Planning and estimation using B-Splines with applications to autonomous vehicles

Randal W Beard
Brigham Young University

Congresso Brasileiro de Automatica
October 18, 2024

Outline

- Motivation
- B-Spline Tutorial
- Applications to Planning
  - Collision Avoidance
  - Cooperative Temporal-Spatial Sequencing
  - Cooperative Informative Path Planning
- Applications to Estimation:
  - Moving horizon estimation exploiting Differential Flatness
  - Calibration

Collaborators and Students

Tim McLain
Professor
Mechanical
Engineering

Cammy Peterson
Professor
Electrical and
Computer
Engineering

David Christensen
MS Graduate

Grant Stagg
PhD Candidate

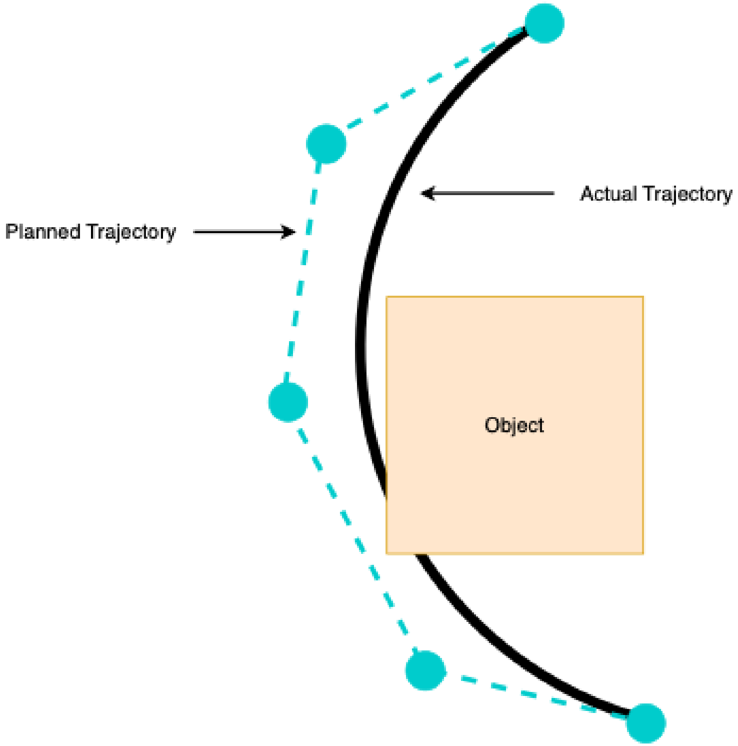
Landon Shumway
MS Candidate

Jake Johnson
PhD Graduate

Motivation

Path planning:

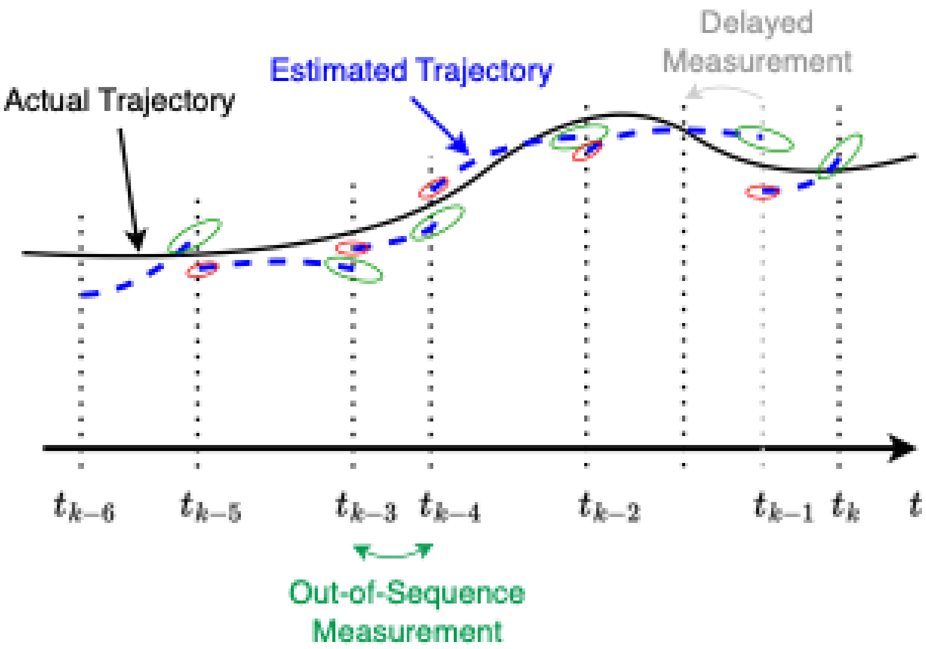
- Discrete parameterization:
    - Always a question of what happens in-between planning points.
    - Cannot guarantee collision avoidance or safety.
- Continuous parameterizations of path
    - Better chance to make provable statements with continuous paths



Motivation

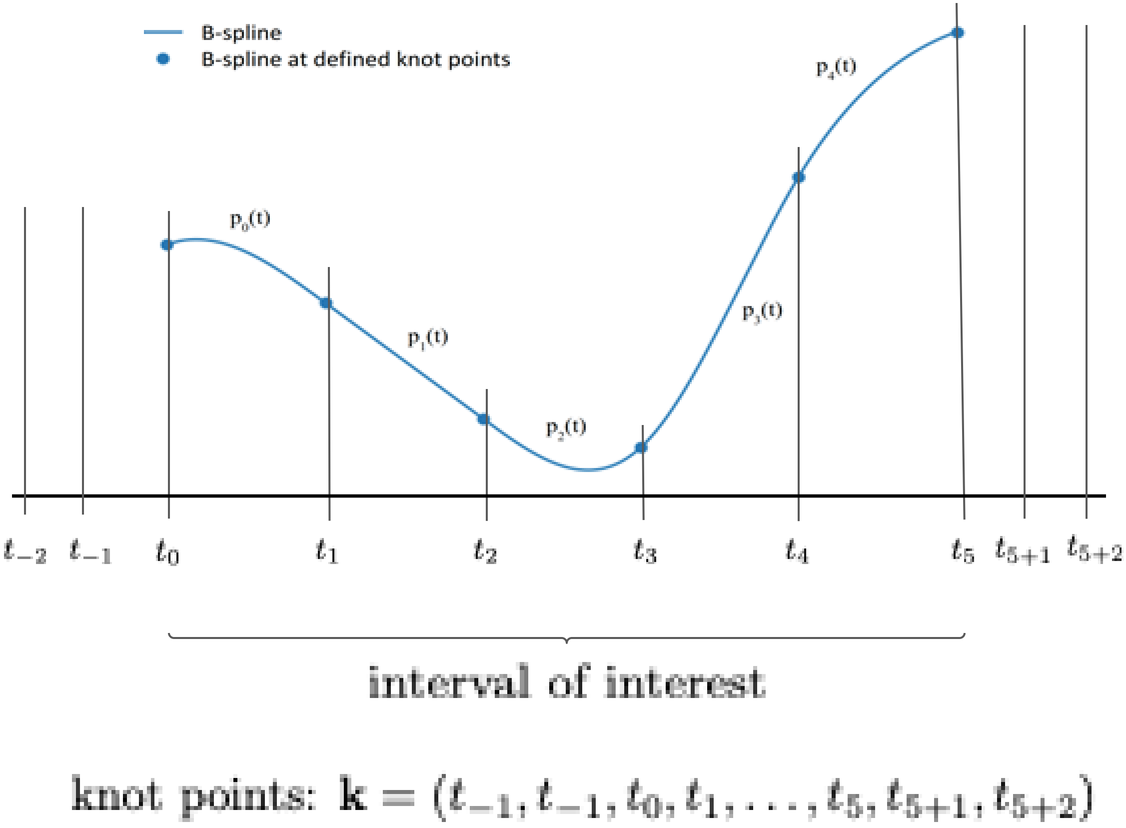
Estimation:

- Discrete parameterization:
  - Typical approach makes assumptions about clocks and sample rate.
- Continuous parameterizations
  - MHE: Easily handles
    - out of sequence measurement
    - delayed measurement
    - non-aligned clocks
    - inaccurate time stamping



What is a B-spline?

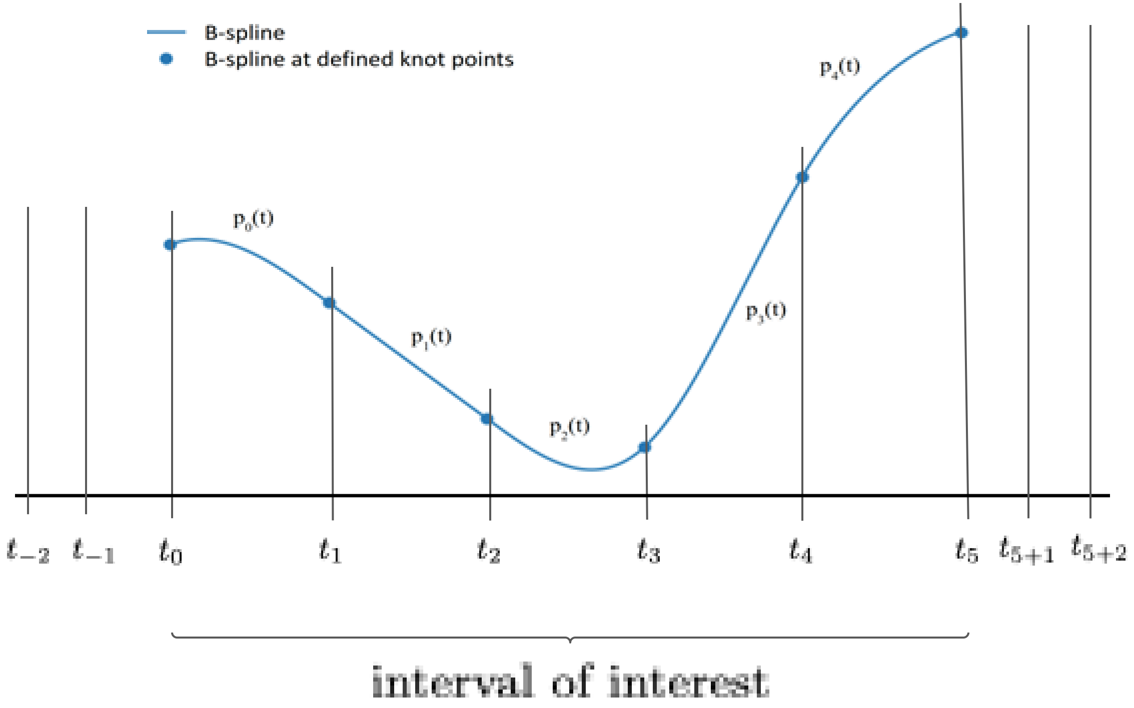
- "B" stands for basis
  - B-spline = Basis-spline
- The bases are Bernstein polynomials
  - So "B" could stand for Bernstein-splines.
- Each segment of B-spline is a Bezier curve
  - So "B" could stand for Bezier-splines



What is a B-spline?

Over each of M intervals of interest, the B-spline is a polynomial of degree d:

$$p(t) = \begin{cases} p_0(t) & \text{for} \quad t_0 \le t < t_1 \\ p_1(t) & \text{for} \quad t_1 \le t < t_2 \\ \quad \vdots & \\ p_{M-1}(t) & \text{for} \quad t_{M-1} \le t < t_M \end{cases}$$

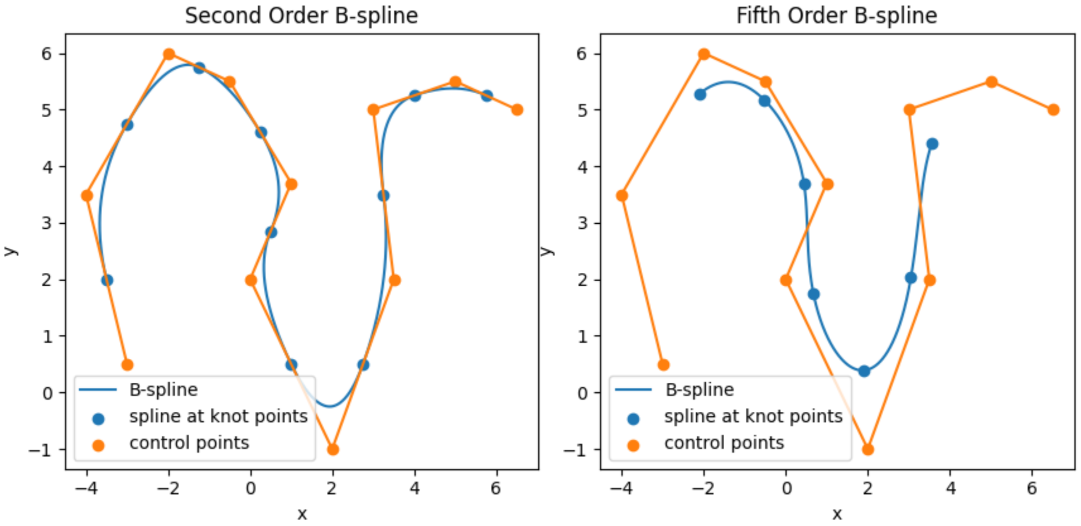


knot points: $\mathbf{k} = (t_{-1}, t_{-1}, t_0, t_1, \ldots, t_5, t_{5+1}, t_{5+2})$

What is a B-spline?

A B-spline of degree d, with M intervals of interest, is defined by

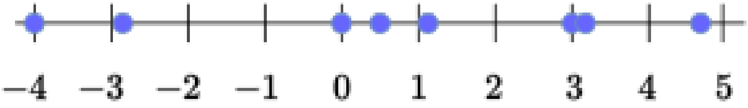
$$\mathbf{p}(t) = \sum_{m=0}^{M+d-1} \underbrace{\mathbf{c}_m}_{\text{Control points}} \underbrace{b_m^d(t, \mathbf{k})}_{\text{Basis functions}}$$



B-Spline: Knot Points

Natural knot vector: $\mathbf{k}_M^d = [\underbrace{t_{-d}, \dots, t_{-1},}_{\substack{d\text{-terms} \\ \text{non-decreasing}}} \quad \underbrace{t_0, t_1, \dots, t_M,}_{\substack{\text{interval of interest} \\ (M+1)\text{-terms} \\ \text{strictly increasing}}} \quad \underbrace{t_{M+1}, \dots, t_{M+d}}_{\substack{d\text{-terms} \\ \text{non-decreasing}}}].$

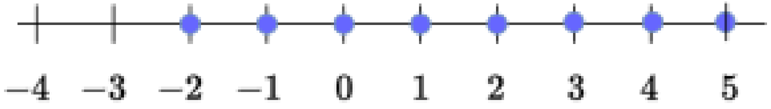
Example: $\mathbf{k}_{M=3}^{d=2} = [-4, -2.8, \vdots\, 0, 0.5, 1.1, 3.0, \vdots\, 3.1, 4.6]$



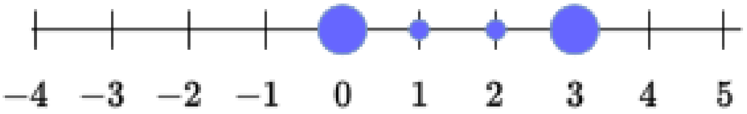
Uniform knot vector:

$$\mathbf{k}_M^d = [\underbrace{t_0 - d\Delta, \dots, t_0 - \Delta,}_{d\text{-terms}} \underbrace{t_0, t_0 + \Delta, t_0 + 2\Delta, \dots, t_0 + M\Delta,}_{\substack{\text{interval of interest} \\ (M+1)\text{-terms}}} \underbrace{t_0 + (M+1)\Delta, \dots, t_0 + (M+d)\Delta}_{d\text{-terms}}]$$

Example: $\mathbf{k}_{M=3}^{d=2} = [-2, -1, \vdots\, 0, 1, 2, 3, \vdots\, 4, 5], \quad \Delta = 1$



Clamped knot vector: $\mathbf{k}_M^d = [\underbrace{t_0, t_0, \dots, t_0,}_{d\text{-terms}} \quad \underbrace{t_0, t_1, t_2, \dots, t_M,}_{\substack{\text{interval of interest} \\ (M+1)\text{-terms} \\ \text{strickly increasing}}} \quad \underbrace{t_M, \dots, t_M}_{d\text{-terms}}],$



Example: $\mathbf{k}_{M=3}^{d=2} = [0, 0, \vdots\, 0, 1, 2, 3, \vdots\, 3, 3], \quad \Delta = 1$

B-Spline: Basis Functions

The B-spline basis function are defined by the recursive formula:

$$b_m^0(t, \mathbf{k}) = \begin{cases} 1 & \text{if } \tau_m \leq t \leq \tau_{m+1} \\ 0 & \text{otherwise} \end{cases}$$

$$b_m^d(t, \mathbf{k}) = w_m^d(t, \mathbf{k}) b_m^{d-1}(t, \mathbf{k}) + \left[1 - w_{m+1}^d(t, \mathbf{k})\right] b_{m+1}^{d-1}(t, \mathbf{k}),$$

$$w_m^d(t, \mathbf{k}) = \begin{cases} \frac{t - \tau_m}{\tau_{m+d} - \tau_m}, & \tau_{m+d} \neq \tau_m \\ 0, & \text{otherwise} \end{cases}.$$

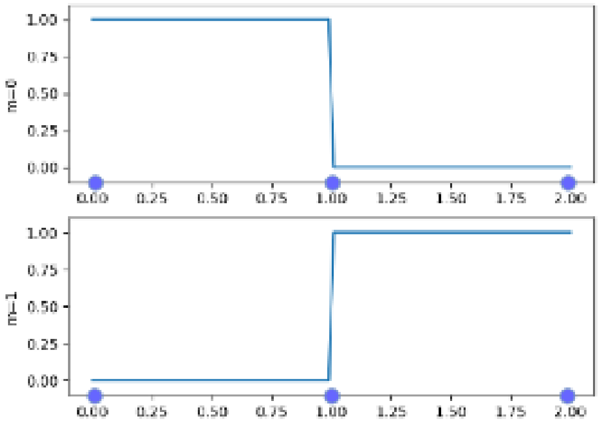
d=degree

m=index of basis

Depends on the knots $\mathbf{k}$, the degree d, and the index m

B-Spline: Degree-0 Basis Functions

$$d = 0$$
$$\mathbf{k} = (0, 1, 2)$$



$b_0^0(t, \mathbf{k})$

$b_1^0(t, \mathbf{k}) = b_0^0(t - t_1, \mathbf{k})$

For example, if the knot vector is given by

$$\mathbf{k}_{M=2}^{d=0} = [t_0, t_1, t_2] \doteq [0, 1, 2],$$

then there are two basis function of degree $d = 0$ given by

$$b_0^0(t, \mathbf{k}_2^0) = \begin{cases} 1 & \text{if } t_0 \leq t \leq t_1 \\ 0 & \text{otherwise} \end{cases}$$

$$b_1^0(t, \mathbf{k}_2^0) = \begin{cases} 1 & \text{if } t_1 \leq t \leq t_2 \\ 0 & \text{otherwise} \end{cases}$$

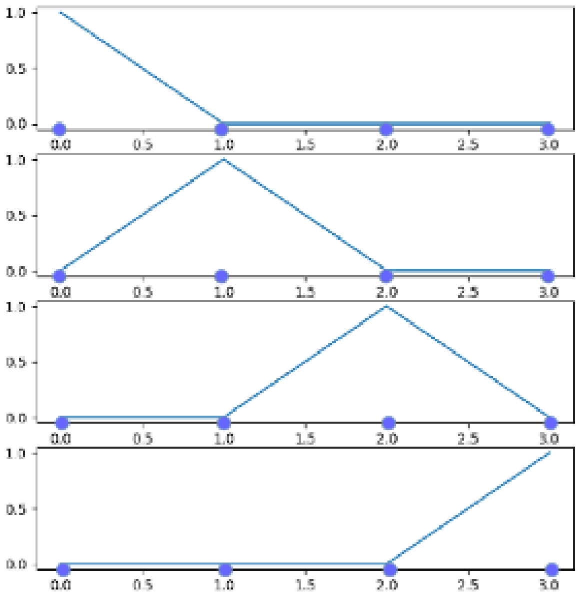
Clamped = Unclamped

All basis functions are shifted and scaled version of $b_0^0(t, \mathbf{k})$.

B-Spline: Degree-1 Basis Functions

$d = 1$

$\mathbf{k} = (-1, 0, 1, 2, 3, 4)$



$b_0^1(t,\mathbf{k})$

$b_1^1(t,\mathbf{k})$

$b_2^1(t,\mathbf{k})$

$b_3^1(t,\mathbf{k})$

Clamped = Unclamped

Knot vector:

$$\mathbf{k}^{d=1}_{M=3} = [t_{-1}, t_0, t_1, t_2, t_3, t_4] \doteq [-1, 0, 1, 2, 3, 4],$$

$$b_0^1(t,\mathbf{k}) = \frac{t - t_{-1}}{t_0 - t_{-1}} b_0^0(t,\mathbf{k}) + \frac{t_1 - t}{t_1 - t_0} b_1^0(t,\mathbf{k}) = \begin{cases} 0 & \text{if } t_{-1} \le t \le t_0 \\ 1 - t & \text{if } t_0 \le t \le t_1 \end{cases}$$

$$b_1^1(t,\mathbf{k}) = \frac{t - t_0}{t_1 - t_0} b_1^0(t,\mathbf{k}) + \frac{t_2 - t}{t_2 - t_1} b_2^0(t,\mathbf{k}) = \begin{cases} t & \text{if } t_0 \le t \le t_1 \\ 2 - t & t_1 \le t \le t_2 \\ 0 & \text{otherwise} \end{cases}$$

$$b_2^1(t,\mathbf{k}) = \frac{t - t_1}{t_2 - t_1} b_2^0(t,\mathbf{k}) + \frac{t_3 - t}{t_3 - t_2} b_2^0(t,\mathbf{k}) = \begin{cases} t - 1 & \text{if } t_1 \le t \le t_2 \\ 3 - t & t_2 \le t \le t_3 \\ 0 & \text{otherwise} \end{cases}$$

$$b_3^1(t,\mathbf{k}) = \frac{t - t_3}{t_4 - t_3} b_3^0(t,\mathbf{k}) + \frac{t_4 - t}{t_4 - t_3} b_4^0(t,\mathbf{k}) = \begin{cases} t - 3 & \text{if } t_2 \le t \le t_3 \\ 0 & \text{if } t_3 \le t \le t_4 \end{cases}$$

All basis functions are shifted and scaled version of $b_1^1(t,\mathbf{k})$.

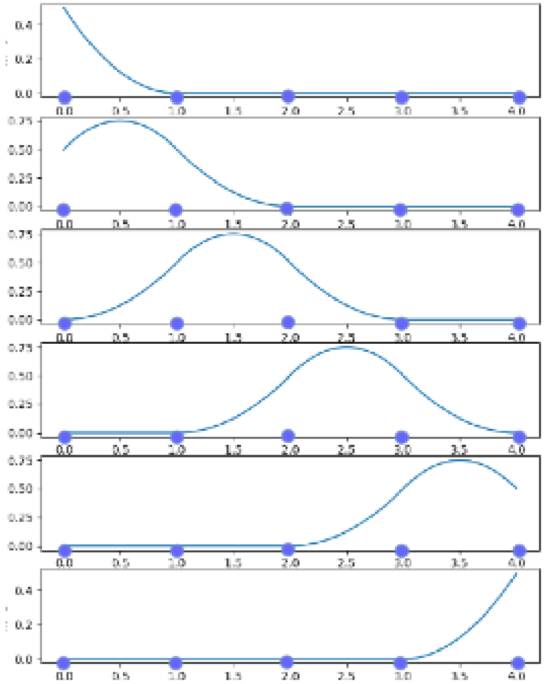
B-Spline: Degree-2 Basis Functions, unclamped

$d = 2$

$\mathbf{k} = (-2, -1, 0, 1, 2, 3, 4, 5, 6)$

Knot vector:

$\mathbf{k}_{M=4}^{k=2} = [t_{-2}, t_{-1}, t_0, t_1, t_2, t_3, t_4, t_5, t_6] \doteq [-2, -1, 0, 1, 2, 3, 4, 5, 6],$



$b_0^2(t, \mathbf{k})$

$b_1^2(t, \mathbf{k})$

$b_2^2(t, \mathbf{k})$

$b_3^2(t, \mathbf{k})$

$b_4^2(t, \mathbf{k})$

$b_5^2(t, \mathbf{k})$

$$b_0^2(t,\mathbf{k}) = \frac{t - t_{-2}}{t_0 - t_{-2}} b_0^1(t,\mathbf{k}) + \frac{t_1 - t}{t_1 - t_{-1}} b_1^1(t,\mathbf{k}) = \begin{cases} (1-t)^2 & \text{if } t_0 \le t \le t_1 \\ 0 & \text{otherwise} \end{cases}$$

$$b_1^2(t,\mathbf{k}) = \frac{t - t_{-1}}{t_1 - t_{-1}} b_1^1(t,\mathbf{k}) + \frac{t_2 - t}{t_2 - t_0} b_2^1(t,\mathbf{k}) = \begin{cases} t(2 - \frac{3}{2}t) & \text{if } t_0 \le t \le t_1 \\ \frac{(2-t)^2}{2} & t_1 \le t \le t_2 \\ 0 & \text{otherwise} \end{cases}$$

$$b_2^2(t,\mathbf{k}) = \frac{t - t_0}{t_2 - t_0} b_2^1(t,\mathbf{k}) + \frac{t_3 - t}{t_3 - t_1} b_3^1(t,\mathbf{k}) = \begin{cases} \frac{t^2}{2} & \text{if } t_0 \le t \le t_1 \\ -\frac{3}{2}t^2 + \frac{7}{2}t - \frac{3}{2} & \text{if } t_1 \le t \le t_2 \\ \frac{(3-t)^2}{2} & \text{if } t_2 \le t \le t_3 \\ 0 & \text{otherwise} \end{cases}$$

$$b_3^2(t,\mathbf{k}) = \frac{t - t_1}{t_3 - t_1} b_2^1(t,\mathbf{k}) + \frac{t_4 - t}{t_4 - t_2} b_4^1(t,\mathbf{k}) = \begin{cases} \frac{(t-1)^2}{2} & \text{if } t_1 \le t \le t_2 \\ -\frac{3}{2}(t-1)^2 + \frac{7}{2}(t-1) - \frac{3}{2} & \text{if } t_2 \le t \le t_3 \\ \frac{(4-t)^2}{2} & \text{if } t_3 \le t \le t_4 \\ 0 & \text{otherwise} \end{cases}$$

$$b_4^2(t,\mathbf{k}) = \frac{t - t_2}{t_4 - t_4} b_3^1(t,\mathbf{k}) + \frac{t_5 - t}{t_5 - t_6} b_4^1(t,\mathbf{k}) = \begin{cases} \frac{(t-2)^2}{2} & \text{if } t_2 \le t \le t_3 \\ -\frac{3}{2}(t-1)^2 + \frac{15}{2}(t-1) - \frac{15}{2} & \text{if } t_3 \le t \le t_4 \\ 0 & \text{otherwise} \end{cases}$$

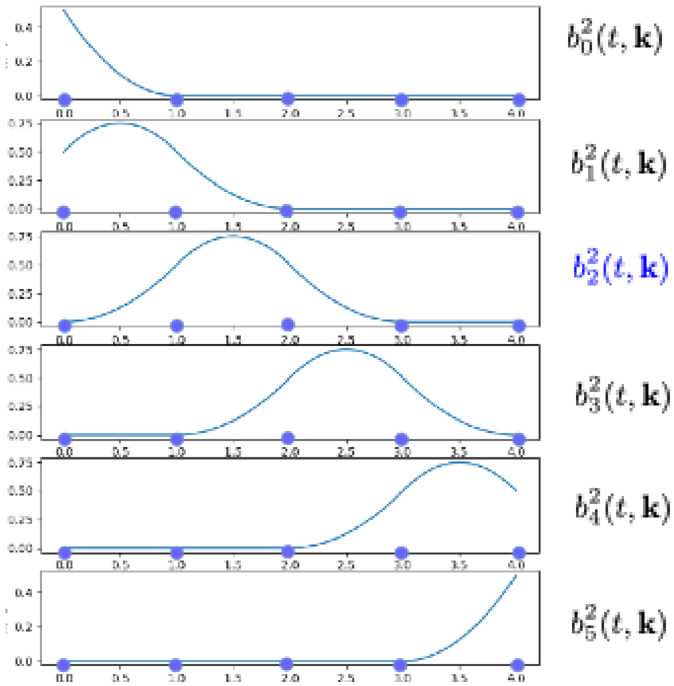
$$b_5^2(t,\mathbf{k}) = \frac{t - t_3}{t_5 - t_3} b_3^1(t,\mathbf{k}) + \frac{t_6 - t}{t_6 - t_4} b_4^1(t,\mathbf{k}) = \begin{cases} (t-3)^2 & \text{if } t_3 \le t \le t_4 \\ 0 & \text{otherwise} \end{cases}.$$

All basis functions are shifted and scaled version of $b_2^2(t, \mathbf{k})$.

B-Spline: clamped vs. unclamped

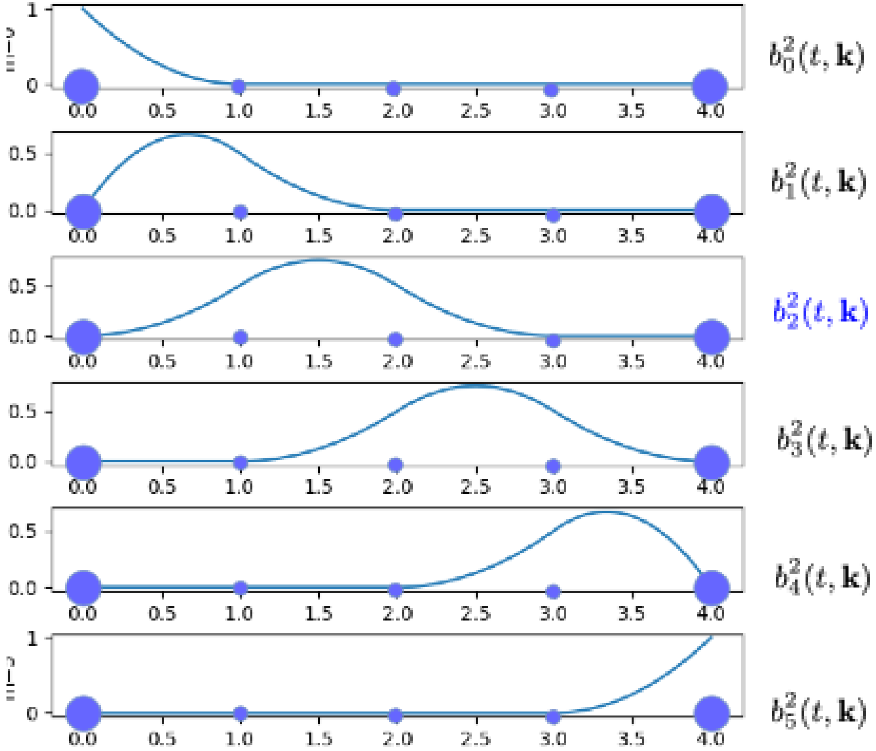
$d = 2$

$\mathbf{k} = (-2, -1, 0, 1, 2, 3, 4, 5, 6)$



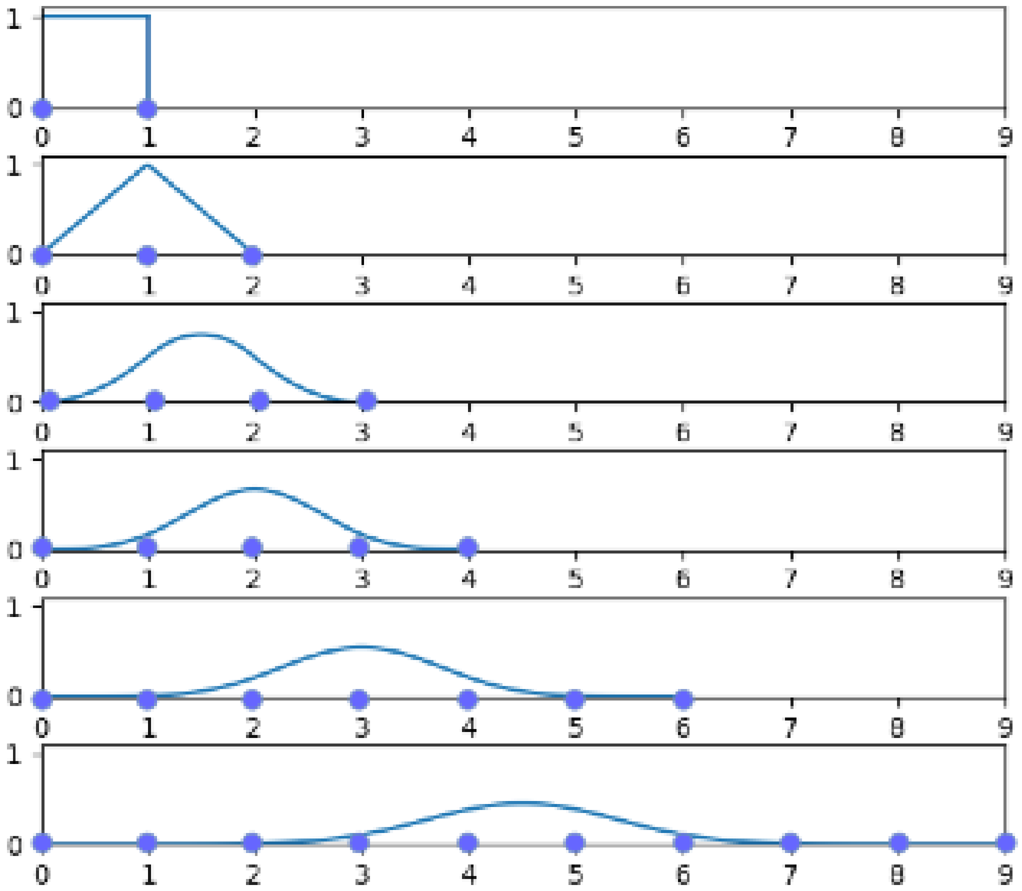
$d = 2$

$\mathbf{k} = (0, 0, 0, 1, 2, 3, 4, 4, 4)$



The basis function $b_2^2(t, \mathbf{k})$ still plays a central role.

B-Spline: Central Basis Functions



$b^{d=0}_{M=0}(t)$

$\mathbf{k} = [0, 1]$

$b^{d=1}_{M=1}(t)$

$\mathbf{k} = [-1, 0, 1, 2, 3]$

$b^{d=2}_{M=2}(t)$

$\mathbf{k} = [-2, -1, 0, 1, 2, 3, 4, 5]$

$b^{d=3}_{M=3}(t)$

$\mathbf{k} = [-3, -2, -1, 0, 1, 2, 3, 4,, 5, 6, 7]$

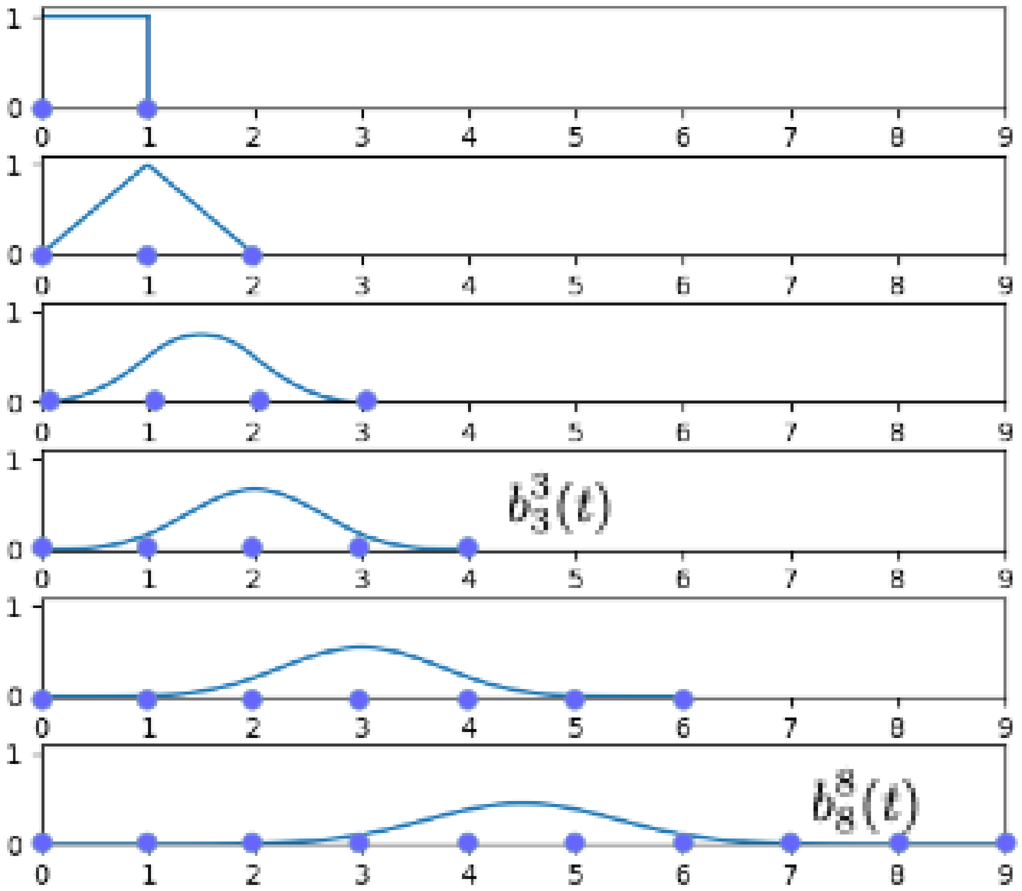
$b^{d=5}_{M=5}(t)$

$\mathbf{k} = [-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]$

$b^{d=8}_{M=8}(t)$

$$\mathbf{k} = [-8, -7, -6, -5, -4, -3, -2, -1, \\ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, \\ 10, 11, 12, 13, 14, 15, 16, 17]$$

B-Spline: Central Basis Functions



Note how little difference there is between $b_3^3(t)$ and higher order bases like $b_8^8(t)$.

Higher order bases don't have a lot of addition expressive power.

But they do add degrees or smoothness!

B-Spline Properties: Shift and Scale Invariance

Lemma. (Shift Invariance)
Shifting the knot sequence:

$$\mathbf{k} = [\tau_0, \tau_1, \ldots \tau_T] \quad \longrightarrow \quad \mathbf{k} + \Delta\mathbf{1} = [\tau_0 + \Delta, \tau_1 + \Delta, \ldots \tau_T + \Delta]$$

is equivalent to shifting the basis functions (or spline):

$$b_j^d(t, \mathbf{k} + \Delta\mathbf{1}) = b_j^d(t - \Delta, \mathbf{k})$$

Lemma. (Scale Invariance)
Scaling the knot sequence:

$$\mathbf{k} = [\tau_0, \tau_1, \ldots \tau_T] \quad \longrightarrow \quad \alpha\mathbf{k} = [\alpha\tau_0, \alpha\tau_1, \ldots \alpha\tau_T]$$

is equivalent to scaling the basis functions (or spline):

$$b_j^d(t, \alpha\mathbf{k}) = b_j^d(t/\alpha, \mathbf{k})$$

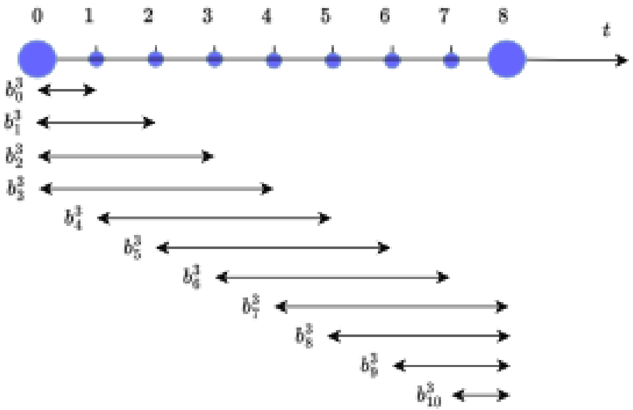
B-Spline Properties: Finite Support

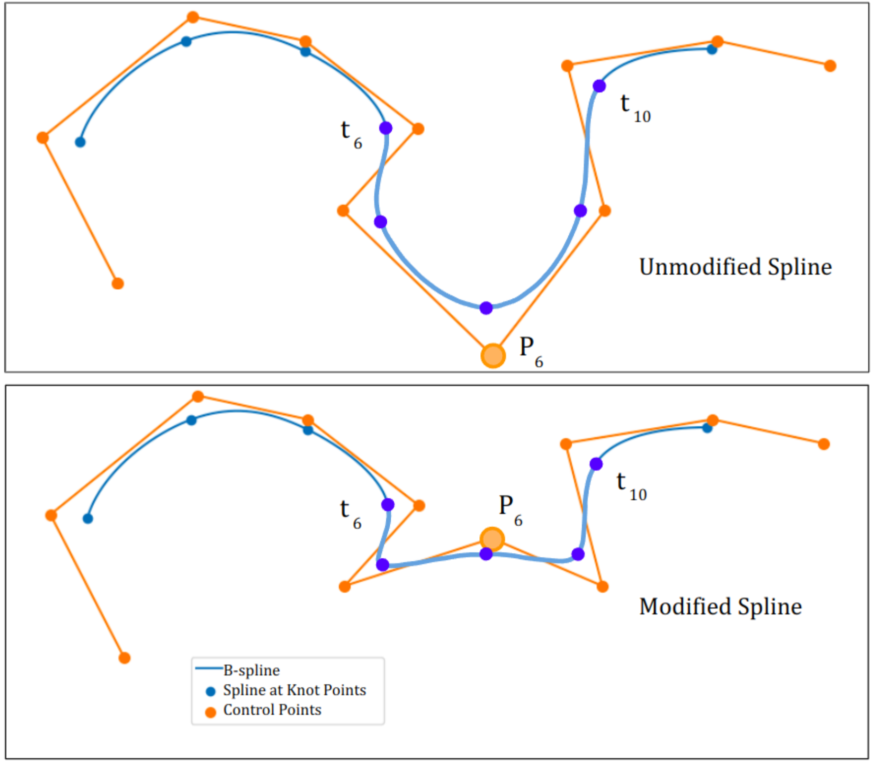
Lemma. (Finite Support)
For any time $t \in [t_s, t_{s+1}]$, there are only $d+1$ non-zero basis functions:

$$p(t) = \sum_{m=0}^{M+d-1} c_m b_m^d(t, k_M^d) = \sum_{m=s}^{s+d} c_m b_m^d(t, k_M^d).$$

Implication: There are only $d+1$ control points that influence $p(t)$ at any instance of time.

$$\mathbf{t}_8^3 = [0, 0, 0, \vdots 0, 1, 2, 3, 4, 5, 6, 7, 8, \vdots 8, 8, 8]$$

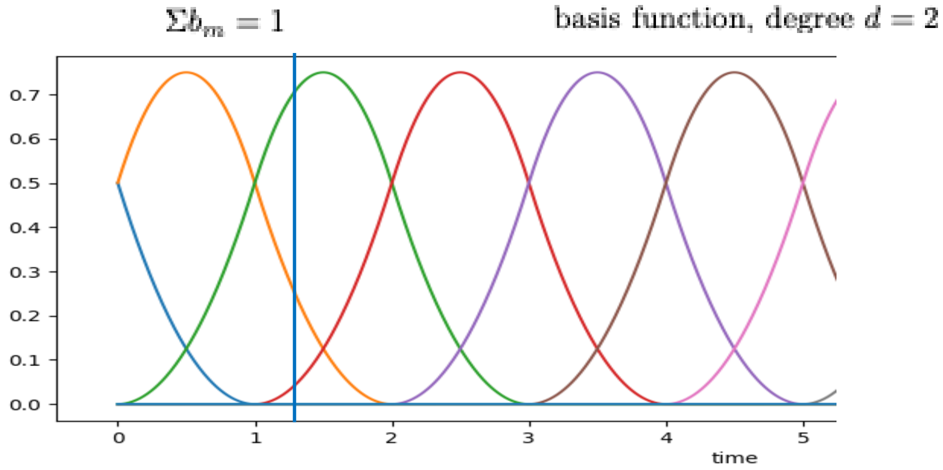


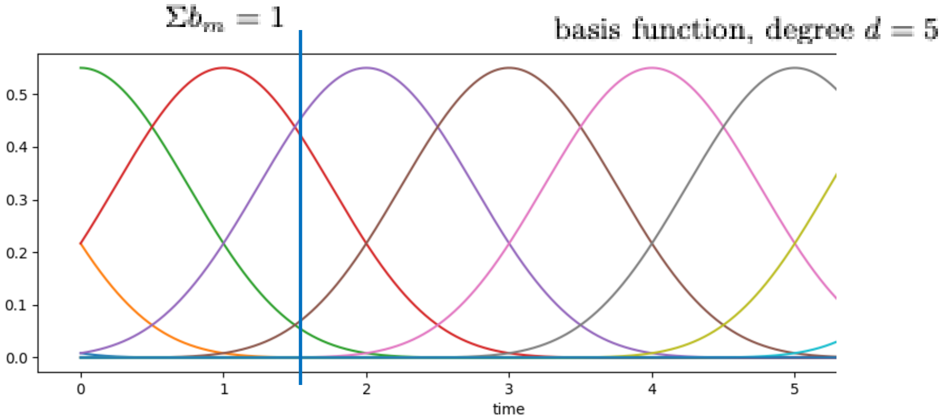


B-Spline Properties: Convex Hull

Lemma. (Partition of Unity)
For any time $t \in [t_s, t_{s+1}]$, the $d+1$ basis functions sum to unity:

$$\sum_{m=0}^{M+d-1} b_m^d(t, \mathbf{k}_M^d) = \sum_{m=s}^{s+d} b_m^d(t, \mathbf{k}_M^d) = 1$$



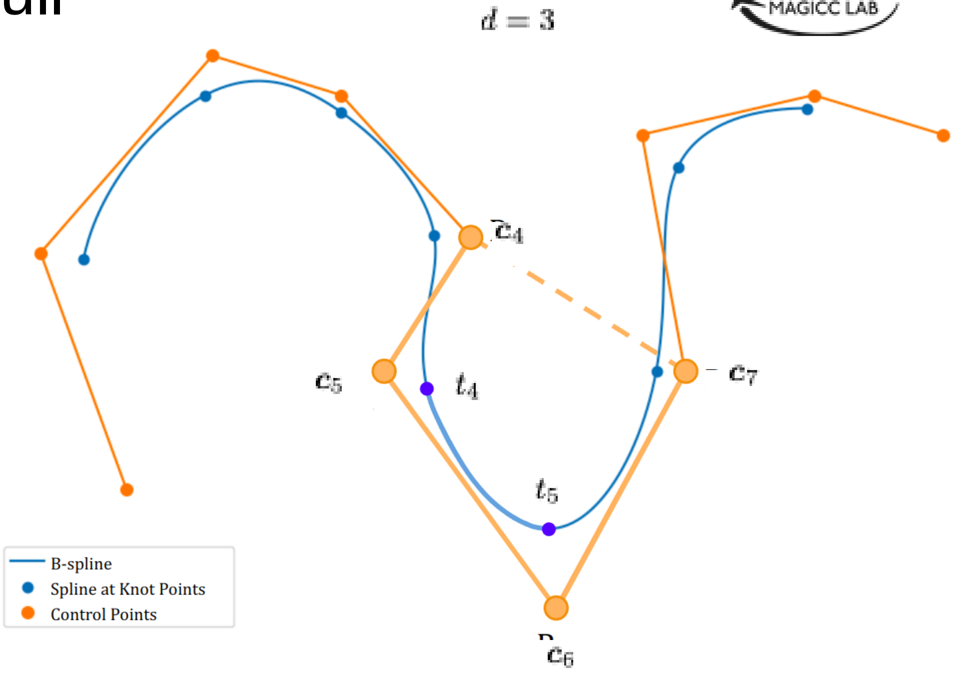


B-Spline Properties: Convex Hull

Corollary. (Convex Hull)
For any time $t \in [t_s, t_{s+1}]$, the spline

$$\mathbf{p}(t) = \sum_{m=0}^{M+d-1} c_m b_m^d(t, \mathbf{k}_M^d) = \sum_{m=s}^{s+d} c_m b_m^d(t, \mathbf{k}_M^d)$$

is contained in the convex hull of $\{c_s, \ldots, c_{s+d}\}$

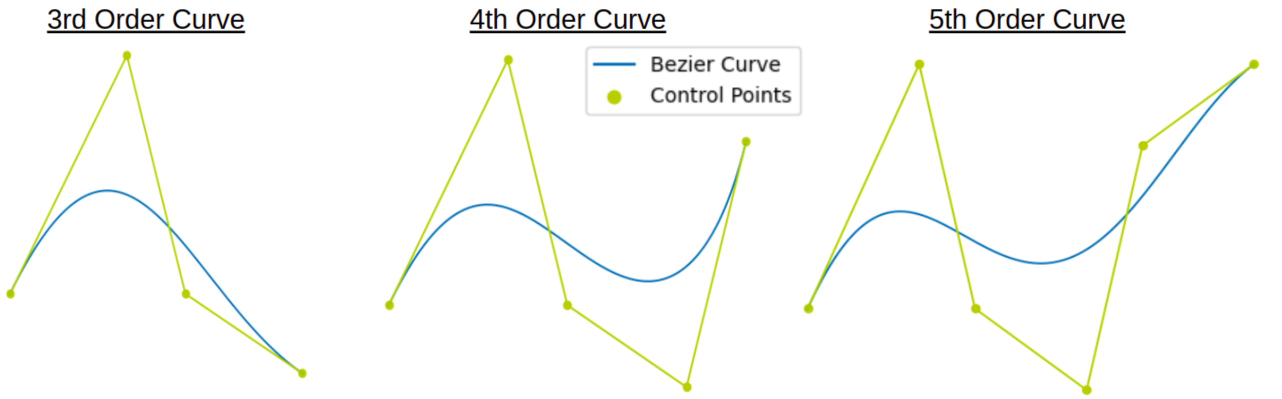


The spline segment from $[t_4, t_5]$ is contained in

$\text{conv}\{c_4, c_5, c_6, c_7\}$

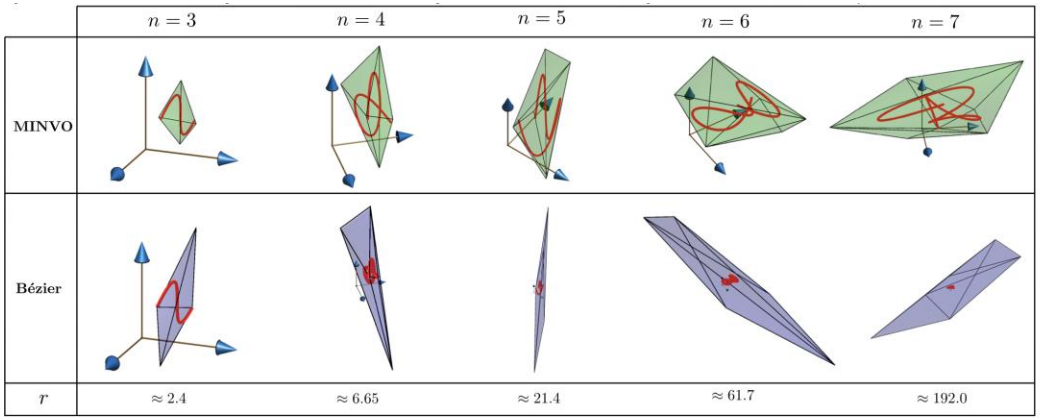
Bezier Curves: Convex Hull

- Every segment of a B-spline is a Bezier curve, but with different control points: the curve always starts and ends on a control point
- Bezier curves also enjoy a convex hull property.
- Often the convex hull is smaller for Bezier curves.

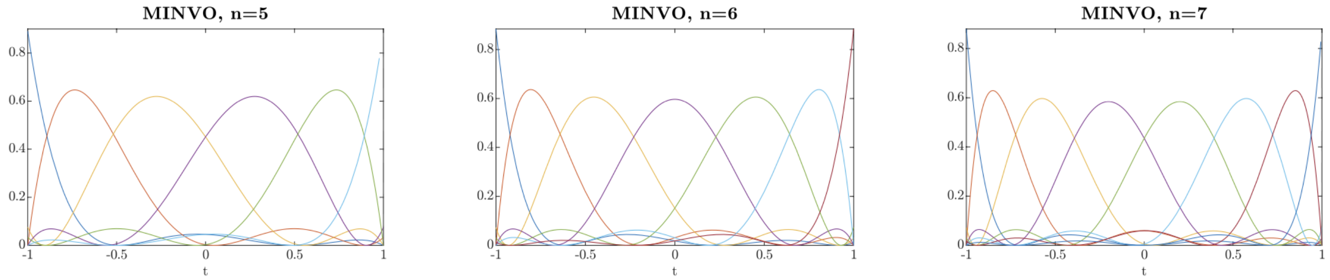


MINVO Curves: Convex Hull

MINVO basis:
Given a polynomial curve $\boldsymbol{p}(t)$ of degree d defined on $[t_0, t_1]$, find the minimum volume simplex (control points) so that $\boldsymbol{p}(t) \subseteq \text{conv}(\boldsymbol{C})$.

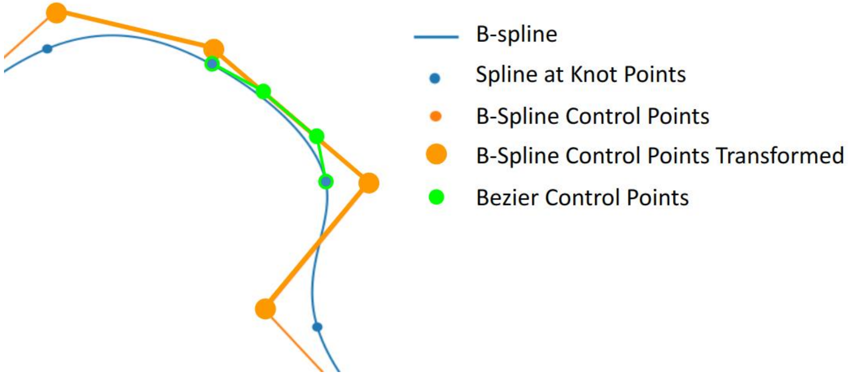


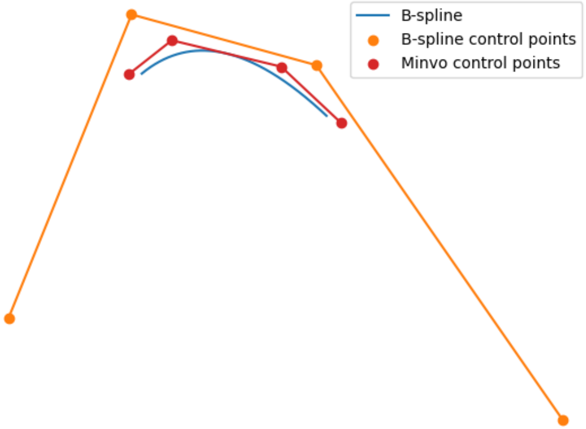
Generates MINVO basis functions, similar to Bezier curves.



MINVO Basis: Finding Simplexes with Minimum Volume Enclosing Polynomial Curves, Jesus Tordesillas and Jonathan P. How, Computer Aided Design, vol 151, 2022

B-Spline vs. Bezier vs MINVO Bounds





- The bounding simplexes for Bezier and MINVO are much tighter than for B-splines
- Bezier and MINVO are only valid over *one* time interval (not splines)
- For each time interval, there are well-defined matrix transformations between control points for B-splines, Bezier, and MINVO

Therefore, for applications like path planning, collision checks can be performed using MINVO simplexes instead of B-spline simplexes.

Derivatives of B-Spline

Stack the basis functions as

$$\boldsymbol{b}_M^j(t, \boldsymbol{k}_M^d) \doteq \begin{pmatrix} b_0^j(t, \boldsymbol{k}_M^d) \\ b_1^j(t, \boldsymbol{k}_M^d) \\ \vdots \\ b_{M+2d-j-1}^j(t, \boldsymbol{k}_M^d) \end{pmatrix}, \qquad 0 \leq j \leq d,$$

Collect the control points as

$$\boldsymbol{C} = \begin{pmatrix} \boldsymbol{c}_0 & \boldsymbol{c}_1 & \dots & \boldsymbol{c}_{M+d-1} \end{pmatrix}$$

Therefore B-spline can be written as

$$\boldsymbol{p}(t) = \boldsymbol{C}\boldsymbol{b}_M^d(t, \boldsymbol{k}_M^d), \qquad t \in \mathrm{span}(\boldsymbol{k}_M^d).$$

Derivatives of B-Spline

Lemma Given spline of degree d

$$\boldsymbol{p}(t) = \boldsymbol{C}\boldsymbol{b}_M^d(t)$$

its time derivative is given by

$$\frac{d\boldsymbol{p}}{dt}(t) = \boldsymbol{C}D_M^d\boldsymbol{b}_M^{d-1}(t)$$

where

$$D_M^d = -\begin{bmatrix} \bar{D}_M^d \\ \mathbf{0}_{1\times(M+d-1)} \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{1\times(M+d-1)} \\ \bar{D}_M^d \end{bmatrix},$$

and where

$$\bar{D}_M^d = \operatorname{diag}\left(\frac{d}{t_{-d+1} - t_1}, \frac{d}{t_{-d+2} - t_2}, \dots, \frac{d}{t_{M+2d-2} - t_{M+d-2}}\right)$$

Take Away: To differentiate, control points changed by matrix multiply and reducing the degree of the spline basis.

Derivatives of B-Spline

For uniform clamped the control points for the derivative are

$$C' \doteq CD_M^k = \begin{pmatrix} \frac{d}{1}(\mathbf{c}_1 - \mathbf{c}_0)^\top \\ \vdots \\ \frac{d}{d-1}(\mathbf{c}_{d-2} - \mathbf{c}_{d-3})^\top \\ (\mathbf{c}_{d-1} - \mathbf{c}_{d-2})^\top \\ \vdots \\ (\mathbf{c}_{M+1} - \mathbf{c}_M)^\top \\ \frac{d}{d-1}(\mathbf{c}_{M+2} - \mathbf{c}_{M+1})^\top \\ \vdots \\ \frac{d}{1}(\mathbf{c}_{M+d-1} - \mathbf{c}_{M+d-2})^\top \end{pmatrix}^\top$$

For uniform unclamped the control points for the derivative are

$$C' \doteq CD_M^k = \begin{pmatrix} (\mathbf{c}_1 - \mathbf{c}_0)^\top \\ \vdots \\ (\mathbf{c}_{d-1} - \mathbf{c}_{d-2})^\top \\ \vdots \\ (\mathbf{c}_{M+1} - \mathbf{c}_M)^\top \\ \vdots \\ (\mathbf{c}_{M+d-1} - \mathbf{c}_{M+d-2})^\top \end{pmatrix}^\top$$

Implication for Path Planning: Clamped, uniform

If the desired B-spline trajectory with $d \geq 3$ has the following desired endpoint conditions:

$$\begin{aligned}
\text{Initial position:} \quad & \boldsymbol{p}(0) \stackrel{des}{=} \boldsymbol{p}_0 \\
\text{Final position:} \quad & \boldsymbol{p}(M) \stackrel{des}{=} \boldsymbol{p}_f \\
\text{Initial velocity:} \quad & \frac{d\boldsymbol{p}}{dt}(0) \stackrel{des}{=} \mathbf{v}_0 \\
\text{Final velocity:} \quad & \frac{d\boldsymbol{p}}{dt}(M) \stackrel{des}{=} \mathbf{v}_f, \\
\text{Initial acceleration:} \quad & \frac{d^2\boldsymbol{p}}{dt^2}(0) \stackrel{des}{=} \mathbf{a}_0 \\
\text{Final acceleration:} \quad & \frac{d^2\boldsymbol{p}}{dt^2}(M) \stackrel{des}{=} \mathbf{a}_f,
\end{aligned}$$

then the first and last three control points satisfy

$$\begin{aligned}
\mathbf{c}_0 &= \boldsymbol{p}_0 \\
\mathbf{c}_1 &= \boldsymbol{p}_0 + \frac{1}{d}\mathbf{v}_0 \\
\mathbf{c}_2 &= \boldsymbol{p}_0 + \frac{3}{d}\mathbf{v}_0 + \frac{2}{(d)(d-1)}\mathbf{a}_0 \\
\mathbf{c}_{M+d-3} &= \boldsymbol{p}_f - \frac{3}{d}\mathbf{v}_f + \frac{2}{(d)(d-1)}\mathbf{a}_f \\
\mathbf{c}_{M+d-2} &= \boldsymbol{p}_f - \frac{1}{d}\mathbf{v}_f \\
\mathbf{c}_{M+d-1} &= \boldsymbol{p}_f.
\end{aligned}$$

Similar techniques for unclamped and non-uniform B-spline trajectories

Implication for Velocity Constraints

Velocity constraints:

$$\left\| \frac{\mathbf{c}_{i+1} - \mathbf{c}_i}{\alpha} \right\|^2 \leq V_{\max}^2$$

$$\|\mathbf{c}_{i+1} - \mathbf{c}_i\|^2 \leq \alpha^2 V_{\max}^2$$

where α is the knot vector scale (from integers $1 : M$).

Use MINVO for tighter bounds:
For each time interval, transform control points as

$$\begin{bmatrix} \boldsymbol{\nu}_i & \cdots & \boldsymbol{\nu}_{i+k} \end{bmatrix} = \begin{bmatrix} \mathbf{c}_i & \cdots & \mathbf{c}_{i+k} \end{bmatrix} F_{\text{b-spline}}^{\text{MINVO}}$$

where $F_{\text{b-spline}}^{\text{MINVO}}$ transforms B-spline control points to MINVO control points.

Constraint becomes

$$\|\boldsymbol{\nu}_{i+1} - \boldsymbol{\nu}_i\|^2 \leq \alpha^2 V_{\max}^2$$

Implication for Acceleration Constraints

Acceleration constraints:

$$\left\|\frac{\boldsymbol{c}_{i+2} - 2\boldsymbol{c}_{i+1} - 2\boldsymbol{c}_i}{\alpha^2}\right\|^2 \leq a_{\max}^2$$

$$\|\boldsymbol{c}_{i+2} - 2\boldsymbol{c}_{i+1} - 2\boldsymbol{c}_i\|^2 \leq \alpha^2 a_{\max}^2$$

Use MINVO for tighter bounds:

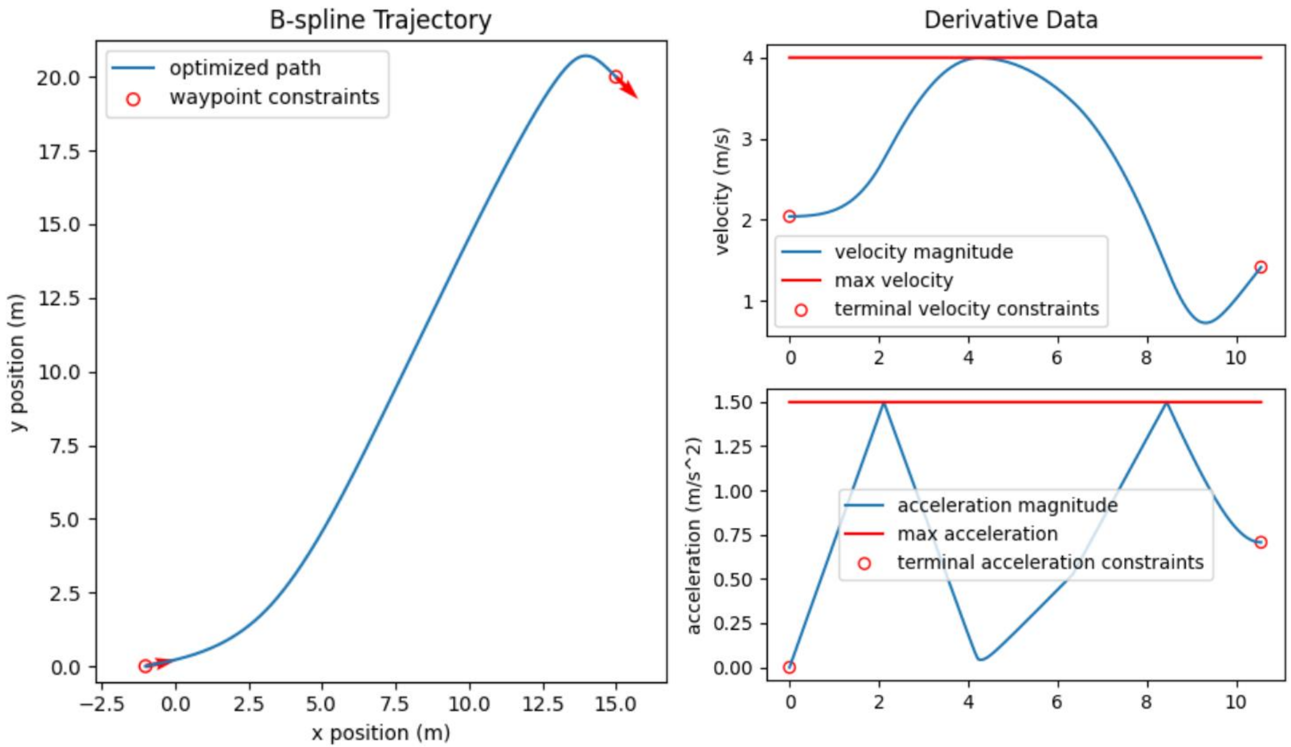
$$\begin{bmatrix}\boldsymbol{\nu}_i & \cdots & \boldsymbol{\nu}_{i+k}\end{bmatrix} = \begin{bmatrix}\boldsymbol{c}_i & \cdots & \boldsymbol{c}_{i+k}\end{bmatrix} F_{\text{b-spline}}^{\text{MINVO}}$$

Constraint becomes

$$\|\boldsymbol{\nu}_{i+2} - 2\boldsymbol{\nu}_{i+1} - 2\boldsymbol{\nu}_i\|^2 \leq \alpha^2 a_{\max}^2$$

Similar (but more complicated) techniques can be derived for constraining curvature


Trajectory Optimization Example



KEY POINT: velocity, acceleration, and velocity constraints at *every time* are guaranteed by checking constraints on the control points, NOT at discrete points along the trajectory.

Trajectory Optimization Example:
Tiltrotor aircraft

Initial and final:
- Position
- Velocity
- Acceleration

Acceleration and velocity constraints.

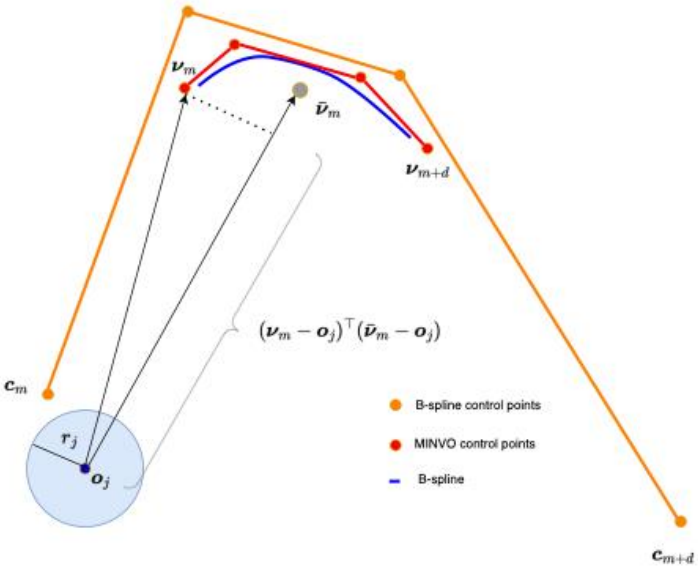
https://www.youtube.com/watch?v=nQ5Z9ri2-AY&t=3s

Nonlinear, Low-Energy-Actuator-Prioritizing Control Allocation for Winged eVTOL UAVs, Mason B. Peterson, Randal W. Beard, Jacob B. Willis, Intermountain Engineering, Technology, and Computing Conference, 2022

Outline

- Motivation
- B-Spline Tutorial
- Applications to Planning
  - Collision Avoidance
  - Cooperative Temporal-Spatial Sequencing
  - Cooperative Informative Path Planning
- Applications to Estimation:
  - Moving horizon estimation exploiting Differential Flatness
  - Calibration

Applications to Planning: Collision Avoidance



- For the m^{th} spline segment, convert to MINVO control points:

$$\underbrace{\begin{pmatrix}\boldsymbol{\nu}_m & \cdots & \boldsymbol{\nu}_{m+d}\end{pmatrix}}_{3\times d} = \underbrace{\begin{pmatrix}\boldsymbol{c}_m & \cdots & \boldsymbol{c}_{m+d}\end{pmatrix}}_{3\times d} \underbrace{F_{\text{b-spline}}^{\text{Minvo}}}_{d\times d}$$

- Define geometric center:

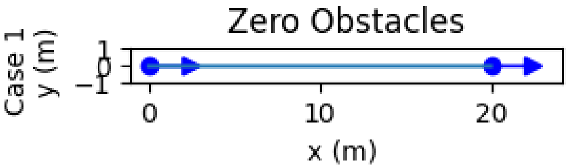
$$\bar{\boldsymbol{\nu}}_m \doteq \frac{1}{d}\sum_{i=m}^{m+d} \boldsymbol{\nu}_i$$

- Projection of relative MINVO control point onto the relative geometric center must be larger than r_j:

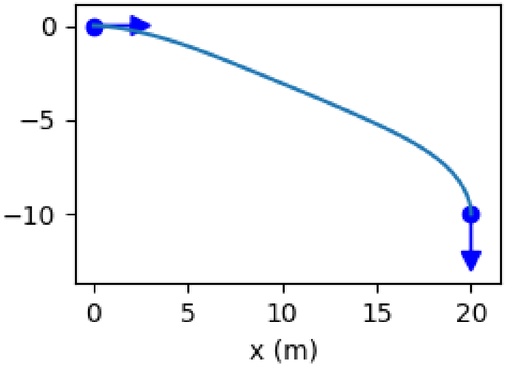
$$(\boldsymbol{\nu}_m - \boldsymbol{o}_j)^\top(\bar{\boldsymbol{\nu}}_m - \boldsymbol{o}_j) > r_j$$

Convex constraint in control points $\boldsymbol{c}_m, \ldots, \boldsymbol{c}_{m+d}$
...but nonlinear

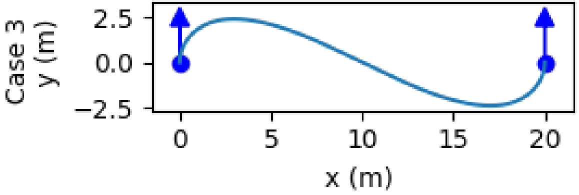
Applications to Planning: Collision Avoidance



evaluation time: 0.1
path length: 20.0
num ctrl pts: 8



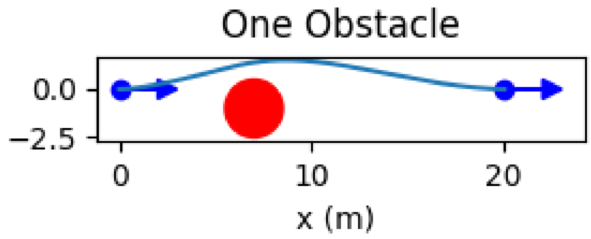
evaluation time: 0.26
path length: 23.22
num ctrl pts: 8



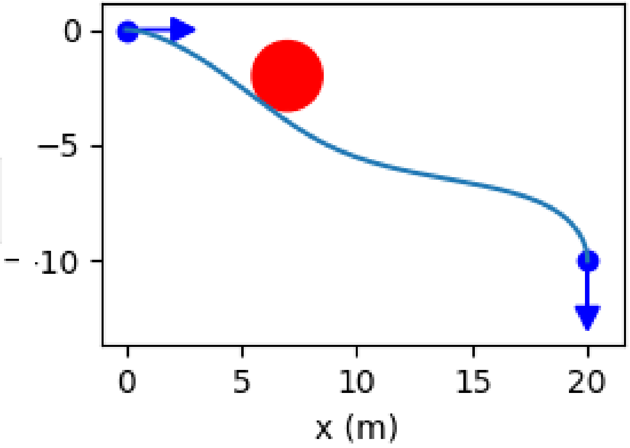
evaluation time: 0.76
path length: 23.46
num ctrl pts: 8

Zero Obstacles

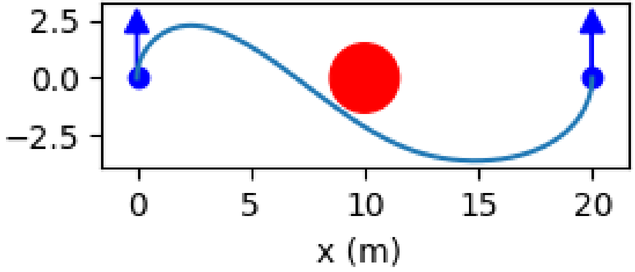
Applications to Planning: Collision Avoidance



evaluation time: 0.26
path length: 20.27
num ctrl pts: 8



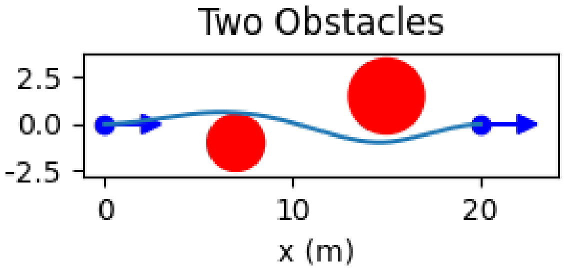
evaluation time: 0.94
path length: 23.18
num ctrl pts: 8



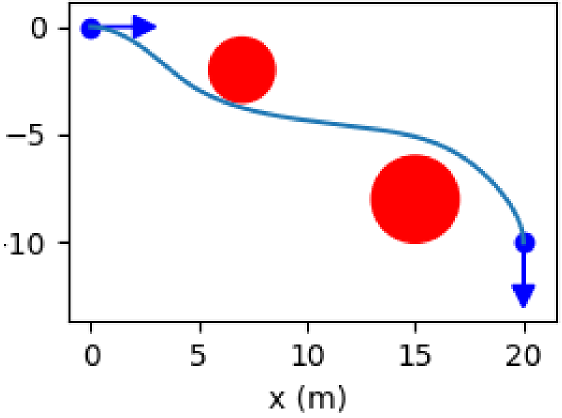
evaluation time: 1.46
path length: 24.73
num ctrl pts: 8

One Obstacle

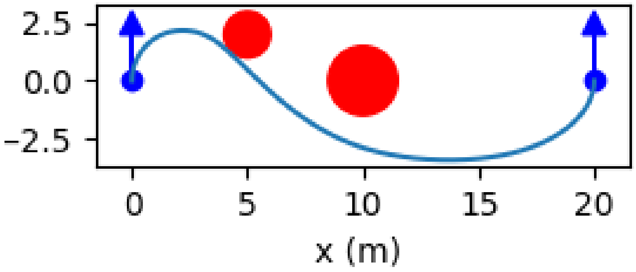
Applications to Planning: Collision Avoidance



evaluation time: 0.65
path length: 20.33
num ctrl pts: 10



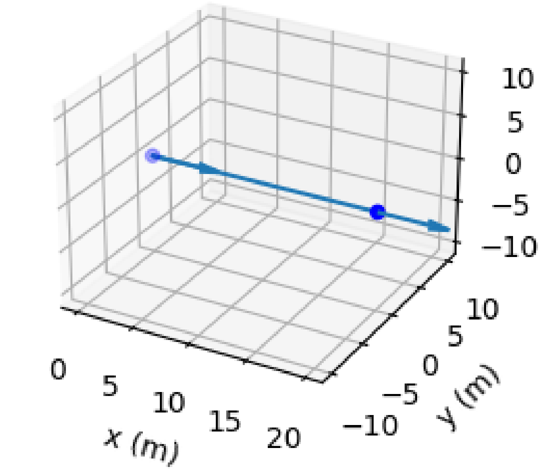
evaluation time: 2.19
path length: 23.59
num ctrl pts: 10



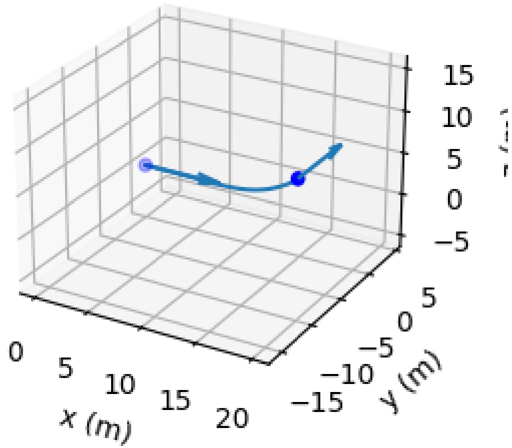
evaluation time: 2.25
path length: 24.52
num ctrl pts: 10

Two Obstacles

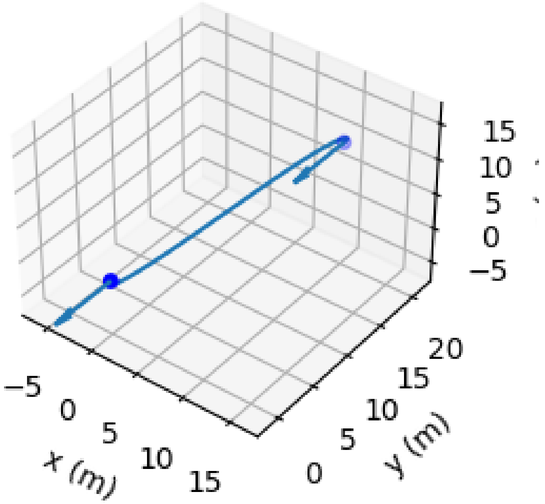

Applications to Planning: Collision Avoidance



evaluation time: 0.17
path length: 20.0
num ctrl pts: 8



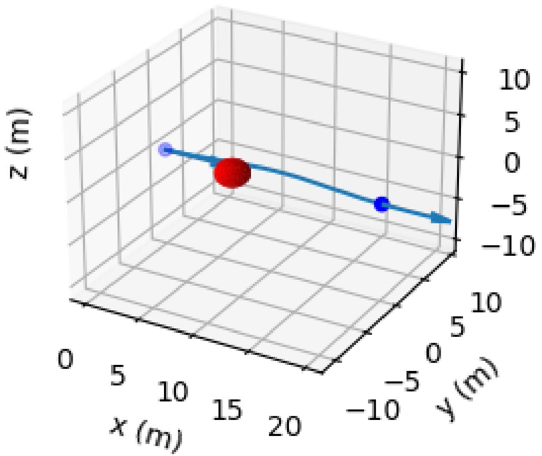
evaluation time: 0.63
path length: 26.25
num ctrl pts: 8



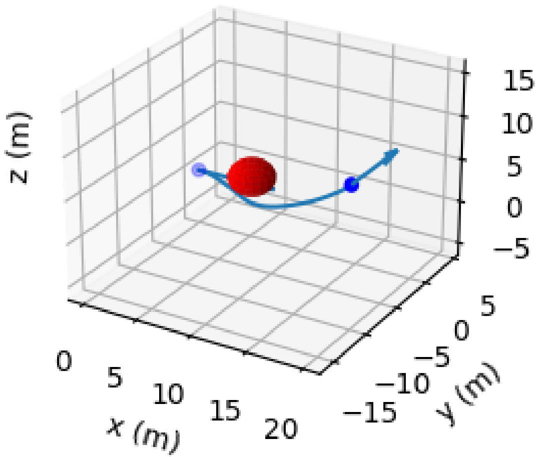
evaluation time: 1.62
path length: 30.52
num ctrl pts: 8

Zero Obstacles

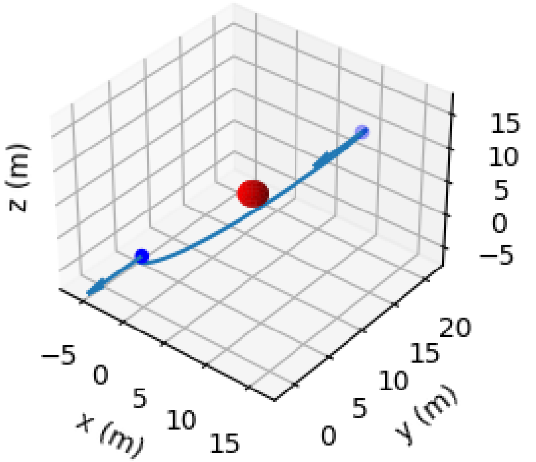
Applications to Planning: Collision Avoidance



evaluation time: 0.54
path length: 20.13
num ctrl pts: 8



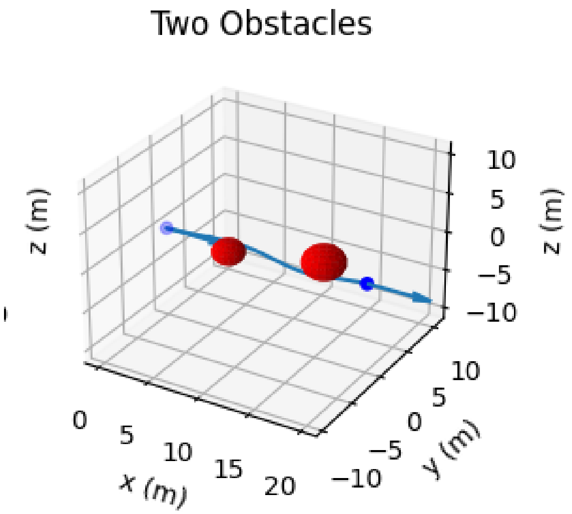
evaluation time: 2.03
path length: 26.73
num ctrl pts: 8



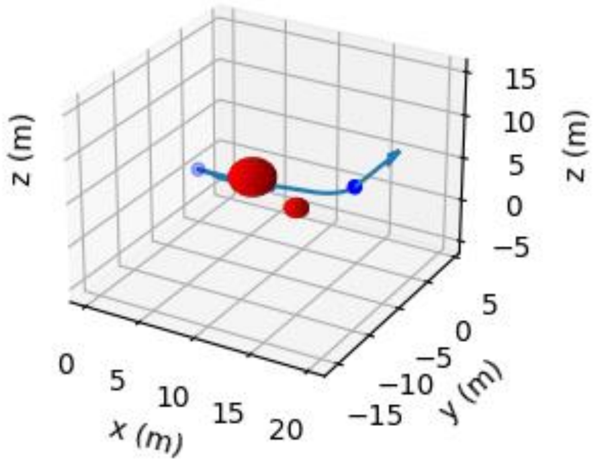
evaluation time: 2.07
path length: 30.72
num ctrl pts: 8

One Obstacle

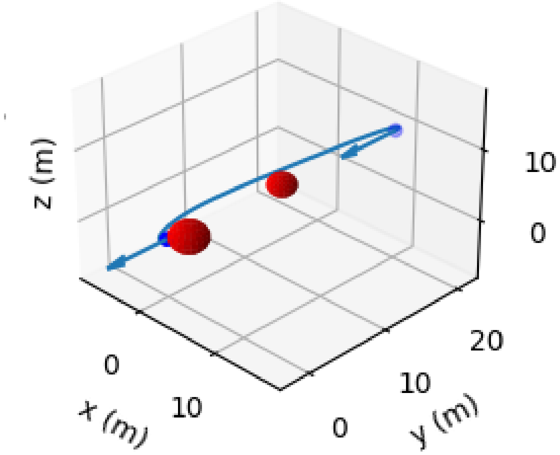
Applications to Planning: Collision Avoidance



evaluation time: 1.31
path length: 20.33
num ctrl pts: 10



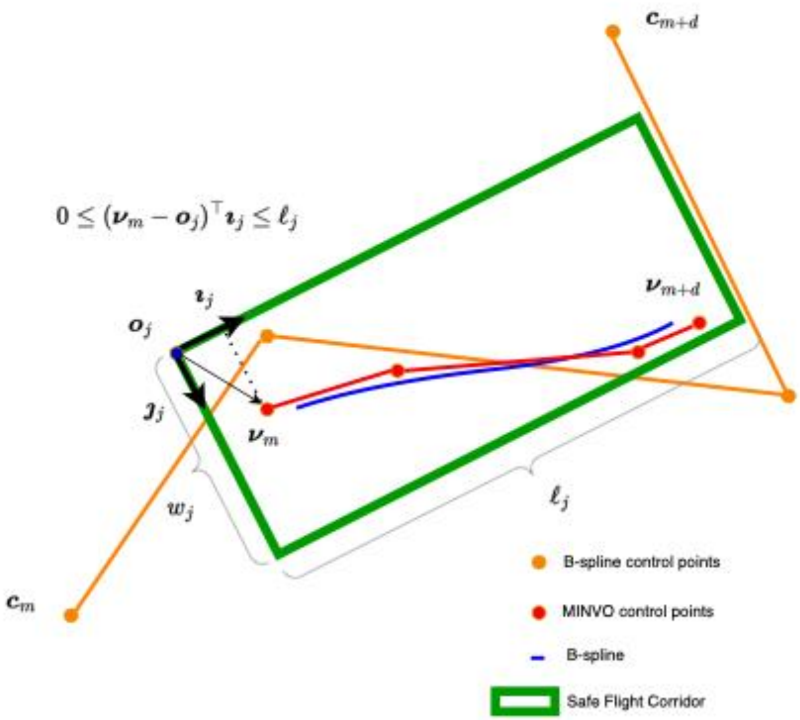
evaluation time: 3.32
path length: 26.27
num ctrl pts: 10



evaluation time: 3.64
path length: 30.13
num ctrl pts: 10

Two Obstacle

Applications to Planning: Safe Flight Corridor



- For the m^{th} spline segment, convert to MINVO control points:

$$\begin{pmatrix} \boldsymbol{\nu}_m & \cdots & \boldsymbol{\nu}_{m+d} \end{pmatrix} = \begin{pmatrix} \boldsymbol{c}_m & \cdots & \boldsymbol{c}_{m+d} \end{pmatrix} F^{\text{Minvo}}_{\text{b-spline}}$$

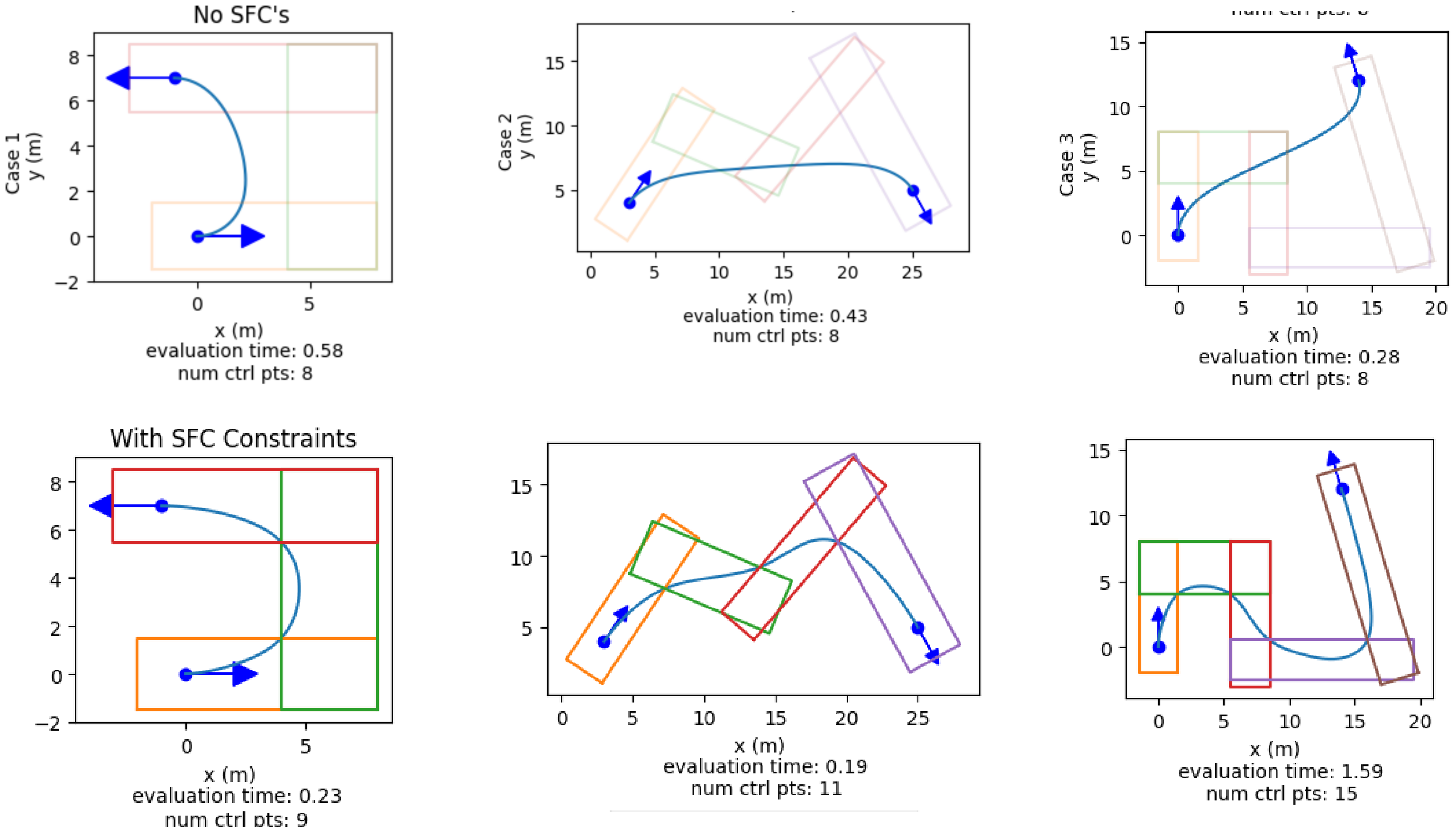
- The j^{th} flight corridor defined by corner $\boldsymbol{o}_j$, direction unit vectors $\boldsymbol{\imath}_j, \boldsymbol{\jmath}_j$, width w_j, and length ℓ_j.

- Corridor constraints

$$0 \leq (\boldsymbol{\nu}_m - \boldsymbol{o}_j)^\top \boldsymbol{\imath}_j \leq \ell_j$$
$$0 \leq (\boldsymbol{\nu}_m - \boldsymbol{o}_j)^\top \boldsymbol{\jmath}_j \leq w_j$$

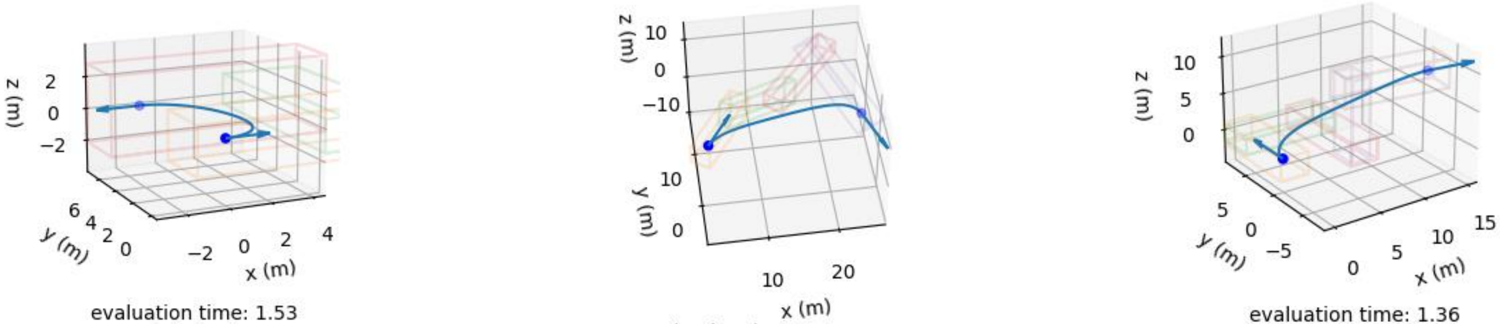
Convex constraint in control points $\boldsymbol{c}_m, \ldots, \boldsymbol{c}_{m+d}$
...and linear!

Applications to Planning: Safe Flight Corridor



Applications to Planning: Safe Flight Corridor

No SFC's

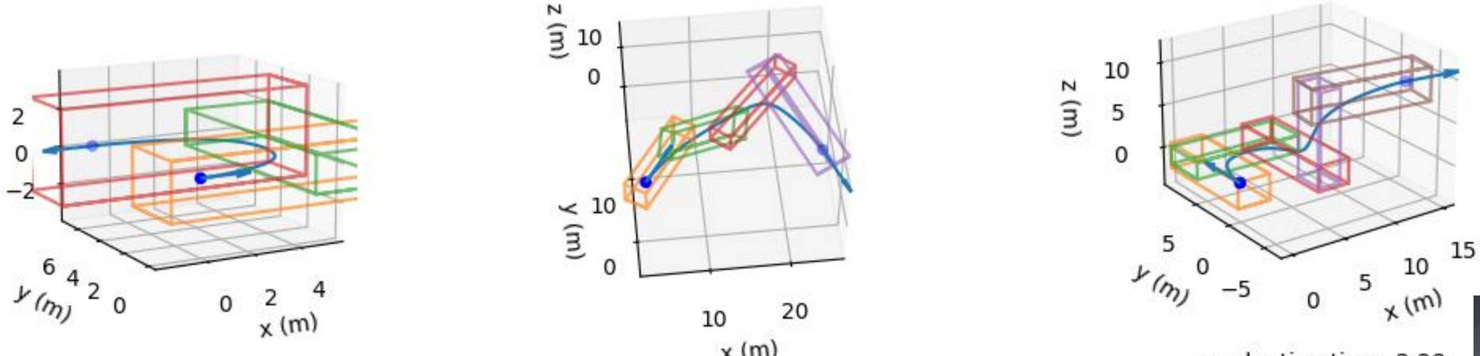


evaluation time: 1.53
num ctrl pts: 8

evaluation time: 0.8
num ctrl pts: 8

evaluation time: 1.36
num ctrl pts: 8

With SFC Constraints

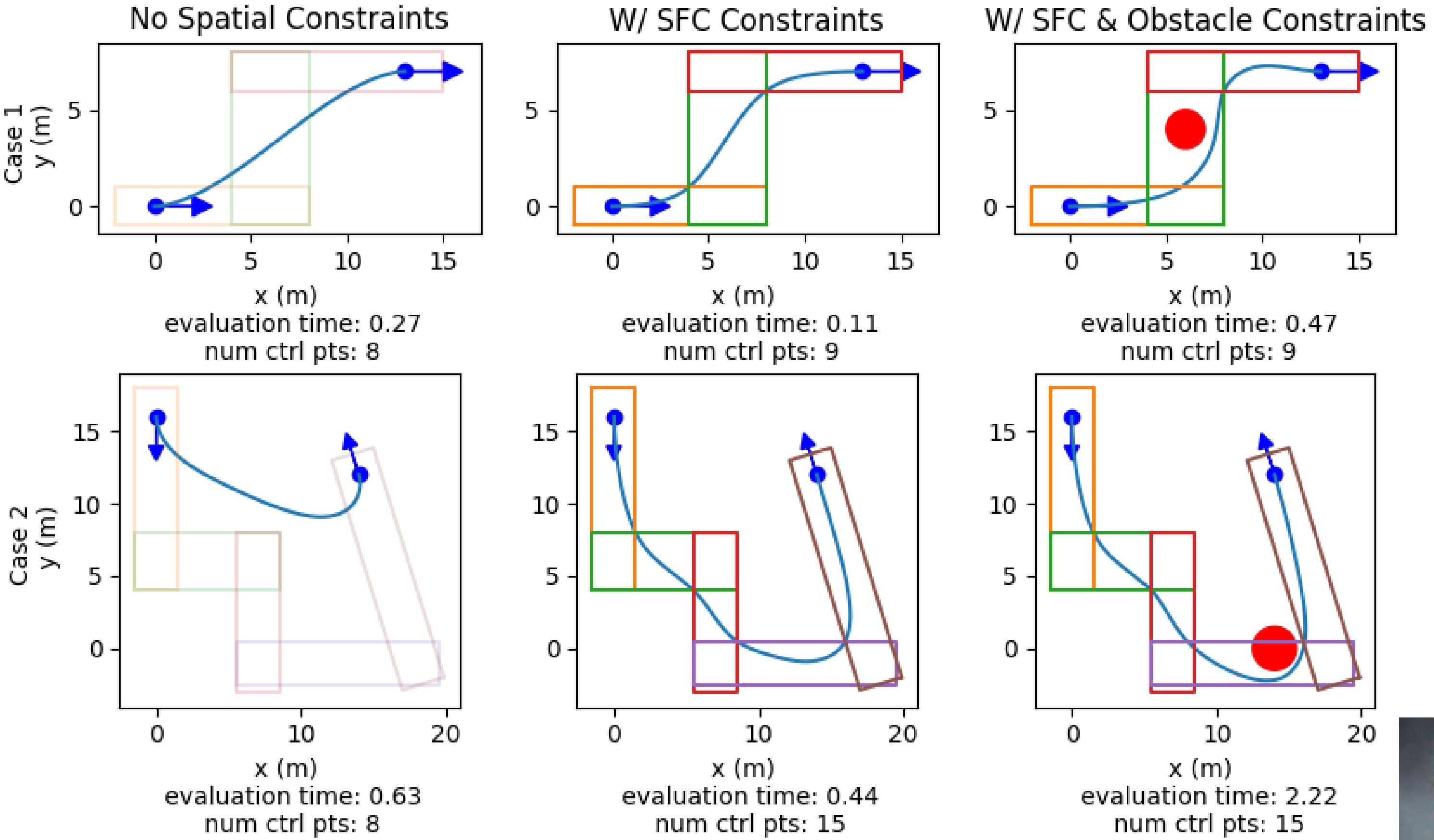


evaluation time: 0.61
num ctrl pts: 9

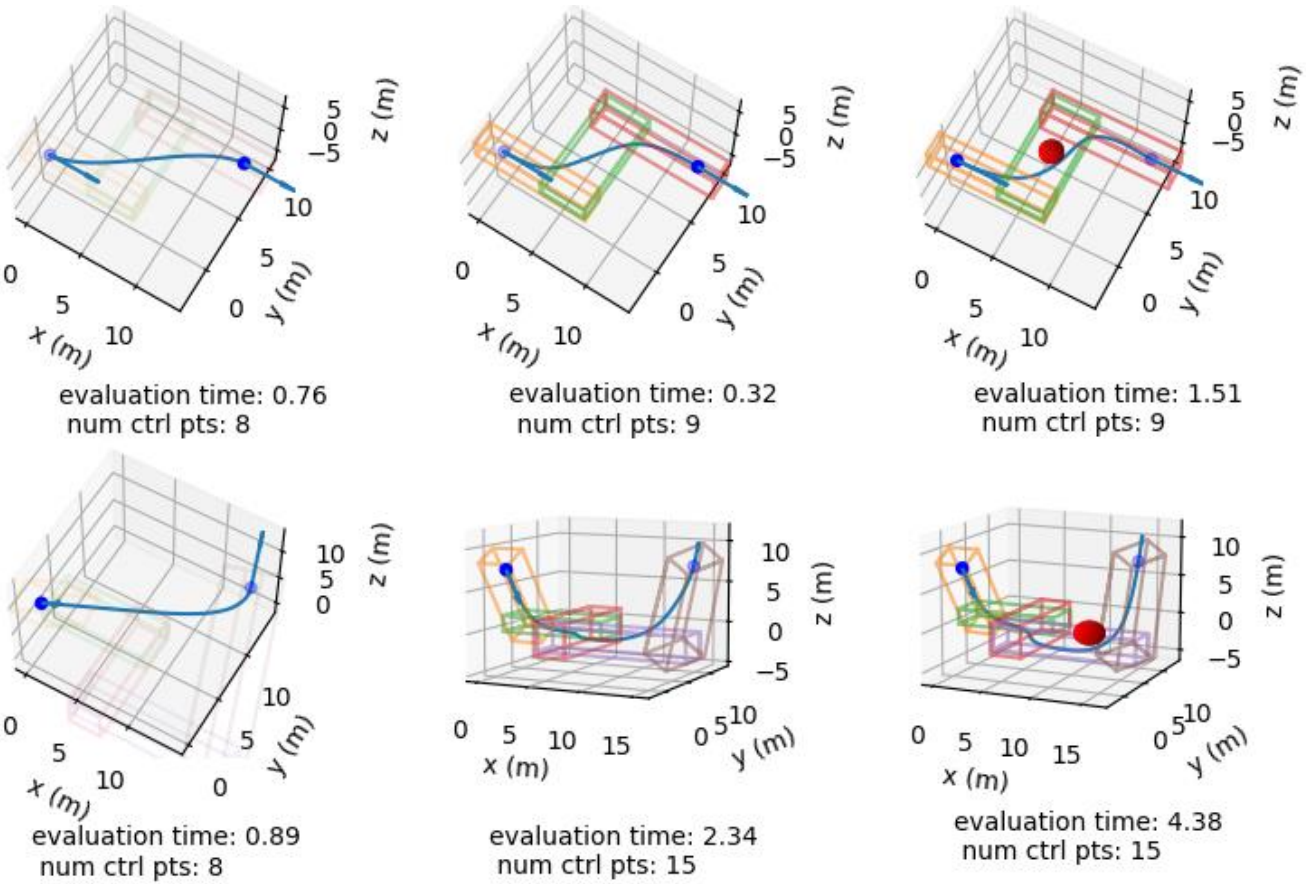
evaluation time: 0.93
num ctrl pts: 13

evaluation time: 2.28
num ctrl pts: 13

Obstacle Avoidance & Safe Flight Corridor



Obstacle Avoidance & Safe Flight Corridor



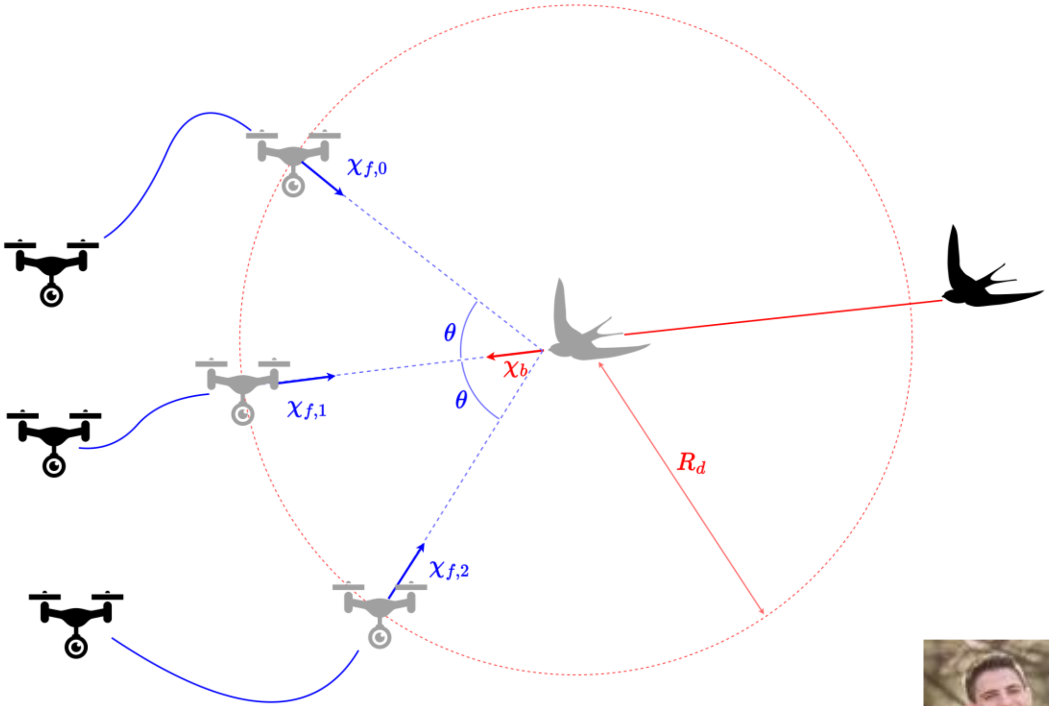
Outline

- Motivation
- B-Spline Tutorial
- Applications to Planning
    - Collision Avoidance
    - Cooperative Temporal-Spatial Sequencing
    - Cooperative Informative Path Planning
- Applications to Estimation:
    - Moving horizon estimation exploiting Differential Flatness
    - Calibration

Applications to Planning: Cooperative Temporal-Spatial Sequencing

Wildlife cinematography:

- Capture video of bird at specified times (e.g. simultaneously) from different angles.
- [Key challenge:] Camera motion has pre-specified, but known, velocity profile (to enhance image quality)



Differentially Flat Systems

Definition: Differential Flatness
The dynamic system

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

is *differentially flat with flat output* $\mathbf{z}$ if there exists function g^x and g^u such that

$$\mathbf{x}(t) = \mathbf{g}^x\left(\mathbf{z}(t), \dot{\mathbf{z}}(t), \ddot{\mathbf{z}}(t), \cdots\right)$$
$$\mathbf{u}(t) = \mathbf{g}^u\left(\mathbf{z}(t), \dot{\mathbf{z}}(t), \ddot{\mathbf{z}}(t), \cdots\right)$$

Examples include:
- Unicycle
- Quadrotor
- Fixed wing Aircraft
  (with reasonable assumptions)
- eVTOL
  (with reasonable assumptions)
- Mobile robot
  (with reasonable assumptions)
- Unmanned surface vehicle
  (with reasonable assumptions)
- …many others…




Informative Path Planning for Differentially Flat Systems

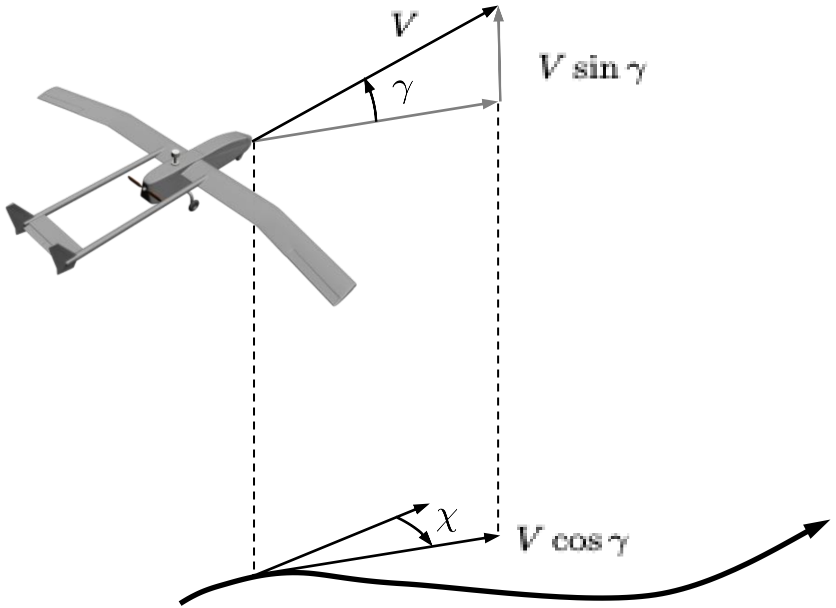
Dubins Airplane

$$\dot{r}_n = V_a \cos\psi$$
$$\dot{r}_e = V_a \sin\psi$$
$$\dot{r}_d = -V_a \sin\gamma$$
$$\dot{\psi} = \frac{g}{V_a}\tan\phi$$
$$\dot{V}_a = \underbrace{s(t)}_{\text{known and pre-defined}}$$

Subject to

$$|\phi| \leq \bar{\phi}$$
$$|\gamma| \leq \bar{\gamma}$$

Without the $\dot{V}_a$ equations, this system is differentially flat with flat output $z = (r_n, r_e, r_d)^\top$.



Flight path projected onto ground

Informative Path Planning for Differentially Flat Systems

Dubins Airplane

$$\dot{r}_n = V_a \cos\psi$$
$$\dot{r}_e = V_a \sin\psi$$
$$\dot{r}_d = -V_a \sin\gamma$$
$$\dot{\psi} = \frac{g}{V_a}\tan\phi$$

Subject to

$$|\phi| \leq \bar{\phi}$$
$$|\gamma| \leq \bar{\gamma}$$

Differentially flat with flat output $\mathbf{p} = (r_n, r_e, r_d)^\top$ since:

$$(r_n, r_e, r_d)^\top(\mathbf{p}) = \mathbf{p}$$
$$V(\dot{\mathbf{p}}) = \sqrt{\dot{p}_n^2 + \dot{p}_e^2 + \dot{p}_d^2}$$
$$\psi(\dot{\mathbf{p}}) = \frac{\dot{p}_e}{\dot{p}_n}$$
$$\gamma(\dot{\mathbf{p}}) = -\sin^{-1}\left(\frac{\dot{p}_d}{\sqrt{\dot{p}_n^2 + \dot{p}_e^2 + \dot{p}_d^2}}\right)$$
$$\phi(\dot{\mathbf{p}}, \ddot{\mathbf{p}}) = \tan^{-1}\left(\frac{(\dot{p}_n\ddot{p}_e - \dot{p}_e\ddot{p}_e)\sqrt{\dot{p}_n^2 + \dot{p}_e^2 + \dot{p}_d^2}}{g(\dot{p}_n^2 + \dot{p}_e^2)}\right)$$

Cooperative Temporal-Spatial Sequencing

B-Spline Approximation of Dubins Paths

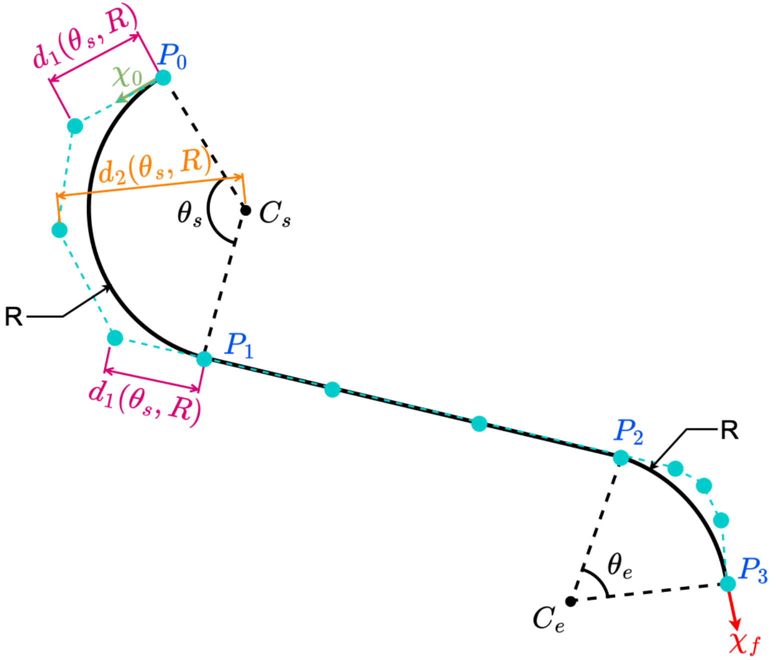
Given a Dubin's path:

- Use degree $d = 3$ (to ensure continuous acceleration), clamped, uniform, B-spline
- Select control points to correspond to P_0, P_1, P_2, P_3 shown in the figure
- Add control points to ensure appropriate specified heading at P_0, P_1, P_2, P_3
- Add addition waypoints on turns according to (empirical fit)

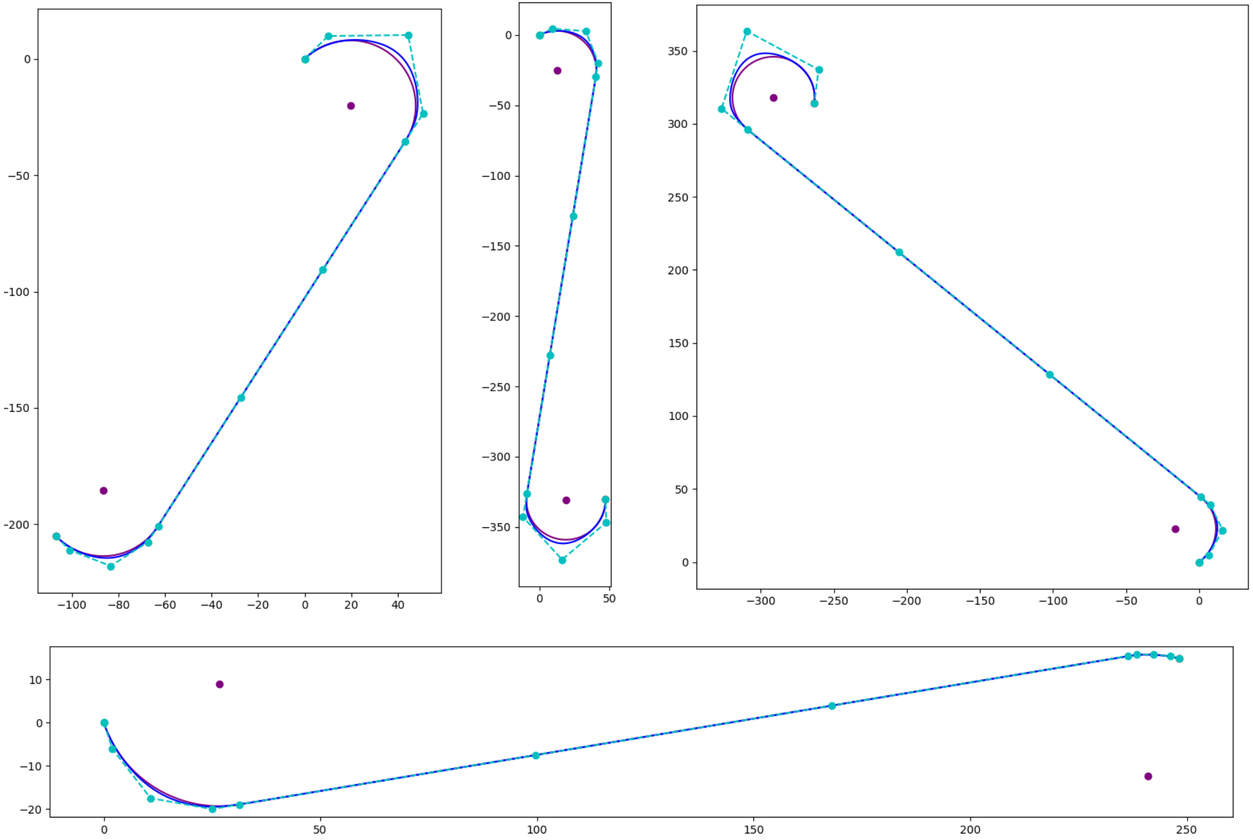
$$d_1 = R(0.004298534\theta_s^3 - 0.010226116\theta_s^2 + 0.163751234\theta_s + 0.00529229)$$

$$d_2 = R(0.002995584\theta_s^4 - 0.031004044\theta_s^3 + 0.142959764\theta_s^2 - 0.097444194\theta_s + 1.01900052)$$

- Add two addition control points along straight-line segment.



B-Spline Approximation of Dubins Paths



Cooperative Temporal-Spatial Sequencing

Path Parameterization for Known Speed Profile

Approximating the Dubin's path results in the B-spline:

$$\mathbf{p}(\sigma) = \sum_{m=0}^{M+d-1} \mathbf{c}_m b_m^d(\sigma, \mathbf{k}),$$

where $d = 3$, and

$$\mathbf{k} = \left[0, 0, 0, 0, \frac{1}{M}, \ldots, \frac{M-1}{M}, 1, 1, 1, 1\right]$$

where we use the path parameter σ instead of t.

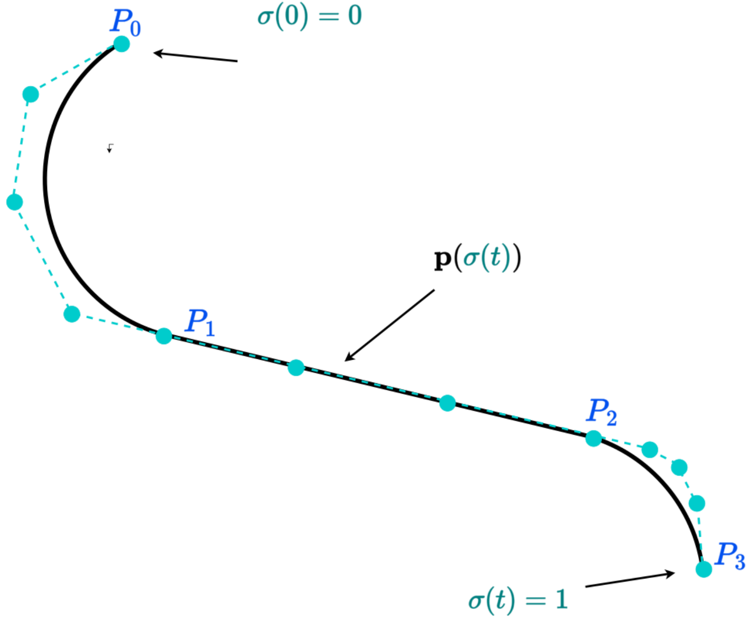
To achieve the desired speed profile, note that

$$s(t) = \|\frac{d}{dt}\mathbf{p}(\sigma(t))\| = \dot{\sigma}\|\frac{d\mathbf{p}}{d\sigma}\|$$

Therefore let

$$\dot{\sigma} = \frac{s(t)}{\|\frac{d\mathbf{p}}{d\sigma}(\sigma(t))\|}, \qquad \sigma(0) = 0$$

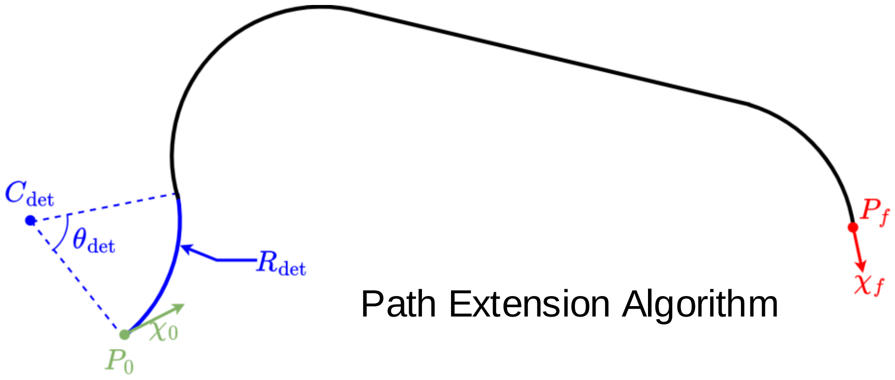
which is integrated until $\sigma(t) = 1$ corresponding to the end of the B-spline path.



Cooperative Temporal-Spatial Sequencing

Cooperation Auction Algorithm

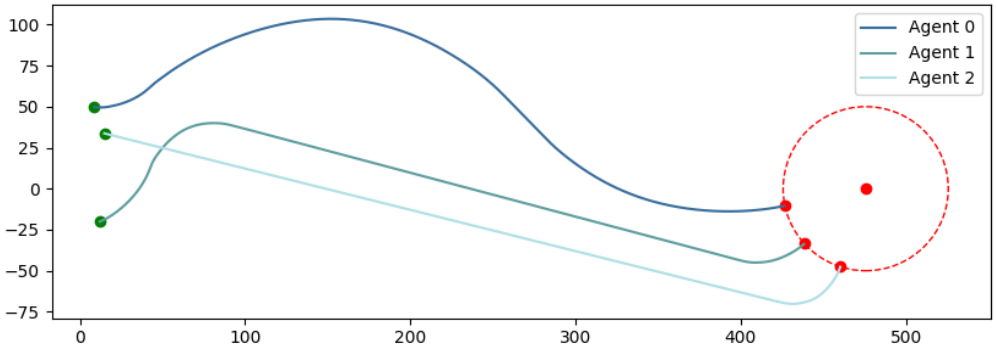
- Each vehicle plans its myopically best path, and submits bid: B-spline parameterization and arrival time
- The vehicle with the longest arrival time $t_{\max}$ is selected
- Remaining vehicles plan myopically best de-conflicted and elongated path matching arrival time $t_{\max}$ an
- Best bid selected, and repeat

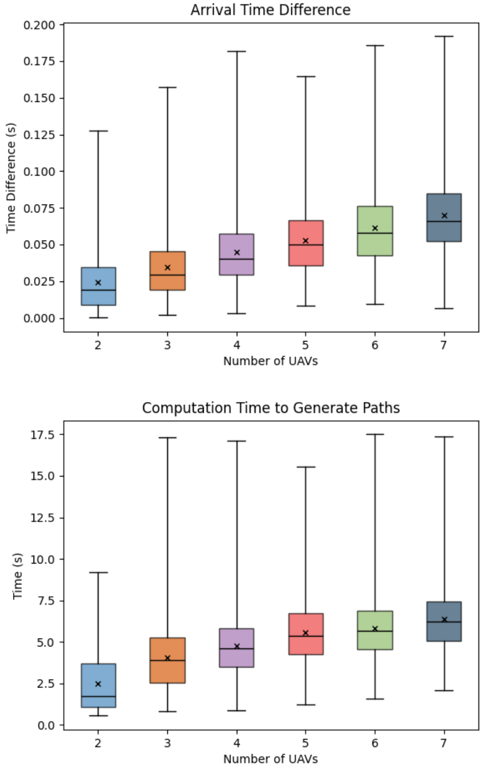


Path Extension Algorithm

Cooperative Temporal-Spatial Sequencing

Monte-Carlo runs:
Each whisker represents 1000 simulations





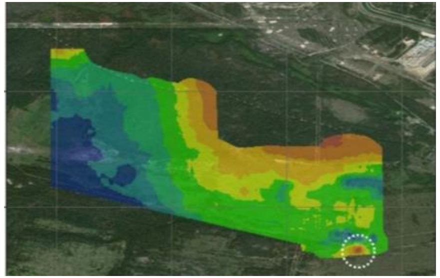
Outline

- Motivation
- B-Spline Tutorial
- Applications to Planning
  - Collision Avoidance
  - Cooperative Temporal-Spatial Sequencing
  - Cooperative Informative Path Planning
- Applications to Estimation:
  - Moving horizon estimation exploiting Differential Flatness
  - Calibration

Applications to Planning: Informative Path Planning

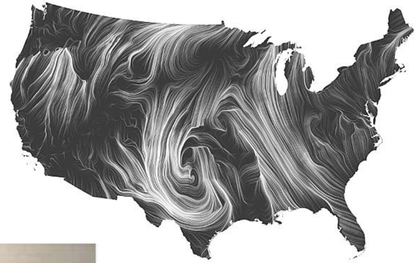
Problem Statement:

- Team of Mobile Agents
- Sensor to measure phenomenon:
- Objective is to create a map of the environment



Radiation

Wind



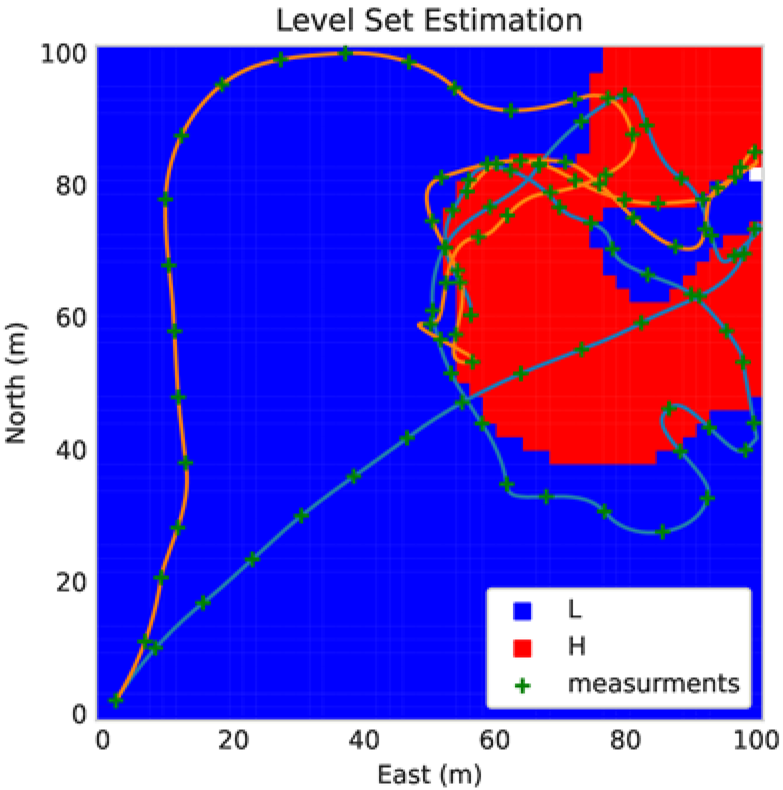
Chemical Spill

Informative Path Planning: Level Set Estimation

- D is the area of interest
- $f(x)$ is an unknown scaler field specifying concentration, where

$$H = \{\mathbf{x} \in D | f(\mathbf{x}) > h\}$$
$$L = \{\mathbf{x} \in D | f(\mathbf{x}) \leq h\}$$

- Sensor measures $z_i = f(\mathbf{x}_i) + \eta, \quad \eta \sim \mathcal{N}(0, \sigma_n^2)$.
- **Goal:** Plan cooperative paths to classify area into high and low sets using noisy measurements of the environment

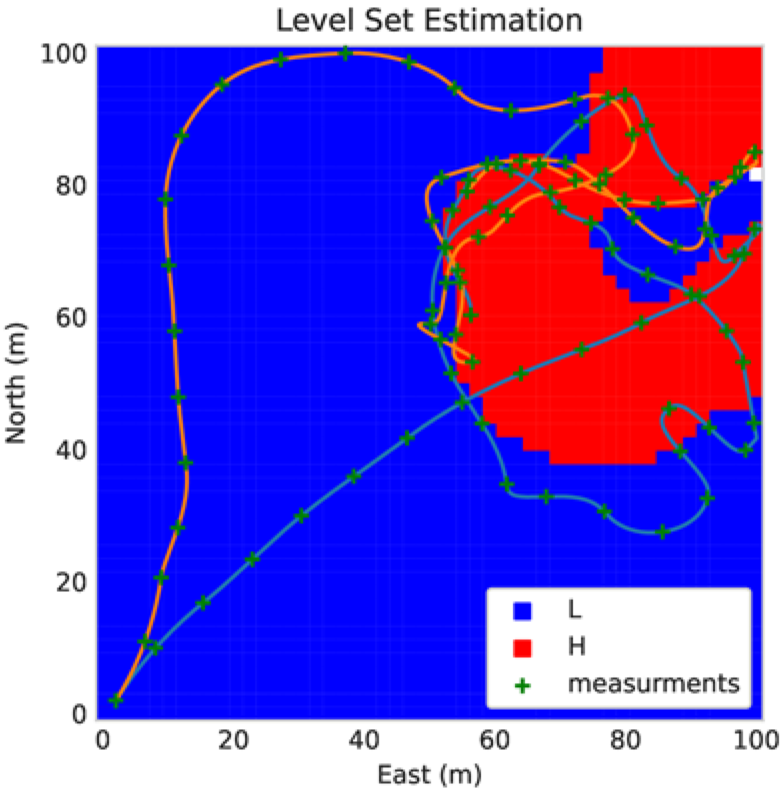


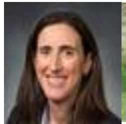
Informative Path Planning: Level Set Estimation

- As the aircraft moves and collects measurement, the scalar field is updated using Gaussian Process Regression:

$$p(f(\bar{X}_*)|z) \sim \mathcal{N}(\mu_p(\bar{X}_*, \mathrm{cov}_p(\bar{X}_*))$$
$$\mu_p(\bar{X}_*) = \mu(\bar{X}_*) + K_{*f}(K_{ff} + \sigma_n^2 I)^{-1}\,(z - \mu(\bar{X}))$$
$$\Sigma_p(\bar{X}_*) = K_{**} - K_{*f}(K_{ff} + \sigma_n^2 I)^{-1} K_{f*}$$

- Cost function for planning algorithm is to maximize the information gain over a receding horizon trajectory.





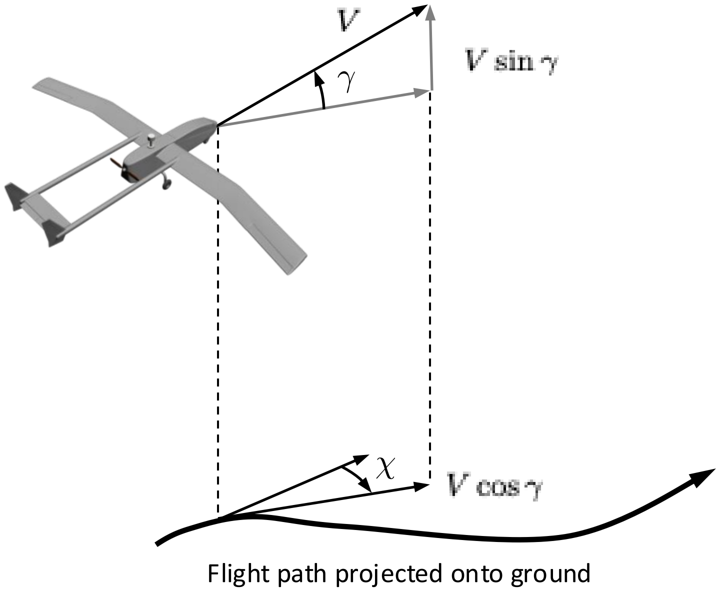
Informative Path Planning for Differentially Flat Systems

Dubins Airplane

$$\dot{r}_n = V_a \cos\psi$$
$$\dot{r}_e = V_a \sin\psi$$
$$\dot{r}_d = -V_a \sin\gamma$$
$$\dot{\psi} = \frac{g}{V_a}\tan\phi$$

Subject to

$$|\phi| \le \bar{\phi}$$
$$|\gamma| \le \bar{\gamma}$$

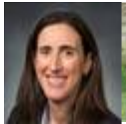


Informative Path Planning: Path Planning

- Differential flatness for feasibility constraints
  - Path specified as $\mathbf{p}$
  - Velocity contraint: $\|\dot{\mathbf{p}}\| \leq \bar{V}$
  - Curvature: $\frac{\|\dot{\mathbf{p}} \times \ddot{\mathbf{p}}\|}{\|\dot{\mathbf{p}}\|^2} \leq \bar{u}$
- Clamped and uniform B-splines to parameterize paths

$$\mathbf{p}(t) = \sum_{m=0}^{M+d-1} \mathbf{c}_m b_m^d(t, \mathbf{k})$$

- Level set estimation specific objective function
- Decentralization using block coordinate ascent




Informative Path Planning:
Path Planning – Optimization

The path optimization problem is therefore:

$$\mathbf{C}^* = \arg\max \sum_{i=1}^{N_c} \Gamma\left(\mathbf{p}(t_c + i\Delta_{tf}, \mathbf{k})\right)$$

subject to:

$$\mathbf{c}_1 = \mathbf{p}_{t_c}$$

$$\mathbf{c}_2 = \mathbf{c}_1 + \left(\frac{t_{k+2} - t_2}{k}\right) v_{t_c} \begin{pmatrix} \cos\theta_{t_c} \\ \sin\theta_{t_c} \end{pmatrix}$$

$$\mathbf{p}(t_i) \in D$$

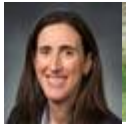
$$\|\dot{\mathbf{p}}(t_i)\| \leq \bar{v}$$

$$\underline{u} \leq \frac{\|\dot{\mathbf{p}}(t_i) \times \ddot{\mathbf{p}}(t_i)\|}{\|\dot{\mathbf{p}}(t_i)\|^2} \leq \bar{u}$$

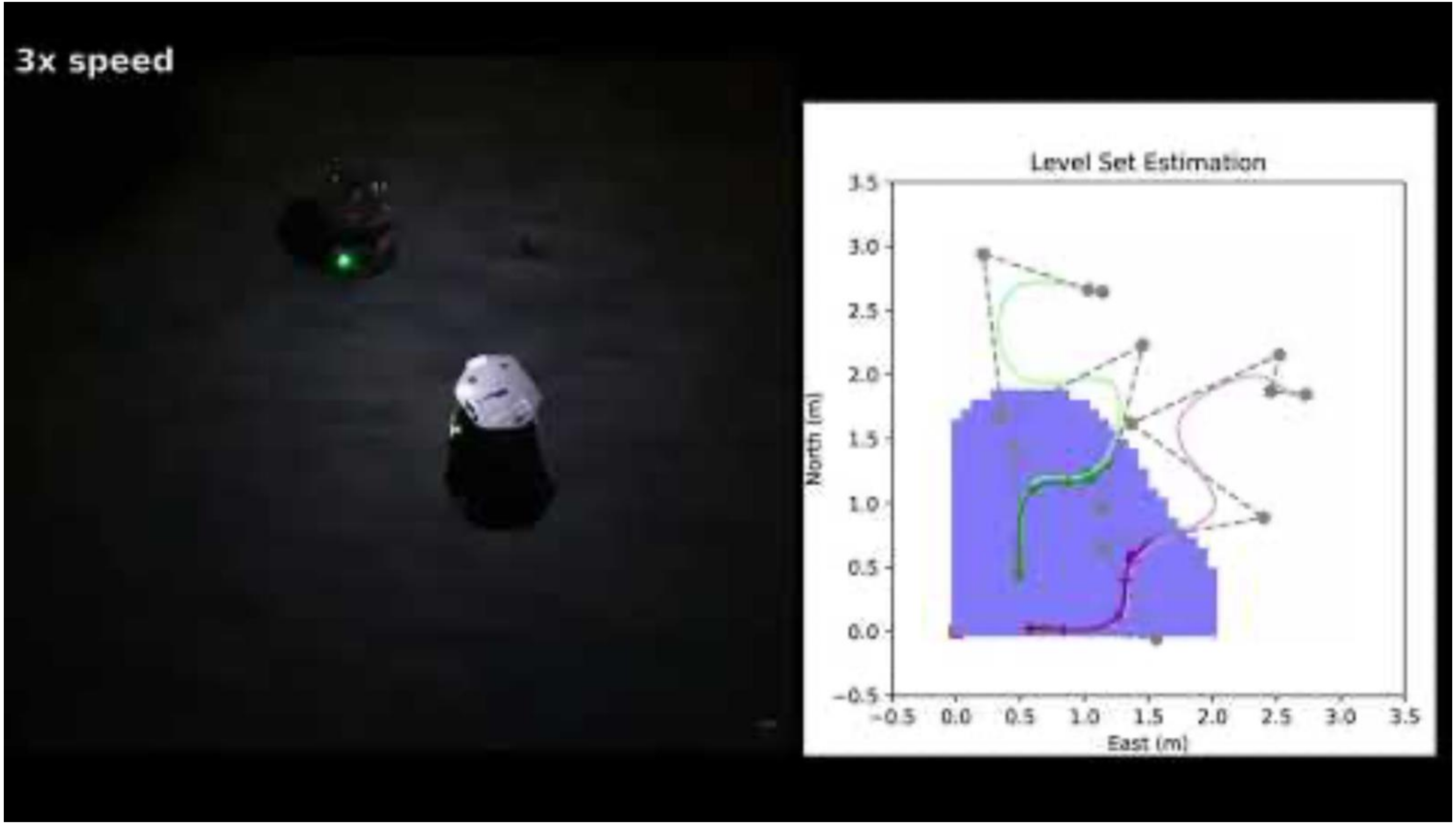
where

$$\Gamma(\mathbf{x}) = \alpha\|\Sigma_{t_c}(\mathbf{x})\| + (1-\alpha)(h - \mu_{t_c}(\mathbf{x}))^2$$

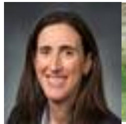
if the information gain at $\boldsymbol{x}$.



Informative Path Planning: Results



https://youtu.be/7jVc9QdYnJs?si=Y1jqu6_lnP3Hdizx&t=20




Outline

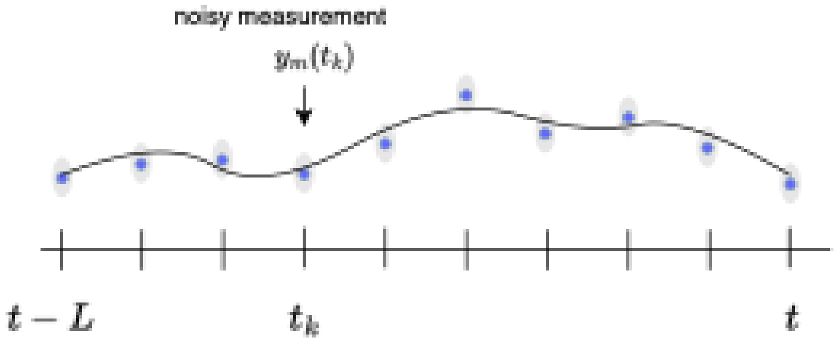
- Motivation
- B-Spline Tutorial
- Applications to Planning
    - Collision Avoidance
    - Cooperative Temporal-Spatial Sequencing
    - Cooperative Informative Path Planning
- Applications to Estimation:
    - Moving horizon estimation exploiting Differential Flatness
    - Calibration

Moving Horizon Estimation (MHE) using B-splines

Given the nonlinear system

$$\dot{x} = f(x, u, w), \qquad x(t-L)$$
$$y = h(x, u) + \xi$$

where L is the window length, and the measurement noise satisfies $\xi \sim \mathcal{N}(0, \Sigma)$
The signal $w(t)$ can be thought of as an unknown noise and/or disturbance term, but with no Gaussian restrictions.



Represent $w(t) = \sum_{m=0}^{M+d-1} \mathbf{c}_m b_m^d(t, \mathbf{k}_M^d)$
and solve differential equation to obtain $x(t, \mathbf{C}, x(t-L))$.

Solve:

$$\mathbf{C}^*, x^*(t-L) = \arg\min \sum \| h(x(t_k, \mathbf{C}, x(t-L)), u(t_k)) - y_m(t_k) \|_{\Sigma^{-1}},$$

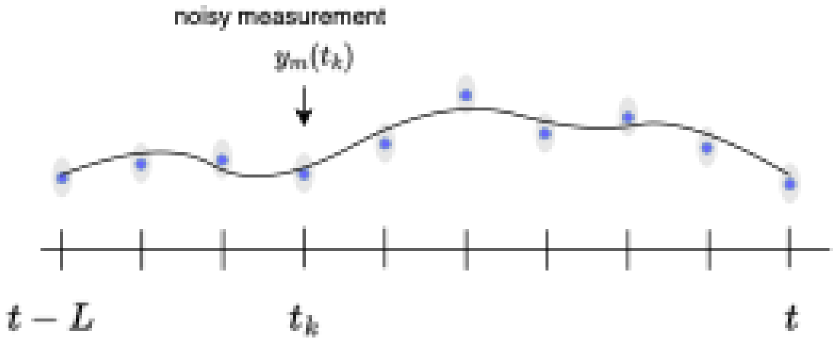
where $y_m(t_k)$ is a measurement at time $t - L \leq t_k \leq t$, and the sum is over all measurements in the window.

Moving Horizon Estimation (MHE) using B-splines

Given the nonlinear system

$$\dot{x} = f(x, u, w), \qquad x(t-L)$$
$$y = h(x, u) + \xi$$

where L is the window length, and the measurement noise satisfies $\xi \sim \mathcal{N}(0, \Sigma)$
The signal $w(t)$ can be thought of as an unknown noise and/or disturbance term, but with no Gaussian restrictions.



Represent $w(t) = \sum_{m=0}^{M+d-1} \mathbf{c}_m b_m^d(t, \mathbf{k}_M^d)$
and solve differential equation to obtain $x(t, \mathbf{C}, x(t-L))$.

Solve:

$$\mathbf{C}^*, x^*(t-L) = \arg\min \sum \| h(x(t_k, \mathbf{C}, x(t-L)), u(t_k)) - y_m(t_k) \|_{\Sigma^{-1}},$$

where $y_m(t_k)$ is a measurement at time $t - L \leq t_k \leq t$, and the sum is over all measurements in the window.

Note optimization over spline coefficients $\mathbf{C}$ and initial condition $x(t-L)$, and that each iteration requires the solution of a differential equation.

Moving Horizon Estimation (MHE) using B-splines

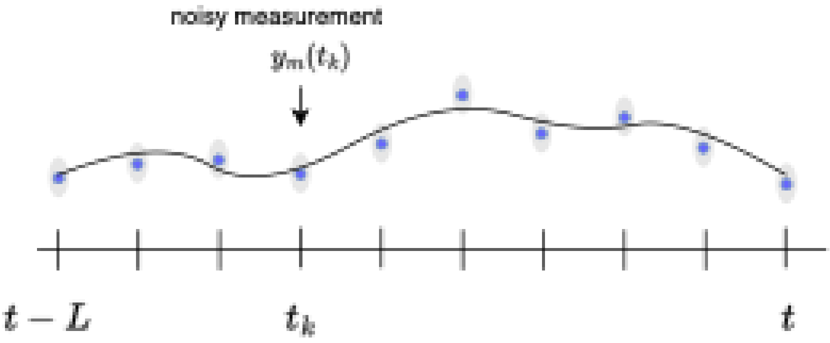
Given the nonlinear system

$$\dot{x} = f(x, u, w), \qquad x(t-L)$$
$$y = h(x, u) + \xi$$

where L is the window length, and the measurement noise satisfies $\xi \sim \mathcal{N}(0, \Sigma)$
The signal $w(t)$ can be thought of as an unknown noise and/or disturbance term, but with no Gaussian restrictions.

B-spline estimates the input disturbance:

Represent $w(t) = \sum_{m=0}^{M+d-1} \mathbf{c}_m b_m^d(t, \mathbf{k}_M^d)$
and solve differential equation to obtain $x(t, \mathbf{C}, x(t-L))$.



Solve:

$$\boxed{\mathbf{C}^*, x^*(t-L)} = \arg\min \sum \| h(x(t_k, \mathbf{C}, x(t-L)), u(t_k)) - y_m(t_k) \|_{\Sigma^{-1}},$$

where $y_m(t_k)$ is a measurement at time $t - L \leq t_k \leq t$, and the sum is over all measurements in the window.

Note optimization over spline coefficients $\mathbf{C}$ and initial condition $x(t-L)$, and that each iteration requires the solution of a differential equation.

Differentially Flat Systems

Definition: Differential Flatness
The dynamic system

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

is *differentially flat with flat output* $\mathbf{z}$ if there exists function g^x and g^u such that

$$\mathbf{x}(t) = \mathbf{g}^x\left(\mathbf{z}(t), \dot{\mathbf{z}}(t), \ddot{\mathbf{z}}(t), \cdots\right)$$
$$\mathbf{u}(t) = \mathbf{g}^u\left(\mathbf{z}(t), \dot{\mathbf{z}}(t), \ddot{\mathbf{z}}(t), \cdots\right)$$

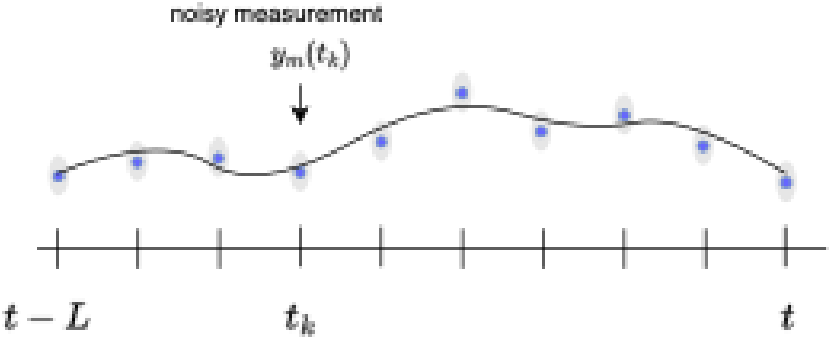
Examples include:
- Unicycle
- Quadrotor
- Fixed wing Aircraft
(with reasonable assumptions)
- eVTOL
(with reasonable assumptions)
- Mobile robot
(with reasonable assumptions)
- Unmanned surface vehicle
(with reasonable assumptions)
- …many others…




Moving Horizon Estimation (MHE) using DF + B-splines

Given the differentially flat nonlinear system with flat output $\mathbf{z}$:

$$\begin{aligned} \dot{x} &= f(x, u, w), \qquad x(t-L) \\ y &= h(x, u) + \xi \end{aligned}$$



Represent the flat output using $\hat{z}(t, \mathbf{C}) = \sum_{m=0}^{M+d-1} \mathbf{c}_m b_m^d(t, \mathbf{k}_M^d)$

giving $\hat{x}(t, \mathbf{C})$ and $\hat{w}(t, \mathbf{C})$

Solve:

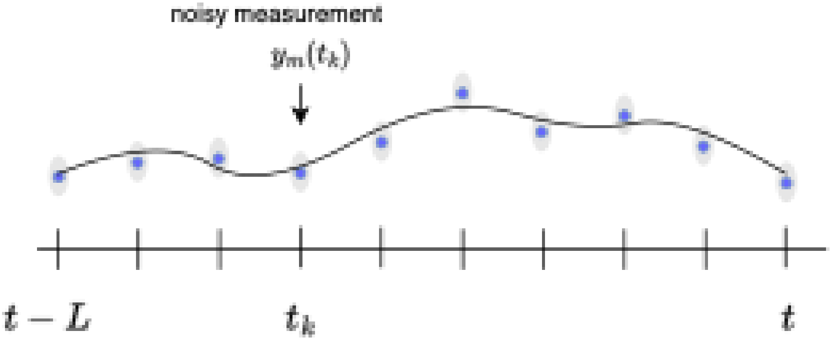
$$\mathbf{C}^* = \arg\min \sum \|h\left(\hat{x}(t_k, \mathbf{C}), u(t_k)\right) - y_m(t_k)\|_{\Sigma^{-1}},$$

where $y_m(t_k)$ is a measurement at time $t - L \leq t_k \leq t$, and the sum is over all measurements in the window.

Moving Horizon Estimation (MHE) using DF + B-splines

Given the differentially flat nonlinear system with flat output $\mathbf{z}$:

$$
\begin{aligned}
\dot{x} &= f(x, u, w), \qquad x(t-L) \\
y &= h(x, u) + \xi
\end{aligned}
$$



Represent the flat output using $\hat{z}(t, \mathbf{C}) = \sum_{m=0}^{M+d-1} \mathbf{c}_m b_m^d(t, \mathbf{k}_M^d)$

giving $\hat{x}(t, \mathbf{C})$ and $\hat{w}(t, \mathbf{C})$

Solve:

$$
\mathbf{C}^* = \arg\min \sum \| h\left(\hat{x}(t_k, \mathbf{C}), u(t_k)\right) - y_m(t_k) \|_{\Sigma^{-1}},
$$

where $y_m(t_k)$ is a measurement at time $t - L \leq t_k \leq t$, and the sum is over all measurements in the window.

Note: No initial condition, & do not solve differential equation

Moving Horizon Estimation (MHE) using DF + B-splines

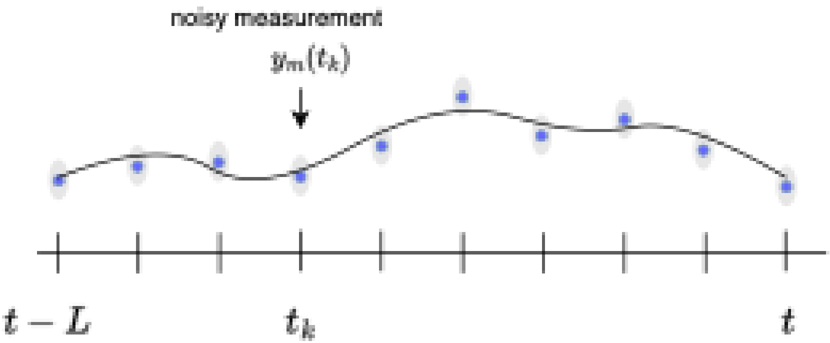
Given the differentially flat nonlinear system with flat output $\mathbf{z}$:

$$\dot{x} = f(x, u, w), \qquad x(t - L)$$
$$y = h(x, u) + \xi$$

Represent the flat output using $\hat{z}(t, \mathbf{C}) = \sum_{m=0}^{M+d-1} \mathbf{c}_m b_m^d(t, \mathbf{k}_M^d)$

giving $\hat{x}(t, \mathbf{C})$ and $\hat{w}(t, \mathbf{C})$

B-spline estimates output trajectory:



Solve:

$$\mathbf{C}^* = \arg\min \sum \left\| h\left(\hat{x}(t_k, \mathbf{C}), u(t_k)\right) - y_m(t_k) \right\|_{\Sigma^{-1}},$$

where $y_m(t_k)$ is a measurement at time $t - L \leq t_k \leq t$, and the sum is over all measurements in the window.

Note: No initial condition, & do not solve differential equation

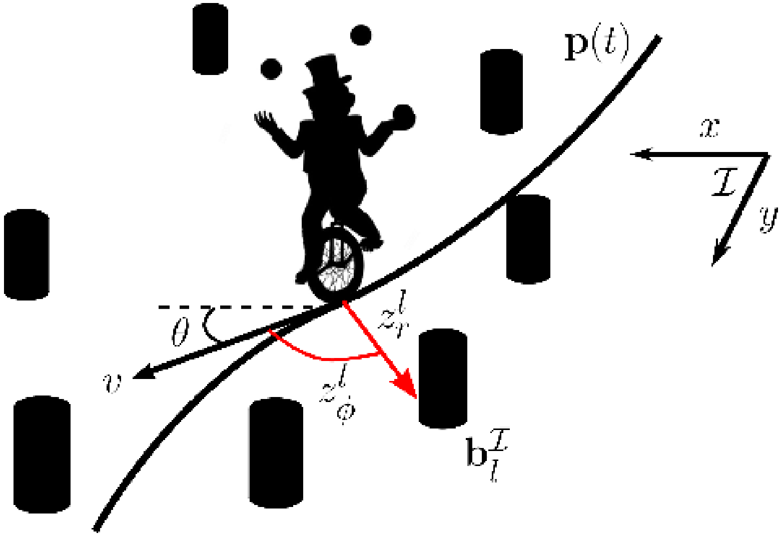
Continuous-time Trajectory Estimation for Differentially Flat Systems

Unicycle Dynamic Model

$$
\begin{aligned}
\dot{\mathbf{T}}^{\mathcal{I}}_{\mathcal{B}} &= \mathbf{T}^{\mathcal{I}}_{\mathcal{B}}\left(\begin{bmatrix} v & 0 & \omega \end{bmatrix}^\top\right)^\wedge \\
\dot{v} &= \frac{1}{m}(F - d_v v + \eta_F), \\
\dot{\omega} &= \frac{1}{J}(\tau - d_\omega \omega + \eta_\tau),
\end{aligned}
$$

where

$$
\mathbf{T}^{\mathcal{I}}_{\mathcal{B}} = \begin{bmatrix} \mathbf{R}^{\mathcal{I}}_{\mathcal{B}} & \mathbf{t}^{\mathcal{I}}_{\mathcal{B}/\mathcal{I}} \\ \mathbf{0} & 1 \end{bmatrix} \in SE(2)
$$



Continuous-time Trajectory Estimation for Differentially Flat Systems

Unicycle Dynamic Model

$$\dot{\mathbf{T}}^{\mathcal{I}}_{\mathcal{B}} = \mathbf{T}^{\mathcal{I}}_{\mathcal{B}} \left(\begin{bmatrix} v & 0 & \omega \end{bmatrix}^{\top} \right)^{\wedge}$$

$$\dot{v} = \frac{1}{m}(F - d_v v + \eta_F),$$

$$\dot{\omega} = \frac{1}{J}(\tau - d_\omega \omega + \eta_\tau),$$

where

$$\mathbf{T}^{\mathcal{I}}_{\mathcal{B}} = \begin{bmatrix} \mathbf{R}^{\mathcal{I}}_{\mathcal{B}} & \mathbf{t}^{\mathcal{I}}_{\mathcal{B}/\mathcal{I}} \\ \mathbf{0} & 1 \end{bmatrix} \in SE(2)$$

Differentially flat with flat output $\mathbf{t}^{\mathcal{I}}_{\mathcal{B}/\mathcal{I}}$ since:

$$\mathbf{T}^{\mathcal{I}}_{\mathcal{B}}(\mathbf{p}, \dot{\mathbf{p}}) = \begin{bmatrix} \mathrm{Exp}\left(\tan^{-1}\left(\frac{\dot{p}_y}{\dot{p}_x}\right)\right) & \mathbf{p} \\ \mathbf{0} & 1 \end{bmatrix}$$

$$v(\dot{\mathbf{p}}) = \|\dot{\mathbf{p}}\|,$$

$$\omega(\dot{\mathbf{p}}, \ddot{\mathbf{p}}) = -\frac{1}{\|\dot{\mathbf{p}}\|^2} \dot{\mathbf{p}}^{\top} 1^{\wedge} \ddot{\mathbf{p}}$$

$$\dot{v}(\dot{\mathbf{p}}, \ddot{\mathbf{p}}) = \frac{1}{\|\dot{\mathbf{p}}\|} \dot{\mathbf{p}}^{\top} \ddot{\mathbf{p}}$$

$$\dot{\omega}(\dot{\mathbf{p}}, \ddot{\mathbf{p}}\, \dddot{\mathbf{p}}) = \frac{2}{\|\dot{\mathbf{p}}\|^4} \left(\dot{\mathbf{p}}^{\top} \ddot{\mathbf{p}}\right) \left(\dot{\mathbf{p}}^{\top} 1^{\wedge} \ddot{\mathbf{p}}\right) - \frac{1}{\|\dot{\mathbf{p}}\|^2} \dot{\mathbf{p}}^{\top} 1^{\wedge} \dddot{\mathbf{p}}$$

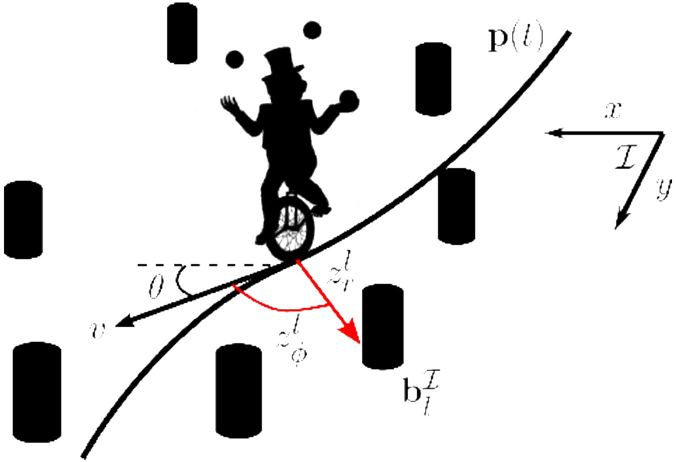
$$F(\dot{\mathbf{p}}, \ddot{\mathbf{p}}) = m\dot{v}(\dot{\mathbf{p}}, \ddot{\mathbf{p}}) + d_v v(\dot{\mathbf{p}})$$

$$\tau(\dot{\mathbf{p}}, \ddot{\mathbf{p}}, \dddot{\mathbf{p}}) = J\dot{\omega}(\dot{\mathbf{p}}, \ddot{\mathbf{p}}, \dddot{\mathbf{p}}) + d_\omega \omega(\dot{\mathbf{p}}, \ddot{\mathbf{p}}).$$

Continuous-time Trajectory Estimation for Differentially Flat Systems

Unicycle Measurement Model:

$$
\begin{aligned}
&\text{Gyroscope:} && z_g = \omega + \eta_g \\
&\text{Accelerometer:} && \mathbf{z}_a = \begin{pmatrix} \frac{1}{m}F - d_v v \\ 0 \end{pmatrix} + \boldsymbol{\eta}_a \\
&\text{Range:} && z_r^l = \|\mathbf{b}_l^B\| + \eta_r \\
&\text{Bearing:} && z_\phi^l = \tan^{-1}\left(\frac{b_{l_y}}{b_{l_x}}\right) + \eta_\phi \\
&\text{Known input force:} && z_F = F + \eta_F \\
&\text{Known input torque:} && z_\tau = \tau + \eta_\tau
\end{aligned}
$$



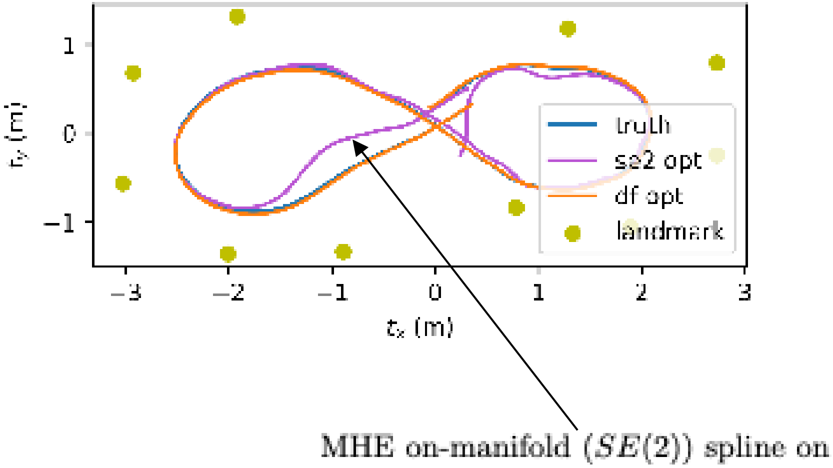
Continuous-time Trajectory Estimation for Differentially Flat Systems: Experimental Results

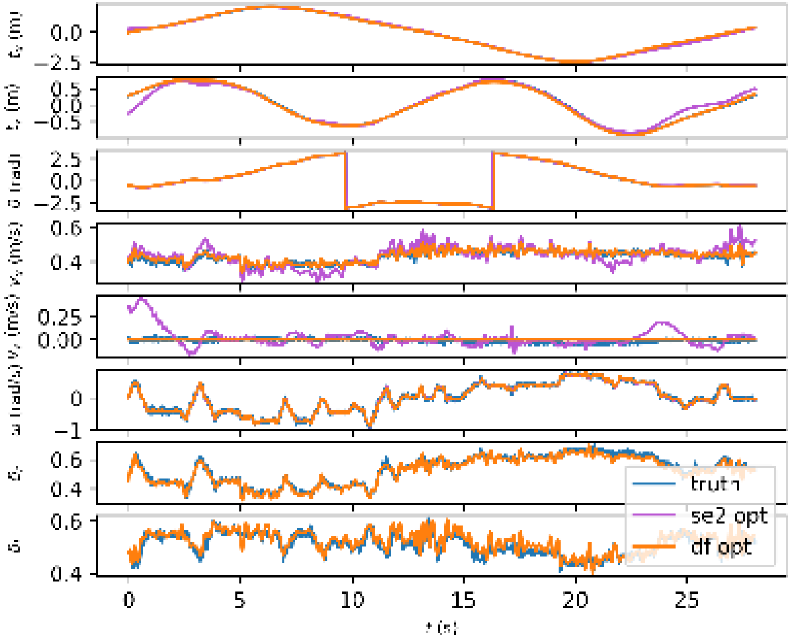
Turtlebot, with Intel RealSense D435i camera
(all measurements produced by RealSense)

MoCap room for truth data

Landmarks identified by ArUco markers

Continuous-time Trajectory Estimation for Differentially Flat Systems: Experimental Results





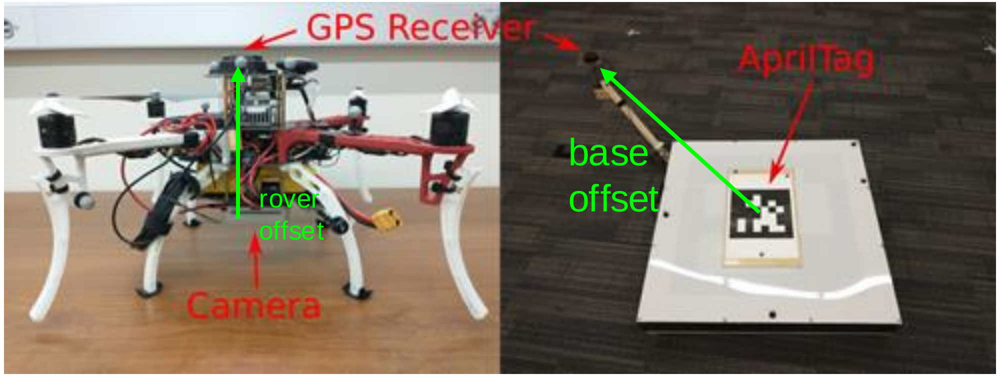
Spline Est. Type	Pose RMSE	Velocity RMSE	Input RMSE	Solve Time (s)	Iterations	Avg. Time per Iteration (s)
DF with IMU	0.07914	0.03440	0.11363	**0.03950**	10.20	**0.00418**
DF with input	0.07934	0.03292	0.11366	0.06195	14.90	0.00433
DF with IMU and input	**0.07801**	**0.02774**	**0.11283**	0.07557	13.44	0.00597
$SE(2)$ with IMU	0.07945	0.03852	NA	0.12425	**5.98**	0.02100
$SO(2) \times \mathbb{R}^2$ with IMU	0.08090	0.03839	NA	0.07891	8.06	0.01009

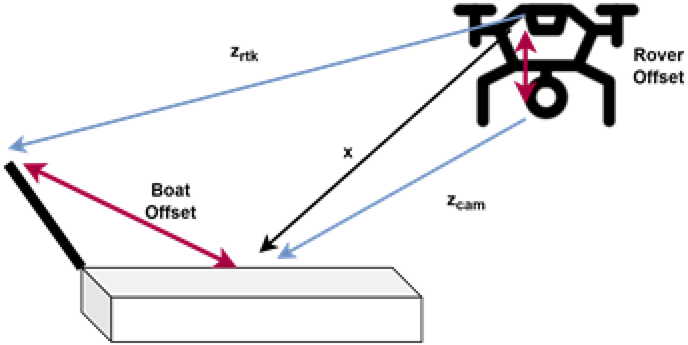
Outline

- Motivation
- B-Spline Tutorial
- Applications to Planning
  - Collision Avoidance
  - Cooperative Temporal-Spatial Sequencing
  - Cooperative Informative Path Planning
- Applications to Estimation:
  - Moving horizon estimation exploiting Differential Flatness
  - Calibration

GNSS/Camera Extrinsic Calibration Using Splines on SE(3)

- State estimation by fusing GNSS and Camera
  - Precision Landing
  - Target geolocation
- Constant offsets needed for measurement models in online estimation
- Offsets not easily measured by hand

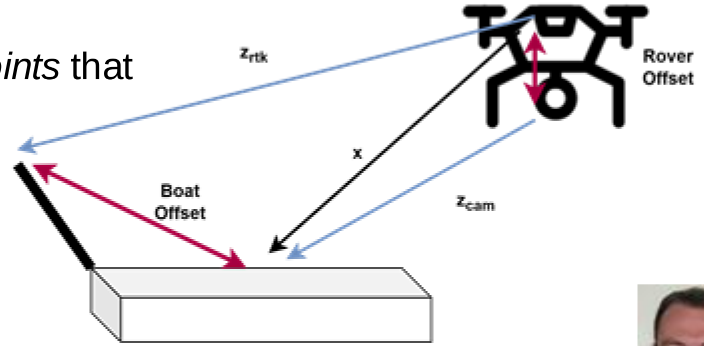




GNSS/Camera Extrinsic Calibration Using Splines on SE(3)

Overview of Method

- Offline full-batch calibration of sensor offsets
- Simultaneously estimate offsets and vehicle trajectory
    - Trajectory represented as a spline on SE(3)
- Result is a Maximum Likelihood estimation problem
    - Given a batch of measurements…
    - Choose the optimal *parameters* and *spline control points* that maximize the likelihood of the measurements.



B-Splines: (differential parameterization)

Splines on $\mathbb{R}^n$:
For any time $t \in [t_s, t_{s+1}]$, the spline

$$\mathbf{p}(t) = \sum_{m=0}^{M+d-1} \boldsymbol{c}_m b_m^d(t, \mathbf{k}_M^d) = \sum_{m=s}^{s+d} \boldsymbol{c}_m b_m^d(t, \mathbf{k}_M^d) = b_s^d \boldsymbol{c}_s + b_{s+1}^d \boldsymbol{c}_{s+1} + \cdots + b_{s+d-1}^d \boldsymbol{c}_{s+d-1} + b_{s+d}^d \boldsymbol{c}_{s+d}$$

because $\sum_{m=s}^{s+d} b_m^d(t) = 1$,

$$\begin{aligned}\mathbf{p}(t) &= \left(1 - (1 - \sum_{m=s}^{s} b_m^d)\right) \boldsymbol{c}_s + \left(1 - \sum_{m=s}^{s} b_m^d - (1 - \sum_{m=s}^{s+1} b_m^d)\right) \boldsymbol{c}_{s+1} + \cdots + \left(1 - \sum_{m=s}^{s+d-2} - (1 - \sum_{m=s}^{s+d-1} b_m^d)\right) \boldsymbol{c} \\ &= \boldsymbol{c}_s + \left(1 - \sum_{m=s}^{s} b_m^d\right)(\boldsymbol{c}_{s+1} - \boldsymbol{c}_s) + \cdots + \left(1 - \sum_{m=s}^{s+d-2} b_m^d\right)(\boldsymbol{c}_{s+d} - \boldsymbol{c}_{s+d-1})\end{aligned}$$

Letting $\beta_m^d(t, \mathbf{k}_M^d) \doteq 1 - \sum_{q=s}^{s+q} b_m^d(t, \mathbf{k}_M^d)$ gives

$$\mathbf{p}(t) = \boldsymbol{c}_s + \sum_{m=1}^{s+d} \beta_m^d(t, \mathbf{k}_M^d)(\boldsymbol{c}_{s+m} - \boldsymbol{c}_{s+m-1})$$

K. Qin, “General matrix representations for B-splines,” *The Visual Computer*, vol. 16, no. 3-4, pp. 177–186, 2000.

B-Splines on SE(3) (generalizes to other Lie Groups)

Splines on $\mathbb{R}^n$: For any time $t \in [t_s, t_{s+1}]$, the spline

$$\mathbf{p}(t) = \boldsymbol{c}_s + \sum_{m=1}^{s+d} \beta_m^d(t, \mathbf{k}_M^d)\,(\boldsymbol{c}_{s+m} - \boldsymbol{c}_{s+m-1})$$

Splines on $SE(3)$: For any time $t \in [t_s, t_{s+1}]$, the spline

$$\mathbf{T}(t) = \bar{\boldsymbol{C}}_s \Pi_{m=1}^{s+d} \mathrm{Exp}\left(\beta_m^d(t, \mathbf{k}_M^d) \mathrm{Log}\left(\bar{\boldsymbol{C}}_{s+m-1}^{-1} \bar{\boldsymbol{C}}_{s+m}\right)\right)$$

where

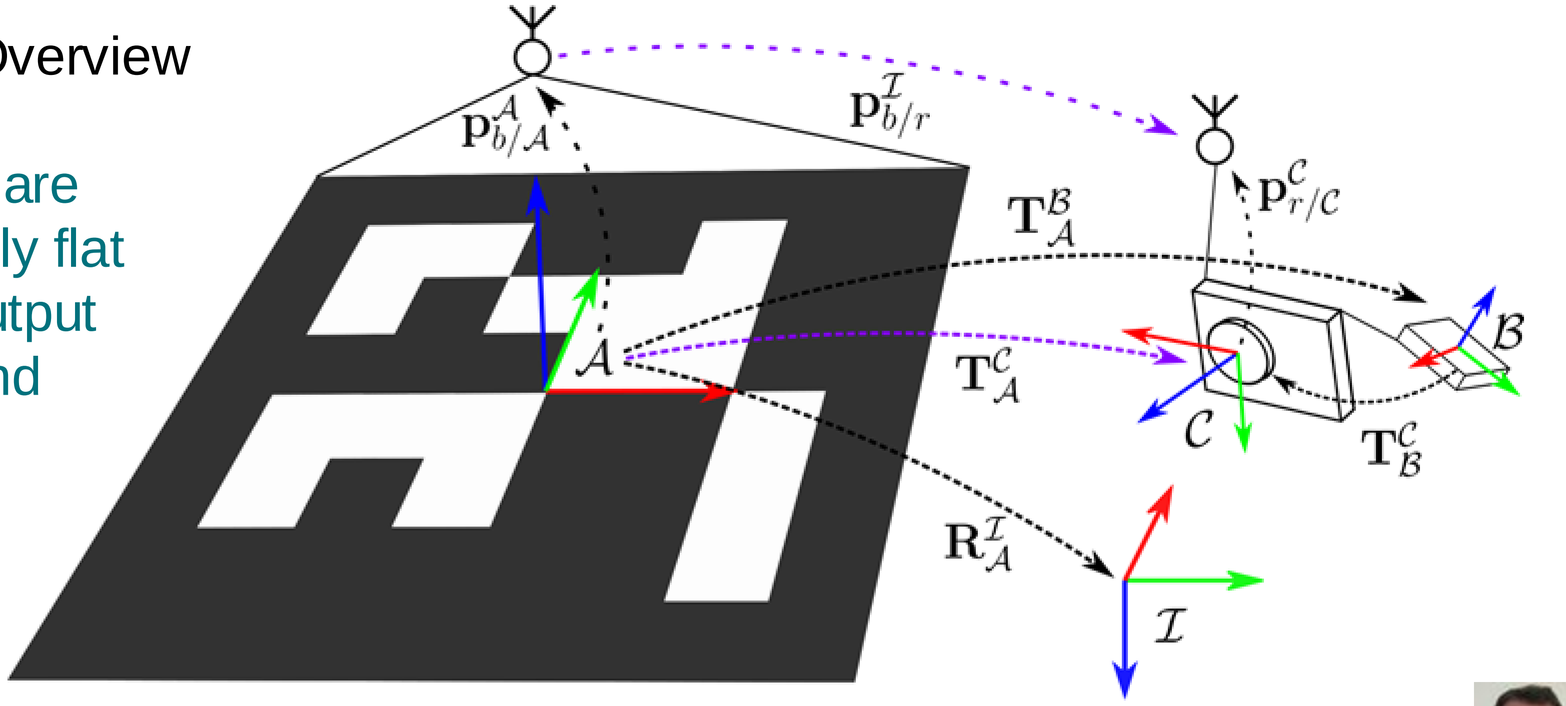
$$\begin{aligned} \bar{\boldsymbol{C}}_m &\in SE(3) && \text{are the control points} \\ \mathrm{Exp} = \exp \circ \wedge &: \mathbb{R}^6 \to SE(3) && \text{is the exponential map} \\ \mathrm{Log} = \vee \circ \log &: SE(3) \to \mathbb{R}^6 && \text{is the logarithmic map} \end{aligned}$$

A. Patron-Perez, S. Lovegrove, and G. Sibley, “A spline-based tra- jectory representation for sensor fusion and rolling shutter cameras,” *International Journal of Computer Vision*, vol. 113, no. 3, pp. 208– 219, 2015.

GNSS/Camera Extrinsic Calibration Using Splines on SE(3)

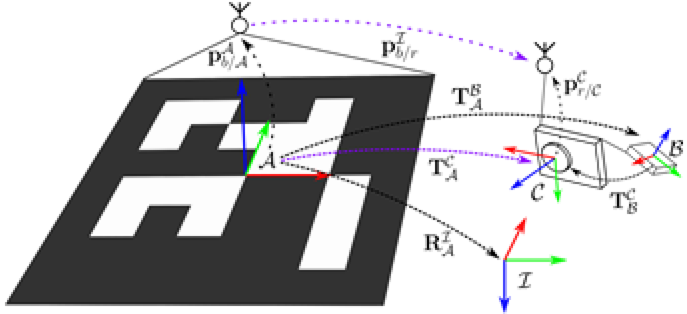
Problem Overview

Dynamics are differentially flat with flat output position and heading



δt : time delay between GPS and camera clocks

Calibration - Measurement Models



AprilTag: $z_{\text{cam}} = \text{Exp}(\eta_{\text{cam}} T_{\mathcal{B}}^{\mathcal{C}} T_{\mathcal{A}}^{\mathcal{B}}(t_{\text{cam}}))$

RTK: $z_{\text{rtk}} = R_{\mathcal{A}}^{\mathcal{I}} \left(-p_{b/\mathcal{A}}^{\mathcal{A}} + T_{\mathcal{B}}^{\mathcal{A}}(t_{\text{gnss}} + \delta t) T_{\mathcal{C}}^{\mathcal{B}} \mathbf{p}_{r/\mathcal{C}}^{\mathcal{C}} \right) + \eta_{\text{rtk}}$

GNSS Velocity: $z_{\text{vel}} = R_{\mathcal{A}}^{\mathcal{I}} R_{\mathcal{B}}^{\mathcal{A}} \left(v_{\mathcal{B}/\mathcal{A}}^{\mathcal{B}} + (\omega_{\mathcal{B}/\mathcal{A}}^{\mathcal{B}})^{\wedge} T_{\mathcal{C}}^{\mathcal{C}} \mathbf{p}_{r/\mathcal{C}}^{\mathcal{C}} \right) + \nu_{\text{vel}}$

Gyro: $z_{\text{gyro}} = \omega_{\mathcal{B}/\mathcal{I}}^{\mathcal{B}}(t_{\text{imu}}) + b_{\text{gyro}}(t_{\text{imu}}) + \eta_{\text{gyro}}$

Accelerometer: $z_{\text{accel}} = a_{\mathcal{B}/\mathcal{I}}^{\mathcal{B}}(t_{\text{imu}}) - g R_{\mathcal{A}}^{\mathcal{B}}(t_{\text{imu}}) R_{\mathcal{I}}^{\mathcal{A}} e_3 + b_{\text{acc}}(t_{\text{imu}}) + \eta_{\text{acc}}$

Calibration - Optimization

Quantities that need to be calibrated:

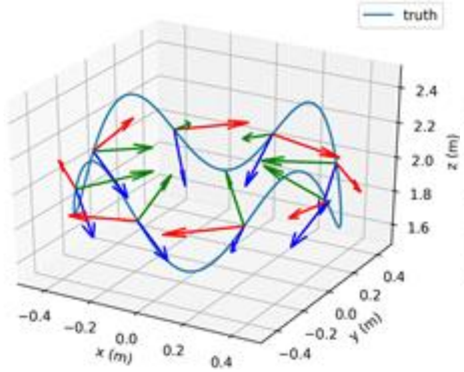
$\{\bar{T}_j\}_{j=0,\ldots,M}$	trajectory control points
$\boldsymbol{p}^{\mathcal{C}}_{r/\mathcal{C}}$	position of rover receiver in camera frame
$\boldsymbol{p}^{\mathcal{A}}_{b/\mathcal{A}}$	position of base receiver in fiducial frame
$R^{\mathcal{I}}_{\mathcal{A}}$	rotation from fiducial to NED frame
δt	time offset between camera and GNSS
$\{\bar{b}_{g_j}\}_{j=0,\ldots,M_{\text{gyro}}}$	gyro bias control points
$\{\bar{b}_{g_j}\}_{j=0,\ldots,M_{\text{gyro}}}$	gyro bias control points

Optimization Problem:

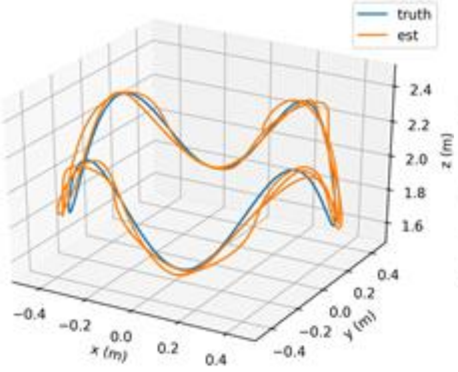
$$\begin{aligned}\arg\min_{\boldsymbol{\Gamma}} \Big\{ &\sum_{i_{\text{cam}}} \|\mathrm{Log}(\boldsymbol{h}_{\text{cam}}(\boldsymbol{\Gamma})\boldsymbol{z}^{-1}_{i_{\text{cam}}})\|^2_{\Sigma_{\text{cam}}^{-1}} \\ &+ \sum_{i_{\text{rtk}}} \|\boldsymbol{z}_{i_{\text{rtk}}} - \boldsymbol{h}_{\text{rtk}}(\boldsymbol{\Gamma})\|^2_{\Sigma^{-1}_{\text{rkt}}} \\ &+ \sum_{i_{\text{vel}}} \|\boldsymbol{z}_{i_{\text{vel}}} - \boldsymbol{h}_{\text{vel}}(\boldsymbol{\Gamma})\|^2_{\Sigma^{-1}_{\text{vel}}} \\ &+ \sum_{i_{\text{gyro}}} \|\boldsymbol{z}_{i_{\text{gyro}}} - \boldsymbol{h}_{\text{gyro}}(\boldsymbol{\Gamma})\|^2_{\Sigma^{-1}_{\text{gyro}}} \\ &+ \sum_{i_{\text{accel}}} \|\boldsymbol{z}_{i_{\text{accel}}} - \boldsymbol{h}_{\text{accel}}(\boldsymbol{\Gamma})\|^2_{\Sigma^{-1}_{\text{accel}}} \Big\}\end{aligned}$$

Optimization is solved iteratively using Levenberg-Marquardt

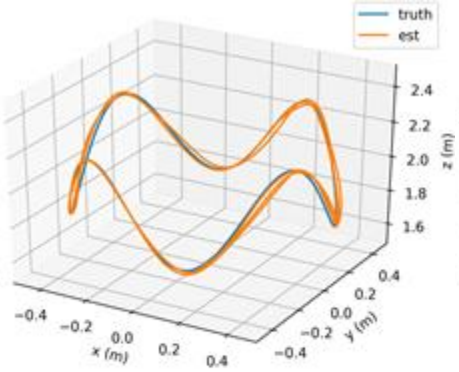
Simulation Results



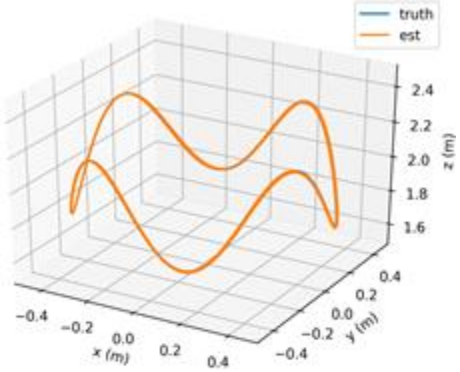
Trajector
y



Camera + GNSS



Camera + GNSS
+ Velocity



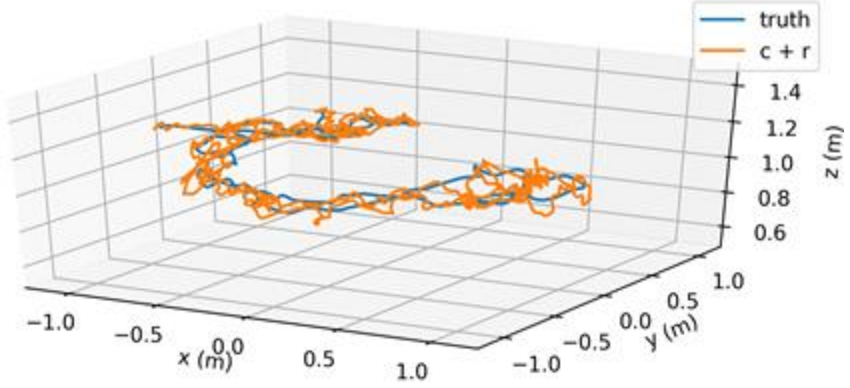
Camera + GNSS
+ Velocity + IMU

Motion Capture Results

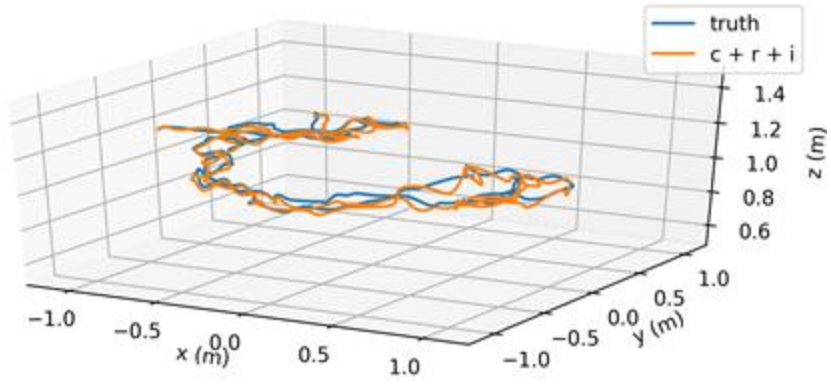
TABLE I

MOTION CAPTURE CALIBRATION RESULTS FOR C+R AND C+R+I FOR A 50 AND 100 s TRAJECTORY.

Type	$\mathbf{p}_{b/\mathcal{A}}^{\mathcal{A}}$ err. (m)	$\mathbf{R}_{\mathcal{A}}^{\mathcal{I}}$ err. (rad)	$\mathbf{p}_{r/\mathcal{C}}^{\mathcal{C}}$ (m)	δt (s)	Pose RMSE	Time (s)
c+r, 50	0.0136	0.0089	[0.100, 0.048, -0.196]	-0.0067	0.0560	2.90
c+r+i, 50	**0.0098**	**0.0067**	[0.099, 0.047, -0.198]	-0.0200	**0.0458**	24.88
c+r, 100	**0.0039**	0.0049	[0.100, 0.046, -0.201]	-0.0160	0.0460	9.90
c+r+i, 100	**0.0039**	**0.0047**	[0.100, 0.046, -0.202]	-0.0206	**0.0421**	35.58



Camera + GNSS



Camera + GNSS
+ IMU

Outdoor Results

TABLE II
OUTDOOR CALIBRATION RESULTS

	$\mathbf{p}^{\mathcal{A}}_{b/\mathcal{A}}$ (m)	$\mathbf{p}^{\mathcal{C}}_{r/\mathcal{C}}$ (m)	δt (s)
mean	[0.766, -0.650, 0.308]	[0.053, 0.005, -0.184]	-0.0570
std. dev.	[0.006, 0.005, 0.004]	[0.003, 0.005, 0.006]	0.0033

Hand Calibrated Values:

- Base offset: [0.78, -0.64, 0.33]
- Rover offset: [0.04, 0, -0.19]

Summary and Future work

- B-splines
    - continuous representation of trajectories and surfaces
    - Discrete parameterization not directly linked to time constants, sampling rates, noise levels, etc.
    - Sparse representation: control points have only local impact
    - Many properties can be guaranteed at every time by checking finite (and selectable) number of constraints

- Applications to Planning
    - Trajectories are in the convex hull of control points (locally)

- Applications to Estimation
    - Flexible parameterization with many advantages for Moving Horizion Estimation