2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2025)
October 19-25, 2025. Hangzhou, China

2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 979-8-3315-4393-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/IROS60139.2025.11247737

Many-Objective Motion Generation Method for Redundant Manipulators by Solving Pathwise Inverse Kinematics

Bin Xie, Jiaming Zhao, Qingfeng Wang and Di Wu*

Abstract— **Modern robots are required to operate in complex environments and perform diverse tasks, resulting in redundant degrees of freedom (DoF) for flexibility. However, managing redundancy is challenging due to the high-dimensional and non-convex nature of robotic kinematics. When executing complex tracking tasks, redundant robots must handle non-convex constraints while maintaining many objectives, such as balancing and obstacle avoidance. This paper models the pathwise inverse kinematics of redundant mechanisms as a multi-objective nonlinear optimization problem. We propose an efficient gradient-free optimization method named MoeIK, which demonstrates strong multi-objective balance, rapid global convergence, and adaptability. Our approach enhances the method by integrating relaxation dominance, adaptive interval search strategies, and a restart strategy, significantly improving performance in overcoming many-objective optimization challenges. We compared MoeIK with RelaxedIK, Trac-IK, and BioIK across multiple trajectories on various redundant robots, and the experimental results demonstrate that our algorithm exhibits better multi-objective balance capabilities and supports real-time computation.**

I. INTRODUCTION

In recent years, the demand for dexterous robots has been quickly increasing in various application domains, including complex industrial scenarios [1], human-robot interaction [2]–[5], aerospace [6], etc. In contrast to traditional industrial robots that operate along fixed trajectories, modern robots are expected to perform more natural and intelligent behaviors and operations within complex environments. This necessitates robots to possess enhanced capabilities for balancing multiple tasks and constraints, which is typically achieved through redundancy, including redundant manipulators, humanoid robots, and bionic structures. Among these, redundant manipulators are the most typical and significant, with other structures considered as their complex variants.

Redundancy means that the manipulator possesses additional degrees of freedom to improve the performance beside the end-effector pose, such as obstacle avoidance, motion continuity, and singular configuration avoidance. When redundant manipualtor is performing continuous trajectory tracking or operations that require satisfying multiple objectives and constraints, how to effectively track the trajectory and balance multiple tasks is referred to as the pathwise inverse kinematics problem [7]. The algorithms capable of handling the pathwise inverse kinematics problem can also solve for single pose, and the smoothness of motion is similarly guaranteed, as the single pose problem can be viewed as a waypoint in a continuous trajectory with a large distance. 'Task' typically encompasses specific operational tasks, such as instructing a robot to navigate to a location and grasp an object. However, in this paper, 'tasks' concretely signify the constraints and optimization objectives that need to be satisfied during the motion of a redundant manipulator.

It is challenging to precisely determine the mathematical expression of a subtask, especially in complex environments and during the execution of multi-stage high-level planning. For instance, obstacle avoidance can be formulated as a constraint, specifying the minimum distance the body needs to maintain from an obstacle. However, setting such hard constraints is problematic for two reasons: firstly, it is difficult to set thresholds that are adaptable to various environments, and secondly, in narrow environments, the motion of the manipulator can appear particularly clumsy. Alternatively, setting obstacle avoidance as both a hard constraint and an optimization objective (soft constraint) can effectively address the aforementioned issue. However, this approach leads to new challenges, that is, how to swiftly adjust the structure of the optimization problem and balance multiple constraints and objectives.

To tackle the aforementioned challenge, an algorithm with the following features is necessary: 1. An efficient and flexible structure that enables the addition, removal, and alteration of the quantity and hierarchy of tasks; 2. Robust ability to balance multiple tasks; 3. Sufficiently lightweight to allow for real-time problem-solving. Although there has been considerable research in this field, the aforementioned issues have not been satisfactorily resolved.

In previous research, the authors proposed a semi-analytic framework for path inverse kinematics, which reduces the dimensionality of the redundant space by obtaining the kinematic analytical expressions for non-redundant joints [8] [9] [10]. However, this method essentially involves strict planning in the kinematic constraint manifold space, leading to its inability to avoid obstacles in complex and dense environments. This paper presents a path inverse kinematics algorithm that possesses stronger multi-task balance capabilities and does not rely on analytical expressions, significantly enhancing the algorithm's generalizability. Since the previous work [9] is unable to address the obstacle avoidance problem, we do not include it in our comparisons, focusing only on comparisons with other state-of-the-art in this paper.

II. RELATED WORK

The pathwise inverse kinematics algorithms for manipulators can be broadly categorized into two types: semi-

Bin Xie, Jiaming Zhao, Qingfeng Wang, and Di Wu are with School of Automation, Central South University, Changsha, China. Di Wu is the corresponding author, email: wudiautomatic@163.com

analytical methods and numerical methods. The semi-analytical method achieves dimension reduction of the configuration space by identifying the geometric relationships between joints. Its primary advantage is that, given parameterized joint angles, it can find all joint angles that satisfy the end-effector pose in analytical time. Lee et al. [11] [12] first proposed the parameterization method, treating the values of parameterized joints as known to obtain analytical solutions for the remaining joints. Although semi-analytical methods offered fast solution speeds, they were limited in terms of application scenarios and required manual derivation of analytical expressions prior to application. Most critically, the inherent nature of the semi-analytical methods determines that they cannot effectively handle end-effector obstacle avoidance because the precision of the end-effector is always the highest priority.

In contrast, numerical methods are more versatile but require longer solution times. Jacobian based methods were preferred to generate the joint motion with geometric constraints for the end-effector [13]. The method established the mapping between the velocities of end-effector movement and joint motion velocities to efficiently obtain position-specific inverse kinematics solutions. However, the Jacobian-based methods are easy to fall into local minima or fail in convergence. The Trac-IK [14] pointed out this problem and improved the success rate by using a random restart strategy and sequence quadratic planning (SQP). Various offline algorithms [7] [15] have also been proposed to solve the pathwise inverse kinematics problem. However, these algorithms demonstrated poor multi-task balance capabilities, especially when the manipulators have higher degree of redundancy.

Evolutionary algorithms (EAs) were promising to solve this problem and were flexible to realize task adjustments [16]. Starke proposed a memetic evolutionary algorithm BioIK [17], which converted the problem into a single-objective optimization problem with nonlinear kinematic constraints to provide feasible solutions for various robots. However, those researches failed to fully utilize the characteristics of the problem to speed up the algorithm and improve its performance. Rakita et al. proposed RelaxedIK [18] to transform the pathwise inverse kinematics problem into a nonlinear optimization task that integrated multiple optimization objectives, allowing for real-time motion generation. Subsequently, they proposed the CollisionIK [19] algorithm to avoid external obstacles effectively. Unfortunately, due to the use of gradient-based methods for solving, they were prone to getting trapped in local optima, and generating smooth motion came at the cost of sacrificing trajectory accuracy.

Extending the multi-objective evolutionary algorithm to the entire joint space effectively prevents convergence to local optima and enhances multi-objective balancing capability, but it also introduces new challenges. Time consumption increases significantly compared to single-objective evolution. Additionally, addressing the dominance impedance problem among multiple tasks, especially with a pose task included, proves difficult. Furthermore, the simultaneous evolution of multiple objectives raises concerns about completing primary tasks.

Based on the above analysis, this paper constructs the MoeIK algorithm within a multi-objective optimization framework. This is because the multi-objective optimization framework allows for flexible modifications of constraints and objectives while providing a foundational architecture for handling multi-objective balance. To address the issue of slow solving time that hinders real-time computation within the multi-objective optimization framework, we have incorporated algorithmic strategies and optimization techniques tailored to the characteristics of multi-objective path inverse kinematics, including relaxation dominance, adaptive interval search, and restart strategies. The proposed method can be conveniently applied to various types of robots for online task execution and supports realtime task switching. We compared our algorithm with the current state-of-the-art methods to demonstrate its superior comprehensive performance. The main contributions of this paper are summarized as follows:

1) A novel many-objective evolutionary algorithm MoeIK for path-wise inverse kinematics problem is proposed by integrating the many-objective algorithm framework with several efficient strategies, including ϵ-dominance ranking, elite inheritance and adaptive interval search with heuristic restarts. The proposed algorithm enables the free and flexible incorporation of new tasks and the adjustment of priorities, including posture, minimal energy consumption, and obstacle avoidance.

2) We conducted experiments on redundant manipualtors with 7 DoF, 8 DoF, and 12 DoF using various trajectories and compared the results with state-of-the-art (SOTA) algorithms, demonstrating the proposed algorithm's superior balance and generality in handling many-task objectives, with solution times comparable to those of the SOTA algorithms.

III. ALGORITHMIC APPROACH

We formulate the pathwise inverse kinematics problem as a multi-objective, multi-constraint optimization problem based on full joint space optimization,

$$\begin{aligned} \text{Minimize} \quad & f_1(x), f_2(x), \ldots, f_m(x) \\ \text{subject to} \quad & g_j(x) \leq 0, \quad j = 1, 2, \ldots, p \\ & h_k(x) = 0, \quad k = 1, 2, \ldots, q \\ & x \in X \end{aligned} \tag{1}$$

$f_1(x), f_2(x), \ldots, f_m(x)$ are the objective functions to be optimized simultaneously, including end-effector pose matching, joint motion continuity, joint limit avoidance and obstacle avoidance. $g_j(x)$ represents inequality constraints, including joint motion constraints, joint obstacle avoidance and external obstacle avoidance. $h_k(x)$ represents equality constraints, including forward kinematics constraints. x is the decision variable vector, representing the joint configuration, which needs to be optimized within the feasible domain X.

The proposed algorithm named MoeIk is improved based on the many-objective evolutionary framework in ϵ-NSGAII [20]. The framework of the MoeIK is illustrated in

Figure 1 and Algorithm 1. First, we design a corresponding fitness function to quantify both the individual objective function values and the constraint conditions. Subsequently, we realize a series of enhancements to the algorithm based on the characteristics of the path pathwise inverse kinematics problem, aiming to improve its capability in handling many objectives and constraints, as well as to reduce the solution time. During the population initialization stage, we incorporate an elite inheritance strategy based on trajectory information. This strategy inherits the final population front from the previous trajectory point, thereby extending the excellent individuals. To enhance the precision and efficiency of the search within the local solution space, an adaptive interval search method is applied to the remaining individuals awaiting initialization. Following the aforementioned steps, the algorithm generates a high-quality population tailored to the current trajectory point.

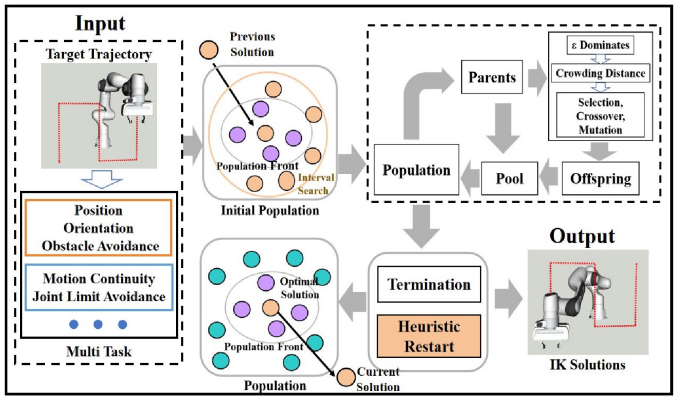


Fig. 1: MoeIK Solutions Procedure

The algorithm is terminated once it achieves the dynamic task completion indicators or reaches the maximum iterations. If the indicator is not achieved when this threshold is reached, a heuristic restart is triggered. After the restart, the population is regenerated based on the results of the previous evolution to continue the computation. Finally, a maximum restart count is set. Once this limit is reached, the optimal solution is selected as the final joint configuration, ensuring the algorithm's effectiveness and feasibility. In the following section, we will elaborate further on the details of the algorithm.

A. Joint Variable Encoding

To address the continuously variable nature of the pathwise inverse kinematics problem in redundant manipulators, we employ real-number encoding. For instance, for a manipulator with n degrees of freedom, an individual is encoded as an n-dimensional vector of real numbers, as shown in

$$x = (x_1|x_2|x_3|...|x_{n-1}|x_n), \tag{2}$$

where each variable must satisfy its joint constraints and remain within the range of the corresponding joint, as shown in

Algorithm 1: MoeIK algorithm

Input: Population Size, ϵ-Weight, Initial Interval Size, Last Population Front, Last Solution
Output: Best Solution and Final Population Front
Determine Elite Individuals and interval Center
Initialize Population
>>**Elite** Inheritance
>>**Adaptive** Interval Search: Generate Random Individuals in Search Interval
Calculate Fitness, **Sort** Individuals Based on ϵ-Weights and **Calculate** Crowd Degree
while *Not* ***Terminated*** **do**
  Create Offspring by Selection, Crossover and Mutation
  Combine Parent and Offspring as Merging Pool
  Calculate Fitness and **Sort** Individuals Based on ϵ-Weights for Merging Pool
  Calculate Crowd Degree
  Extract Elitist form Merging Pool and Form New Population
end
Return Best Solution and **Save** Final Population Front

$$\theta_{i\min} \leq x_i \leq \theta_{i\max}, \forall i = 1, ..., n. \tag{3}$$

B. Design of Objectives and Constraints

We designed the following fitness functions for the objectives and constraints: position objective, orientation objective, joint continuity, joint limit avoidance, and obstacle avoidance.

1) Position objective:

$$T = f(\theta_1, \theta_2, ..., \theta_n), \tag{4}$$

where n denotes the number of joints in the manipulator, and T denotes the forward kinematics result of the manipulator.

$$PosError = ||Object^{pos} - T^{pos}|| \tag{5}$$

For the forward kinematics of the manipulator, $PosError$ represents the position error, where $Object^{pos}$ denotes the target position and T^{pos} denotes the actual position. The position error is quantified by the L2-norm.

2) Orientation objective:

$$RotError = ||Object^{rot} - T^{rot}|| \tag{6}$$

Similarly, $RotError$ represents the orientation error, where $Object^{rot}$ denotes the target orientation and T^{rot} denotes the actual orientation. The orientation error is also quantified by the L2-norm.

3) Joint Continuity:

$$disC = 1 - e^{-(\frac{\Theta - \Theta_c}{fiaC})^2}, \tag{7}$$

where Θ_c denotes the last solution, and $fiaC$ is the scaling ratio. This objective also implies minimal energy loss, thereby avoiding excessive joint motions.

4) Joint Limit Avoidance:

$$disL = \frac{2}{1+e^{\frac{\theta-\theta_{min}}{fiaL}}} + \frac{2}{1+e^{\frac{\theta_{max}-\theta}{fiaL}}}, \tag{8}$$

where Θ_{min} and Θ_{max} denote the minimum and maximum joint angles, respectively, and $fiaL$ is the scaling ratio.

5) Obstacle Avoidance: In this paper, we employ the QuickHull algorithm to efficiently construct a convex hull model and utilize the GJK algorithm to calculate the shortest distance ($disO$) between the convex hulls of the manipulator and the obstacle for collision detection. The fitness function for the obtained distance values is defined as

$$f(disO) = \begin{cases} \frac{1}{disO}, & \text{if } disO > 0, \\ \infty, & \text{if } disO \leq 0. \end{cases} \tag{9}$$

To avoid the high computational cost of treating obstacle avoidance as both an objective and a constraint, we designed a **pruning strategy**: during trajectory point planning, obstacle avoidance is activated only when the avoidance distance is below a threshold; once activated, joint-obstacle pairs within the threshold are updated and collected, and used for planning subsequent points.

C. Initialization

In the initialization phase, we introduce an elite inheritance strategy and an adaptive interval search strategy.

- **Elite Inheritance**: The initial population inherits the Pareto front from the final population at the previous trajectory point, thereby retaining high-quality individual information from the previous trajectory point to provide guidance.
- **Adaptive Interval Search**: We use trajectory information to randomly generate individuals within the initially set interval, which is centered on the final solution at the previous point.

$$range = \frac{(\Theta_{\max} - \Theta_{\min})2^m}{2^{m_{max}}}, \tag{10}$$

$$\Theta = rand(\Theta_c - range, \Theta_c + range), \tag{11}$$

where m and $m_{\max}$ denote the current interval size and the maximum interval size, respectively, and $range$ denotes the search interval. Finally, Θ denotes the final interval initialization for joint values. The adaptivity is primarily reflected in the adaptive adjustment of the interval size. The initial value is predefined based on the relationship between the current end position and the target point position. The interval range is adaptively adjusted through a restart mechanism,

D. ϵ Dominance Ranking and Feasibility Principle

We use ϵ-dominance instead of Pareto-dominance. In ϵ-dominance, if an individual's fitness value is better than that of other individuals after weighting with the weight vector, it is considered to dominate the other individuals. It represents a relaxed dominance relationship.

$$\epsilon_i f_i(x) < f_i(y), \quad \forall i \in \{1, \ldots, p\} \Rightarrow x \prec_\epsilon y \tag{12}$$

The purpose of this dominance relation is to expand the dominance region of non-dominated solutions and effectively mitigate the dominance resistance problem encountered in many-objective optimization.

For constraints, we adopt the feasibility principle. For instance, in the case of obstacle avoidance constraints, individuals colliding with obstacles are deemed infeasible. During ranking, we follow the principle that feasible individuals dominate infeasible ones: feasible individuals are compared among themselves, and infeasible individuals are compared among themselves. This ensures that the Pareto front satisfies the constraints while maximizing the utilization of individual information.

E. Selection Crossover and Mutation

Selection is performed using a binary tournament strategy, crossover is implemented using simulated binary crossover (SBX), and mutation is applied using polynomial mutation.

F. Termination and Heuristic Restart

This paper replaces the maximum iteration count with dynamic task completion metrics as the termination criterion. The algorithm stops when a solution meets the primary task metric, improving efficiency. Since many-objective algorithms produce a solution set (Population Front), we select the solution with the minimum weighted sum of primary task fitness as the final output. This ensures primary task completion and optimization. To avoid local optima, a maximum iteration limit is set. If no valid solution is found by then, a **heuristic restart** is triggered.

Algorithm restart implies regenerating and evolving a population specifically for the target point. The heuristic utilizes information from the previous evolution rather than trajectory information for guidance. The elite inheritance strategy is adapted to inherit the population front from the last evolution. Adaptive interval search uses the optimal solution from the final population of the previous evolution as the center, while exponentially decreasing the interval size by reducing m by one, thereby narrowing the search range. This ensures that each restart improves upon the previous evolution. Finally, a maximum restart count is set. If this limit is reached and the termination condition remains unmet, the optimal solution is selected from the final population to ensure the feasibility and effectiveness of the output.

IV. EXPERIMENTS AND ANALYSIS

The experiments were conducted on a Dell G15 laptop with a 2.3 GHz processor, utilizing the ROS platform, the MoveIt framework, and the Rviz tool for simulation testing. All statistical results are based on data collected from 100 repeated trials of the trajectory execution and evaluated accordingly. This section first discusses parameter selection, then elaborates on the experimental design, and finally analyzes the performance of the proposed algorithm in comparison with several state-of-the-art algorithms.

TABLE I: Panda(7 DoFs) "IROS" Trajectory Performance Evaluation

Alg	PE(mm)	OE(°)	M1	M2	M3	M4	M5	M6	M7	TCL	Time(ms)
RelaxedIK	**8.52**	**6.25**	**0.15**	**0.32**	**0.19**	**0.33**	**0.08**	**0.53**	**0.33**	**0**	**56**
Trac-IK	**0.08**	**0.05**	**1.46**	**0.91**	**1.36**	**0.39**	**1.04**	**0.89**	**0.83**	**8**	**10**
BioIK	0.39	0.16	1.00	0.51	0.89	**0.35**	0.34	0.57	**0.33**	**24**	**9.8**
MoeIK(A)	0.92	1.62	1.15	0.67	1.43	1.37	2.06	2.84	1.81	1	32
MoeIK(B)	0.33	0.29	0.42	0.38	0.46	0.33	0.44	0.52	0.38	0	27
MoeIK	**0.12**	**0.14**	**0.17**	**0.34**	**0.18**	**0.35**	**0.09**	**0.54**	**0.29**	**0**	31

Alg: algorithm, PE: average position error (mm), OE: average orientation error (°), $M(i)$: average movement distance of joint i (°), TCL: times close to joint limits, Time: the average time required for the end-effector pose to match each trajectory point (ms). MoeIK(A) is a variant of MoeIK that excludes heuristic restarts and adaptive interval search, while MoeIK(B) incorporates fixed interval search and removes heuristic restart. Green indicates optimal performance for the corresponding metric, orange represents sub-optimal performance, and red denotes poor performance.

TABLE II: KC30(8 DoFs) Tunnel Trajectory Performance Evaluation

Alg	PE(mm)	OE(°)	M1	M2	M3	M4	M5	M6	M7	M8	TCL	Time(ms)
RelaxedIK	**53.1**	**19.8**	**0.52**	**0.30**	**11.2**	**0.57**	**0.96**	**11.4**	**1.27**	**1.53**	**89**	**191**
Trac-IK	0.96	**0.08**	**3.37**	**4.12**	**228**	**7.46**	**8.40**	**201**	**1.87**	**6.44**	19	67
BioIK	**0.77**	0.12	1.57	1.04	64.9	**1.93**	3.74	78.5	1.47	2.24	**14**	**26**
MoeIK(A)	1.35	1.68	2.23	2.06	137	4.0	5.3	127	1.89	3.77	21	45
MoeIK(B)	0.46	0.72	0.34	0.22	17.6	0.51	0.58	13.8	1.35	0.62	2	41
MoeIK	**0.11**	**0.06**	**0.36**	**0.29**	**28.7**	**0.57**	**0.71**	**22.2**	**1.30**	**0.62**	**1**	**57**

M(3) and M(6) both represent prismatic joints (mm).

TABLE III: SSRMS12(12 DoFs) "U" Trajectory Performance Evaluation

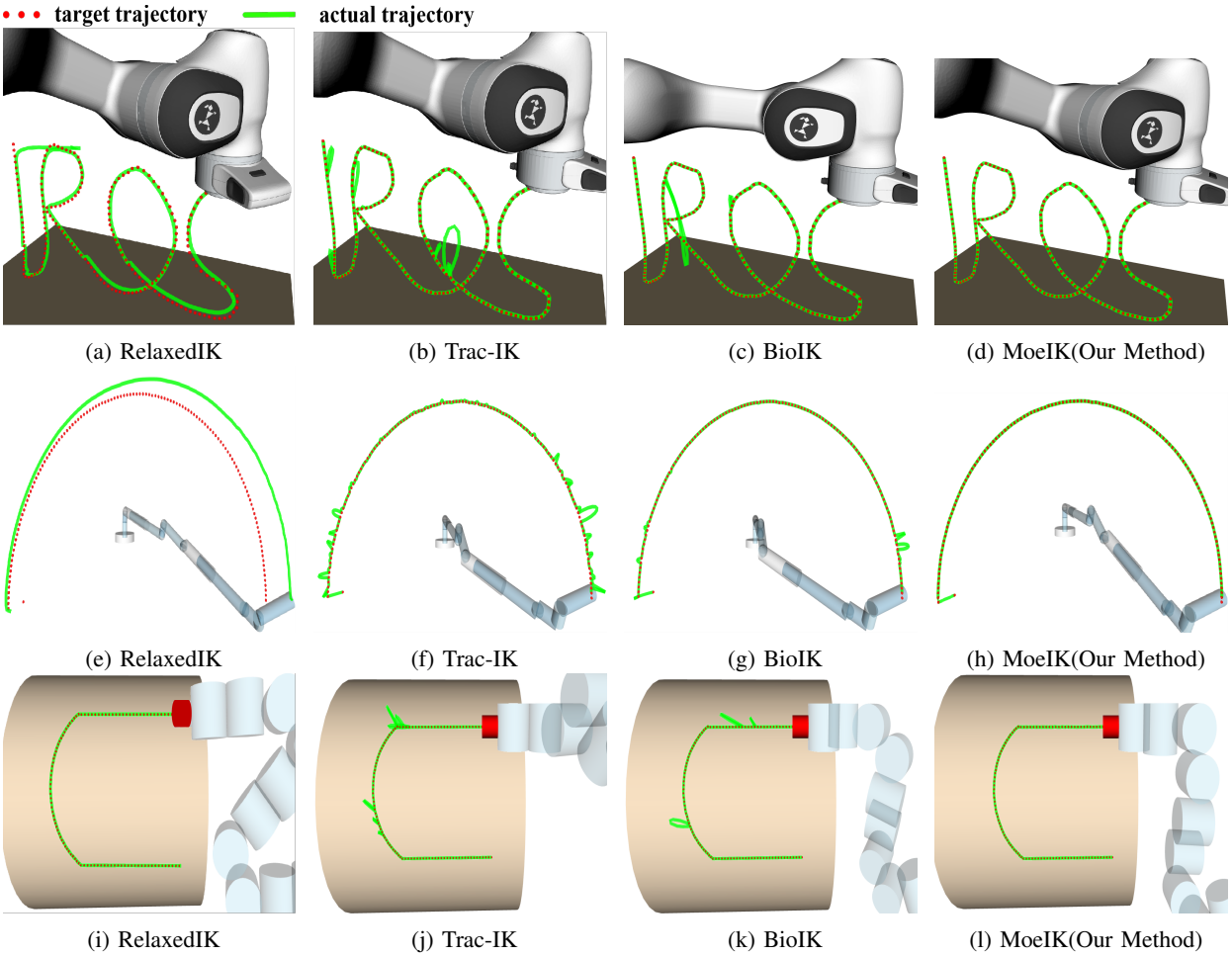
Alg	PE(mm)	OE(°)	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	TCL	Time(ms)
RelaxedIK	**1.9**	**0.15**	**0.19**	**0.12**	**0.06**	**0.04**	**0.36**	**0.31**	**0.06**	**0.04**	**0.25**	**0.30**	**0.04**	**0.05**	**0**	**99**
Trac-IK	**0.08**	**0.05**	**0.85**	**0.79**	**0.58**	**0.84**	**1.62**	**2.09**	**1.02**	**1.41**	**2.19**	**1.70**	**2.28**	**2.02**	**12**	**11**
BioIK	0.15	0.12	0.66	0.50	0.54	0.59	0.80	1.18	0.41	0.63	1.77	1.34	0.63	0.61	**5**	**9.8**
MoeIK(A)	0.99	1.14	2.14	2.57	2.01	2.47	2.23	2.55	2.25	2.49	2.58	2.38	2.55	2.25	4	47
MoeIK(B)	0.46	0.39	0.36	0.35	0.28	0.30	0.47	0.47	0.30	0.31	0.47	0.46	0.36	0.34	2	37
MoeIK	**0.11**	**0.10**	**0.33**	**0.33**	**0.06**	**0.08**	**0.41**	**0.55**	**0.13**	**0.15**	**0.46**	**0.47**	**0.58**	**0.50**	**0**	41

A. *Parameter Setting*

We set the population size to 50, with position accuracy and orientation accuracy identified as the main tasks. The crossover probability was set to 0.9, with a distribution exponent of 20. The polynomial mutation probability was $1/n$, where n represents the number of manipulator joints, and the distribution exponent was 40. The ϵ-weight vectors for position accuracy, orientation accuracy, joint continuity maintenance, joint limit avoidance, and external obstacle avoidance were set to $[1, 1, 0.7, 0.5, 0.5]$. Pose accuracy was designated as the primary task and assigned the highest weights. Joint continuity was given the second-highest priority due to its impact on energy loss. Joint limit avoidance was treated as a secondary objective, while obstacle avoidance, primarily governed by the feasibility principle, was assigned lower weights. For the adaptive interval search, the parameter $m_{\max}$ was set to 10, with an initial interval size m of 3. The value of m was adjusted based on the trajectory point spacing, increasing as the spacing increased. For global searches, m was initialized as $m_{\max}$. During the iterative process, m was adaptively adjusted through the restart strategy, enabling the solution to be obtained. In the restart strategy, the maximum number of evolutionary iterations was set to 100 generations, and the maximum number of restarts was set to 50.

B. *Experiment Setting*

In this paper, we construct a multi-scenario simulation experiment system for three redundant manipulators with varying degrees of freedom. Specifically, the system includes the general-purpose Panda manipulator (7 DoFs, all revolute joints), the KC30 manipulator modified from a tunnel shotcrete machine (8 DoFs, including 2 prismatic joints), and the super-redundant manipulator SSRMS12 (12 DoFs, all revolute joints). Different test trajectories are designed according to the characteristics of each manipulator type. For the Panda manipulator, a handwritten 'IROS' trajectory is adopted to verify its ability to track complex paths. For the KC30 manipulator, a semi-circular tunnel shotcrete trajectory is implemented, with the end-effector maintaining a tangential attitude to meet engineering requirements. For the



(a) RelaxedIK (b) Trac-IK (c) BioIK (d) MoeIK(Our Method)

(e) RelaxedIK (f) Trac-IK (g) BioIK (h) MoeIK(Our Method)

(i) RelaxedIK (j) Trac-IK (k) BioIK (l) MoeIK(Our Method)

Fig. 2: Visualization results of experiments configured for RelaxedIK, Trac-IK, BioIK, and MoeIK. The red points represent the target trajectory points, while the green lines represent the actual trajectories executed by each algorithm.

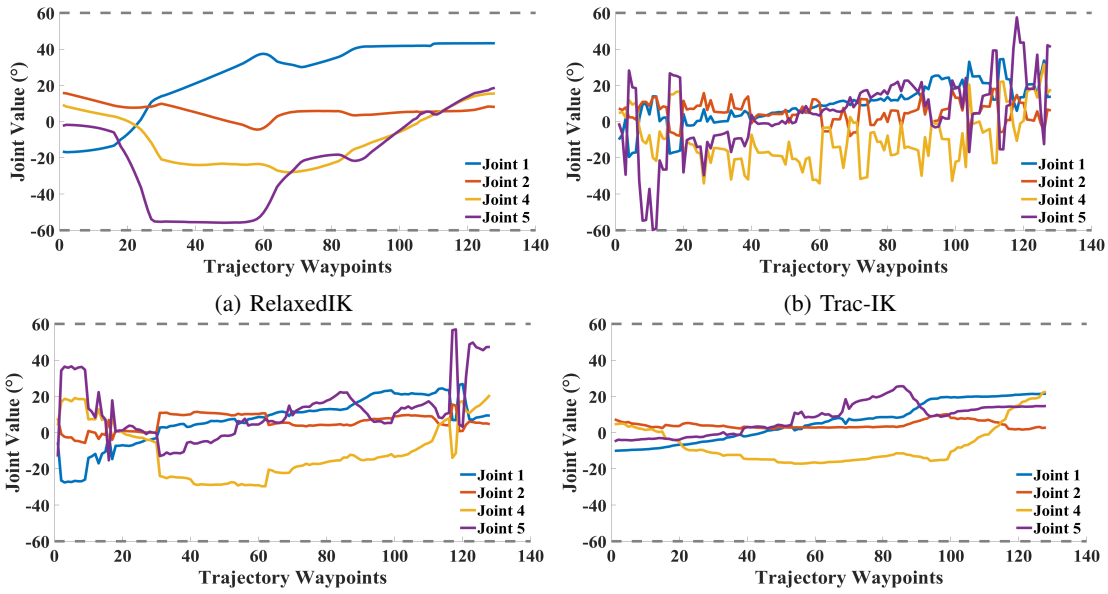
SSRMS12 manipulator, a 'U'-shaped exploration trajectory inside a pipeline is designed to test the space traversal performance of the super-redundant system. In the experiments, RelaxedIK, Trac-IK, and BioIK are used as benchmark algorithms, and two variants of MoeIK are developed through ablation studies. For MoeIK(A), the adaptive interval search and restart strategies were removed compared to the original MoeIK. For MoeIK(B), the fixed interval search was used, and the restart mechanism was removed. During the execution of all algorithms, the initial joint configuration of the manipulator was kept consistent to eliminate the influence of the initial configuration.

C. Evaluation

Our simulation results are shown in Figure 2, while Figure 3 illustrates the changes in some joints of the KC30 during the execution of the planned trajectory. To evaluate the planning results, several metrics are employed, including the average position error (PE), the average orientation error (OE), the average moving distance of joint i (M_i), the number of times close to the joint limit (TCL), and the average time required for the end-effector pose to match each trajectory point (Time), as shown in Tables 1 2 and 3. We then analyzed the results based on these metrics.

RelaxedIK combines multiple objectives through weighted summation and solves them using gradient descent, exhibiting excellent smoothness in joint motion. However, due to the limitations of the local search algorithm, it is prone to getting trapped in local optima, leading to poor performance when the solution space is constrained near the workspace boundary. Additionally, the algorithm is highly sensitive to the spacing between trajectory points. Smaller spacing improves solution accuracy but increases computation time. As shown in the experimental results, the algorithm excels in maintaining joint continuity and trajectory smoothness, but this comes at the cost of accuracy. In particular, for the KC30, when some trajectory points are near the workspace boundary and the spacing is large, finding a solution becomes challenging.

As a traditional inverse kinematics solver, Trac-IK employs Jacobian matrix iteration and the SQP algorithm to compute solutions. It focuses on efficiently computing joint configurations for trajectory points but neglects aspects such as joint continuity and joint limit avoidance. However, as a local search algorithm, Trac-IK struggles to find solutions near singularities and workspace boundaries. For the



(a) RelaxedIK (b) Trac-IK

(c) BioIK (d) MoeIK(Our Method)

Fig. 3: Values of joint 1,2,4,5 by RelaxedIK, Trac-IK, BioIK, and MoeIK during the operation of KC30. The joint ranges are all [-60, 60]. This figure is used to display the joint continuity indices and avoidance of joint limit indices.

Panda and SSRMS12 manipulators, it can quickly achieve high-precision inverse kinematics solutions, but abrupt joint changes during motion may result in poor continuity. Moreover, when some trajectory points of the KC30 are near workspace boundaries, Trac-IK often fails to find solutions, causing a significant drop in overall accuracy. These issues cannot be resolved by increasing computation time, as the algorithm's speed is inherently determined by its design.

BioIK also combines multiple objectives through a weighted approach. It employs a memetic evolutionary algorithm for global exploration and accelerates the solution process using multi-threading and an optimized forward kinematics tree structure. Due to its single-objective evolutionary algorithm, BioIK struggles to effectively enforce the assigned weights, even when they are explicitly defined. Experimental results show that while the algorithm can achieve multiple objectives, its performance in each is suboptimal, and conflicts arise between objectives. For example, improving pose accuracy often leads to degraded joint continuity, and this issue cannot be resolved by increasing computation time.

MoeIK(A) introduces elite inheritance based on a many-objective evolutionary algorithm, enabling the algorithm to achieve preliminary solution capabilities. MoeIK(B) incorporates a fixed interval search strategy, enhancing local exploration and improving computational efficiency. Finally, MoeIK integrates a heuristic restart mechanism and adaptive interval search strategy. On the one hand, the restart mechanism reduces the randomness inherent in evolutionary algorithms, thereby improving robustness. On the other hand, after a restart, the algorithm leverages previous evolutionary results for guidance and adjusts the search interval range, accelerating the process and enhancing computational efficiency. Experimental results show that, compared to other algorithms, MoeIK achieves a more balanced performance across all objectives. It excels in balancing multiple objectives, and even when some trajectory points of the KC30 are near workspace boundaries, it can leverage its global exploration capability to obtain high-quality solutions.

D. Comparison with the Semi-Analytical Framework

Finally, to validate the flexible addition of tasks and demonstrate the complementary advantages of MoeIK over the semi-analytical search framework, we designed a scenario with multiple obstacles. As illustrated in Figure 4, a dynamic ball is placed on the planned trajectory of the Panda manipulator. When the Panda reaches the latter part of its trajectory, the ball blocks its path, requiring emergency obstacle avoidance. If the semi-analytical framework is used, the algorithm can only compute exact solutions and thus becomes trapped in an unsolvable state. However, when the MoeIK method is adopted, it relies on the feasibility principle to ensure that the final joint configuration strictly satisfies the obstacle avoidance constraints. Meanwhile, this method incorporates obstacle avoidance into both the con-

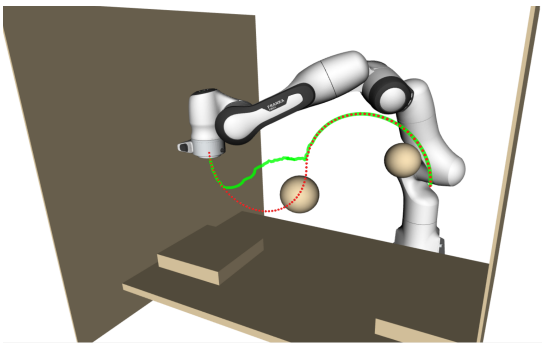


Fig. 4: Visualization of Panda in Collision Scenarios

straint conditions and the optimization objectives, enabling the manipulator to activate the obstacle avoidance mechanism in advance within the preset threshold range, thereby obtaining a broader joint optimization space. Furthermore, in terms of setting the ϵ-weight of the obstacle avoidance optimization objective, by making it smaller than the weight of the pose objective, the manipulator can synchronously complete obstacle avoidance actions while closely tracking the actual trajectory. The MoeIK method replaces invalid exact solutions with effective approximate solutions, successfully achieving real-time motion generation.

V. CONCLUSIONS

This paper proposes the MoeIK algorithm to solve the pathwise inverse kinematics problem for redundant manipulators in complex scenarios. The algorithm is based on an enhanced many-objective evolutionary algorithm featuring elite inheritance and an adaptive interval search strategy. It employs dynamic task metrics for termination and a heuristic restart strategy, enabling robust and efficient planning by optimizing the primary task while considering sub-optimal objectives. The algorithm delivers consistently favorable results across a variety of test cases, demonstrating its effectiveness for multi-constraint, multi-objective problems in high-dimensional spaces. Nevertheless, for robots with higher degrees of freedom and a larger number of concurrent tasks—such as whole-body manipulators and bionic robots—further evaluation and refinement are required to furnish a stable and reliable path-wise motion planner.

REFERENCES

[1] Y. Qin, A. Escande, K. Adrien and E. Yoshida, "Dual-Arm Mobile Manipulation Planning of a Long Deformable Object in Industrial Installation," in IEEE Robotics and Automation Letters, vol. 8, no. 5, pp. 3039-3046, 2023.

[2] J. Laplaza, N. Rodríguez, J. E. DomínguezVidal, F. Herrero, S. Hernández, A. López, A. Sanfeliu and A. Garrell, "IVO Robot: A New Social Robot for Human-Robot Collaboration," 2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pp. 860-864, 2022.

[3] Y. Sun, M. Liu, J. Liu, "Comfortable Following Evaluation Index System for Service Robots in Dynamic Environments," 2021 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 1016-1021, 2021.

[4] Y. Qin, S. Feyzabadi, M. Allan, J. Burdick and M. Azizian, "daVinciNet: Joint Prediction of Motion and Surgical State in Robot-Assisted Surgery," 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2921-2928, 2020.

[5] Z. Sun,H. Yang, Y. Ma, X. Wang, Y. Mo, H. Li and Z. , Jiang, "BIT-DMR: A Humanoid Dual-Arm Mobile Robot for Complex Rescue Operations," in IEEE Robotics and Automation Letters, vol. 7, no. 2, pp. 802-809, 2022.

[6] Z. Jiang, J. Xu, H. Li and Q. Huang, "Stable Parking Control of a Robot Astronaut in a Space Station Based on Human Dynamics," in IEEE Transactions on Robotics, vol. 36, no. 2, pp. 399-413, 2020.

[7] D. Rakita, B. Mutlu and M. Gleicher, "STAMPEDE: A Discrete-Optimization Method for Solving Pathwise-Inverse Kinematics," 2019 International Conference on Robotics and Automation (ICRA), pp. 3507-3513, 2019.

[8] D. Wu, W. Zhang, M. Qin and B. Xie, "Interval Search Genetic Algorithm Based on Trajectory to Solve Inverse Kinematics of Redundant Manipulators and Its Application," 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 7088-7094, 2020.

[9] D. Wu, G. Hou, W. Qiu and B. Xie, "T-IK: An Efficient Multi-Objective Evolutionary Algorithm for Analytical Inverse Kinematics of Redundant Manipulator," in IEEE Robotics and Automation Letters, vol. 6, no. 4,pp. 8474-8481, 2021.

[10] B. Xie, Q. Wang and D. Wu, "Optimal Parameterized Joints Selection to Improve Motion Planning Performance of Redundant Manipulators," 2023 IEEE International Conference on Robotics and Automation (ICRA), pp. 9672-9678, 2023.

[11] L. Sukhan and K. Antal, "Redundant arm kinematic control based on parameterization," Proceedings. 1991 IEEE International Conference on Robotics and Automation, vol. 1, pp. 458-465, 1991.

[12] M. Shimizu, H. Kakuya, W. Yoon, K. Kitagaki and K. Kosuge, "Analytical Inverse Kinematic Computation for 7-DOF Redundant Manipulators With Joint Limits and Its Application to Redundancy Resolution", in IEEE Transactions on Robotics, vol. 24, no. 5, pp. 1131-1142, 2008.

[13] Buss S., "Introduction to inverse kinematics with jacobian transpose,pseudoinverse and damped least squares methods," in IEEE Journal of Robotics , pp. 1-19, 2004.

[14] P. Beeson and B. Ames, "TRAC-IK: An open-source library for improved solving of generic inverse kinematics," 2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids), pp. 928-935, 2015.

[15] M. Kang, H. Shin, D. Kim and S. -E. Yoon, "TORM: Fast and Accurate Trajectory Optimization of Redundant Manipulator given an End-Effector Path," 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 9417-9424, 2020.

[16] R. Falconi, R. Grandi, and C. Melchiorri, "Inverse kinematics of seria -manipulators in cluttered environments using a new paradigm of particle swarm optimization," in IFAC Proceedings Volumes, vol. 47, no. 3,pp. 8475–8480, 2014.

[17] S. Starke, N. Hendrich, and J. Zhang, "Memetic Evolution for Generic Full-Body Inverse Kinematics in Robotics and Animation," in IEEE Transactions on Evolutionary Computation, vol. 23, no. 3,pp. 406-420, 2019.

[18] D. Rakita, B. Mutlu and M. Gleicher, "An analysis of RelaxedIK: an optimization-based framework for generating accurate and feasible robot arm motions," in AUTONOMOUS ROBOTS, vol. 44, no. 7, pp. 1341-1358, 2020.

[19] D. Rakita, H. Shi, B. Mutlu and M. Gleicher, "CollisionIK: A Per-Instant Pose Optimization Method for Generating Robot Motions with Environment Collision Avoidance," 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 9995-10001, 2021.

[20] D. Hadka, P. Reed, and T. Simpson, "Diagnostic assessment of the borg moea for many-objective product family design problems," 2012 IEEE Congress on Evolutionary Computation (CEC), pp. 1-10, 2012.