Safe Flight Corridor Planner for Fast and Robust Snap Minimized Trajectories

Alejandro Pelaez & Randal Beard

IMMERSE Presentation

July 2026



Outline

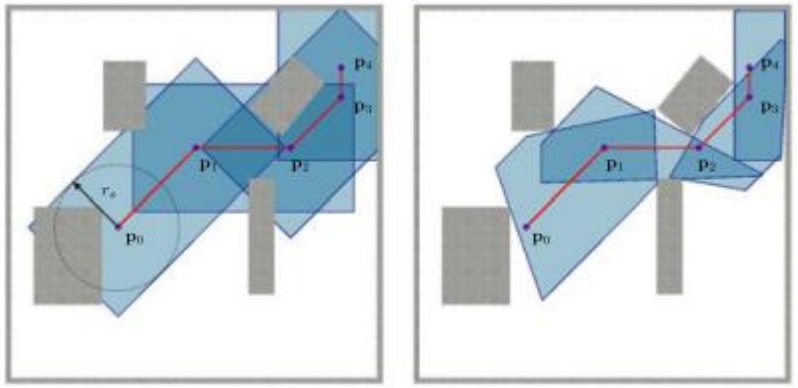
- Motivation & Goal
- Background
  - Safe Flight Corridors
  - B-Splines
    - Properties
    - Snap Minimization w/ B-Splines
    - MINVO Spline Boundaries
- Pipeline Implementation
- Current Progress
- Next Steps

Outline

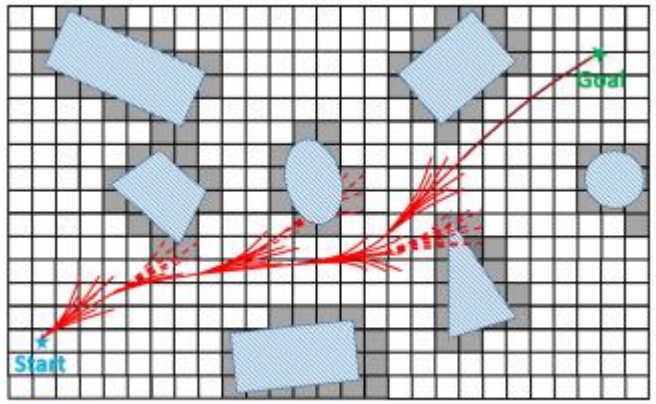
- Motivation & Goal
- Background
  - Safe Flight Corridors
  - B-Splines
    - Properties
    - Snap Minimization w/ B-Splines
    - MINVO Spline Boundaries
- Pipeline Implementation
- Current Progress
- Next Steps

Motivation & Goal

- **Online Trajectory Planner**
  - How aircraft will physically traverse from a to b using only onboard sensors for environmental understanding
- **Current online trajectory planners balance different tradeoffs:**
  - Mathematical guarantees at expense of computational speed
    - Liu, Sikang, et al. "Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-d complex environments." *IEEE Robotics and Automation Letters* 2.3 (2017): 1688-1695.
  - Computational speed at expense of mathematical guarantees
    - Zhou, Boyu, et al. "Robust and efficient quadrotor trajectory generation for fast autonomous flight." *IEEE Robotics and Automation Letters* 4.4 (2019): 3529-3536.
- **What if we could incorporate both methodologies for a hyper-efficient planner that mathematically respects obstacle boundaries?**



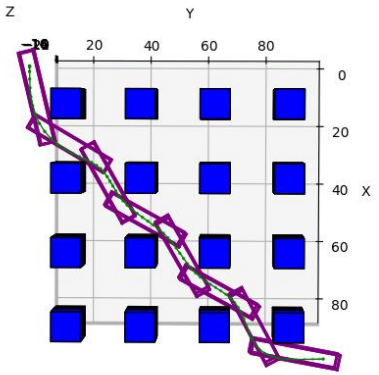
Liu, Sikang

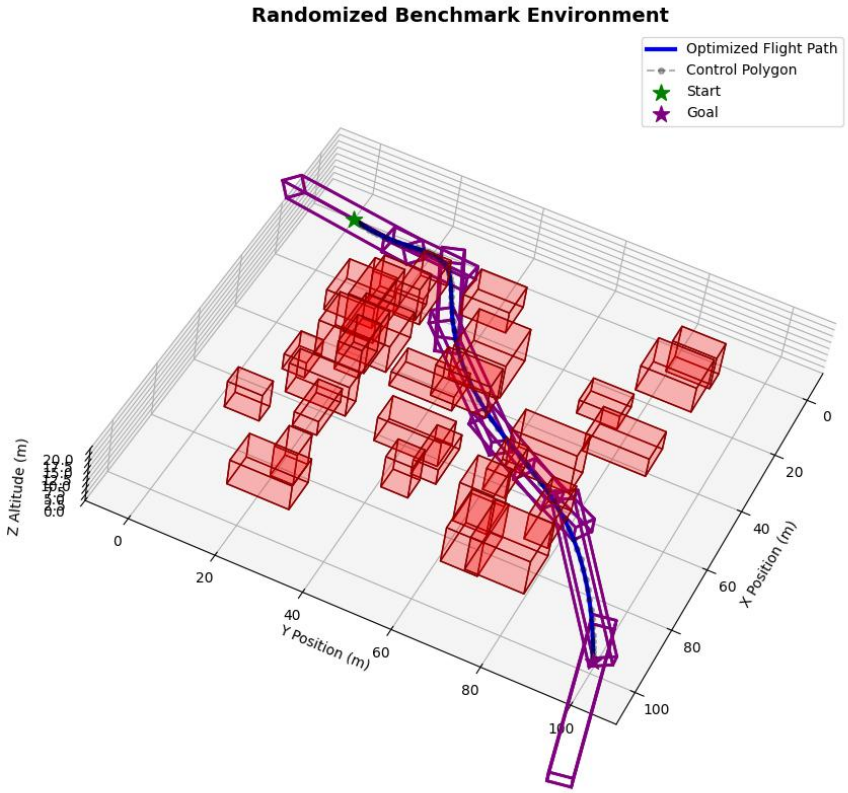


Zhou, Boyu

Motivation & Goal (Continued)

- **COAST (Corridor-Optimized Aerially-Safe Trajectories)**
  - Spatiotemporally-bounded path and trajectory planner
    - Trajectory formulated within mathematically proven obstacle-free rectangular prisms
    - Trajectory formulated with time-parameterized B-Spline
  - Simple bounds and Matrix B-Spline representations aid with computational speed
  - Ideally plan trajectories in real time within 20ms





Outline

- Motivation & Goal
- Background
  - Safe Flight Corridors
  - B-Splines
    - Properties
    - Snap Minimization w/ B-Splines
    - MINVO Spline Boundaries
- Pipeline Implementation
- Current Progress
- Next Steps

Background – Safe Flight Corridors (SFCs)

- **Rectangular prism (or convex 3D shape) mathematically guaranteed to be collision-free within its volume/bounds**
  - Christensen, David LaVae. "Real-time dynamically feasible b-spline trajectory generation for unmanned aircraft." (2024).
  - Anderson, Dean Benjamin. "Path Planning and Control for Vertical Takeoff and Landing Aircraft." (2026).
- **Defined as 6 half-plane intersections**
  - Half-plane: Mathematical inequality $\boldsymbol{a^t x \leq b}$
    - '$\boldsymbol{a}$': Unit vector perpendicular to boundary
    - 'b': Scalar distance to boundary
    - '$\boldsymbol{x}$': Coordinates of interest
      - Dot product with 'a' determines whether coordinate is within 'b'
  - Represented in matrix form $\boldsymbol{Ax \leq b}$ for multiple planes
- **Formulated to respect aircraft constraints, such as turn radius for fixed-wing**

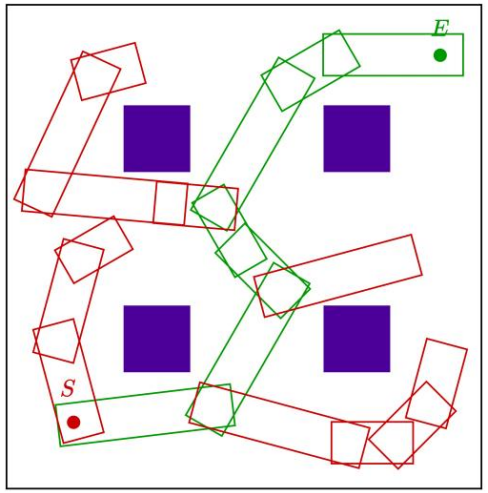
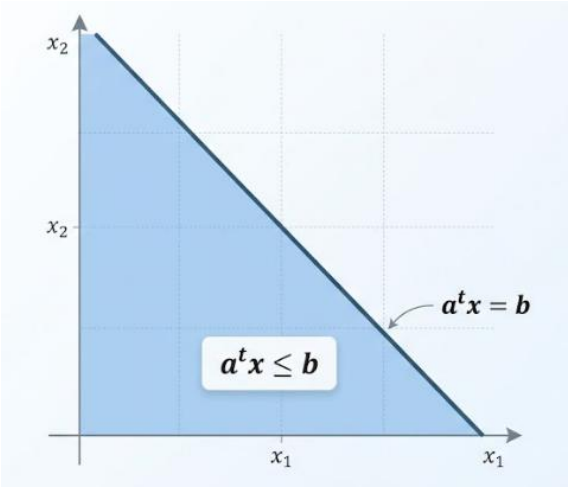


Figure 5.10: Example of RRT tree.

Anderson, Dean



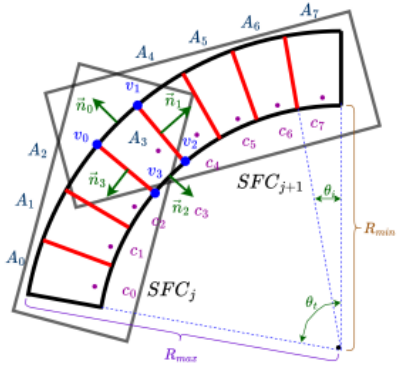


Figure 5.15: Diagram of sampled annulus constraints

Anderson, Dean

$$A = \begin{bmatrix} a_1^T \\ a_2^T \\ a_3^T \\ a_4^T \\ a_5^T \\ a_6^T \end{bmatrix}, \quad x = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \end{bmatrix}$$

Gemini

Outline

- Motivation & Goal
- Background
  - Safe Flight Corridors
  - B-Splines
    - Properties
    - Snap Minimization w/ B-Splines
    - MINVO Spline Boundaries
- Pipeline Implementation
- Current Progress
- Next Steps

Background – B-Splines

- **B-Spline: Basis spline**
- **Piecewise polynomial (basis) function that can be summed and scaled using control points**

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

  - $b_m^d(t,k)$: Time-parameterized basis function
    - *'d'* defines degree of function
  - c_m: control point for given basis function
    - Scalar value that affects height of basis
  - Sumed basis functions always equal 1 before applying control points
- **In context of path planning x-axis represents time, y-axis represents position**
  - 3D trajectories require 3 splines (x,y,z)

BASIS FUNCTION BY DEGREE

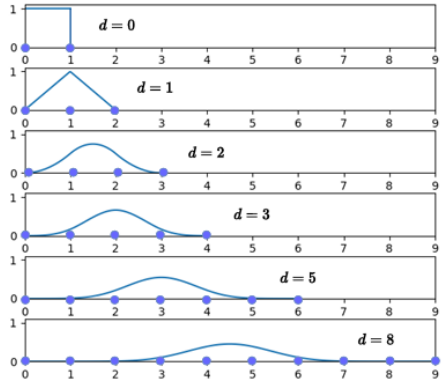
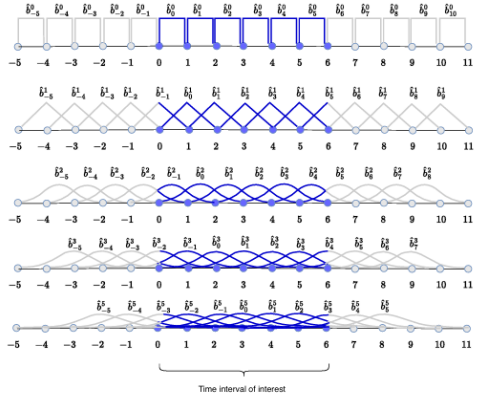


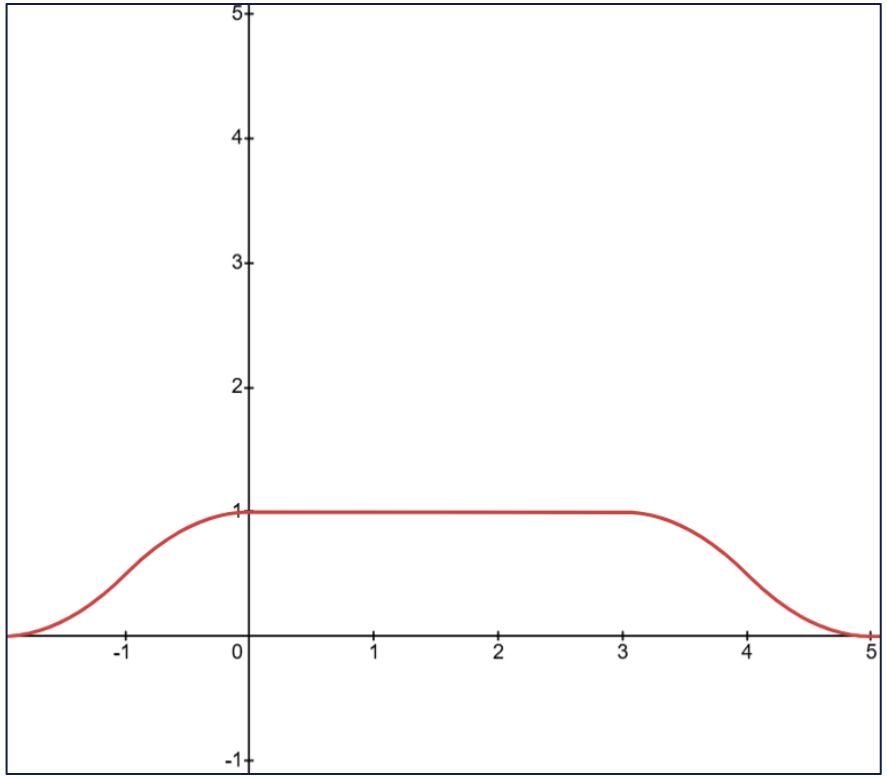
Figure 2: Central basis functions for different degrees d.

SUMMED FUNCTIONS (EQUALS 1)

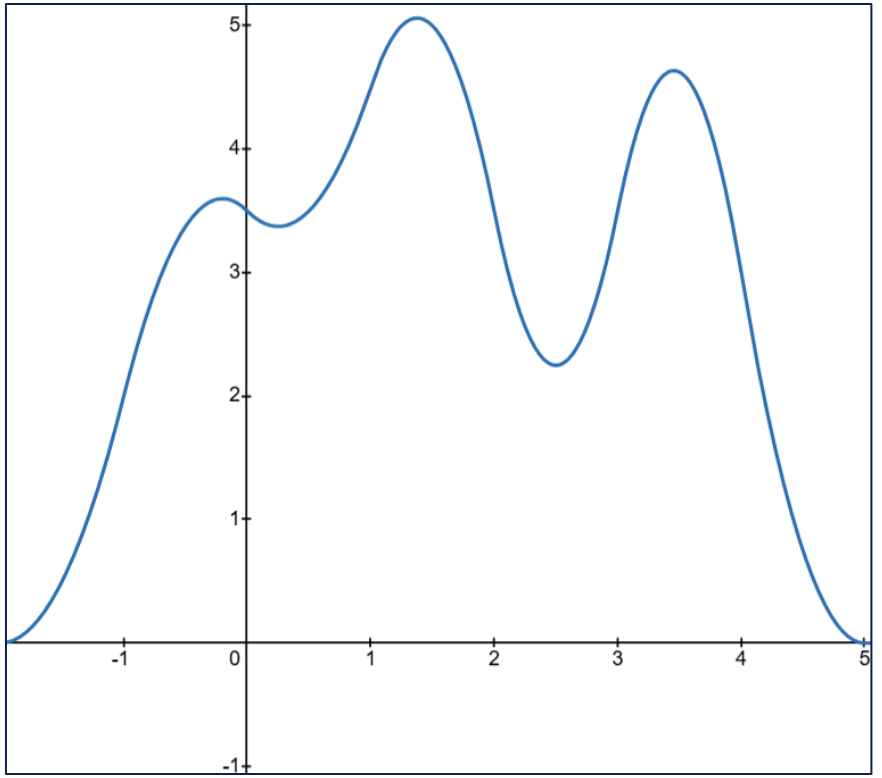


Background – B-Splines

NO CONTROL POINTS



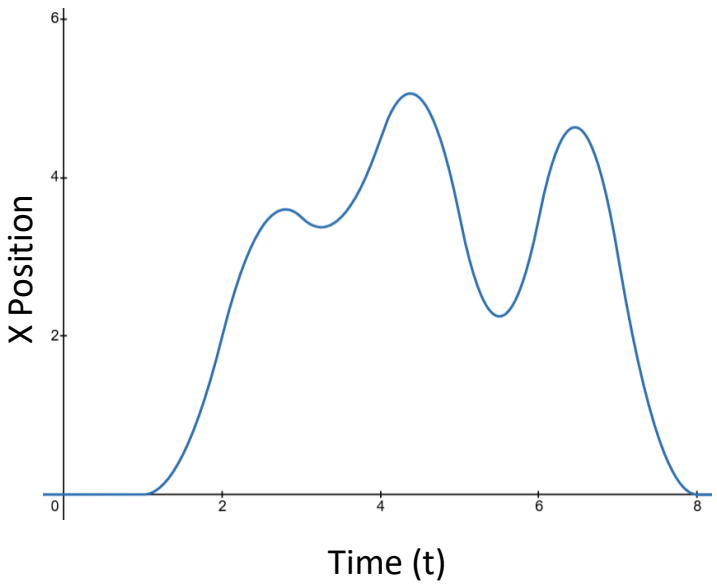
CONTROL POINTS [4,3,6,1,6]



Background – B-Splines

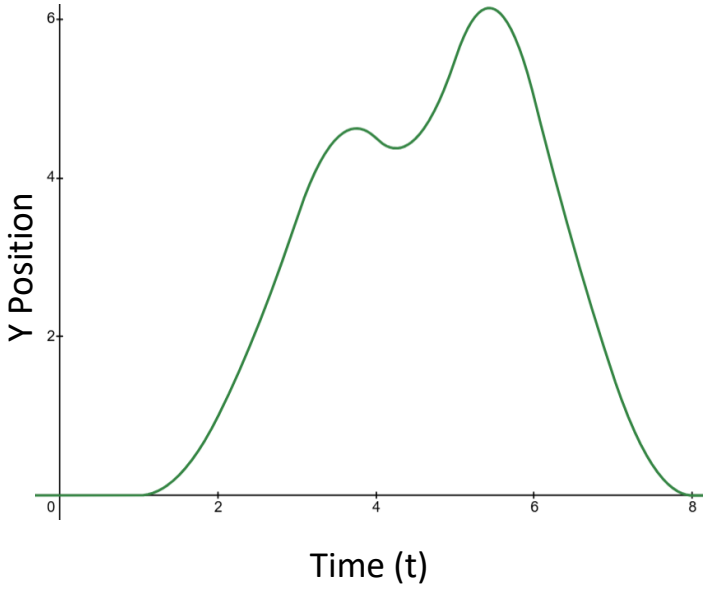
X-AXIS TRAJECTORY

CP: [4,3,6,1,6]

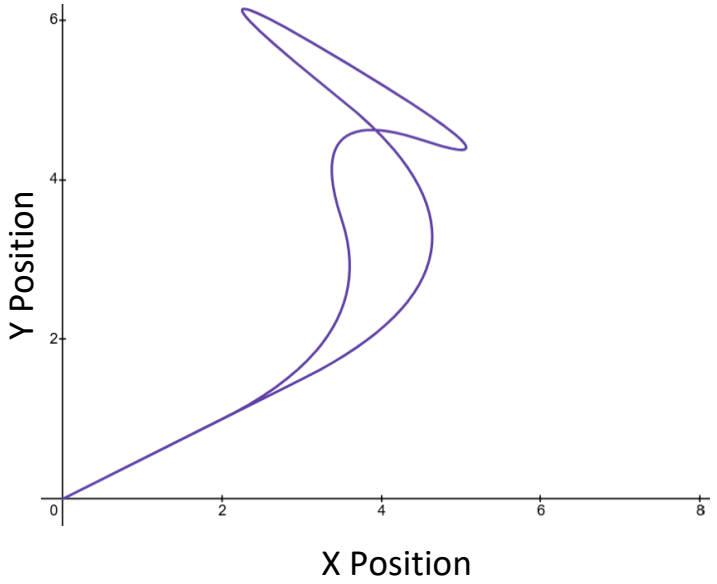


Y-AXIS TRAJECTORY

CP:[2,5,4,7,3]

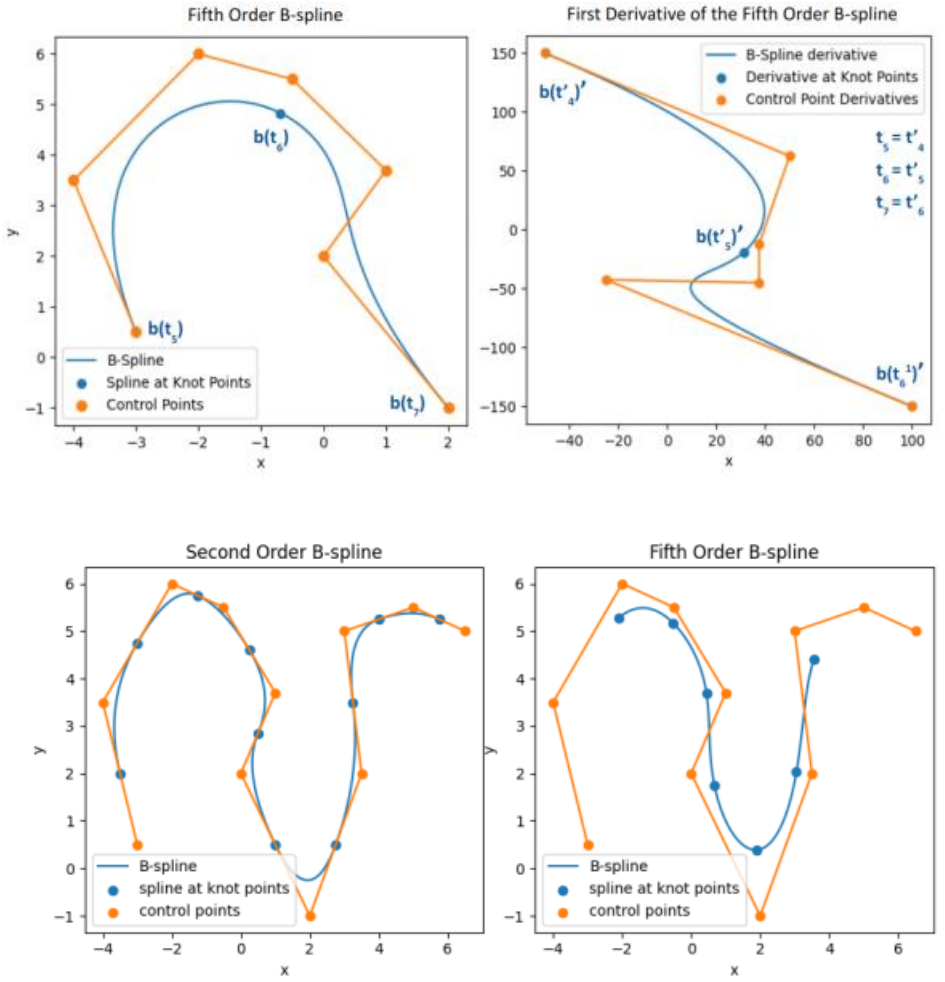


(x(t),y(t))



Background – B-Splines: Properties

- **Differentiable**
  - Can get velocity and acceleration by evaluating derivatives at different timestamps
  - Can be estimated by taking the difference between control points
- **Convex-hull property**
  - Curve will never extend beyond control points
  - For trajectory planning: Can assure curve will never exit SFCs if control points are bounded within



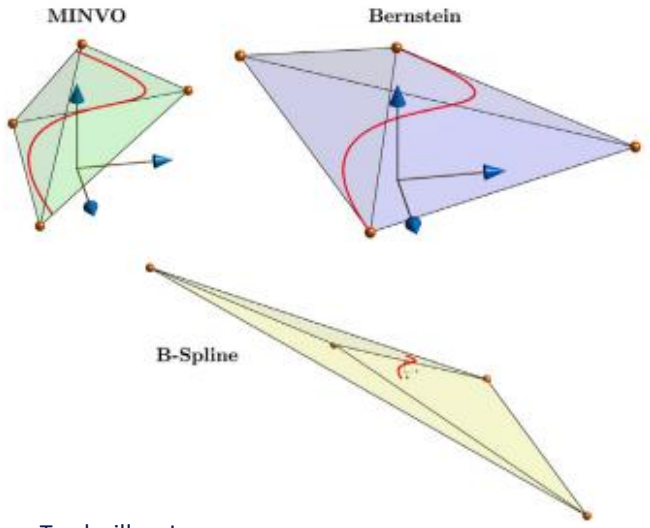
Christensen, David

Background – B-Splines: MINVO

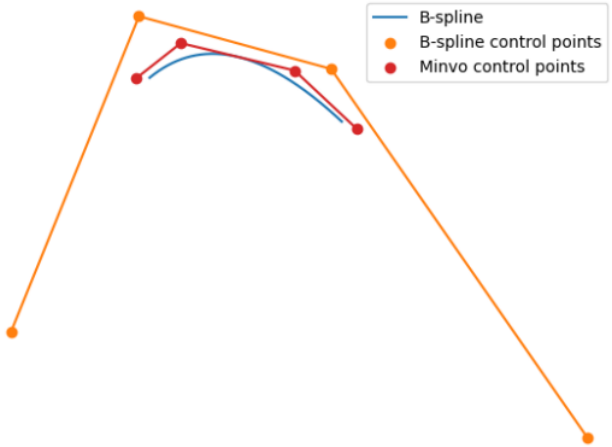
- **Minimum volume simplex enclosure**
  - Tordesillas, Jesus, and Jonathan P. How. "MINVO basis: Finding simplexes with minimum volume enclosing polynomial curves." *Computer-Aided Design* 151 (2022): 103341.
- **Found through applying a matrix transformation to original control points**
  - Q/V: Control points, A: Matrix transform

$$Q_j^{\mathrm{MV}} = Q_j^{\mathrm{BS}} A_{\mathrm{pos}}^{\mathrm{BS}}(j) \left(A_{\mathrm{pos}}^{\mathrm{MV}}\right)^{-1}$$
$$V_j^{\mathrm{MV}} = V_j^{\mathrm{BS}} A_{\mathrm{vel}}^{\mathrm{BS}}(j) \left(A_{\mathrm{vel}}^{\mathrm{MV}}\right)^{-1}$$

    - Tordesillas, Jesus, and Jonathan P. How. "MADER: Trajectory planner in multiagent and dynamic environments." *IEEE Transactions on Robotics* 38.1 (2021): 463-476.
  - Works for natural and clamped splines
- **Leverages B-Spline properties for more accurate reflection of spline trajectory and constraints**
  - Since splines are differentiable, derivative of position trajectory is velocity trajectory
    - Can use this for less conservative estimates

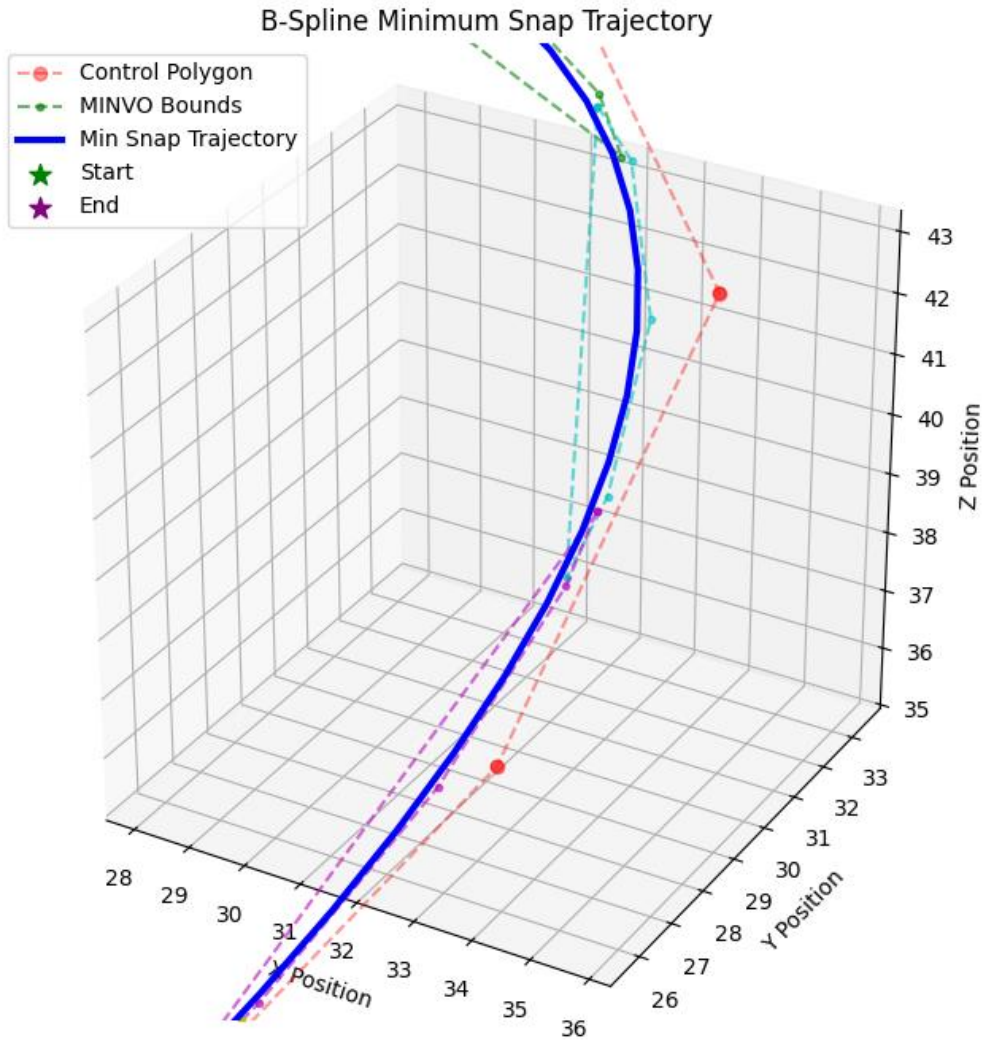


Tordesillas, Jesus



Christensen, David

Background – B-Splines: MINVO (Continued)



Background – B-Splines: Snap Minimization

- **Snap: 4th derivative of position**
- **Directly affects jerk (3rd derivative) and acceleration (2nd derivative), minimizing it leads to smoother, less taxing flight**
  - Mellinger, Daniel, and Vijay Kumar. "Minimum snap trajectory generation and control for quadrotors." *2011 IEEE international conference on robotics and automation*. Ieee, 2011.
- **Can design trajectories prioritizing lower energy consumption or motor overexertion**
- **Can use 4th, and larger, degree B-Splines to involve snap**
  - Beard, R. W., "Robotics applications of uniform b-spline trajectories", Manuscript in preparation, 2025.
- **Calculated as a penalty matrix (W_M^d) using matrix representations of B-Splines ($\boldsymbol{b}_M^k(t)$) and differentiation ($D_M^{d,\ell-1}$)**

$$\rho^0 \neq 0, \rho^j = 0 \text{ for } j > 0 \implies \text{Minimize integral of path}$$
$$\rho^1 \neq 0, \rho^j = 0 \text{ for } j \neq 1 \implies \text{Minimize the path length}$$
$$\rho^4 \neq 0, \rho^j = 0 \text{ for } j \neq 4 \implies \text{Minimize snap.}$$

Define

$$S_M^k \triangleq \int_0^M \boldsymbol{b}_M^k(t)\boldsymbol{b}_M^k(t)^\top dt$$
$$W_M^{d,\ell} \triangleq D_M^{d,\ell-1} S_M^{d-\ell} D_M^{d,\ell-1\top},$$

and note that $W_M^{d,\ell}$ are constant matrices that can be pre-computed and stored in memory, and define

$$W_M^d(\boldsymbol{\rho}) \triangleq \sum_{\ell=0}^{L} \rho^\ell W_M^{d,\ell}.$$

Beard, R. W.

Outline

- Motivation & Goal
- Background
  - Safe Flight Corridors
  - B-Splines
    - Properties
    - Snap Minimization w/ B-Splines
    - MINVO Spline Boundaries
- Pipeline Implementation
- Current Progress
- Next Steps

Pipeline Implementation

IN

- Map Data (LIDAR)
- Start/End coordinates
- Max. Vel/Accel
- B-Spline Type & Degree
- Aircraft Radius
- Max. SFC Width/Height

Trajectory Planner

Voxelized World Map

Fixed-Wing

- Dubins A* for turn radius constraints and [(SFC Width)/2] obstacle expansion
- Arbitrary fixed-size SFC height & width
- Calculates # of control points per SFC based on SFC length

Multirotor

- A* with drone radius obstacle expansion
- Dynamically resizing SFC heights & widths
  - For indoor/cluttered traversal
- Calculates # of control points per SFC based on SFC length

SFC Bounds & Intersections

Min. Snap Planner

Calculate Snap Minimization Penalty Matrix

Optimizer

- Takes in the SFC bounds, penalty matrix, starting/ending coordinates, and MINVO bounds for maximum velocity and acceleration constraints
- Returns control points meeting these constraints

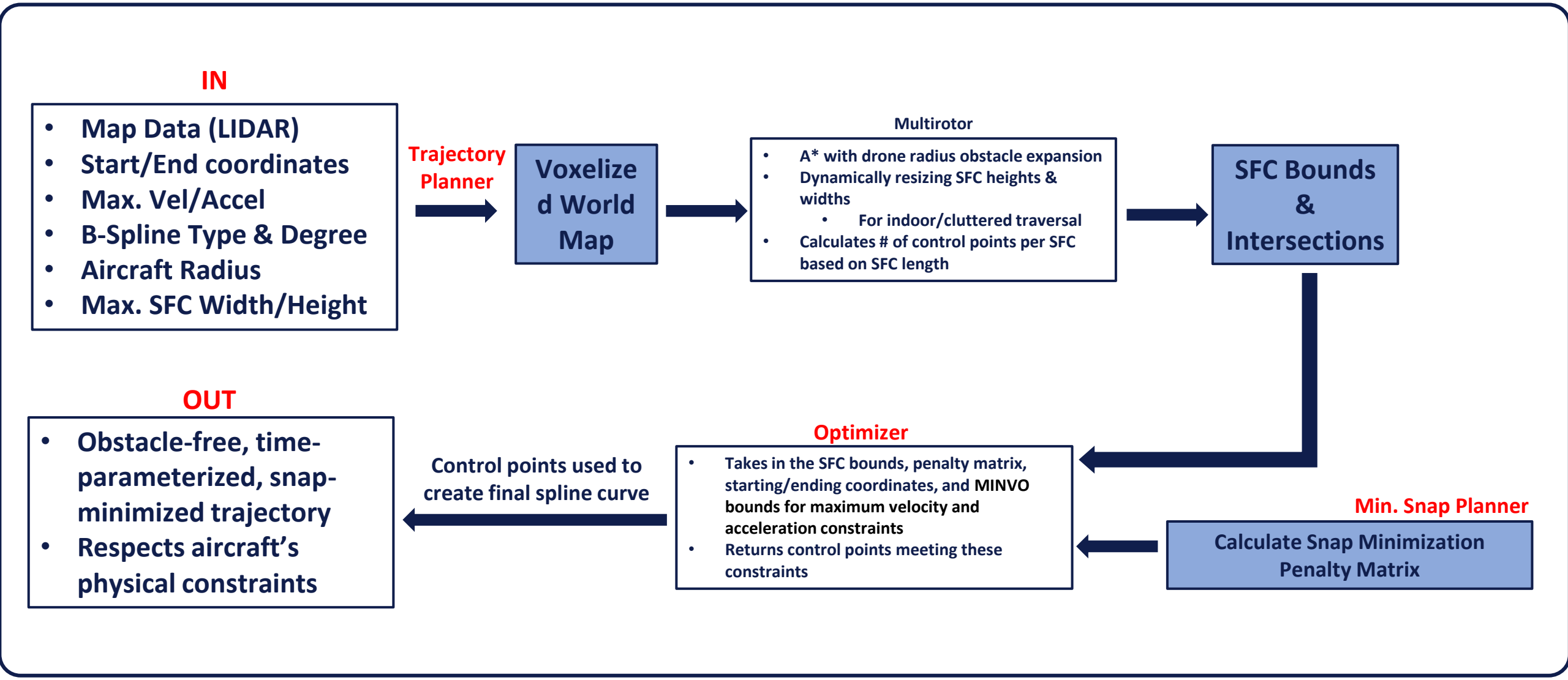
Control points used to create final spline curve

OUT

- Obstacle-free, time-parameterized, snap-minimized trajectory
- Respects aircraft's physical constraints

Outline

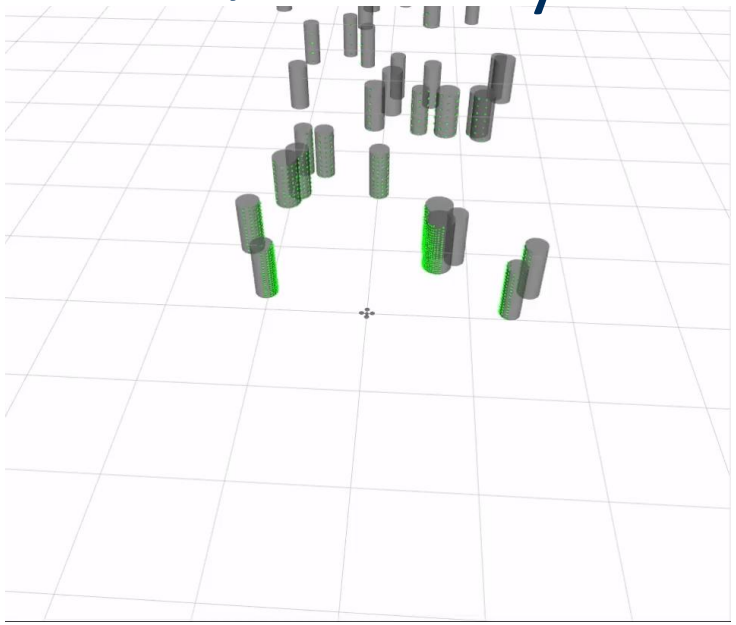
- Motivation & Goal
- Background
  - Safe Flight Corridors
  - B-Splines
    - Properties
    - Snap Minimization w/ B-Splines
    - MINVO Spline Boundaries
- Pipeline Implementation
- Current Progress
- Next Steps


IN
Map Data (LIDAR)
Start/End coordinates
Max. Vel/Accel
B-Spline Type & Degree
Aircraft Radius
Max. SFC Width/Height
Trajectory Planner
Voxelized World Map
Multirotor
A* with drone radius obstacle expansion
Dynamically resizing SFC heights & widths
For indoor/cluttered traversal
Calculates # of control points per SFC based on SFC length
SFC Bounds & Intersections
Min. Snap Planner
Calculate Snap Minimization Penalty Matrix
Optimizer
Takes in the SFC bounds, penalty matrix, starting/ending coordinates, and MINVO bounds for maximum velocity and acceleration constraints
Returns control points meeting these constraints
Control points used to create final spline curve
OUT
Obstacle-free, time-parameterized, snap-minimized trajectory
Respects aircraft's physical constraints

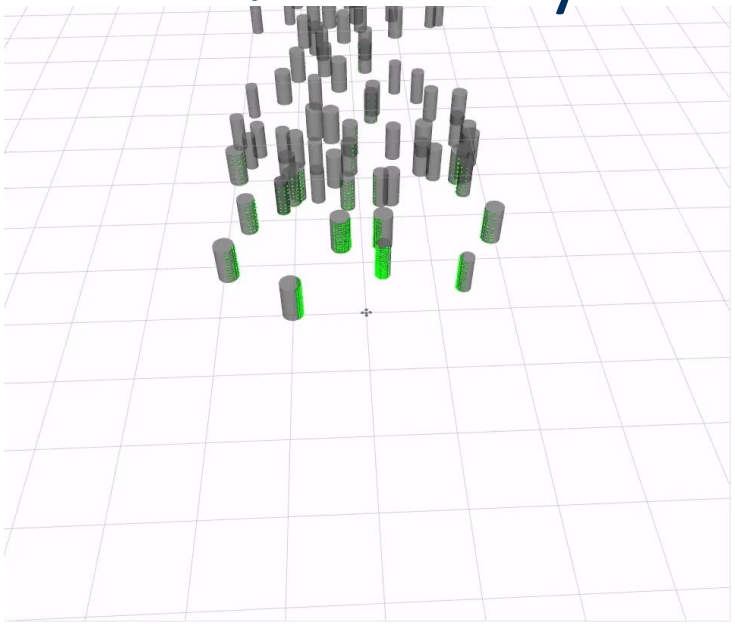
Current Progress

Density	Success	Comp. Time		Performance			Constraint Viol.	
	R_{succ} [%]	T_{opt} [ms]	T_{replan} [ms]	T_{trav} [s]	L_{path} [m]	S_{jerk} [m/s^2]	ρ_{vel} [%]	ρ_{acc} [%]
5%	100.0	0.89	2.46	47.48	109.79	64.58	0.00	0.00
10%	90.0	1.10	4.25	50.16	119.68	129.20	0.00	0.00

5% Density



10% Density



Outline

- Motivation & Goal
- Background
  - Safe Flight Corridors
  - B-Splines
    - Properties
    - Snap Minimization w/ B-Splines
    - MINVO Spline Boundaries
- Pipeline Implementation
- Current Progress
- Next Steps

Next Steps

- **Create ‘pip’ installable library and polish repo for open source use**
  - Potentially ROS package as well?
- **Extend ACC work into journal level publication**
  - Extensive hardware testing in indoor and outdoor settings
  - Address stall issue
  - Head-to-head comparison against other SOTA planners

Conclusion

BYU MAGICC LAB

Questions?