Rods and Rings: Soft Subdivision Planner for $R^3 \times S^2$

Ching-Hsiang Hsu* **Yi-Jen Chiang**** Chee Yap*

* CS Department, Courant, New York University, USA
** CSE Department, Tandon, New York University, USA

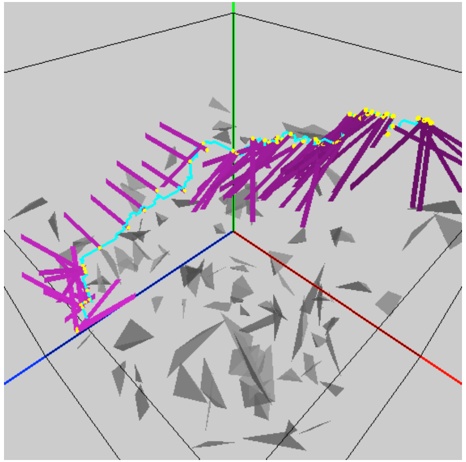
SoCG 2019, Portland, OR, USA. June 2019

Motion Planning

- A central problem in robotics

– There is a fixed rigid robot: $R_0 \subseteq R^k$ (k = 2,3)

– Configuration: pos. & orientation of a point p in R_0



INPUT : (α, β, Ω)

- Start and Goal configurations α, β
- Polyhedral obstacle set $\Omega \subseteq R^k$ (k = 2,3)

OUTPUT:

- A path from α to β avoiding all obstacles in Ω, if it exists.
- Else report “NO PATH”.

State of the Art

(A) Exact Methods

+ Strong theoretical guarantees
- High complexity

e.g., roadmap is single exponential time [Canny 93]

basic path planning is semi-algebraic (book of [Basu-Pollack-Roy])
- Complex to implement & expensive to compute

(rarely implemented and not practical)

(B) Subdivision Methods (e.g., [Zhu-Latombe 91], [Zhang et al 08])

Fairly popular but ``do not scale''

Often degenerate into "grid method"

State of the Art (cont.)

(C) Sampling Methods

* Probabilistic Road Map (PRM) [Kravraki 96];
  many variants: EST, RRT, SRT, etc.
* Dominate the field in the last 2 decades.

Major Issue: Halting Problem ("Narrow Passage" problem) ---
Don't know how to halt when there is no path (except for artificial cut-off)

- Some subdivision work (e.g., [Zhang et al 08]) can detect non-existence of paths, but cannot guarantee to always detect that (sol. is partial).

State of the Art (cont.)

Resolution-Exact Algorithms

- We initiated in [Wang-Chiang-Yap SoCG13], [Yap 13]
- Use subdivision and soft predicates --- Soft Subdivion Search (SSS)
- Avoid exact computation, easy to implement correctly, run fast, always halt, with theoretical guarantees (see paper for details).

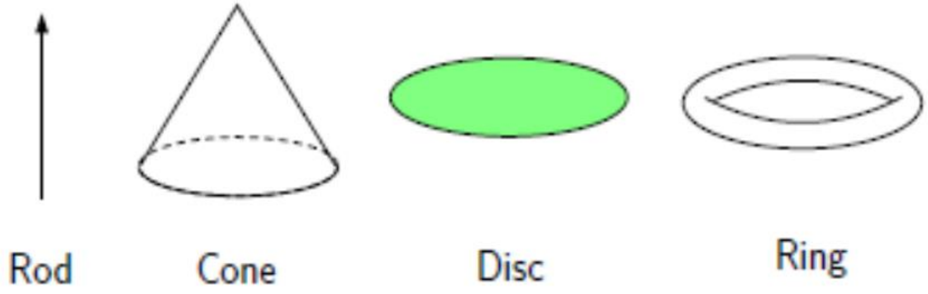
- Extended for 2-link planar robots with 4 degrees of freedom (4 DOFs) [Luo-Chiang-Yap WAFR14], [Chee-Luo-Hsu WAFR16], 2D complex robots [Zhou-Chiang-Yap ESA18]. We are all much faster than sampling methods even for PATH cases (where there is a path)!

- In this paper, we work on 3D, 5-DOF robots under this framework.

New Results: SSS for Rods and Rings ($R^3 \times S^2$)

- 3D rigid robots with an axis of symmetry --- configuration space $C_{space} = R^3 \times S^2$ (5 DOFs)

  Examples:



- Correct, complete and practical path planning for such robots is a long standing challenge.

Review of SSS: Resolution Exactness

- An resolution-exact planner takes an extra input parameter $\varepsilon > 0$. It always halts and outputs either a path or NO-PATH. The output satisfies:

There is an accuracy constant $K > 1$, s.t.

– If exists a path of clearance $K\varepsilon$, it must output a path;

– If there is no path of clearance ε/K, it must output NO-PATH.

– Indeterminacy allowed (small price for avoiding exact computation)

Review of SSS: Basic Search Framework

- Maintain a subdivision tree T rooted at a box B_0 (input domain ⊆ C_{space})
- Each internal node is a box B, which is split into 2^i ($1 \le i \le d$) congruent subboxes (T/R-split: intuitively, first split on R^3 only then on S^2 only)
- Each box B is classified as
  free (each t ∈ B is a free configuration),
  stuck (each t ∈ B is in the exterior of the free space), or
  mixed (otherwise).

- We maintain connected components of free boxes via a Union-Find data structure (α, β ∈ same component → path found)
- Priority Queue Q for mixed boxes to be expanded later

New Major Technical Contributions

I. "Forbidden orientations" used in 2D robots is too complicated in 3D
→ Instead we use approximate footprints of boxes (for rods & rings)

II. Σ_2-Set representation of the approximate footprints
* easy to implement
* allows an easy extension to "thick" versions of rods & rings

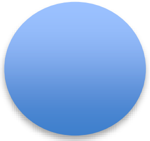
III. For subdivision of $R^3 \times S^2$ --- S^2 is a non-Euclidean space
→ introduce the square model of S^2 * it also avoids singularities

IV. Use Voronoi heuristic for efficient search

Subdivision of Non-Euclidean Space S^2

- Usual representation (singularities at N/S poles, severe distortions): spherical polar coordinates: $(\theta, \phi) \in [0, 2\pi] \times [-\pi/2, \pi/2]$
- Solution: Subdivision charts
  Use invertible map from $S^2 \to \widehat{S^2}$.

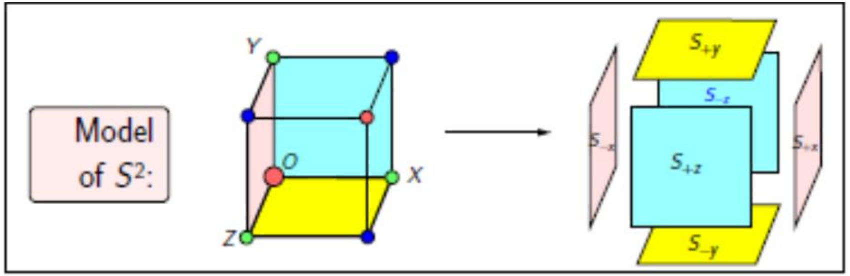
$$\boxed{q \in S^2 \mapsto \widehat{q} := \frac{q}{\|q\|_\infty}}$$



unit sphere S^2

$\downarrow$

$\widehat{S^2}$ Model of S^2



6 faces of enclosing cube *[-1,1]*^{3}

Lemma: The max distortion

$$C_0 := \max_{q \neq p \in S^2} \left\{ \frac{d(p,q)}{\widehat{d}(\widehat{p},\widehat{q})}, \frac{\widehat{d}(\widehat{p},\widehat{q})}{d(p,q)} \right\}$$

is $\sqrt{3}$

Feature Set of a Box

- Box B: we use its feature set to classify B as free/stuck/mixed.
- Let $\phi(\Omega)$ be set of obstacle boundary features f (corners, edges, walls)
- *Fp(B)*: union of the footprints of the robot when its configuration is in B.
- Feature set $\phi(B) := \{ f \in \phi(\Omega) : f \cap Fp(B) \neq \emptyset \}$

  (i.e., all *f* that are potentially in conflict with the robot when its configuration is in B.)
- **Inheritance:** $\phi(child(B)) \subseteq \phi(B)$ → *split box until its $\phi()$ is empty*
- **Classification:**

  $\phi(B)$ is empty: B is in no conflict with any obstacle boundary ➔

  B is **free** or **stuck** (use parent feature set to decide)
- **Softness**: replace $\phi(B)$ with approx. feature set $\tilde{\phi}(B)$ to classify B.

Key: Softness --- Approximate Footprint & Feature Set

- *Fp(B)*: the footprint of B. Let $\widetilde{\boldsymbol{Fp}}(B)$ be the **approximate** footprint of B.
- We require the fundamental inclusions (for some global constant $\sigma > 1$):

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

2[nd] inclusion: *conservative*
1[st] inclusion: *effective*

- Recall: feature set $\phi(B) := \{ f \in \phi(\Omega) : f \cap Fp(B) \neq \emptyset \}$
  Approximate feature set $\tilde{\phi}(B) := \{ f \in \phi(\Omega) : f \cap \widetilde{\boldsymbol{Fp}}(B) \neq \emptyset \}$ (a)

- We require: (**) $\tilde{\phi}(B/\sigma) \subseteq \phi(B) \subseteq \tilde{\phi}(B)$ (like (*))
  also want: **[inheritance]** $\tilde{\phi}(child(B)) \subseteq \tilde{\phi}(B)$ (like $\phi(B)$)

Note: Def. (a) fulfills (**) but not [inheritance].

Key: Softness --- Approximate Footprint & Feature Set (cont.)

- Recall: the fundamental inclusions: (*) $\boxed{\widetilde{Fp}(B/\sigma) \subseteq Fp(B) \subseteq \widetilde{Fp}(B)}$

  approximate feature set $\tilde{\phi}(B) := \{ f \in \phi(\Omega) : f \cap \widetilde{\boldsymbol{Fp}}(B) \neq \emptyset \}$ (a)

- Re-define approx. feature set: ($\widetilde{\phi}'(B/\sigma)$ is defined similarly) [inheritance]

$$\boxed{\widetilde{\phi}'(B) := \begin{cases} \left\{ f \in \Phi(\Omega) : f \cap \widetilde{Fp}(B) \neq \emptyset \right\} & \text{if } B \text{ is the root,} \\ \left\{ f \in \widetilde{\phi}'(parent(B)) : f \cap \widetilde{Fp}(B) \neq \emptyset \right\} & \text{else.} \end{cases}}$$

Lemma: If approx. footprint $\widetilde{\boldsymbol{Fp}}(B)$ fulfills fundamental inclusions (*), then $\widetilde{\phi}'(B)$ satisfies (**) (like (*); see previous slide), as desired.

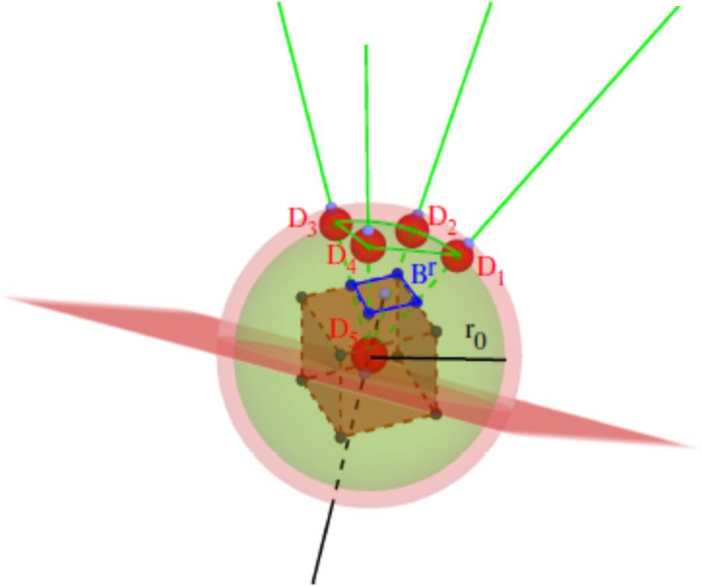
* We will write "$\tilde{\phi}(B)$" to refer to $\widetilde{\phi}'(B)$ *(ignore (a))*

Σ_2-Set Representation of the Approx. Footprints

- An *elementary set* is one of
  half space, ball, ring, cone, or cylinder,
  or the complement of one.
- A Σ_2-set has the form $\bigcup_{i=1}^{n} \bigcap_{j=1}^{m_i} S_{ij}$
  where each S_{ij} is elementary
- Represent each approx. footprint by a Σ_2-set $\rightarrow$
  intersection test $(f \cap \widetilde{\boldsymbol{Fp}}(B) \neq \emptyset?)$ becomes very simple $(f \cap S_{ij} \neq \emptyset?, etc)$
  [* also allows an easy extension to "thick" versions of rods & rings]

Exact & Approx. Footprints of a Rod Robot

- A rod robot: length r_0 & one endpoint p as the rotation center
- A box $B = B^t \times B^r$ ($B^t \subseteq R^3$: translational box, center m_B , radius r_B ;
  $B^r \subseteq \widehat{S^2}$: rotational box)

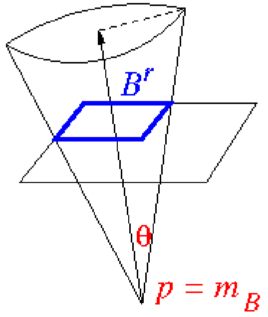


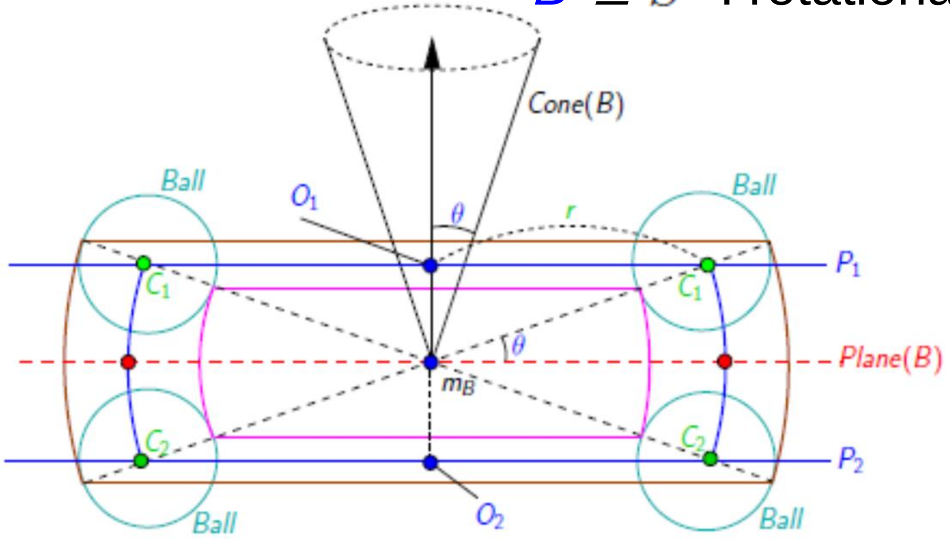
- $Fp(m_B \times B^r)$: square cone: 4 green rays & green ball (center m_B, radius r_0) = $H_1 \cap H_2 \cap H_3 \cap H_4 \cap$ *green ball* $B(m_B, r_0)$
  (H_i : half space)

  (Minkowski sum)
- $Fp(B^t \times B^r) = Fp(m_B \times B^r) \oplus$ *ball* $B(r_B)$
  [$D_1 \sim D_5$: balls of radius r_B]
- $\widetilde{\boldsymbol{Fp}}(B) := \cap_i$ (H_i expanded by r_B) $\cap$ *pink ball* $B(m_B, r_0 + r_B) \cap H_5$ *thru pink plane*

Exact & Approx. Footprints of a Ring Robot

- A ring robot: embedded circle with center p and radius r_0
  Orientation: normal of the embedding plane (*Plane(B)*).
- A box $B = B^t \times B^r$ ($B^t \subseteq R^3$: center m_B , radius r_B
  $B^r \subseteq \widehat{S^2}$: rotational box)





* Central cross section of $Fp_1(B)$ appears as two blue arcs.

* $\widetilde{\boldsymbol{Fp}}(B) :=$ the union of two "thick rings" and a "truncated annulus".

* The axis of *Cone(B)* is shown as a vertical ray. Each *Ball* has radius r_B.

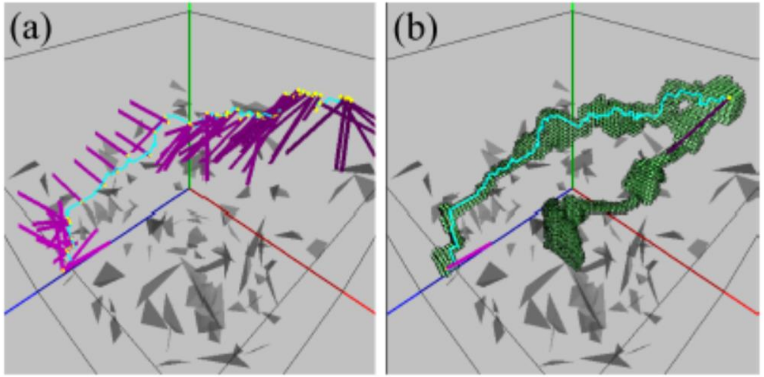
Properties

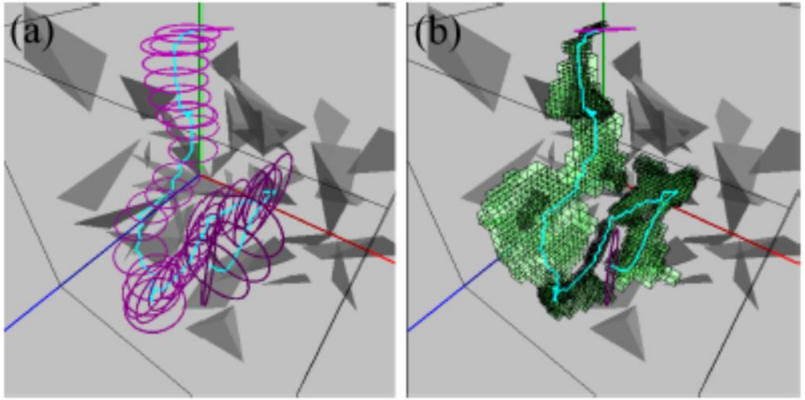
- Recall: the fundamental inclusions: (*) $\boxed{\widetilde{Fp}(B/\sigma) \subseteq Fp(B) \subseteq \widetilde{Fp}(B)}$

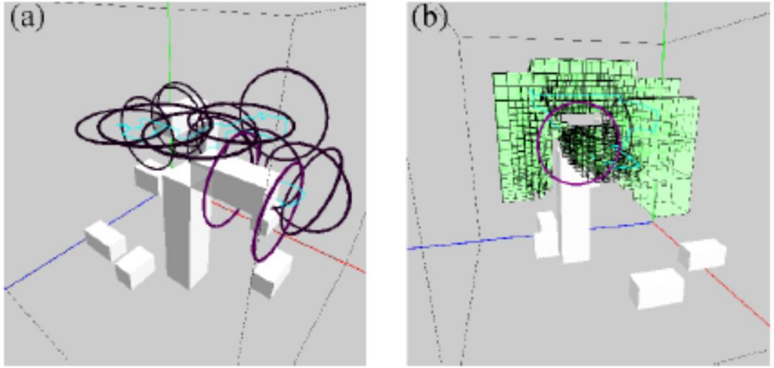
- **Theorem:** The approx. footprint $\widetilde{\boldsymbol{Fp}}(B)$ defined for the rod robot fulfills the fundamental inclusions (*).

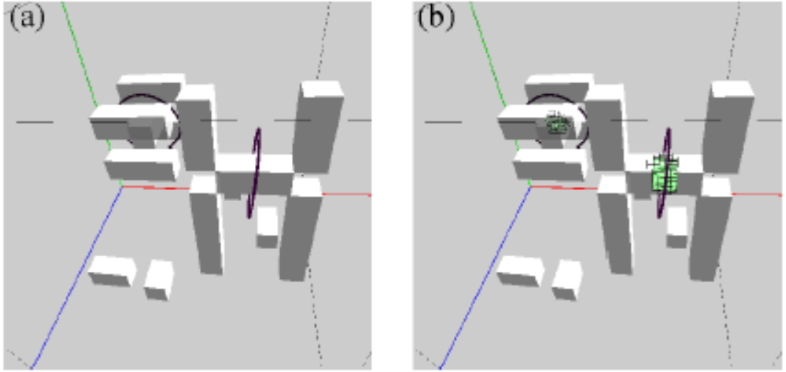
- **Theorem:** The approx. footprint $\widetilde{\boldsymbol{Fp}}(B)$ defined for the ring robot fulfills the fundamental inclusions (*).

Experimental Results: Some Screen Shots









Experimental Results: Characteristics of Our Methods

Rod Robot								
Exp. #	Envir.	Length	ε	Start Conf.	Goal Conf.	Path	Time (s)	#Boxes (K)
1/2	Rand100	120	16/8	(240, 120, 360, -0.5, -0.5, -1)	(220, 50, 80, 0.1, 0.8, 1)	Y/Y	1.05/2.82	8.1/22.1
3/4	Rand100	120	16/8	(400, 60, 380, -1, 0, 0)	(200, 200, 240, 0, 1, 0)	Y/Y	1.43/3.92	19.3/62.2
5/6	Rand40	160	16/8	(80, 32, 480, 0, 0, -1)	(240, 440, 200, 1, 0, 0)	Y/Y	16.12/90.65	244.7/ 1138.2
7/8	Rand40	160	16/8	(400, 480, 80, 0, -1, 0)	(30, 80, 480, 0.5, 0.1, -1)	Y/Y	14.54/9.4	217.5/113.0
9/10	Posts	60	16/8	(160, 480, 190, 0, 0.1, -1)	(390, 60, 420, 1, 0, 0)	Y/Y	0.07/0.13	2.1/3.7
11/12	Posts	60	16/8	(320, 120, 320, 0, 1, 0)	(200, 360, 60, 0, -1, 0)	N/N	1.77/242.7	25.6/3790.3

Ring Robot								
Exp. #	Envir.	Radius	ε	Start Conf.	Goal Conf.	Path	Time (s)	#Boxes (K)
1/2	Rand100	40	16/8	(240, 120, 360, -0.5, -0.5, -1)	(220, 50, 80, 0.1, 0.8, 1)	Y/Y	0.66/0.57	2.72/2.63
3/4	Rand100	40	16/8	(400, 60, 380, -1, 0, 0)	(160, 240, 240, 0, 1, 0)	Y/Y	0.25/0.24	0.92/0.92
5/6	Rand40	60	16/8	(80, 120, 480, 0, 0, -1)	(240, 440, 200, 1, 0, 0)	Y/Y	9.38/30.61	38.37/71.05
7/8	Rand40	60	16/8	(400, 480, 80, 0, -1, 0)	(100, 80, 480, 0.5, 0.1, -1)	Y/Y	2.07/3.76	10.61/13.90
9/10	Posts	60	16/8	(200, 320, 190, 0, 0.1, -1)	(390, 320, 320, 1, 0, 0)	Y/Y	35.68/89.7	114.3/139.3
11/12	Posts2	60	16/8	(410, 90, 190, 0, 0.1, -1)	(315, 220, 325, 0, 1, 0)	N/N	7.03/271.3	8.1/539.1

Experimental Results: Comparison with OMPL Sampling Methods

Rod Robot

Exp. #	Ours	PRM	Lazy PRM	RRT	Lazy RRT	RRT Connect	PDST	BFMT	Lazy Bi-KPIECE
1	1.05/Y	0.036/0.027/1	**0.017**/0.024/1	1.18/0.74/1	0.019/0.023/1	0.22/0.043/1	0.058/0.055/1	1.11/0.18/1	0.58/0.36/1
3	1.43/Y	0.05/0.047/1	0.028/0.019/1	1.73/0.82/1	**0.024**/0.024/1	0.23/0.023/1	0.1/0.056/1	1.51/0.2/1	0.59/0.28/1
5	16.12/Y	0.044/0.036/1	0.051/0.025/1	22.1/43/1	**0.036**/0.032/1	0.99/0.36/1	0.21/0.11/1	1.74/0.33/1	0.44/0.18/1
7	14.54/Y	0.077/0.04/1	**0.03**/0.02/1	10.31/6.08/1	0.033/0.023/1	1.26/0.5/1	0.26/0.2/1	1.74/0.32/1	0.5/0.21/1
9	0.07/Y	0.0058/0.002/1	0.0038/0.0044/1	1.17/0.78/1	**0.003**/0.002/1	0.084/0.017/1	0.025/0.025/1	0.3/0.053/1	0.065/0.032/1
11	**1.77**/N	300/0/0	300/0/0	300/0/0	300/0/0	300/0/0	300/0/0	300/0/0	300/0/0

Ring Robot

Exp. #	Ours	PRM	Lazy PRM	RRT	Lazy RRT	RRT Connect	PDST	BFMT	Lazy Bi-KPIECE
1	0.66/Y	0.0057/0.0026/1	**0.0056**/0.005/1	1.15/0.93/1	0.0061/0.009/1	0.037/0.005/1	0.015/0.01/1	0.15/0.01/1	0.077/0.035/1
3	0.25/Y	0.0085/0.0067/1	**0.003**/0.002/1	24.16/54/1	0.012/0.0085/1	0.052/0.011/1	0.008/0.008/1	0.145/0.023/1	0.068/0.032/1
5	9.38/Y	0.019/0.014/1	**0.01**/0.004/1	300/0/0	150.07/10.05/0.5	0.53/0.03/1	0.057/0.023/1	0.22/0.04/1	0.093/0.022/1
7	2.07/Y	0.024/0.0066/1	**0.013**/0.006/1	3/1.49/1	0.068/0.007/1	1.46/0.14/1	0.059/0.044/1	0.27/0.048/1	0.12/0.027/1
9	35.68/Y	1.25/1.6/1	12.45/14.7/1	67.56/116.6/0.8	290.1/129.5/0.2	1.77/0.69/1	2.74/2.73/1	0.26/0.06/1	**0.072**/0.0246/1
11	**7.03**/N	300/0/0	300/0/0	300/0/0	300/0/0	300/0/0	300/0/0	300/0/0	300/0/0

Summary of Experiments

Comparing with 8 sampling methods (PRM, Lazy PRM, RRT, Lazy RRT, RRT Connect, PDST, BFMT, Lazy Bi-KPIECE) in open-source library OMPL.

- We achieve near real-time, and can report NO-PATH.
- We usually outperform RRT in cases of PATH.
- In cases of PATH, RRT and Lazy RRT may have unsuccessful runs and need to time out, while we find paths & are much faster. Otherwise, OMPL methods are typically fast.
- In cases of NO-PATH, all OMPL methods timed out (300s) while we stopped at real time (much faster).

Video Demo

- Video is available at (updated from the paper's link) **https://cs.nyu.edu/exact/gallery/rod-ring/rod_ring.html**

- Code is available (updated from the paper's link): **Core Library** **https://cs.nyu.edu/exact/core_pages/downloads.html**

Conclusions

- We have "reached" 5 DOFs!
- There are many interesting 3D robots in this class.
- We can easily extend to "fat rods" and "fat rings".
- No comparable rigorous algorithm is known.
- Even with much stronger theoretical guarantees, our SSS methods are still typically 1-2 orders of magnitude faster than OMPL sampling methods even for PATH cases for ≤ 4 DOFs. We don't see that for 5 DOFs now. We believe that better search methods/heuristics are more crucial now. This is a largely open area.