Optimisation-based Manipulation Planning in Convex Decompositions of C-free

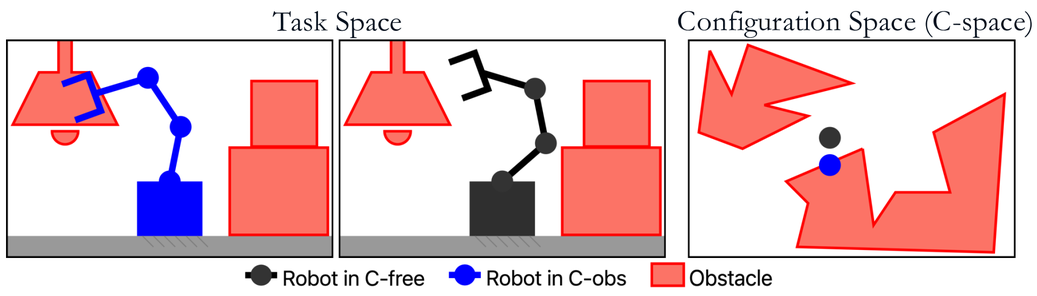
VIDEO DEMOS

Julia López Gómez Dr. Steve Tonneau

Robots navigate in environments with obstacles. Traditionally, trajectory generation has relied on sampling techniques, which cannot guarantee non-collision and only reach asymptotic optimality. Recent algorithms have utilised convex decompositions to describe the *collision-free space* in which robots can move, providing certificates of non-collision and global optimality within the decomposition; however, these methods have not been explored in manipulation yet. We propose a novel optimisation-based manipulation planner that uses these decompositions to generate globally optimal manipulation trajectories.

1. CONFIGURATION SPACE OBSTACLES

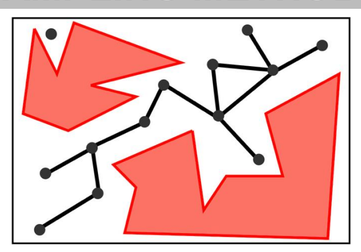
The **Configuration Space (C-space)** represents each robot posture *(configuration)*, which can be in collision (in C-obs) or not (C-free).



2. C-free DECOMPOSITION

We want to represent the Collision-free Configuration Space (C-free).

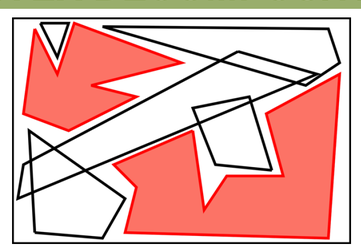
SAMPLING METHODS



Traditional

- Probabilisic Cert. of non-collision
- Probabilistically complete
- (At most) asymptotic optimality

CONVEX DECOMPOSITION



Since 2023

- Certificates of non-collision
- Completeness
- Global optimality [1][2]

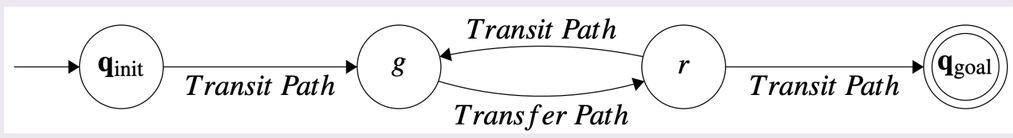
4. OUR MANIPULATION ALGORITHM

Our manipulation planner operates in 3 steps:

1. Convex Decomposition and CG/CP Generation

2. Find Grasp-Release actions that solve the Manipulation Task

Since manipulation trajectories are alternating sequences of transit and transfer paths, we find the *grasp* and *release* configurations that connect them.



We use a Mixed-Integer Quadratic Program to solve this:

$$\min_{\mathbf{x}_0,\dots,\mathbf{x}_{2\ell+1}} \quad \sum_{i=0}^{2\ell} \|\mathbf{x}_{i+1}-\mathbf{x}_i\|_2^2$$

Minimise path length in C-space

$$\text{s.t.} \quad \mathbf{q}_{\min} \le \mathbf{x}_i \le \mathbf{q}_{\max}$$

Joint limits

$$\text{Transfer Constraints}$$

$$\text{Transit Constraints}$$

Ensure that object is at rest during transit and moves with robot in transfer

$$\text{Connectivity Constraints}$$

Ensures there exist valid regions that connect the generated configurations

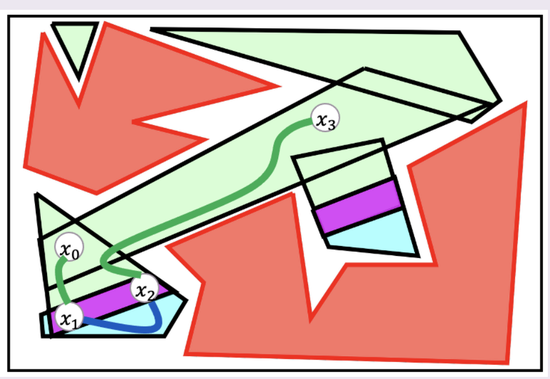
3. Trajectory Generation

$$\tau = \bigcup_{i=0}^{2\ell} \tau_i \quad \text{where} \quad \tau_i = \text{GCS}(\mathbf{x}_i, \mathbf{x}_{i+1}),$$

Use Graphs of Convex Sets (GCS) [3] to generate transfer and transit motions τ_i that connect the grasp and release configurations from 2.

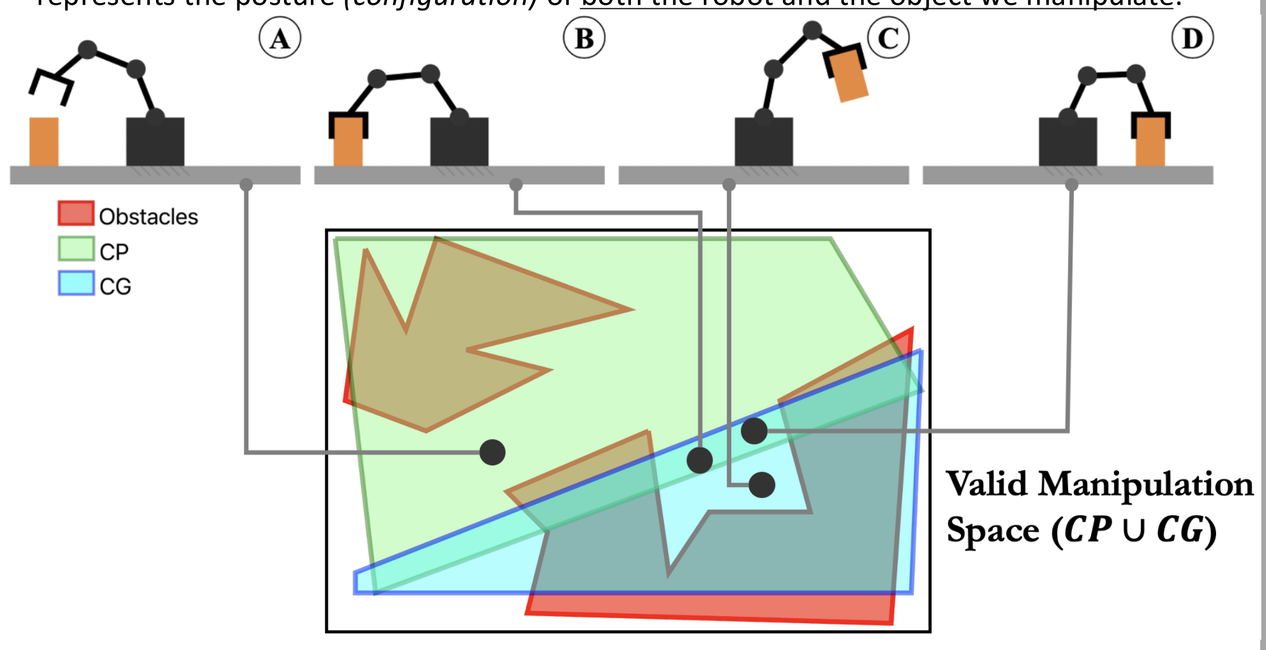
$$\tau_i = \begin{cases} \textit{Transit Path} & \textit{if } i \textit{ is even} \\ \textit{Transfer Path} & \textit{if } i \textit{ is odd} \end{cases}$$

The transit and transfer paths are concatenated to generate the final trajectory.



3. MANIPULATION IN C-SPACE

For manipulation we define a **composite configuration space (CS)**, where each point represents the posture *(configuration)* of both the robot and the object we manipulate.



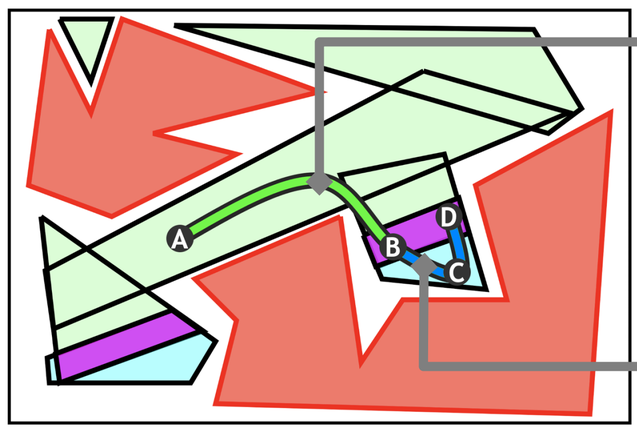
We define two submanifolds:

- **Placement Space (CP):** Configurations where the object is at rest (A,B,D).
- **Grasping Space (CG):** Configurations where the robot is grasping the object (B,C,D).

Valid manipulations occur in $CP \cup CG$: either the object is either at rest (*CP*), or grasped (*CG*).

MANIPULATION TRAJECTORIES

We can combine $CP \cup CG$ with the generated collision-free convex regions, and compute trajectories there. Manipulation trajectories are alternate sequences of two paths:



1. Transit Paths: The robot moves while the object is at rest. Contained in *CP*.

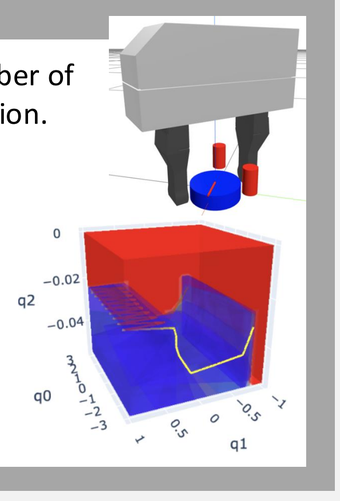
2. Transfer Paths: The robot moves while grasping the object. Contained in *CG*.

5. RESULTS

We ran performance experiments on 3 and 4 DOF tasks, analysing number of free regions generated, and time taken in region and trajectory generation.

DOF	Coverage (Actual)	# Regions	Coverage Time (s)	Traj. Time (s)
3	0.70-0.90 (0.94)	4	29.8	29.5
3	0.96 (0.98)	7	54.6	34.0
3	0.99 (0.996)	11	87.7	112.6
4	0.70 (0.77)	4	16.5	*
4	0.90 (0.95)	5	31.2	**
4	0.96 (0.98)	10	78.5	29.0
4	0.99 (0.996)	14	115.9	95.9

* Could not find a valid Grasp-Release sequence
** Could find a valid Grasp-Release sequence, but not connecting transfer and transit motions



CONCLUSIONS

We demonstrate that manipulation planning can be formulated as a global optimisation problem by using certified convex polytopes and Mixed-Integer Programming. This provides a structured way to reason over meaningful manipulation states (placements and grasps) and generate feasible transit and transfer motions between them.

Current Limitations: Polytope formulation does not deal with dynamics or contact physics. Also, scaling to higher dimensions is computationally challenging and requires generating grasping and placement polytopes in high dimensions, which is not trivial.

FUTURE WORK

- Explore how newer convex decomposition variants affect performance.
- Use these methods to generate grasping/placement regions to scale to higher dimensions.
- Incorporate dynamics into polytope representation.
- Use learning methods for polytope generation.
- **Open question: Can polytopes define richer, meaningful sequences beyond grasp/place to support more general long-horizon motion?**

REFERENCES

[1] Dai, H., Amice, A., Werner, P., Zhang, A., & Tedrake, R. (2024). Certified polyhedral decompositions of collision-free configuration space. The International Journal of Robotics Research, 43(9), 1322-1341.

[2] Werner, P., Cohn, T., Jiang, R.H., Seyde, T., Simchowitz, M., Tedrake, R., & Rus, D. (2024). Faster Algorithms for Growing Collision-Free Convex Polytopes in Robot Configuration Space. ArXiv, abs/2410.12649.

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