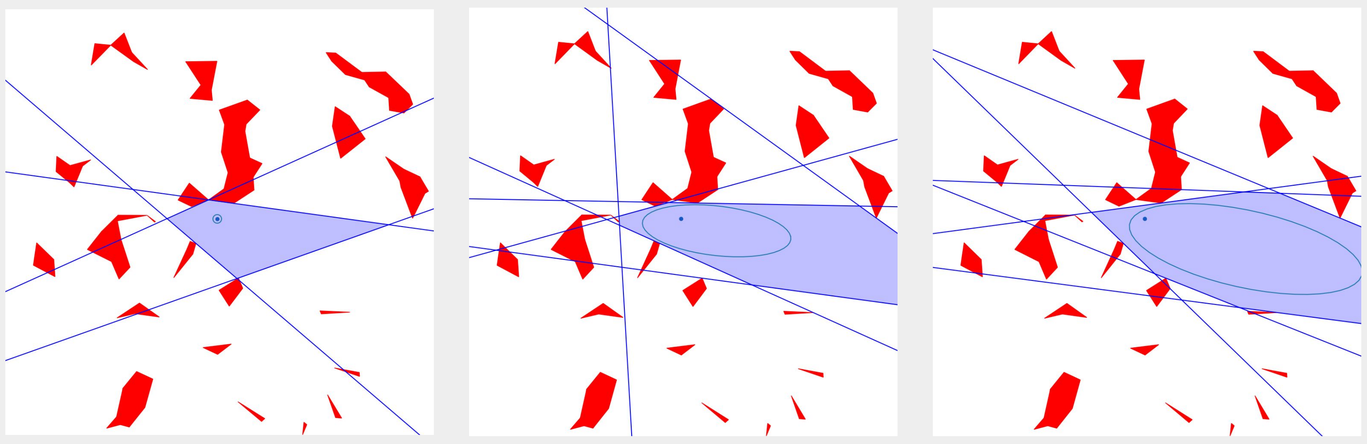
Faster Algorithms for Growing Collision-Free Regions of Constrained Bimanual Configuration Spaces

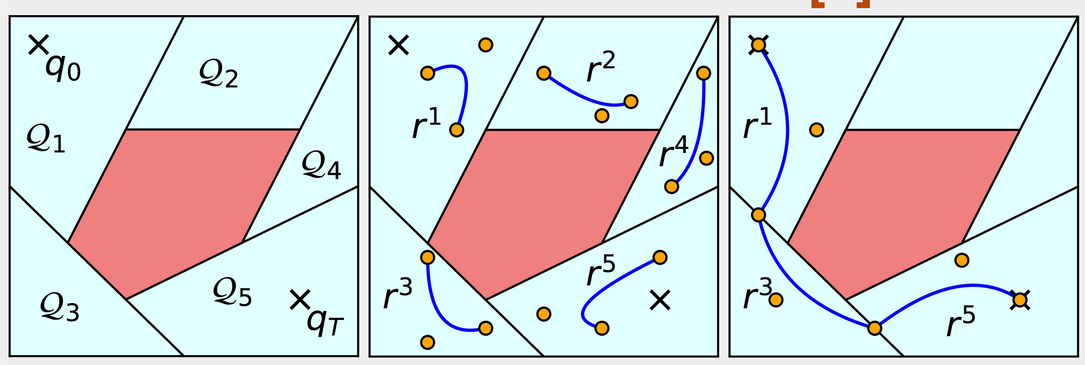
Thomas Cohn[1], Peter Werner[1], Russ Tedrake[1]

[1]Massachusetts Institute of Technology

IRIS [1] Algorithms Build Convex Collision-Free Sets



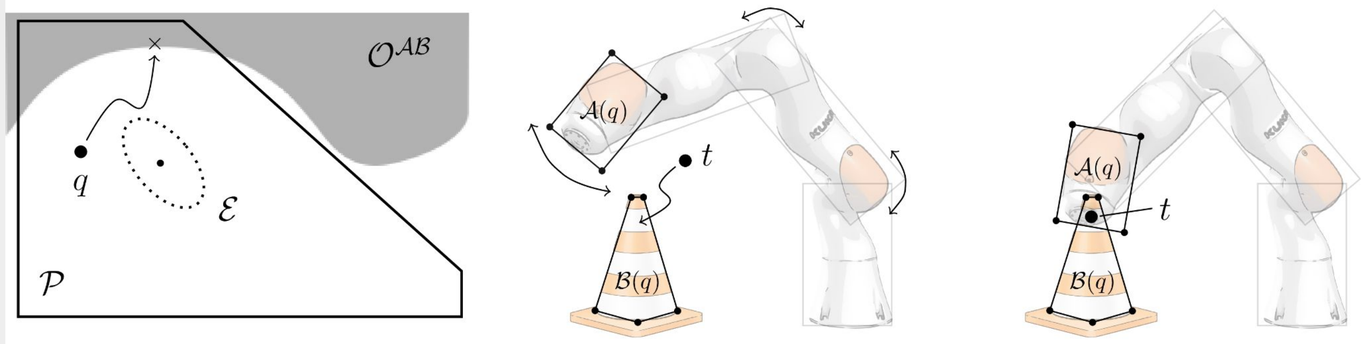
Collision-Free Motion Planning through Convex Sets with GCS [4]



A collision-free trajectory planned through convex sets. Figure reproduced from [4].

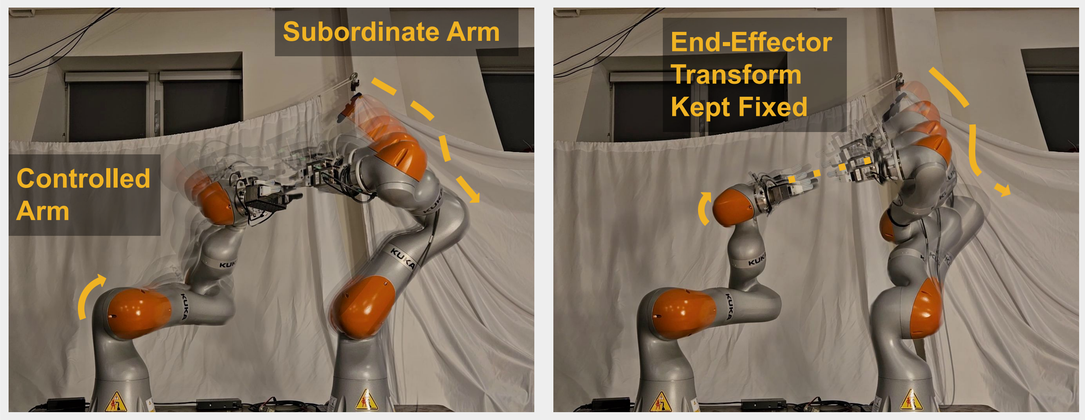
IRIS Can Use Nonlinear Programming for C-Space [2]

$$\underset{q,\,t}{\textbf{minimize}}\ ||q-c||_E^2 \ \textbf{subject to}\ t \in \mathcal{A}(q) \cap \mathcal{B}(q),\ q \in \mathcal{P}$$



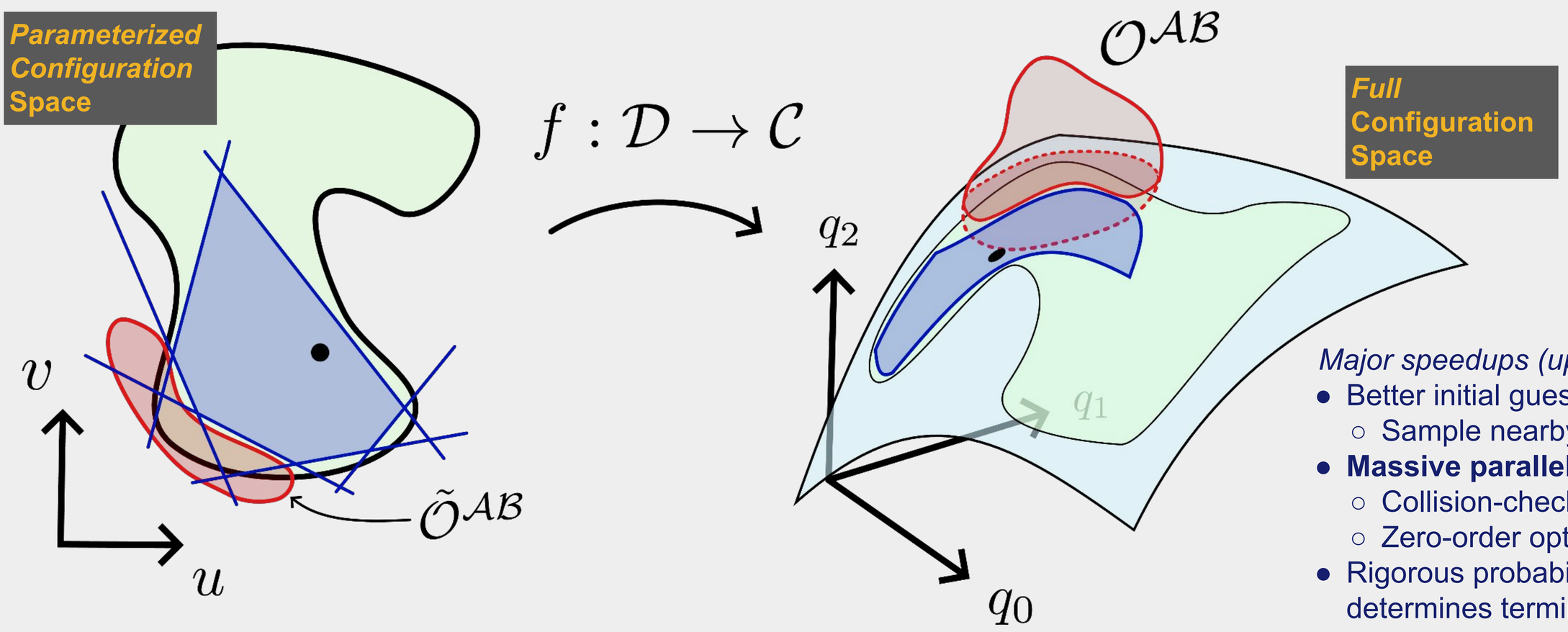
Counterexample search program used by IRIS-NP [2]. Figure reproduced from [3].

Minimal Coordinates of a Constrained Bimanual Configuration Space [5]



Parametrizing the constraint manifold removes nonlinear equality constraints.

General Setup: Growing Regions in a Parametrized Configuration Space



*Major speedups (up to **500x**!)*

- Better initial guess for optimizers
  - Sample nearby collisions first
- **Massive parallelization**
  - Collision-checking
  - Zero-order optimization
- Rigorous probabilistic certificate determines termination

Comparison with Baseline [5]

Seed	Time [s]				Number of Hyperplanes				Relative Volume				Fraction in Violation [%]			
Point	NP	ZO	Greedy	Ray	NP	ZO	Greedy	Ray	NP	ZO	Greedy	Ray	NP	ZO	Greedy	Ray
1	273.80	1.55	**1.02**	1.89	369	69	**53**	**53**	1	2.19×10^{-3}	4.23×10^{2}	$\mathbf{6.81 \times 10^{2}}$	5.87	0.00	0.01	0.29
2	360.70	1.30	**1.03**	1.49	416	62	54	**41**	1	4.36×10^{-4}	3.04×10^{2}	$\mathbf{5.03 \times 10^{2}}$	2.14	0.13	0.02	0.22
3	273.87	2.98	**1.35**	3.00	179	129	54	**52**	1	4.62	5.54×10^{3}	$\mathbf{5.81 \times 10^{3}}$	0.00	0.29	0.09	0.30
4	296.34	**3.79**	4.01	6.26	302	153	63	**56**	1	1.64×10^{-1}	1.83×10^{1}	$\mathbf{3.59 \times 10^{1}}$	0.00	0.11	0.09	0.42
5	280.35	3.14	**1.47**	2.62	370	132	70	**60**	1	2.29×10^{-1}	1.05×10^{1}	$\mathbf{1.16 \times 10^{1}}$	0.08	0.23	0.03	0.23
6	375.20	**1.17**	1.34	1.34	481	53	51	**36**	1	6.49×10^{-3}	1.91×10^{1}	$\mathbf{2.16 \times 10^{2}}$	6.31	0.01	0.16	0.55
7	233.16	**1.50**	2.17	1.75	185	90	57	**46**	1	1.60×10^{-1}	$\mathbf{2.42 \times 10^{2}}$	2.42×10^{2}	0.00	0.11	0.04	0.46
8	428.44	1.97	**1.61**	3.38	234	102	56	**50**	1	5.71	3.42×10^{3}	$\mathbf{3.93 \times 10^{3}}$	0.00	0.17	0.06	0.06
9	390.54	2.84	**1.32**	1.40	216	130	50	**49**	1	2.18	$\mathbf{2.25 \times 10^{3}}$	1.73×10^{3}	0.00	0.32	0.06	0.32
10	353.06	**1.29**	1.79	2.47	579	61	62	**47**	1	0.99×10^{-2}	1.08×10^{1}	$\mathbf{1.26 \times 10^{1}}$	3.09	0.03	0.06	0.39
11	300.84	**0.74**	1.65	1.89	436	**38**	54	39	1	6.72×10^{-3}	3.96×10^{2}	$\mathbf{4.61 \times 10^{2}}$	3.59	0.14	0.03	0.31
12	295.49	**1.22**	1.33	1.81	396	66	53	**46**	1	5.03×10^{-3}	$\mathbf{2.48 \times 10^{2}}$	1.89×10^{2}	3.79	0.09	0.03	0.09
13	433.61	**1.18**	1.27	1.73	420	51	57	**50**	1	5.66×10^{-3}	$\mathbf{2.82 \times 10^{1}}$	2.30×10^{1}	2.49	0.04	0.04	0.12
14	254.72	**1.34**	2.03	3.71	454	72	56	**53**	1	9.21×10^{-2}	2.62×10^{1}	$\mathbf{4.30 \times 10^{1}}$	0.35	0.15	0.15	0.47
15	243.63	1.52	**1.06**	2.06	346	64	51	**44**	1	1.04×10^{-1}	2.97×10^{4}	$\mathbf{4.47 \times 10^{4}}$	0.45	0.04	0.02	0.24
16	290.49	**1.41**	1.71	1.44	226	72	58	**42**	1	9.21×10^{-1}	2.41×10^{3}	$\mathbf{4.66 \times 10^{3}}$	0.00	0.16	0.05	0.15
17	248.97	2.14	**1.98**	2.63	252	107	61	**59**	1	1.05	2.96×10^{2}	$\mathbf{3.69 \times 10^{2}}$	0.00	0.21	0.04	0.15
18	263.98	2.22	**1.52**	2.31	299	109	63	**55**	1	1.08	4.45×10^{2}	$\mathbf{2.92 \times 10^{3}}$	0.10	0.20	0.06	0.47
19	216.20	**1.35**	1.72	1.60	383	66	52	**50**	1	4.01×10^{-2}	7.84×10^{1}	$\mathbf{9.09 \times 10^{1}}$	1.04	0.07	0.07	0.24
Min	216.20	**0.74**	1.02	1.34	179	38	50	**36**	1	3.11×10^{-3}	3.54×10^{2}	$\mathbf{3.99 \times 10^{2}}$	0.00	0.00	0.01	0.06
Mean	305.97	1.82	**1.65**	2.36	344	85	56	**48**	1	2.30×10^{-1}	0.99×10^{2}	$\mathbf{1.22 \times 10^{2}}$	1.54	0.13	0.06	0.29
Max	433.61	**3.79**	4.01	6.26	579	153	70	**60**	1	1.64×10^{-1}	3.18×10^{1}	$\mathbf{3.59 \times 10^{1}}$	6.31	0.32	0.16	0.55

Our convex regions are...

- ***Constraint-Satisfying***
- ***Collision-Free***
- ***Amenable to Planning***

And our algorithms are ***faster*** and ***more reliable***

References

[1] *Deits, R. and Tedrake, R., 2015, April. Computing large convex regions of obstacle-free space through semidefinite programming. In Algorithmic Foundations of Robotics XI: Selected Contributions of the Eleventh International Workshop on the Algorithmic Foundations of Robotics (pp. 109-124). Cham: Springer International Publishing.*

[2] Petersen, M. and Tedrake, R., 2023. Growing convex collision-free regions in configuration space using nonlinear programming. arXiv preprint arXiv:2303.14737.

[3] Werner, P., Cohn, T., Jiang, R.H., Seyde, T., Simchowitz, M., Tedrake, R. and Rus, D., 2024. Faster algorithms for growing collision-free convex polytopes in robot configuration space. arXiv preprint arXiv:2410.12649.

[4] Marcucci, T., Petersen, M., von Wrangel, D. and Tedrake, R., 2023. Motion planning around obstacles with convex optimization. Science robotics, 8(84), p.eadf7843.

[5] Cohn, T., Shaw, S., Simchowitz, M. and Tedrake, R., 2024, May. Constrained bimanual planning with analytic inverse kinematics. In 2024 IEEE International Conference on Robotics and Automation (ICRA) (pp. 6935-6942). IEEE.