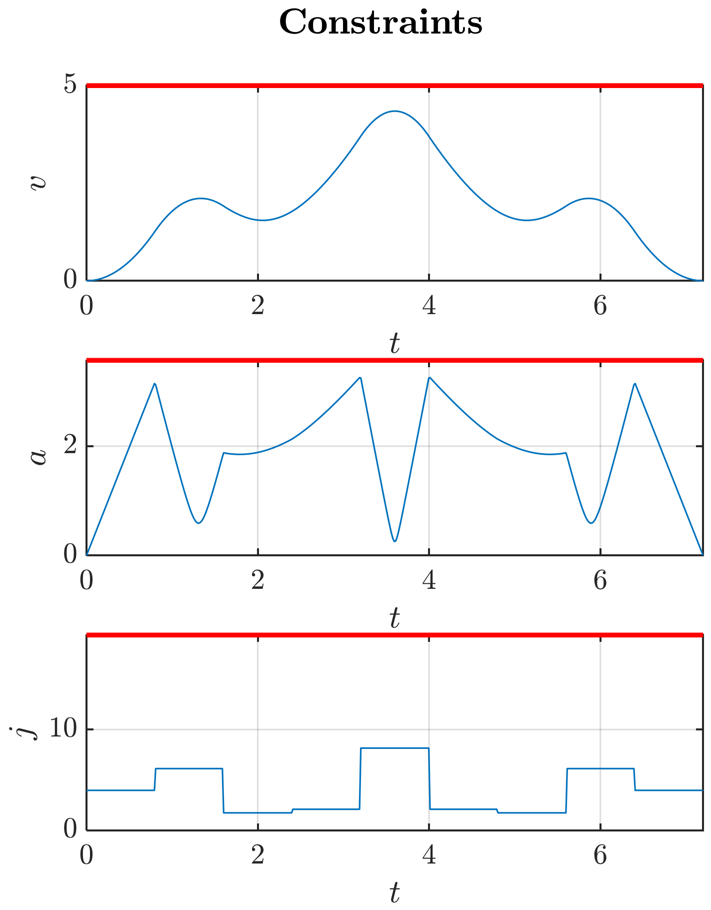
Trivial solution

$$\Delta\tau^* = \max(\boldsymbol{\Delta\tau}_{\min})$$

Trajectory duration

Optimization problem





Feasible trajectory!

Trajectory duration

Optimization problem

$n_1 = n_2 = n_3 = 2 \qquad \sigma = 0.5$