

Optimal Parameterized Joints Selection to Improve Motion Planning Performance of Redundant Manipulators

Bin Xie, Qingfeng Wang and Di Wu*

Abstract—The redundant manipulators’ analytical solutions can be obtained by the parameterization method. Multiple parameterized joints and their corresponding parametric representations exist for a redundant manipulator. However, how to select the optimal parameterized joints has yet to be well-addressed. This paper delves into the mechanism of the parameterization method and proposes a method to select the optimal parametric representations to improve the motion planning performance of manipulators. We tested the proposed method on an 8-degree-of-freedom (DOF) manipulator. First, all feasible parametric representations are derived, followed by an approach to obtain solution manifolds. We then introduce a metric called the “feasible rate,” which characterizes the percentage of the solution manifold in the joint space. This metric is used to rapidly assess the influence of different parameterized joints on the manipulator’s motion planning performance. To verify the proposed method’s correctness, we evaluated the performance of different representations with the MOEA/D algorithm in solving the same path optimization problems based on the algorithm running time and overall motion magnitude of the manipulator. Our simulation results demonstrate that different selections of parameterized joints affect the motion planning performance, and the performance planned by the optimal parametric representation is up to four times greater than that of the worst one.

I. INTRODUCTION

Redundant manipulators are now widely used in various fields and are expected to perform many complex tasks. Compared to industrial manipulators, redundant manipulators are more flexible and can realize fault tolerance when dealing with problems. Obtaining the inverse kinematic solution in real-time is critical for motion planning [1]–[3]. Obtaining analytical inverse kinematic (IK) solutions can be challenging for non-redundant manipulators that do not meet the Pieper criterion [4], and even more so for redundant manipulators, which admit an infinite number of solutions.

Several numerical approaches such as jacobian transpose, pseudoinverse, and damped least square [5] had been proposed. They translated the IK problems into the velocity domain by the Jacobian matrix, then linearized the joint space around a point and obtained the feasible solutions in the linearized velocity domain. Generally, the Jacobian matrix of redundant manipulators has null space, which is effectively applied to enable the manipulator to perform many subtasks, such as obstacle avoidance [6], singularity avoidance [7], and maximize manipulability [8]. In particular, these constraint subtasks are usually accomplished within the framework of task prioritization [9]. These jacobian-based

approaches can dynamically obtain well-behaved solutions by adding a closed-loop scheme when tracking the end-effector’s trajectory. However, due to the existence of joint limits, they tend to fall into local minima or fail to reach convergence if the beginning point is poorly situated [10]. Furthermore, such methods cannot analyze the nature of the global joint space [12].

The position domain IK has the advantage of high precision and repeatability, which is valuable for application. In [11], the method to describe redundancy characteristics in terms of arm angle was first proposed. Shimizu et al. [12] followed the conception of the arm angle and successfully applied the arm angle parameterization method to the S-R-S manipulators. This method is widely used because of its ability to characterize the self-motion of redundant manipulators by the arm angle. However, this method is only applicable to humanoid manipulators with rotating joints only and arm angle characteristics.

Evolutionary algorithms can efficiently handle IK problems for manipulators with joint limits. The particle swarm optimization (PSO) [13] and genetic algorithm (GA) [14], for example, demonstrated their capability to solve this problem. The memetic evolutionary algorithm [15] was applied to robotics and animation. These methods can obtain optimal feasible solutions from a global perspective and can easily add subtasks. Unfortunately, with the manipulator’s DOF increase, these algorithms’ computational complexity grows geometrically.

Lee and Bejczy [16] proposed the parameterization method to derive the analytical representation of the manipulator, using the gradient descent method to optimize the cost functions iteratively in their application. However, for manipulators with joint limits, their solution manifold regarding a specific end pose is usually non-convex, which causes the disappearance of gradients in certain regions. The parameterization method reduces decision variables in manipulators’ motion planning by using redundant joints to control the mapping between task space and joint space. It seems to be a natural process to combine this method with optimization algorithms to solve the planning problem of manipulators.

Wu. et al. [17] extended the application of the parameterization method by deriving the analytical solutions for a shotcrete manipulator with two prismatic joints. They parameterized several specific joints to obtain analytical solutions and used the parameterized variables as decision variables for inverse kinematics optimization. However, this paper did not analyze how to select parameterized joints. To

Bin Xie, 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

the best of our knowledge, few studies are currently on this issue. This article is the first to select parameterized joints by developing a specific metric.

This paper further explores the parameterization method based on the work in [17]. The main contributions of this paper include the following: 1) the application mechanism of the parameterization method is analyzed deeply, and a method for ranking parameterized joints based on the geometric characteristics of the solution manifold is proposed. 2) The proposed method was used to select the best parameterized joints for an 8-DOF manipulator, and it was confirmed by simulation that the resulting parametric representation improves the manipulator's kinematic performance. 3) The simulation results provide some parameterized joint selection schemes for manipulators with similar structures.

II. THE PARAMETERIZED ANALYTICAL SOLUTIONS

In this section, the parameterization method is illustrated in detail, and the parameterized analytic IK expressions of an 8-DOF manipulator are derived. First, how to represent the redundancy of the manipulator by the parameterization method is described. Then, all feasible parametric representations are obtained and the analytical IK expressions of a parametric representation are derived.

A. Parameterization of Redundant Joints

Generally, the manipulator's redundancies mean that the λ DOFs of the manipulator are greater than the μ DOFs required to complete the task. The parameterization method selects specific joints as redundancy parameters and utilizes these joints to obtain "parameterized" analytical solutions for the redundant manipulator. In Fig.1, the solving procedure of the parameterized analytical solution is shown. The parameterization method divides the full joint space of the manipulator into a parameterized joint space $\theta_p \in \mathbb{R}^{\lambda-\mu}$ and a non-parameterized joint space $\theta_r \in \mathbb{R}^{\mu}$, and the analytical expressions for the latter are derived from the forward kinematic model. Therefore, the mapping between the target pose matrix T and the non-parameterized space θ_r is written as (1). This mapping is carried out within the parameterized space θ_p .

$$\theta_r = f^{-1}(T, \theta_p) \quad (1)$$

where f^{-1} denotes the analytical inverse kinematics derived using the parameterization method, known as a parametric representation in this paper.

To further elaborate the parameterization method, we transform (1) into μ linearly independent equations of the following form:

$$g_i(\theta_1, \theta_2, \dots, \theta_\lambda) = h_i(\theta_1, \theta_2, \dots, \theta_\lambda), i = 1, 2, \dots, \mu \quad (2)$$

where g_i and h_i represent transcendental equations with joint angles θ_i as variables. Since θ_i does not appear on both sides of the equation, the left and right sides of (2) are always unequal. The essence of the parameterization method is to reduce the number of variables to match constraints

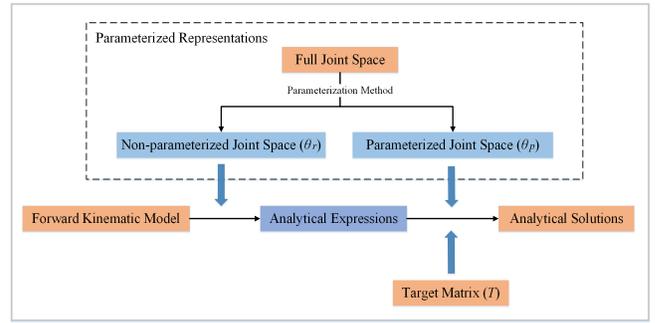


Fig. 1: Parametric analytical inverse kinematic solutions procedure

and DOFs in the equation system. Ideally, we can obtain finite analytical solutions by parameterizing only $\lambda - \mu$ joint variables. Unfortunately, the joint variables in (2) are strongly coupled. The analytic expressions cannot always be found even if this equation system satisfies the $\lambda = \mu$ condition. We can continue increasing the number of parameterized joints. However, the increase will over-constrain the equation system, which easily provides wrong solutions and give a higher challenge to obtain feasible solutions in arithmetic power consumption.

The method for determining the minimum number of parameterized joints is as follows: For a certain manipulator, we find the equations with the least number of variables in (2) and then derive expressions of the rest joints one by one. If there is no solution, the number of parameterized joints increases until all the analytic expressions of non-parameterized joints are obtained. In this article, only the $\lambda - \mu$ joint variables of the manipulator are parameterized.

B. Inverse Kinematics Using Parameterization Method

We applied the above method to an 8-degree-of-freedom shotcrete manipulator with a nozzle as the end-effector. As shown in Fig. 2, the third and the sixth joint are prismatic joints, and the rest are all revolute types. The detailed D-H parameters can be referred from [18]. The posture of the end-effector can be presented by the product of a series of homogeneous transformation matrices as shown in (3).

$${}^e_0T = \prod_{i=1}^8 {}^i_{i-1}T \cdot {}^e_8T \quad (3)$$

where the ${}^i_{i-1}T \in \mathbb{R}^{4 \times 4}$ is the transformation matrix from the coordinate system $\{i-1\}$ to the $\{i\}$. The $\{0\}$ and the $\{e\}$ represent the coordinates fixed to the base and end-effector, respectively, and the e_0T describes the posture of the end-effector in the form shown in (4).

$${}^e_0T = \begin{bmatrix} n & o & a & p \\ - & O & - & | \\ & & & 1 \end{bmatrix} \quad (4)$$

where the $[n \ o \ a] \in SO(3)$ and the $p \in \mathbb{R}^{3 \times 1}$ specify the orientation and position of end coordinate system relative to base one, respectively. Moreover, we denote the elements located in the m_r -th row and m_c -th column of the matrix ${}^i_{i-1}T$ as (m_r, m_c) .

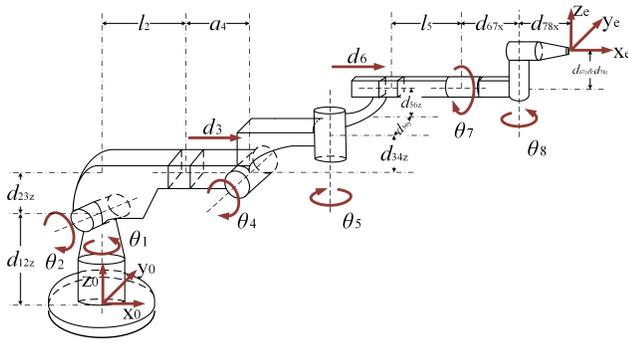


Fig. 2: KC-30 manipulator model

The KC-30 requires only five DOFs for the shotcrete task, so three redundant DOFs can be parameterized: 1) As the full task space of the KC-30 has six DOFs, there is one redundant DOF in the task space for the shotcrete task. We use the XZX Euler angle to determine the nozzle's pose of the manipulator. As the second X (recorded as x_e) is related to the nozzle's movement, its value is not fixed and should be parameterized. 2) The remaining two redundant DOFs are naturally chosen from the joint space and recorded as θ_x and θ_y . Thus, we define the parameterized joints as $G_{xy} = \{\theta_x, \theta_y, x_e\}$ and its corresponding parametric representation as R_{xy} .

Generally, multiple parametric representations exist for the KC-30. To obtain a parametric representation of the KC-30, we must find three-variable equations with two selected parameterized joints in forward kinematics (FK). By treating the parameterized joints as known quantities, we can derive analytical expressions for the rest joints. Only four initial equations with three joint variables are found by performing a series of coordinate transformations on (3). These four initial equations can be derived from the four expressions in the second row of eT in the form shown in (5a) - (5d).

$$\begin{cases} As_1 + Bc_1 = (d_{67x} + l_5 + d_6)s_5 + d_{56y}c_5 & (5a) \\ s_1a_x + c_1a_y + c_5s_7 = 0 & (5b) \\ (s_1n_x + c_1n_y)c_8 + (s_1o_x + c_1o_y)s_8 = s_5 & (5c) \\ Cs_7 + (s_1a_x + c_1a_y)c_7 = 0 & (5d) \end{cases}$$

where $A = (p_x - d_{78x}n_x - d_{78x}a_x - d_{67z}a_x)$, $B = (p_y - d_{78x}n_y - d_{78z}a_y - d_{67z}a_y)$, $C = [(s_1o_x + c_1o_y)c_8 - (s_1n_x + c_1n_y)s_8]$, and the c_i and s_i represent the $\sin(\theta_i)$ and $\cos(\theta_i)$, respectively.

Therefore, we only obtain 12 kinds of parametric representations for KC-30 based on the parameterization method, which are grouped into 8 optional parameterized joints shown in Fig.3. Next, let us select $G_{56} = \{\theta_5, d_6, x_e\}$ as an example to derive the parametric representation.

1) **Joint 1:** The θ_1 is obtained from (5a) that

$$\begin{cases} \theta_1 = atan2(m_1, \pm\sqrt{A^2 + B^2 - m_1^2}) - atan2(B, A) & (6) \\ m_1 = (d_{67x} + l_5 + d_6)s_5 + d_{56y}c_5 \end{cases}$$

where $atan2(y, x)$ is a bivariate arctangent function.

2) **Wrist Joints:** Once the first joint is determined, the solutions for the two wrist joints can be obtained. Turning the view to (5b) which can derive θ_7 .

$$\begin{cases} \theta_7 = atan2(s_7, \pm\sqrt{1 - s_7^2}) & (7) \\ s_7 = -(s_1a_x + c_1a_y)/c_5 \end{cases}$$

Then, we select the element (2,1) of eT to obtain θ_8 .

$$\begin{cases} \theta_8 = atan2(s_5, c_5c_7) - atan2(m_2, \pm\sqrt{s_5^2 + (c_5c_7)^2 - m_2^2}) & (8) \\ m_2 = s_1n_x + c_1n_y \end{cases}$$

3) **Shoulder Joints:** First, we derive the expression of θ_{24} (i.e. $[\theta_2 + \theta_4]$) by element (1,2) of eT .

$$\begin{cases} \theta_{24} = atan2(k_1, s_7c_8) - atan2(m_3, \pm\sqrt{k_1^2 + (s_7c_8)^2 - m_3^2}) & (9) \\ k_1 = c_5c_8 - s_5c_7c_8 \\ m_3 = c_1o_x - s_1o_y \end{cases}$$

Then, the elements (1,4) and (2,4) of eT are combined to calculate the value of θ_2 and θ_4 .

$$\begin{cases} \theta_4 = atan2(k_2, k_3) - atan2(-d_{23z}, \pm\sqrt{k_2^2 + k_3^2 - d_{23z}^2}) & (10) \\ \theta_2 = \theta_{24} - \theta_4 \\ k_2 = k_4s_{24} - k_5c_{24} + d_{56z} + d_{34z} \\ k_3 = k_4c_{24} + k_5s_{24} - (l_5 + d_6)c_5 + d_{56y}s_5 - a_4 \end{cases}$$

Finally, we select the element (1,4) of eT to derive the analytical expression for d_3 .

$$d_3 = k_6/c_2 - l_2 \quad (11)$$

In (10) and (11), the formulas of the k_4 , k_5 and k_6 are complex to be shown here. Then, by giving values to the parameterized joints, the angles of the other joints can be generated from these expressions. Furthermore, other parametric representations of KC-30 can be derived by a similar process.

III. SOLUTION MANIFOLD ANALYSIS OF PARAMETRIC REPRESENTATIONS

For redundant manipulators, an end-effector pose corresponds to a unique solution manifold. However, the different parametric representations impact the geometric characteristics of the feasible solution manifolds. This section discusses

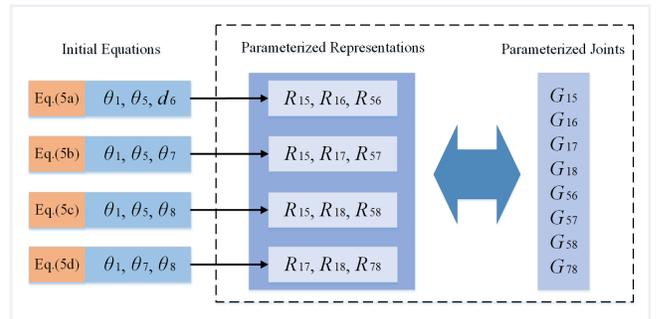


Fig. 3: Optional parameterized joints and parametric representations for the KC-30

how to select the optimal parameterization representation with the solution manifold. First, the ill-conditioned mapping of the parametric representation is discussed. Then, an algorithm for obtaining the complete feasible solution manifold is presented. Finally, the effect of ill-conditioned mapping relations on the solution manifold is verified, and the optimal parameterized joints are preliminarily selected.

A. Ill-conditioned Mapping

Theoretically, when the end-effector pose is specified, its feasible configurations should be determined. However, changing the parameterized joints alters the derivation order of the non-parameterized ones in different representations, which may result in unfavourable mappings between all joint variables. For example, in (5a), the d_6 is always mapped to the θ_5 by (12) when the θ_1 and the d_6 are chosen as the part of parameterized joints. The θ_5 has no solution only if both independent variables in all $\text{atan2}(y, x)$ are zero.

$$\begin{cases} \theta_5 = \text{atan2}(m_4, \pm\sqrt{d_{56}^2 + k_7^2 - m_4^2}) - \text{atan2}(d_{56y}, k_7) \\ k_7 = d_{67x} + l_5 + d_6 \\ m_4 = As_1 + Bc_1 \end{cases} \quad (12)$$

Conversely, if the θ_5 is chosen as the parameterized joint and d_6 is not, their mapping will follow the form of (13). When the independent variable θ_5 is zero, the d_6 has no solutions.

$$d_6 = (As_1 + Bc_1 - d_{56y}c_5)/s_5 - d_{67x} - l_5 \quad (13)$$

The mappings among joint variables may be ill-conditioned, resulting in incomplete mappings between task space and joint space for certain parametric representations. To investigate the impact of different parameterized joints on this issue, we can obtain the solution manifold of the manipulator for an end-accuracy task.

B. Acquisition of A Solution Manifold

An approach for obtaining a solution manifold by traversing all the parameterized joint variables in their joint ranges is proposed, shown in Algorithm 1.

The functions *calmatrix*, *ikine* and *fkine* in Algorithm 1 are used to generate the target matrix, obtain the analytic IK solutions and calculate the FK, respectively.

1) **Target Matrix T** : An end-effector pose X contains five elements. The first three elements represent the tip position, and the rest are the first two elements of the XZX Euler angles. x_e is the remaining Euler angle to be traversed. With these parameters, we can get the target matrix T by the function *calmatrix*.

2) **IK Solutions Sets S** : The function *ikine* usually generates multiple sets of solutions S . Considering the properties of trigonometric functions, we transform the IK solutions in S into $[-180^\circ, 180^\circ]$.

3) **Feasible IK solutions**: A feasible solution must satisfy two requirements: 1) All joints can not violate the joint limit; 2) The matrix norm error between the forward kinematic matrix T_c obtained by the IK solution and the desired matrix T should be less than a minimal value ϵ .

Algorithm 1:

Input: tip pose $X = [p_x, p_y, p_z, \alpha, \beta]$,
parameterized joints G_{xy} ,
joints limits $\theta_i \in [\theta_i^l, \theta_i^u]$,
traversal step length L_i ,
reachable space Θ_{lim}

Output: set of feasible configurations P

```

1 for  $\theta_x \leftarrow \theta_x^l$  to  $\theta_x^u$  by  $L_x$  do
2   for  $\theta_y \leftarrow \theta_y^l$  to  $\theta_y^u$  by  $L_y$  do
3     for  $x_e \leftarrow x_e^l$  to  $x_e^u$  by  $L_{x_e}$  do
4        $T \leftarrow \text{calmatrix}(X, x_e)$ ;
5        $S[1..j] \leftarrow \text{ikine}(\theta_x, \theta_y, T)$ ;
6       for  $k \leftarrow 1$  to  $j$  do
7          $T_c \leftarrow \text{fkine}(S[k])$ ;
8         if  $S[k] \in \Theta_{lim}$  &  $\|T_c - T\| \leq \epsilon$  then
9            $P \leftarrow S[k]$ ;
10        end
11      end
12    end
13  end
14 end
```

C. Optimal Parameterized Joints

To obtain solution manifolds for all parametric representations, we start by setting some parameters. First, we randomly select an end-effector's pose X_1 as shown in the first row of TABLE II. Then, set the traversal step of the revolute joints and prismatic joints as 0.1° and 1 mm, and fix the step length $L_{x_e} = 1$, respectively. All joint movable ranges are listed in TABLE I. We select a three-dimensional space $\{\theta_5, d_6, x_e\} \in \mathbb{R}^3$ to display a partial characteristic of the solution manifolds. The running results of algorithm 1 are shown in Fig. 4.

In Fig. 4, the solution manifolds have the same contour but varying fullness. A metric called the "Feasibility Rate" (FR) is proposed to quantify the differences between these manifolds, which is defined by (14):

$$\mathbf{FR} = \frac{N_m}{N_j} \times 100\% \quad (14)$$

where N_m and N_j denote the count of feasible configurations within the solution manifold corresponding to an end-effector's pose and the number of points traversed in the joint space, respectively. To ensure the generalizability of experimental results, we randomly sampled 50 poses in the reachable space to calculate their FR. However, only three sets of results are presented in Table II due to article length constraints.

The experimental data reveal that: 1) Different parametric representations affect the **FR** of the solution manifolds. The

TABLE I: Upper and Lower Limits of Each Joint

	$\theta_1(^{\circ})$	$\theta_2(^{\circ})$	$d_3(\text{mm})$	$\theta_4(^{\circ})$	$\theta_5(^{\circ})$	$d_6(\text{mm})$	$\theta_7(^{\circ})$	$\theta_8(^{\circ})$	$x_e(^{\circ})$
θ_1^u	45	18	2000	40	60	2000	180	124	180
θ_1^l	-45	-8	0	-40	-60	0	-180	-116	-180

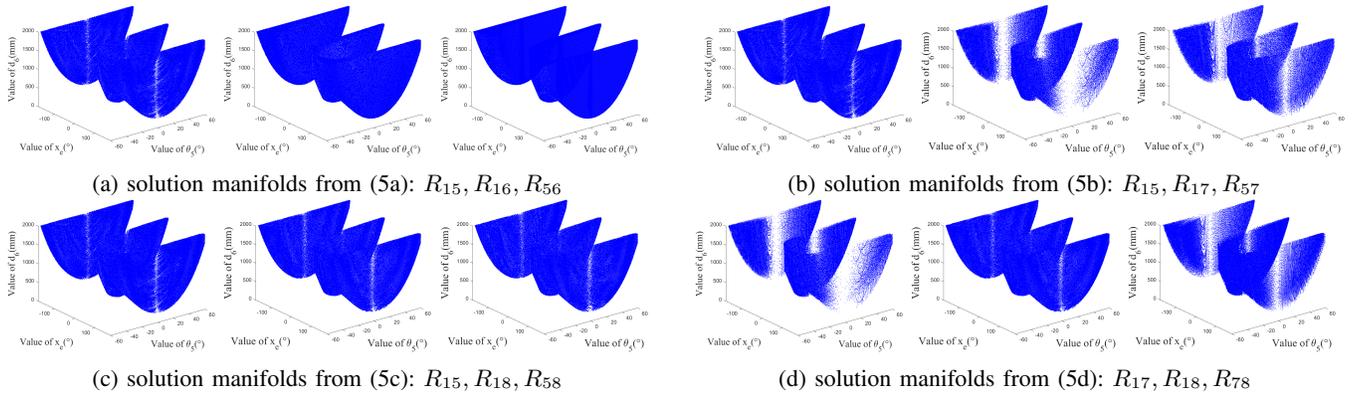


Fig. 4: Solution manifolds of twelve parametric representations

TABLE II: The Feasible Rate of the Solution Manifold of Different Parametric Representations Corresponding to Different Target Points

Target Postures	Parametric Representations											
	(5a)			(5b)			(5c)			(5d)		
	R_{15}	R_{16}	R_{56}	R_{15}	R_{17}	R_{57}	R_{15}	R_{18}	R_{58}	R_{17}	R_{18}	R_{78}
$X_1 = [11223.67, -4928.52, 2581.57, 19.00, 50.00]$	0.610%	6.41%	12.8%	0.612%	0.122%	0.0972%	0.616%	0.416%	0.311%	0.122%	0.418%	0.0154%
$X_2 = [9999.23, 620.94, 2373.58, 45.30, 61.20]$	0.277%	1.60%	3.57%	0.276%	0.124%	0.0912%	0.277%	0.229%	0.172%	0.123%	0.228%	0.0139%
$X_3 = [9397.67, -3493.10, -240.52, 173.02, 104.00]$	0.325%	2.45%	4.94%	0.320%	0.0173%	0.0303%	0.325%	0.131%	0.116%	0.0173%	0.131%	0.0175%

¹ (5a)-(5d) is the four initial equations to derive analytical expressions.

² the parametric representations of R_{xy} obtained from different initial equations are different.

representations with the d_6 have the fullest manifolds, while those with the θ_7 generate the most sparse. 2) Solution manifolds with the same parameterized joints have similar **FR** values. 3) The relative size among the FR of some parametric representations varies across different postures. In column (5b) of Table II, the relative size between the FR of the solution manifolds generated by R_{17} and R_{57} for the target posture X_3 is different from the other two postures.

A complete mapping between task space and joint space is crucial for motion planning, especially for the path-ik problem, as it provides a large number of feasible configurations [19]. According to the experimental results, the solution manifold of the R_{56} has the largest FR, indicating its potential to provide the largest number of feasible configurations. Therefore, we select the parameterized joints G_{56} as the optimal choice.

IV. EXPERIMENT

To verify the effectiveness of the previously proposed method, we apply all parametric representations to solve a path optimization problem. Moreover, parameter settings and performance analysis are both illustrated in this section.

A. Task Description

The task optimizes an existing path in the joint space to obtain a smooth path away from the joint limits, and the problem can be described by

$$\begin{aligned}
 & \text{minimize} && F(\Theta) = f_{smooth}(\Theta) + f_{limit}(\Theta) \\
 & \text{subject to} && H(\Theta) = T_{target} \\
 & && \Theta^l \leq \Theta \leq \Theta^u
 \end{aligned} \quad (15)$$

where $F(\Theta)$ is the cost function that consists of two terms: f_{smooth} is a path-smoothing term that describes the continu-

ity of the manipulator's motion at the joint space, and f_{limit} is a limit term that measures the cost of approaching the joint limit. These terms are fully described in [17]. $H(\Theta)$ is the forward kinematic function. In [17], the proposed T-ik algorithm has shown excellent performance on this problem, but the algorithm itself has many parameters related to the joint properties, such as the interval search operator. Therefore, we decide to solve our problem with the MOEA/D algorithm [20], which has fewer parameters to adjust.

B. Parameter Setting

We choose binary codes to describe the decision variables for each individual and set the resolution of the joint corresponding to each binary code to be less than 0.1° or 1 mm. The crossover probability and mutation probability are 0.7 and 0.1, respectively, and the probability of each neighboring individual being selected for the update is 0.6. Then, the weights of each joint are $[0.125, 0.125, 0.125, 0.125, 0.125, 0.125, 0.125, 0.125]$ in the f_{smooth} function, while the weights of each joint are $[0.143, 0.143, 0.143, 0.143, 0.143, 0.143, 0, 0.143]$ in the f_{limit} function. All the experiments were conducted on a Legion R9000 laptop with a 3.20GHz processor core.

C. Simulation

We generate two different types of paths and sample the path points uniformly on them. Then, the desired tip pose of each point is set according to the task requirements.

1) *Circular Arc Path*: The first path is a circular arc path with 129 discrete path points, and the desired tip pose for each path point is that the tip motion axis coincides with the circular arc radius.

TABLE III: Comparison of the Performance of Different Parameterized Joints

Parameterized Joints	Average Joint Motions								Planning Time(s)	Overall Motion Magnitude	Synthetic Performance
	$\theta_1(^{\circ})$	$\theta_2(^{\circ})$	$d_3(mm)$	$\theta_4(^{\circ})$	$\theta_5(^{\circ})$	$d_6(mm)$	$\theta_7(^{\circ})$	$\theta_8(^{\circ})$			
Circular Arc Path											
G_{15}	3.48	1.82	229.47	4.08	9.43	202.51	33.69	34.07	9.37	0.69	1.28
G_{16}	2.82	1.88	224.13	3.80	7.67	220.63	27.77	29.31	5.22	0.64	0.97
G_{17}	2.90	1.99	199.81	4.55	7.92	173.00	32.42	30.80	43.66	0.64	3.38
G_{18}	3.26	1.98	206.14	4.40	9.07	175.32	38.67	37.93	8.01	0.70	1.20
G_{56}	2.84	1.86	215.70	3.91	8.11	198.85	31.73	32.70	2.52	0.65	0.81
G_{57}	2.44	1.84	197.29	4.15	6.87	162.37	24.70	24.50	34.02	0.56	2.70
G_{58}	3.46	2.09	218.47	4.60	9.89	165.93	36.31	36.54	11.36	0.70	1.42
G_{78}	/	/	/	/	/	/	/	/	/	/	/
Straight-line Path											
G_{15}	3.35	1.10	210.50	1.90	8.90	185.63	22.21	23.42	39.68	0.54	4.26
G_{16}	4.81	1.25	255.67	2.09	12.6	240.91	37.68	40.39	17.08	0.75	2.36
G_{17}	3.51	1.27	193.66	2.40	9.60	175.66	28.61	30.08	21.20	0.59	2.58
G_{18}	4.36	1.41	204.45	2.61	12.05	189.28	37.20	39.70	12.15	0.70	1.84
G_{56}	4.31	1.16	233.44	1.96	11.73	208.67	35.45	37.92	9.02	0.69	1.54
G_{57}	3.38	1.28	202.20	2.43	9.51	161.73	26.41	27.71	29.75	0.56	3.35
G_{58}	4.15	1.38	218.70	2.55	11.92	179.05	33.64	35.61	16.53	0.67	2.22
G_{78}	/	/	/	/	/	/	/	/	/	/	/

* Average Joint Motion, the average motion of each joint between two adjacent desired tip pose after 100 runs; Planning Time, the average planning time of 100 runs.

2) *Straight-line Path*: This path is a straight-line grasping path with 150 discrete path points, and the tip always faces vertically downward.

As previously mentioned, representations with the same parameterized joints have similar **FRs**. Therefore, we only perform experiments for 8 optional parameterized joints, and their planning performance is shown in TABLE III. Moreover, all planning results stay within the joint limits.

To visually describe the smoothness of the path, we specified an indicator function, as shown in (16).

$$E_m = \sum_{i=1}^8 \frac{\Delta\theta_i}{\theta_i^u - \theta_i^l} \quad (16)$$

where E_m means overall motion magnitude, and the $\Delta\theta_i$ is the average motions of each joint. In the TABLE III, the G_{57} and G_{15} allow the KC-30 to have minimal motion in the circular arc and straight-line paths, respectively, because mainly mobilized joints to complete two paths are different. However, they both have the worst performance on planning time, while the G_{56} has the best. In addition, the G_{78} is unable to complete the task in the desired time.

To consider the performance of both aspects together, we first need to normalize the planning time by

$$E_n = \frac{pt_i}{\sqrt{\frac{1}{(n-1)} \sum_{i=1}^n (pt_i - \frac{1}{n} \sum_{i=1}^n pt_i)^2}} \quad (17)$$

where pt_i is the i_{th} element of the ‘‘Planning Time’’ column. Then, the synthetic performance is described by

$$E_s = E_m + E_n \quad (18)$$

The results are listed in the last column of TABLE III, where the G_{56} has the smallest value. For those parameterized joints with the smallest overall motion magnitude in the table, the planning is very time-consuming, which results in poor overall performance.

According to the experimental results, we obtained the following three points: 1) The different parameterized joints selection will affect the task planning results; 2) A suitable

parameterized joint can improve the manipulator’s motion planning performance by obtaining the optimal solution manifold, with the optimal performance being up to four times better than the worst; 3) The feasible rate (**FR**) can be used to select the most suitable parameterized joints rapidly, and for the KC-30 manipulator, the revolute joint θ_5 and the prismatic joint d_6 are the best choices.

V. CONCLUSION

This paper proposed a method to select optimal parameterized joints to improve the motion planning performance of the manipulator. We first analyzed the parameterization method’s mechanism and developed a method to find the minimum number of parameterized joints. We then applied this method to an 8-DOF shotcrete manipulator and found twelve parametric representations that could be grouped into eight optional parameterized joints. We characterized their solution manifolds using the feasible rate to distinguish between different parametric representations and initially selected G_{56} , which had the fullest solution manifold, as the best choice.

To verify that well-chosen parameterized joints can improve the manipulator’s motion planning performance, we combined the parametric representations with the MOEA/D algorithm to handle the same path-planning task. Comparing simulations were carried out on circular arc path and straight-line path, which demonstrated that different parameterized joints affect the planning performance. In addition, the proposed method can provide optimal parameterized joints G_{56} for the KC-30.

As far as we know, this is the first article to provide a detailed discussion on selecting parameterized joints and their effects. In future work, we plan to identify common patterns in the parameterized joints selection by applying this method to various types of manipulators.

REFERENCES

- [1] F. Marić, M. Giamou, A. W. Hall, S. Khoubyarian, I. Petrović and J. Kelly, ‘‘Riemannian Optimization for Distance-Geometric Inverse Kinematics,’’ IEEE Transactions on Robotics, vol. 38, no. 3, pp. 1703-1722, 2022.

- [2] P. Trutman, M. S. E. Din, D. Henrion and T. Pajdla, "Globally Optimal Solution to Inverse Kinematics of 7DOF Serial Manipulator," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6012-6019, 2022.
- [3] B. Ames, J. Morgan and G. Konidaris, "IKFlow: Generating Diverse Inverse Kinematics Solutions," in *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7177-7184, 2022.
- [4] Pieper and D. L. Peiper, "The kinematics of manipulators under computer control", *Stanford Artif. Intell. Project*, no. AI-72, 1968.
- [5] S. R. Buss, "Introduction to inverse kinematics with jacobian transpose pseudoinverse and damped least squares methods", *IEEE Transaction on robotics and automation*, vol. 17, no. 1, pp. 1-19, 2004
- [6] Y. Ding and U. Thomas, "Collision Avoidance with Proximity Servoing for Redundant Serial Robot Manipulators," *IEEE International Conference on Robotics and Automation*, pp. 10249-10255, 2020.
- [7] Y. -H. Chang, Y. -C. Liu and C. -C. Lan, "An End-Effector Wrist Module for the Kinematically Redundant Manipulation of Arm-Type Robots," *IEEE International Conference on Robotics and Automation*, pp. 6075-6080, 2020.
- [8] B. H. Meng, I. S. Godage and I. Kanj, "Smooth Path Planning for Continuum Arms," *IEEE International Conference on Robotics and Automation*, pp. 7809-7814, 2021.
- [9] M. Karimi and M. Ahmadi, "A Reinforcement Learning Approach in Assignment of Task Priorities in Kinematic Control of Redundant Robots," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 850-857, 2022.
- [10] M. Kang, H. Shin, D. Kim and S. -E. Yoon, "TORM: Fast and Accurate Trajectory Optimization of Redundant Manipulator given an End-Effector Path," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 9417-9424, 2020.
- [11] K. Kreuz-Delgado, M. Long and H. Seraji, "Kinematic analysis of 7 DOF anthropomorphic arms," *Proceedings., IEEE International Conference on Robotics and Automation*, pp. 824-830 vol. 2, 1990.
- [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," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1131-1142, 2008.
- [13] Yang Y, Hong-ze X, Shao-hua L, Ling-ling Z, Xiu-ming Y, "Time-optimal trajectory optimization of serial robotic manipulator with kinematic and dynamic limits based on improved particle swarm optimization," *The International Journal of Advanced Manufacturing Technology*, vol. 120, no. 1-2, pp. 1253-1264, 2022.
- [14] M. Ahmad, N. Kumar, and R. Kumari, "A hybrid genetic algorithm approach to solve inverse kinematics of a mechanical manipulator," *International Journal of Scientific & Technology Research*, vol. 8, pp. 1777-1782, 2019.
- [15] S. Starke, N. Hendrich and J. Zhang, "Memetic Evolution for Generic Full-Body Inverse Kinematics in Robotics and Animation," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 3, pp. 406-420, 2019.
- [16] S. Lee and A. K. Bejczy, "Redundant arm kinematic control based on parameterization," *Proceedings., IEEE International Conference on Robotics and Automation*, vol. 1, pp. 458-465, 1991.
- [17] D. Wu, G. Hou, W. Qiu and B. Xie, "T-IK: An Efficient Multi-Objective Evolutionary Algorithm for Analytical Inverse Kinematics of Redundant Manipulator," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 8474-8481, 2021.
- [18] 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," *IEEE International Conference on Robotics and Automation*, pp. 7088-7094, 2020.
- [19] D. Rakita, B. Mutlu and M. Gleicher, "STAMPEDE: A Discrete-Optimization Method for Solving Pathwise-Inverse Kinematics," *IEEE International Conference on Robotics and Automation*, pp. 3507-3513, 2019.
- [20] Q. Zhang and H. Li, "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712-731, 2007.