ICRA2026 Late Breaking Results Poster presented at
2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Optimal Motion Planning for Object Picking in Industrial Contexts with Optimal Control

Dries Dirckx[1, 2], Jan Swevers[1, 2], Wilm Decré[1, 2]
[1]MECO Research Team, KU Leuven
[2]Flanders Make @ KU Leuven
Email: dries.dirckx@kuleuven.be

I. Introduction

Industrial manufacturing is transitioning from rigid, high-volume production towards small-batch, highly variable, and rapidly reconfigurable assembly processes. This shift demands motion planning methods that can quickly adapt to changing product and workspace configurations. Traditional manual waypoint specification is time-consuming and unable to fully leverage a robot's capabilities. Graph-based and sampling-based planners often lack predictability due to their inherent randomness. On the other hand, optimal control problems (OCPs) can be solved deterministically, while incorporating (safety) guarantees and desired behaviour into the final solution, but they need to be properly formulated and initialised. Current state-of-the-art GPU-based trajectory optimisers, such as cuRobo [1], offer fast solutions but often rely on conservative geometric approximations or penalty-based constraint enforcement that may compromise accuracy or reliability in industrial settings and could lead to costly manual interventions and disrupt production flow. Furthermore, GPU-based planners are at a practical disadvantage in assembly cells where dedicated GPU hardware is often unavailable.

This work addresses these challenges by formulating motion planning for pick-and-place tasks as a hard-constrained optimal control problem (OCP) with geometric modelling using geometric representations commonly available in industrial robotic software (cuboids, capsules). The method ensures accurate placement of the end-effector, automatic handling of varied gripper and object geometries, and collision-free motion through cluttered workcells while maintaining competitive computational efficiency. To fully exploit the repetitive nature of object picking tasks, a near-optimal warm-starting strategy is proposed that stores and reuses previously computed trajectories for similar problem instances, further reducing computation times and improving reliability.

II. Methodology

The motion planning problem is expressed as a time-optimal OCP with hard constraints on joint limits, final pose accuracy, collision avoidance (robot–environment and robot self-collision), and kinematic feasibility. Collision avoidance is ensured with bilinearseparating hyperplanes constraints defined between capsules and cuboids, representing both robot links and obstacles. A final pose constraint is enforced as a hard constraint on the end-effector position and orientation at the final timestep, guaranteeing that the object is picked and placed within specified tolerances. The OCP is solved using a direct transcription approach and an interior-point method solver, Fatrop [2].

This high-dimensional, non-linear, and non-convex OCP is prone to local minima, making the choice of initialisation critical for convergence. Two complementary strategies are proposed to provide good initial guesses for the solver, as visualised in Figure 1. The first strategy constructs a configuration-space trajectory by mapping a straight Cartesian line between the desired start and end positions, while the second strategy stores primal–dual OCP solutions from previous executions and reuses or mirrors them for new but similar tasks. If no previous solution is available, the best-guess initialisation is used as a fallback.

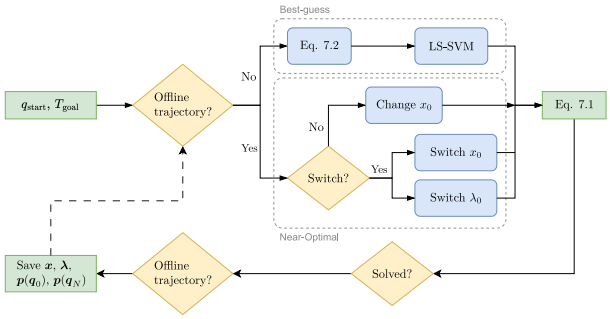

Fig. 1: Automated initialisation strategies for trajectory generation.

III. Results

A. Benchmark Comparison

The proposed planner is benchmarked against cuRobo over 220 trajectories across 11 cluttered environments, each containing six cuboid obstacles, using a UR10e robot. The benchmark results are given in Table I, where T_{motion} denotes the reduction in execution time relative to cuRobo.

This work was supported by the Flanders Make ICON-project FROGS (Flexible and Robust Robotic Gripping Solutions) and SBO-project LearnOpTra (Learning meets optimization for robust and multimodal trajectory planning).

Firstly, the proposed planner achieves a success rate of 98.33 %, compared to 81.67 % for cuRobo and 79.58 % for cuRoboNTA (the version without the terminal action trick for final pose satisfaction). The lower success rate of cuRobo is caused by upstream failures: it cannot always find a feasible solution to the inverse kinematics problem for the desired end pose, it fails to finetune the trajectory or incorrectly classifies the start state as in collision due to its conservative sphere-based geometry approximation.

Secondly, a more fundamental limitation of penalty-based planners is exposed by the goal-reaching accuracy results. The hard constraints in the proposed formulation guarantee that the final pose is reached within 1 mm and 1° for every successful trajectory. cuRoboNTA, which relies solely on non-exact penalty functions to enforce the terminal pose, fails this criterion for 8.75 % of its successful trajectories. Since the same penalty-based mechanism are used for other constraints such as collision avoidance, this result shows that these non-exact penalty formulations provide no formal safety or accuracy guarantees. The terminal action trick in the standard cuRobo release partially compensates for this, but represents a workaround rather than a principled solution.

Finally, the time-optimal OCP formulation yields trajectories whose execution time is $0.70\times$ lower than that of cuRobo's minimal-jerk trajectories, directly reducing cycle times. Although the proposed planner's median computation time (103 ms) is higher than cuRobo's (34.67 ms), it remains well below the minimal motion time of 1.33 s over all 220 trajectories, allowing a new trajectory to be computed during execution of the current one and hence sufficient for online deployment. Even when computation and execution are performed sequentially, the combined time is lower for the proposed planner than for cuRobo's trajectories, confirming the overall cycle time advantage of the proposed time-optimal formulation. It should be noted that cuRobo's faster computation times are achieved through a highly parallelised GPU-based solver, while the proposed planner runs on a CPU, hardware far more commonly available in industrial workcells.

B. Near-Optimal Initialisation

The near-optimal initialisation strategy is validated on a real-world flexible assembly cell with five cuboids and two capsule obstacles, a more complex environment than the environments used in the previous benchmark. Results across 100 trajectories per gripper–object geometry combination are given in Table II. Compared to the best-guess initialisation ((1) in Table II), the near-optimal warm-start ((2) in Table II) raises the success rate to 99–100 % across all gripper and object geometries (up from 92–95 %), as the solver is initialised substantially closer to the feasible solution, reducing the risk of converging to an infeasible local minimum. Computation times are simultaneously reduced by 28–57 %, with the largest gains observed for the most geometrically complex configurations (box gripper with box object), where the reduction reaches $2.3\times$. This improvement scales with geometric complexity because a higher non-convex constraint count (the bilinear separating hyperplane constraints) makes a high-quality initialisation increasingly valuable for rapid convergence. This increase comes at no additional cost, as the trajectories are stored during normal operation and can be reused without any reconfiguration or re-transcription of the OCP, demonstrating the practical benefits of trajectory reuse in repetitive industrial tasks.

Method	Median (ms)	Success	T_{motion}	$\boldsymbol{T}_{\text{goal}}$
Planner	103.04	**98.33 %**	**70**%	**0.00 %**
cuRobo	34.67	81.67 %	100%	0.00 %
cuRoboNTA	**33.94**	79.58 %	100%	8.75 %

TABLE I: A comparison between cuRobo and the proposed planner for a six-DOF robotic manipulator for eleven environments, each containing six cuboids.

	Gripper & Object	- -	Cp -	Cp Cp	Cp Box	Box -	Box Cp	Box Box
(1)	Med.(ms)	168.4	297.8	366.8	744.1	845.2	1056.6	1887.3
	Succ. (%)	95	92	93	94	93	94	93
(2)	Med.(ms)	**116.1**	**168.3**	**235.8**	**530.6**	**429.7**	**463.1**	**802.8**
	Succ. (%)	**100**	**100**	**99**	**100**	**99**	**100**	**100**
	Red.	31.0	43.5	35.7	28.7	49.2	56.2	57.5

TABLE II: A comparison of initialisation approaches over 100 trajectories for different gripper and object geometries (Cp = Capsule, Box = Cuboid; Med. = Median; Succ. = Success; Red. = Reduction).

IV. Conclusion

The proposed OCP-based planner provides a reliable and reconfigurable solution for industrial pick-and-place motion planning, assuming static environments and known grasp poses. Its combination of guaranteed constraint satisfaction, expressive geometric modelling of the robot's geometry, time-optimal formulation and efficient initialisation strategies enables accurate and time-optimal motion generation on standard CPU hardware, suitable for modern production cells. Compared to a state-of-the-art GPU-based trajectory optimiser, the planner achieves higher success rates and lower cycle times, albeit with increased computation times that are still feasible for online deployment. The near-optimal warm-starting strategy further reduces computation times up to a factor of 2.3 in repetitive tasks, demonstrating the practical benefits of trajectory reuse in industrial settings.

References

[1] B. Sundaralingam, S. K. S. Hari, A. Fishman, C. Garrett, K. Van Wyk, V. Blukis, A. Millane, H. Oleynikova, A. Handa, F. Ramos, N. Ratliff, and D. Fox, "Curobo: Parallelized collision-free robot motion generation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[2] L. Vanroye, A. Sathya, J. De Schutter, and W. Decré, "Fatrop: A fast constrained optimal control problem solver for robot trajectory optimization and control," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10036–10043.