Theory and Explicit Design of a Path Planner for an SE(3) Robot

Zhaoqi Zhang, Yi-Jen Chiang, Chee Yap

Oct 7,2024

WAFR 2024, Chicago

NYU

Problem and Notations

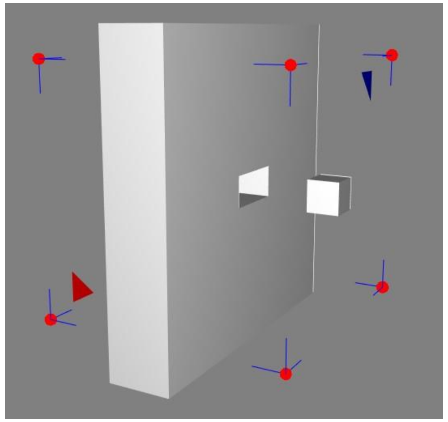
- We deal with kinematic path planning problem.
- Our robot is an isosceles right triangle $\mathcal{AOB}$ in $\mathbb{R}^3$ (Delta robot).
    - We call the area in physical space possessed by the robot under a given configuration γ the **footprint** of γ, denoted by $Fp(\gamma)$.

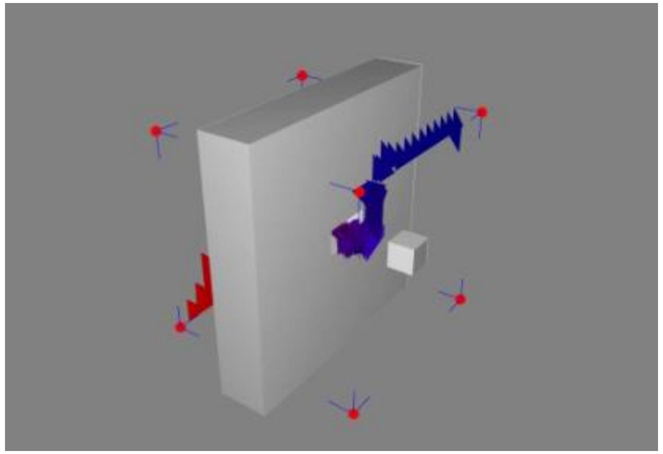
Input

- In physical space $\mathbb{R}^3$, an **obstacle set**, denoted by $\Omega \subseteq \mathbb{R}^3$.
- **Start** and **goal** configurations α and β in configuration space $\mathcal{C}space$.
- **Resolution parameter** $\varepsilon > 0$

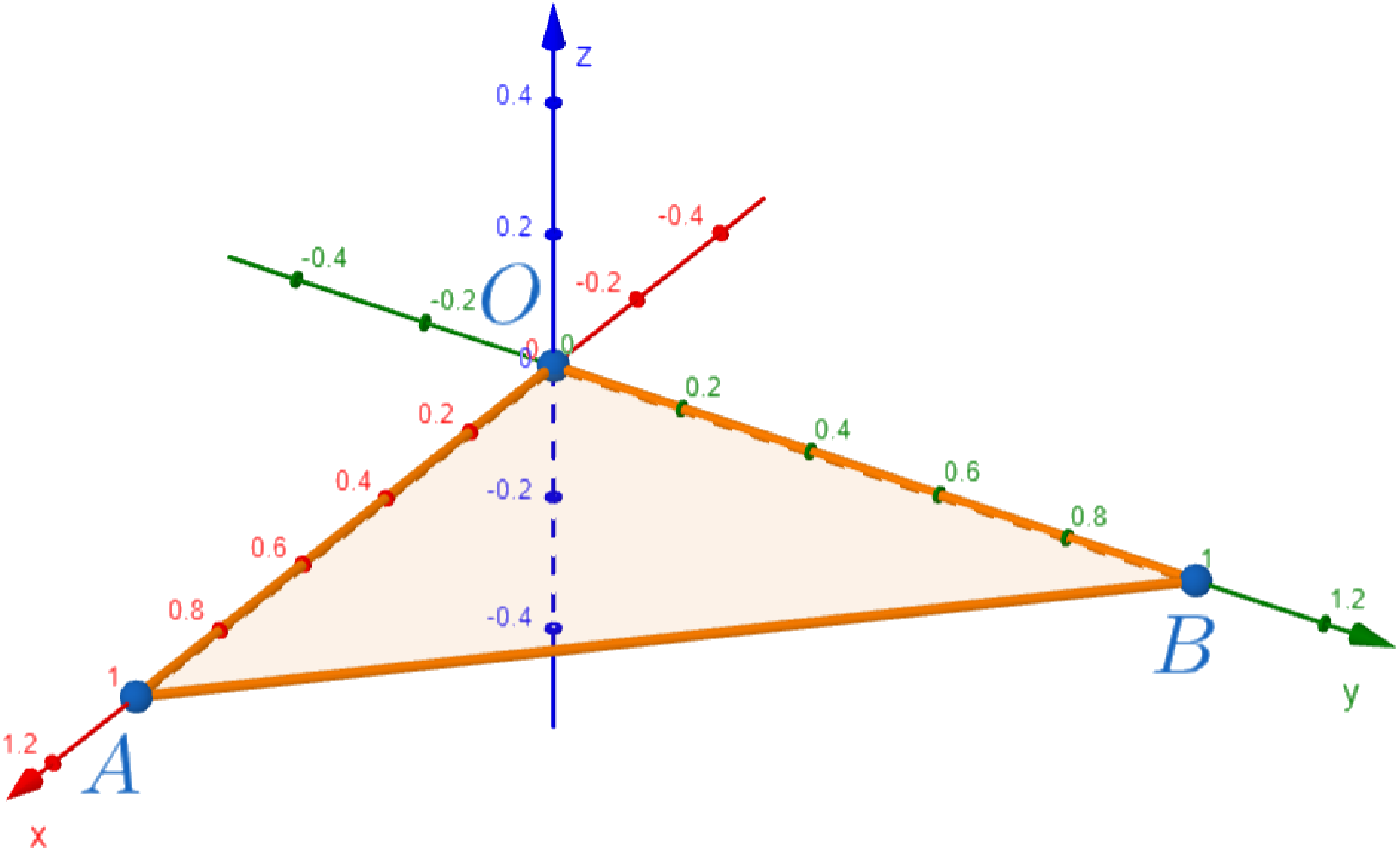
Output

- A **path** (continuous map) from α to β.
- Or **NO-PATH**.





Delta Robot ($\mathcal{AOB}$)



Resolution Exactness

- We use SSS (Soft Subdivision Search) framework. The output of SSS framework is **resolution exact**, i.e.,

- There exists some $\boldsymbol{K} > 1$ (independent of input), such that:
  - (P) if there is a path of clearance $\boldsymbol{K}\varepsilon$, it returns a path;
  - (N) if there is no path of essential clearance $\varepsilon/\boldsymbol{K}$, it returns NO-PATH.

- The SSS framework is currently the only complete method for path planning (other than exact computations) that does not have the halting problem.

- Resolution exactness of SSS is guaranteed by our Fundamental Theorem which depends on 5 axioms (see next).

NYU

SSS Axioms (constants σ, D_0, L_0, C_0)

- (A0) Softness.
  - The predicate $\tilde{C}$ is a soft classifier for $\mathcal{C}space$;
  - The SSS is **effective** if the predicate is σ-effective.
- (A1) Bounded Expansion.
  - There is a **subdivision constant** $D_0 \geq 1$ such that each box can be subdivided into at most D_0 children and the aspect ratio of each box is no more than D_0.
- (A2) Lipschitz Clearance.
  - The footprint satisfies a **Lipschitz constant** $L_0 > 0$:
    $$d_H(Fp(\gamma), Fp(\gamma')\,) \leq L_0 d(\gamma, \gamma')$$
    where d_H is the Hausdorff distance in $\mathbb{R}^k$.
- (A3) Good Atlas.
  - The subdivision atlas has an **atlas constant** $C_0 \geq 1$.
- (A4) Translational Cell.
  - The boxes are **translational**.

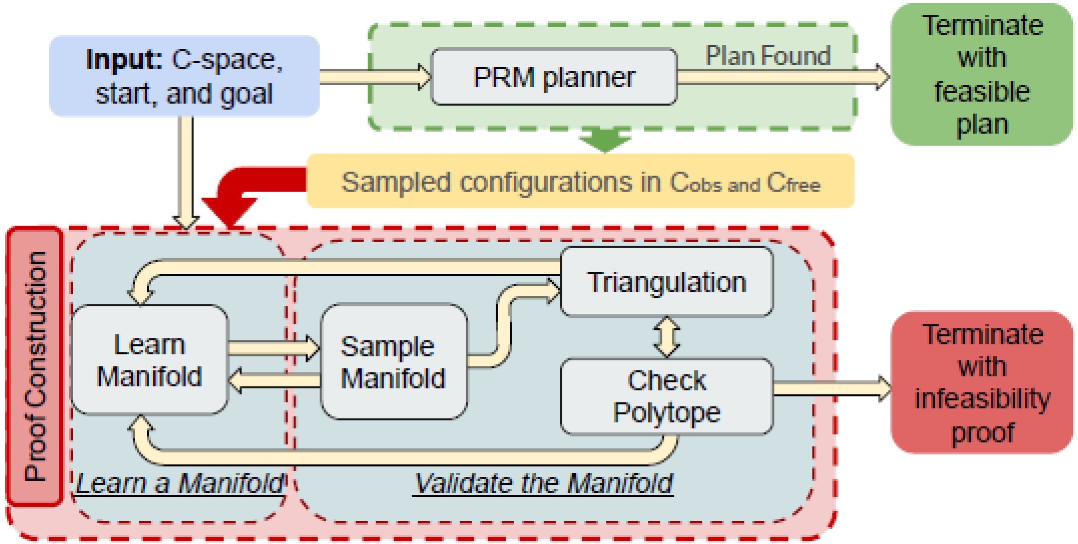
Compare to Sampling Approach

- Finding a path:
  - PRM/RRT/EST/SRT/etc.
  - Sampling functions, local planners, tree/graph-based planners [2].
  - The C_{free} must satisfy ε-goodness [3] and δ-clearance [4].

- Checking NO-PATH (infeasibility proof):
  - Learn and validate C_{obs} manifold [5].
  - The C_{obs} must be entirely ε-blocked [5].

- Requires "promise input"

From Li and Dantam [5]:

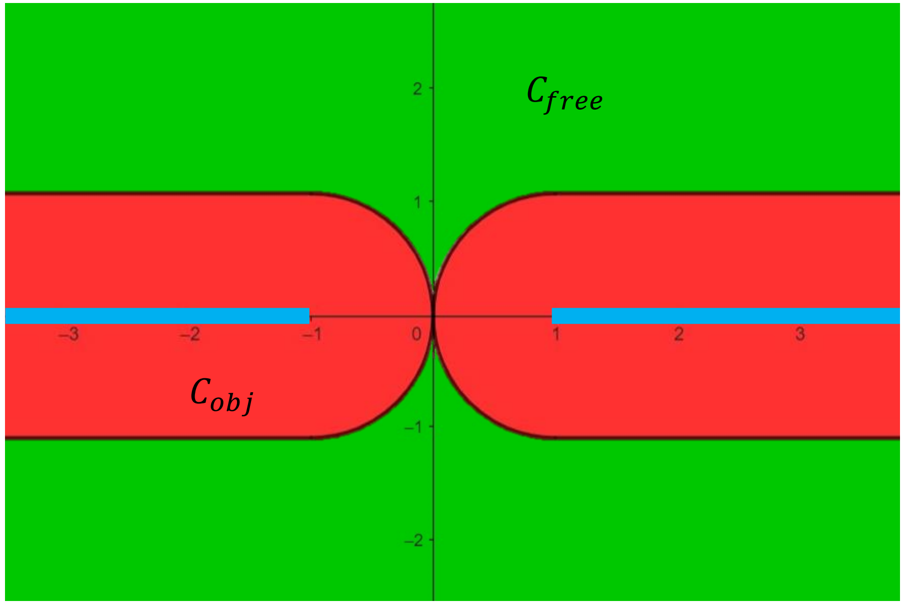


NYU

Zero Problem

- Consider a planar disc robot with radius 1 in $\mathbb{R}^2$.
    - The configuration space is $\mathbb{R}^2$.
    - Let the obstacle set be $\Omega = \{(x, y): x < -1 \text{ or } x > 1, y = 0\}$.
    - The start configuration $\alpha = (0, 1)$, goal configuration $\beta = (0, -1)$.
- Neither C_{free} is ε-good, nor C_{obj} is ε-blocked.
- To determine if configuration $(0,0)$ is free,
    - We must solve this zero problem.
        - $\{(x, y): x^2 + y^2 \leq 1\} \cap \Omega = \emptyset$?
    - We must use exact computation.
        - In general, it is at least single exponential time in the degree of freedom.
- SSS planner will return NO-PATH.

$\mathcal{C}space$



NYU

Predicates in SSS framework

- A predicate C classifies each configuration box B into FREE/MIXED/STUCK (a.k.a. empty/mixed/full) :

$$C(B) = \begin{cases} \text{FREE} & \forall \gamma \in B, \gamma \in C_{free} \\ \text{STUCK} & \forall \gamma \in B, \gamma \notin C_{free} \\ \text{MIXED} & otherwise \end{cases}$$

- A soft predicate $\tilde{C}$ is used for implementations that gives weaker but correct classifications.
  - Conservative:

    $\tilde{C}(B) \neq$ MIXED implies $C(B) = \tilde{C}(B)$;
  - Convergent:
    - If $\{B_i\}$ is a sequence of boxes such that $B_{i+1} \subseteq B_i$ and $\bigcap_{i=1}^{\infty} B_i = \{p\}$ for some $p \in Cspace$, then

      $\tilde{C}(B_i) = C(p)$ for i large enough.

NYU

SSS Framework

- **Priority queue Q:**
  - Controls the search of MIXED boxes.
  - GetNext() may adopt different strategies.
- **Find:**
  - Union find method preserves connected components.
- **Expand:**
  - Subdivide boxes into subboxes;
  - Classify each children:
    - If FREE, then add into the Union Find;
    - If MIXED, then add into the Q;
    - If STUCK or ε-small, then discard.

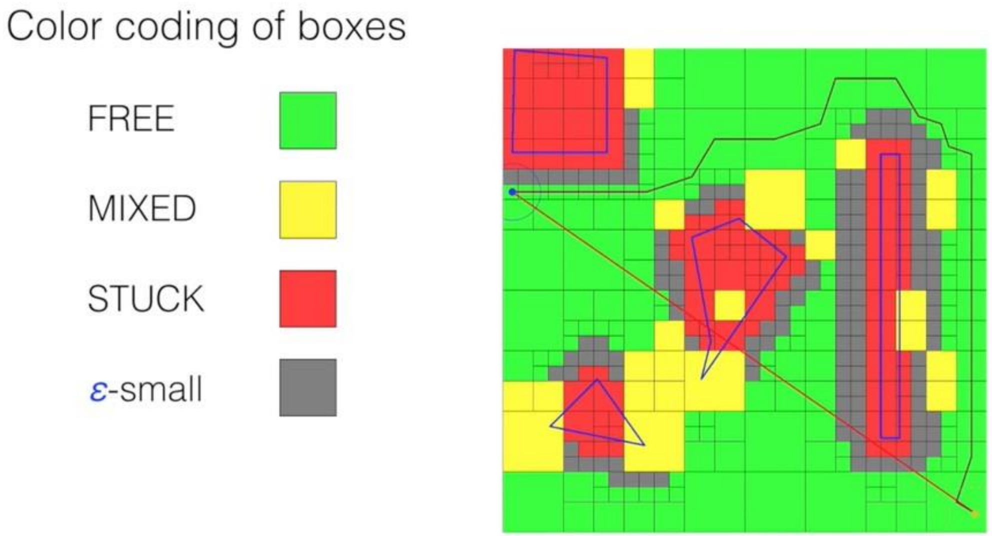
SSS Framework

Input: Start configuration α, goal configuration β, obstacle Ω, resolution parameter ε.
Output: A path $\bar{P}$ or NO-PATH.

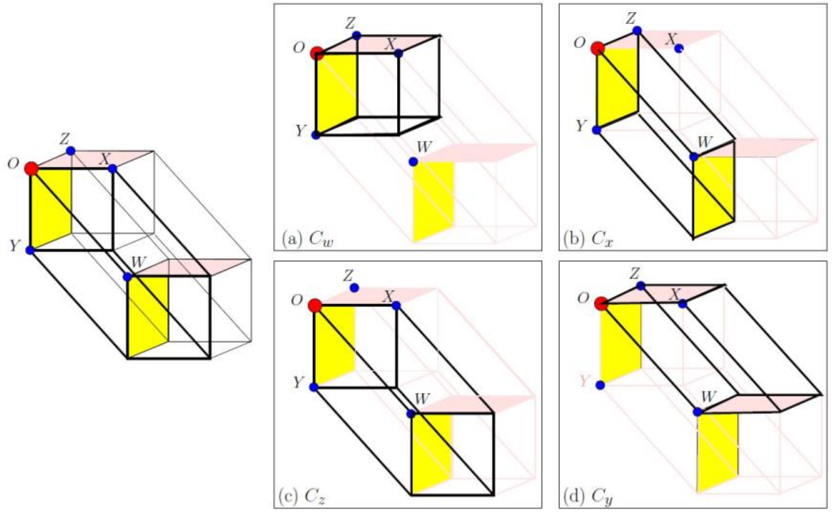
1. $\triangleright$ *Initialization*
   While ($\widetilde{C}(\text{Box}(\alpha)) \neq$ FREE),
   if $l(\text{Box}(\alpha)) < \varepsilon$, return NO-PATH;
   else, Expand(Box(α)).
   While ($\widetilde{C}(\text{Box}(\beta)) \neq$ FREE),
   if $l(\text{Box}(\beta)) < \varepsilon$, return NO-PATH;
   else, Expand(Box(β)).

2. $\triangleright$ *Main Loop*
   While (Find(Box(α)) $\neq$ Find(Box(β))),
   if Q is empty, return NO-PATH
   $B \leftarrow Q$.GetNext()
   Expand(B).

3. $\triangleright$ *Search*
   Compute a FREE channel P from Box(α) to Box(β)
   Generate and return the canonical path $\bar{P}$ inside P.



Subdivision Process



(a) C_w (b) C_x (c) C_z (d) C_y

- Box space:
  - This is a correspondence from $\mathbb{R}^7$ to the configuration space $SE(3)$.
  - The configuration space is $SE(3) \cong \mathbb{R}^3 \times SO(3)$.
    - Boxes in $\mathbb{R}^3$ are **translational** boxes.
    - Boxes in $SO(3)$ are **rotational** boxes (embedded into $\mathbb{R}^4$).
- Subdivide and classify:
  - Expand the boxes containing α and β until they are contained in FREE boxes;
  - Expand the “next” box in Q;
  - Stop when the boxes containing α and β are in the same connected components, or the Q is empty.
- Build a FREE channel from the connected components of α and β.

NYU

Approximate Footprint

- The soft predicate will be given by an approximate footprint $\widetilde{Fp}$.
  - The obstacle set Ω will be inputted as a set of **features** Φ consists of points, edges, triangles and polyhedrons.
  - The soft predicate is defined as

$$\tilde{C}(B) = \begin{cases} \text{FREE} & \widetilde{Fp}(B) \wedge \Phi = \emptyset \text{ and } \widetilde{Fp}(B) \nsubseteq \Omega \\ \text{STUCK} & \widetilde{Fp}(B) \wedge \Phi = \emptyset \text{ and } \widetilde{Fp}(B) \subseteq \Omega \\ \text{MIXED} & \widetilde{Fp}(B) \wedge \Phi \neq \emptyset \end{cases}$$

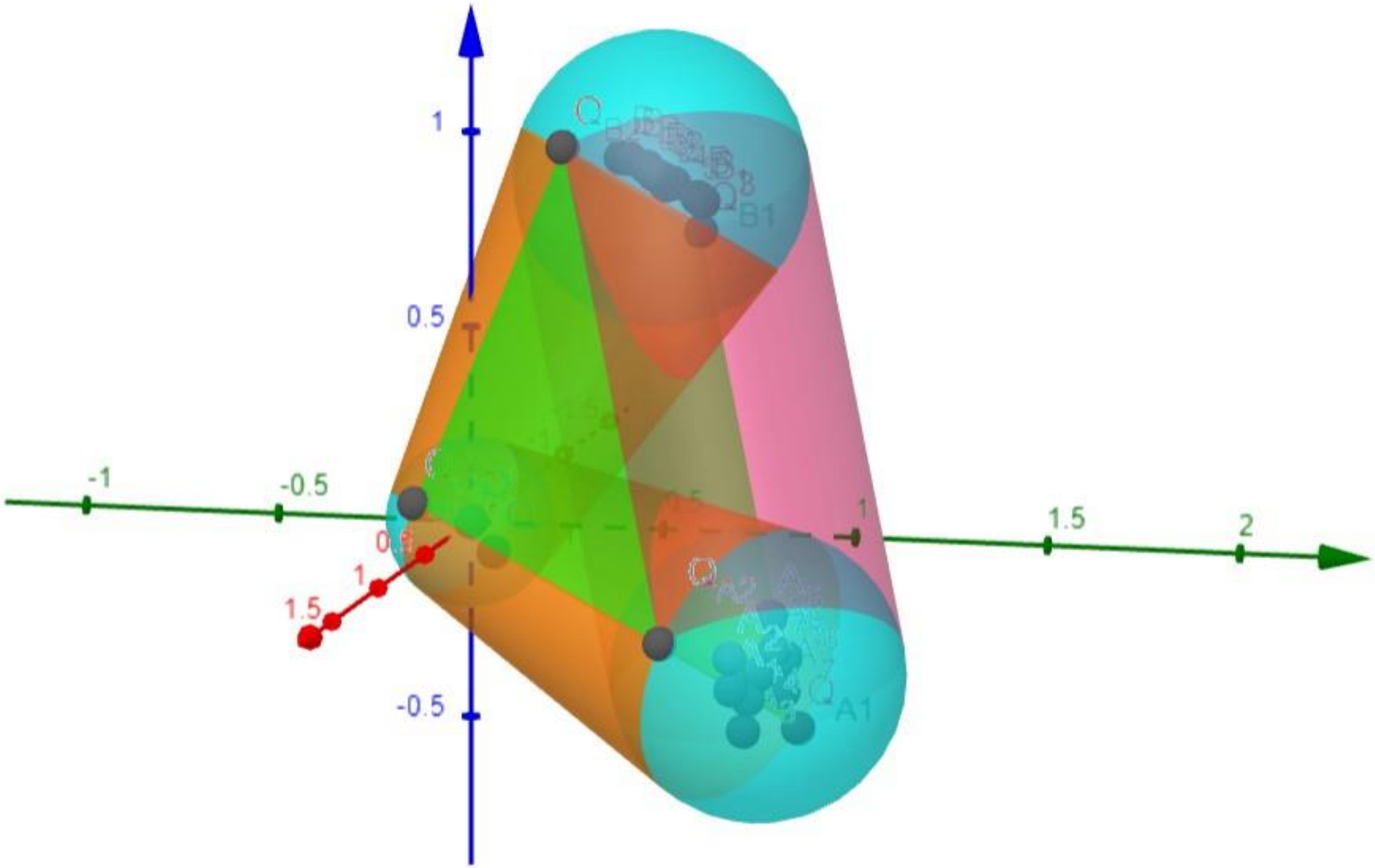
- To make the soft predicate conservative and convergent, the approximate footprint satisfies : there is some $\sigma > 1$ such that

$$\widetilde{Fp}(B/\sigma) \subseteq Fp(B) \subseteq \widetilde{Fp}(B)$$

  - This property is called **σ-effectivity**.

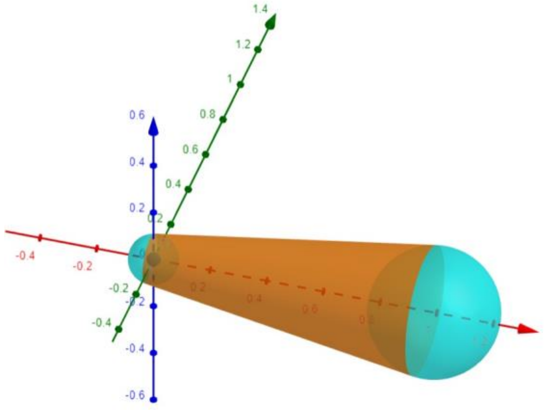
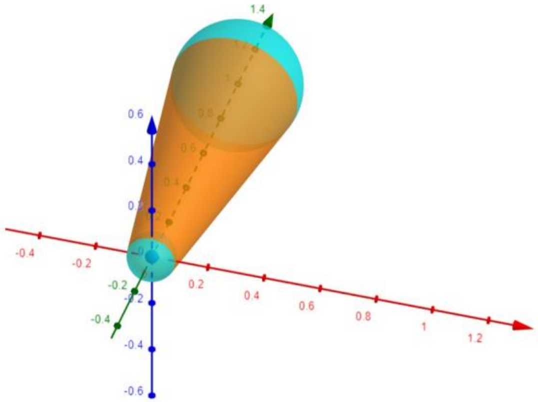
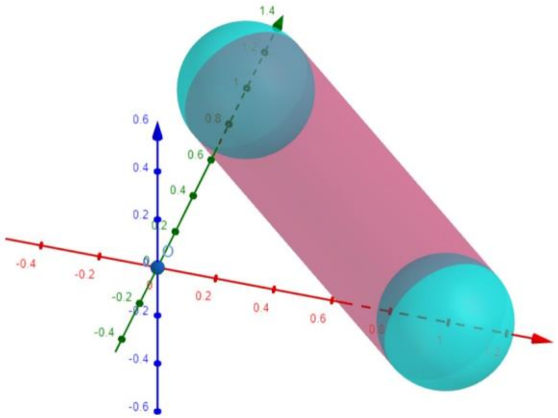
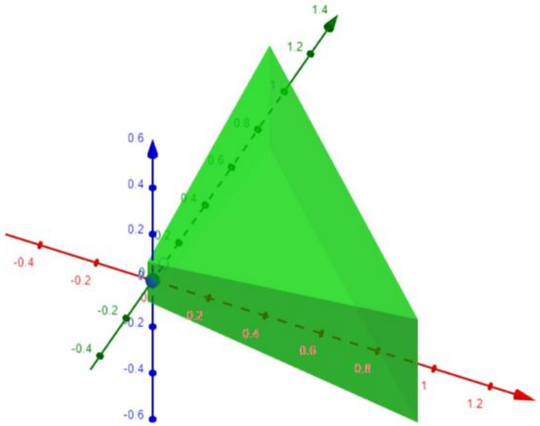
NYU

Approximate Footprint for Delta Robot

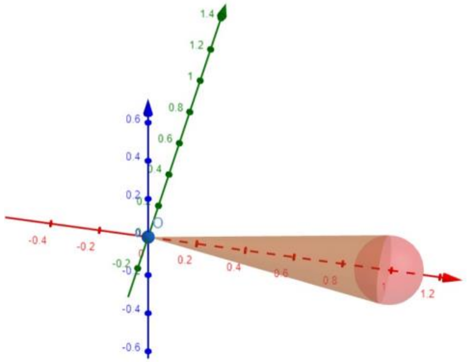


Apply to Delta robot

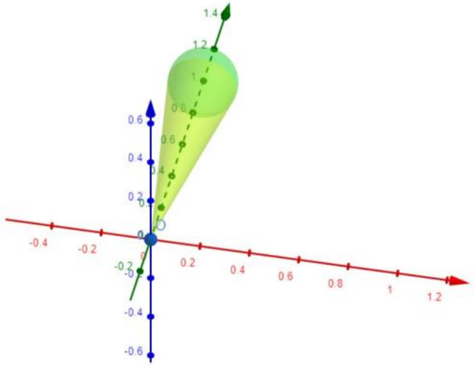
- Regard the approximate footprint as the union of 4 fat sets:

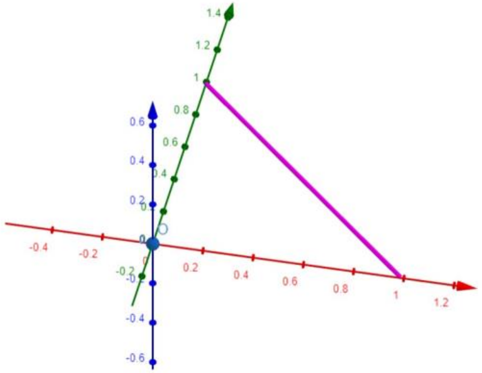
- Turn into Parametric Separation Queries:



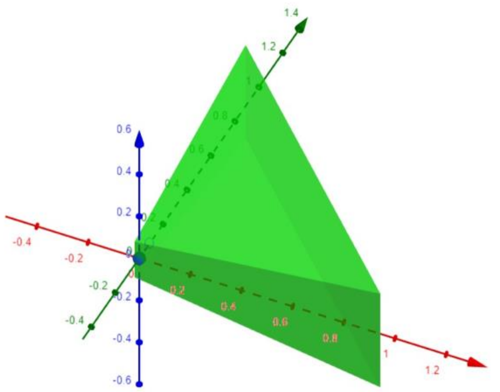
$\mathrm{Sep}(*, f) > r(B)?$



$\mathrm{Sep}(*, f) > r(B)?$

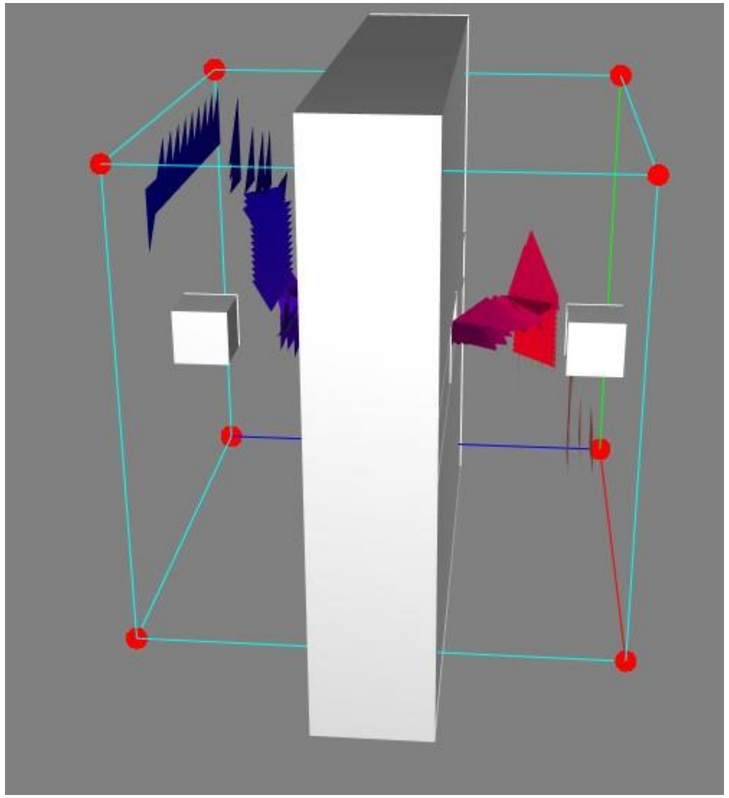


$\mathrm{Sep}(*, f) > d(B)?$



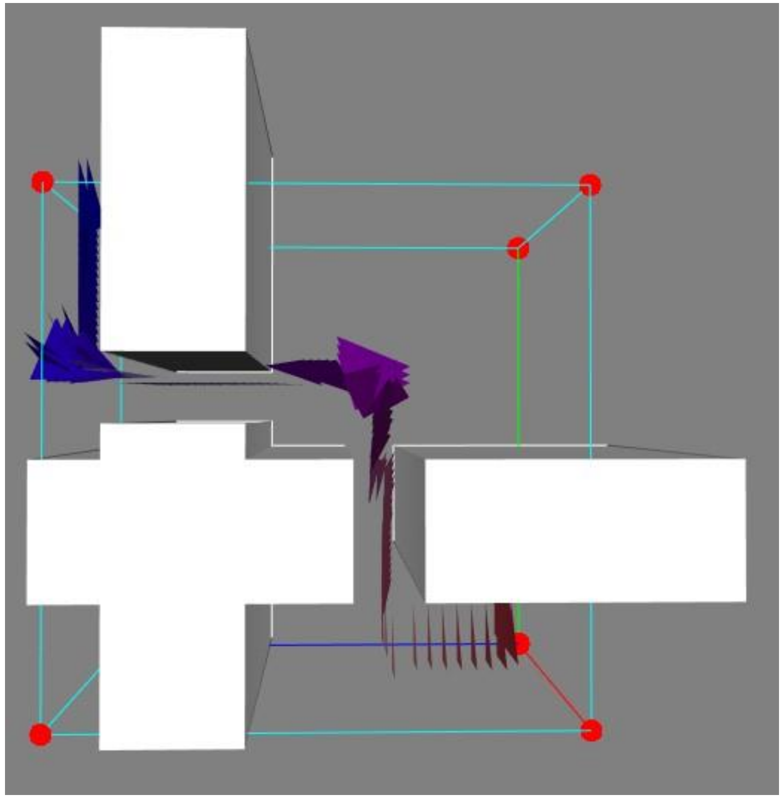
$\mathrm{Sep}(*, f) > 0?$

Performance (Very preliminary)



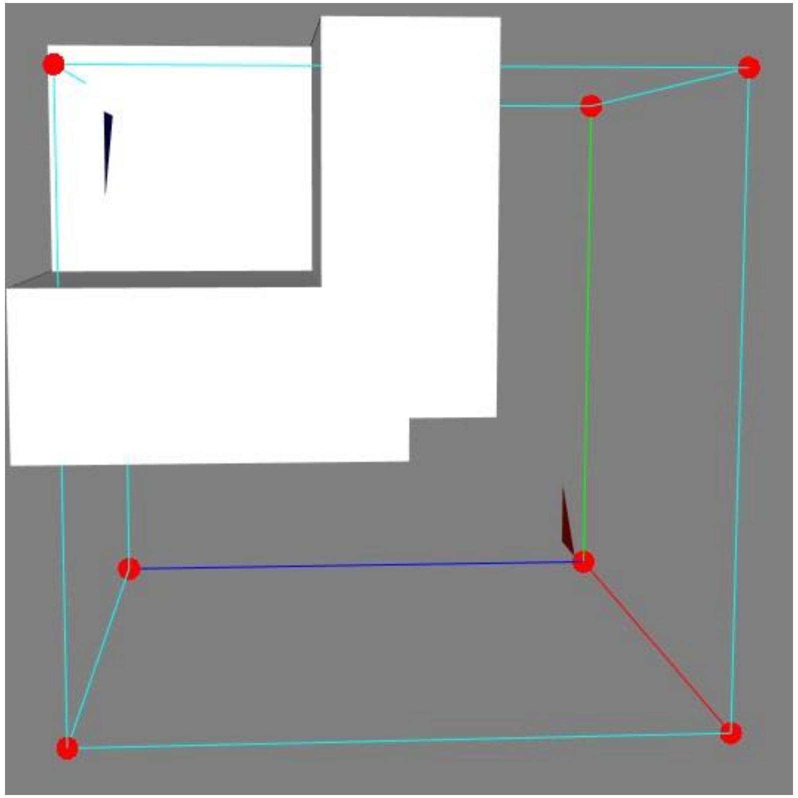
RGB - xyz

This environment find a path with 2996 boxes in 9.52297s.



RGB - xyz

This environment find a path with 8642 boxes in 20.9203s.



RGB - xyz

This environment find NO-PATH with 1866 boxes in 2.90841s.

Conclusion

- This is the first explicit complete $SE(3)$ path planner.
  - Explicit:
    - No invocation of an optimizer.
    - No Newton iteration.
    - No machine learning.
  - All computations are reduced to semi-algebraic tests.

- A full-scale implementation will require additional search techniques (on-going work).

NYU

Reference

[1] C. Wang, Y.-J. Chiang, and C. Yap. On soft predicates in subdivision motion planning. *Comput. Geometry: Theory and Appl. (Special Issue for SoCG'13),* 48(8):589–605, Sept. 2015.
[2] Andreas Orthey and Constantinos Chamzas and Lydia E. Kavraki. Sampling-Based Motion Planning: A Comparative Review. *Annual Reviews.* Vol. 7:285-310, Nov. 2023.
[3] Kavraki, L.E., Latombe, J.C., Motwani, R., Raghavan, P. Randomized query processing in robot path planning. JCSS 57(1), 50–60 (1998)
[4] Karaman, S., Frazzoli, E.: Sampling-based algorithms for optimal motion planning. IJRR 30(7), 846–894 (2011)
[5] Sihui Li and Neil T. Dantam. Exponential Convergence of Infeasibility Proofs for Kinematic Motion Planning. WAFR 22, 294–311 (2023)

Thanks for Listening!

NYU