ICRA2024
YOKOHAMA | JAPAN

IKLink: End-Effector Trajectory Tracking with Minimal Reconfigurations

Yeping Wang, Carter Sifferman, and Michael Gleicher

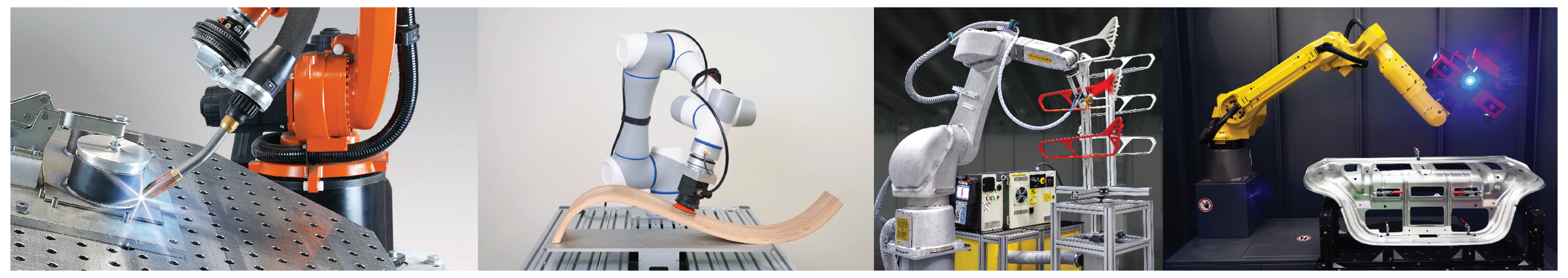
Synopsis

IKLink enables a robot manipulator to track reference end-effector trajectories of any complexity with minimal reconfigurations

Motivation

End-Effector Tracking

In applications such as welding, sanding, painting, or visual tracking, a robot needs to accurately track a reference trajectory with the end-effector.



Most existing approaches assume that the end-effector trajectory can be tracked as a single and continuous path, without the need to segment the trajectory into shorter parts. The uninterrupted tracking of some trajectories can be infeasible due to robot kinematics constraints such as self-collision, joint limits, and singularities.

Reconfiguration

For a complex trajectory, it becomes necessary to divide it into shorter segments. Each division introduces a reconfiguration, in which the robot deviates from the reference trajectory, repositions itself in configuration space, and then resumes task execution.

A: Tracking a welding trajectory B: Pause to avoid self-collision C: Perform a *reconfiguration* D: Resume task execution E: Finish the welding task

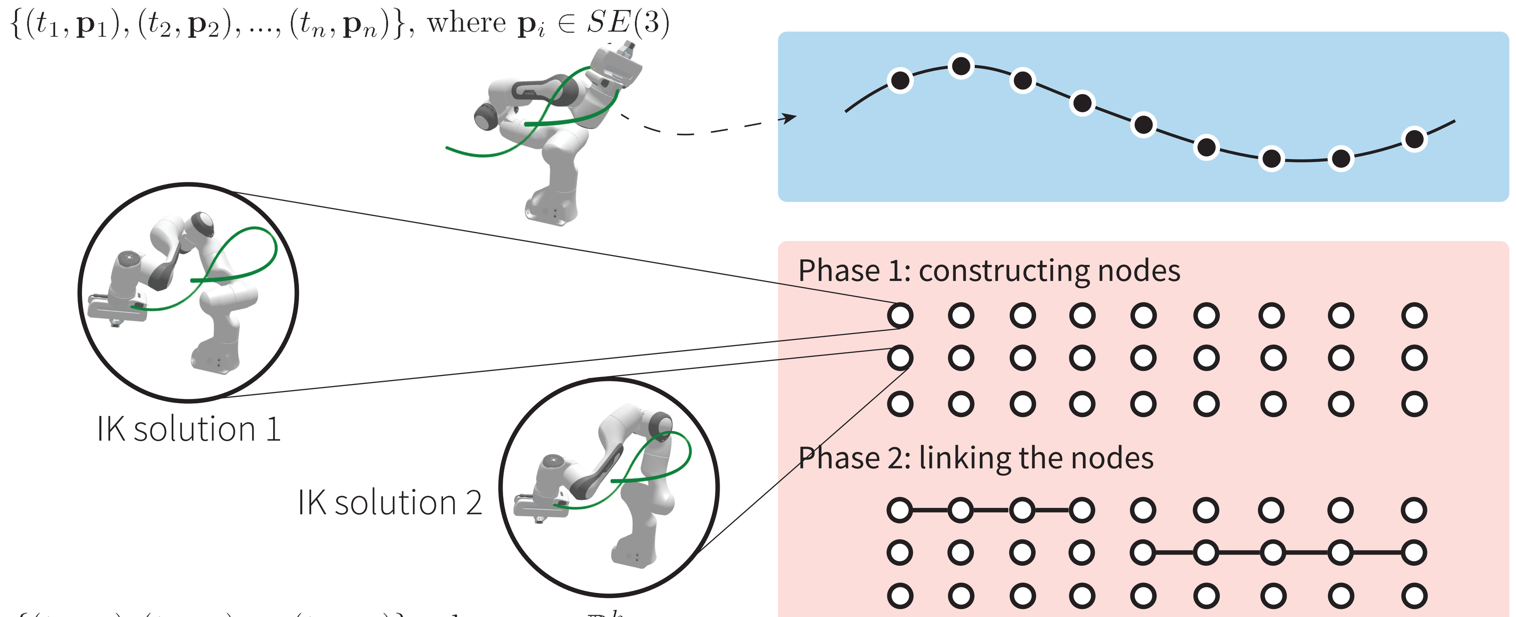
The occurrence of reconfigurations should be minimized because they increase time and energy usage.

IKLink

Overview

Input: an end-effector trajectory represented by waypoints

$\{(t_1, \mathbf{p}_1), (t_2, \mathbf{p}_2), ..., (t_n, \mathbf{p}_n)\}$, where $\mathbf{p}_i \in SE(3)$

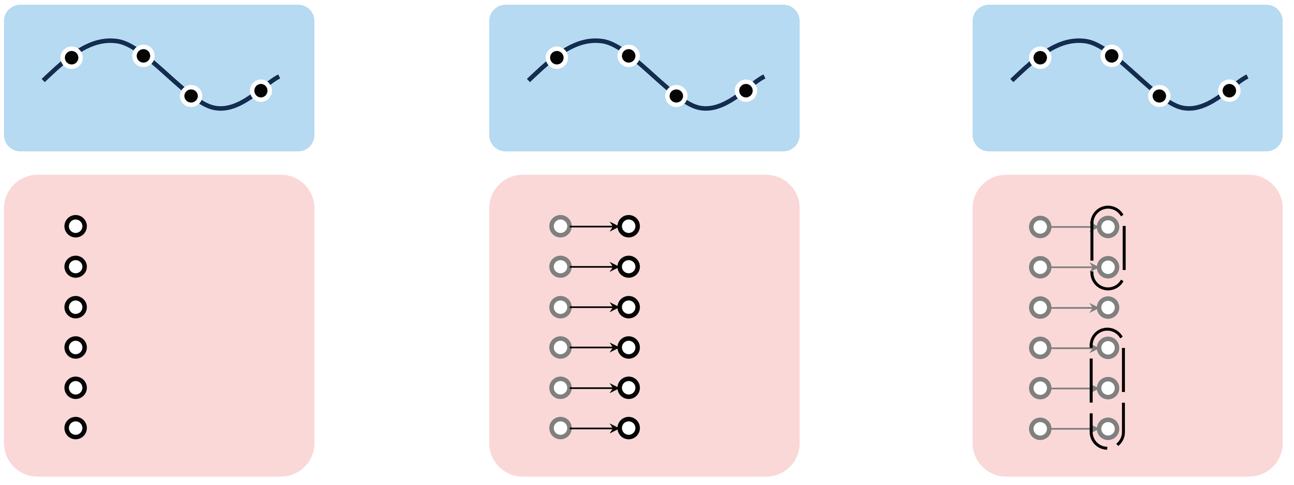


$\{(t_1, \mathbf{q}_1), (t_2, \mathbf{q}_2), ..., (t_n, \mathbf{q}_n)\}$, where $\mathbf{q}_i \in \mathbb{R}^k$

Output: multiple continuous joint motion segments

Phase 1: Constructing IK solutions

with a combination of these 3 methods (inspired by Stampede [1])



Random Sampling
to get diverse IK solutions

Greedy Propagation
to generate smooth motions
(using RelaxedIK [2])

Clustering
to merge similar IK solutions
(using DBSCAN)

Phase 2: Linking IK solutions

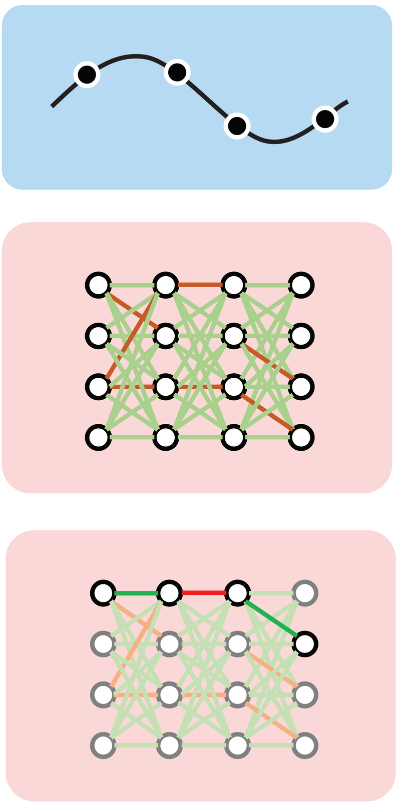
to find the shortest path in a multiple-layer graph

Construct a fully-connected multiple-layer graph where edge weights are number of reconfigurations:

0 within joint velocity limit

1 exceed joint velocity limit

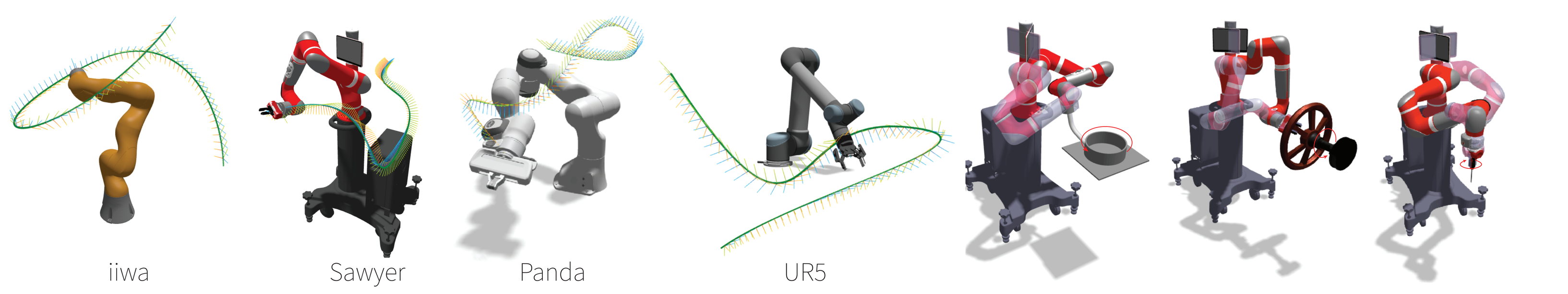
Find the shortest path from the first layer to the last layer using a dynamic programming algorithm



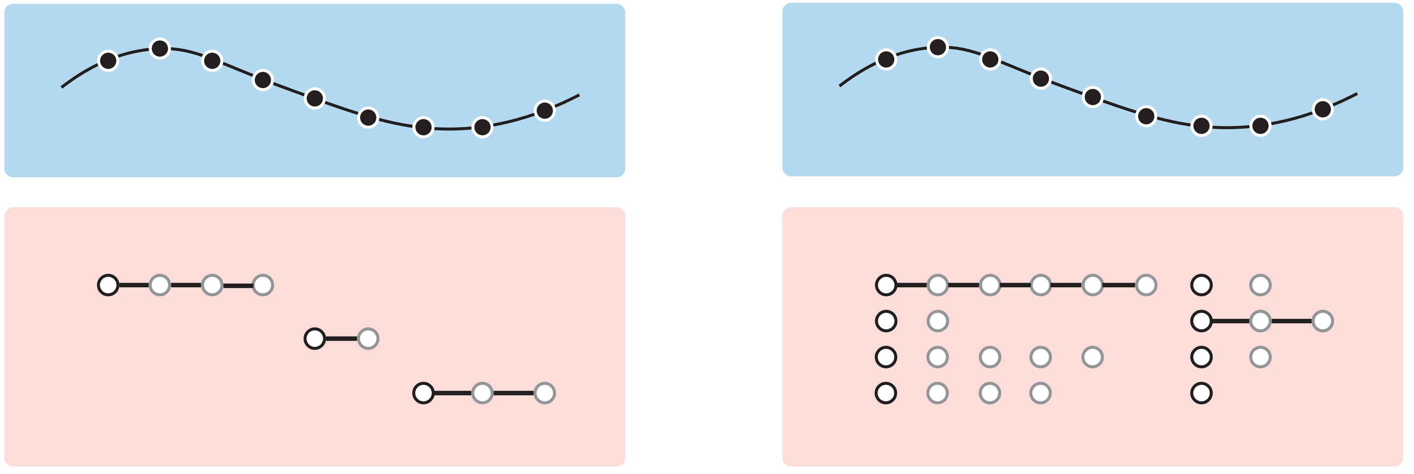
(Algorithm described in the paper also computes a secondary goal that minimizes joint movement)

Evaluation

We compare IKLink with two baseline approaches in simulation



Baseline approaches

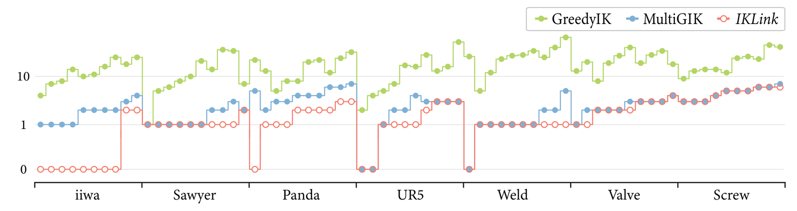


GreedyIK
Use an IK solver to track the trajectory; when it's stuck, reinitialize the IK solver from a random starting config

MultiGIK
Initialize multiple IK solvers with random starting configurations and keeps the longest solution

Results

- IKLink consistently finds motions with equal or fewer reconfigurations than the baselines
- All these approaches generate accurate and smooth motions
- While speed may not be the most important criterion for offline algorithms, IKLink is slower than GreedyIK and MultiGIK.



EXPERIMENT RESULTS AND METRICS OF THE INPUT TRAJECTORIES

Benchmark		Method	Mean Num of Reconfig	Mean Computation Time (s)†	Output Joint Motion Metrics: Mean Joint Vel. (rad/s)	Max Pos. Error (m)‡	Max Rot. Error (rad)‡	Input End-Effector Trajectory Metrics: Mean # of Waypoints	Mean Len. (m)	Mean Rot. (rad)
Random Trajectory Tracking	iiwa	GreedyIK	13.80±6.87	0.13±0.03	0.17±0.04	1e-3±6e-6	1e-2±9e-6	685.6 ±128.2	2.28 ±0.43	9.91 ±2.50
		MultiGIK	1.90±0.94	6.34±2.75	0.18±0.04	1e-3±1e-6	1e-2±5e-6			
		└MultiGIK×30	1.60±1.02	186.01±70.0	0.18±0.04	1e-3±3e-6	1e-2±1e-5			
		IKLink	**0.40±0.80**	166.55±140	0.18±0.04	1e-3±3e-5	1e-2±5e-6			
	Sawyer	GreedyIK	14.20±11.6	0.12±0.03	0.22±0.04	1e-3±2e-5	1e-2±1e-5	787.5 ±169.6	2.62 ±0.57	11.60 ±2.97
		MultiGIK	1.50±0.67	7.05±1.71	0.21±0.06	1e-3±1e-5	1e-2±2e-5			
		└MultiGIK×30	1.70±1.85	212.68±72.5	0.21±0.04	1e-3±6e-6	1e-2±3e-5			
		IKLink	**1.10±0.30**	119.25±33.2	0.19±0.06	1e-3±1e-5	1e-2±3e-5			
	Panda	GreedyIK	16.60±8.24	0.14±0.04	0.23±0.04	1e-3±5e-6	1e-2±1e-5	669.3 ±107.2	2.23 ±0.36	11.95 ±2.66
		MultiGIK	4.40±1.50	10.81±5.11	0.23±0.03	1e-3±3e-5	1e-2±1e-5			
		└MultiGIK×30	3.70±1.55	376.52±150	0.24±0.03	1e-3±5e-5	1e-2±1e-5			
		IKLink	**1.70±0.90**	219.92±61.1	0.24±0.04	1e-3±7e-6	1e-2±3e-5			
	UR5	GreedyIK	16.00±14.1	0.11±0.03	0.26±0.03	1e-3±9e-6	1e-2±2e-5	713.0 ±148.2	2.37 ±0.49	11.77 ±1.89
		MultiGIK	2.10±1.30	6.62±3.57	0.27±0.03	1e-3±3e-5	1e-2±2e-5			
		└MultiGIK×30	1.60±1.20	162.88±61.6	0.27±0.04	1e-3±4e-5	1e-2±2e-5			
		IKLink	**1.50±1.12**	121.87±29.0	0.25±0.03	1e-3±6e-5	1e-2±8e-6			
Weld	Sawyer	GreedyIK	28.50±15.4	0.13±0.02	0.16±0.02	1e-3±5e-5	1e-2±3e-6	450.0 ±0.0	0.86 ±0.17	6.27 ±0.00
		MultiGIK	1.50±1.28	5.30±1.13	0.16±0.02	1e-3±3e-5	1e-2±2e-6			
		└MultiGIK×30	1.20±0.98	142.12±31.4	0.16±0.02	1e-3±2e-5	1e-2±1e-5			
		IKLink	**0.90±0.30**	88.98±11.3	0.14±0.03	9e-4±8e-5	1e-2±3e-6			
Valve		GreedyIK	22.60±9.17	0.16±0.04	0.35±0.02	1e-3±7e-6	1e-2±2e-6	585.0 ±105.0	3.67 ±0.66	24.46 ±4.40
		MultiGIK	2.50±0.81	7.77±1.68	0.36±0.03	1e-3±3e-5	1e-2±6e-6			
		└MultiGIK×30	**2.30±0.90**	227.53±57.4	0.36±0.02	1e-3±2e-5	1e-2±5e-6			
		IKLink	**2.30±0.90**	110.38±22.7	0.35±0.02	1e-3±4e-5	1e-2±7e-6			
Screw		GreedyIK	22.10±11.8	0.28±0.11	0.25±0.01	1e-3±4e-5	1e-2±6e-6	1155.0 ±285.0	0.03 ±0.01	48.34 ±11.94
		MultiGIK	4.70±1.35	21.58±5.94	0.24±0.01	1e-3±3e-5	1e-2±8e-6			
		└MultiGIK×30	**4.60±1.20**	647.97±176	0.25±0.01	1e-3±8e-6	1e-2±1e-5			
		IKLink	**4.60±1.20**	230.59±80.3	0.26±0.01	1e-3±1e-5	1e-2±2e-5			

The range values are standard deviations.
†: All methods were implemented in Python and we expect reduced computation time using a more efficient compiled language such as C++.
‡: In our prototype system, we set the positional and rotational tolerance of IK solvers to be 1e-3 m and 1e-2 rad, respectively. As discussed in §IV-A, the accuracy can be improved by allowing more computing time for greedy propagation.

Conclusion

- IKLink enables a robot arm to track trajectories of any complexity while performing minimal reconfigurations
- IKLink eliminates the need to manually segment a long or complex trajectory and is beneficial in real-life scenarios that involve end-effector trajectory tracking, such as welding, sweeping, scanning, painting, and inspection.

[1] Rakita, Daniel, Bilge Mutlu, and Michael Gleicher. "Stampede: A discrete-optimization method for solving pathwise-inverse kinematics." ICRA 2019.
[2] Rakita, Daniel, Bilge Mutlu, and Michael Gleicher. "RelaxedIK: Real-time Synthesis of Accurate and Feasible Robot Arm Motion." Robotics: Science and Systems. Vol. 14. 2018.

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

This work was supported by
Los Alamos National Laboratory and
the Department of Energy