A Sampling-Based Motion Assignment Strategy With Multi-Performance Optimization for Macro-Micro Robotic System

Yaohua Zhou, Chin-Yin Chen, *Member, IEEE*, Guilin Yang, *Member, IEEE*, and Yaonan Li

***Abstract*—This article proposes a sampling-based motion assignment strategy for coordinated motion planning of macro-micro robotic systems. It is used to achieve performance enhancements while solving joint trajectories. The sampling strategy is implemented by traversing a series of feasible sets generated by the trajectory constraints of the micro robot. Meanwhile, two kinds of performance index maps are introduced to achieve normalization and integration of multiple performance indices. They are used for iterative generation of trajectories and overall performance evaluation, respectively. Comparative numerical results prove the validity of the proposed strategy.**

***Index Terms*—Industrial robots, macro-micro robotic system, motion assignment, redundant robots.**

I. Introduction

MULTI-ROBOT operation modes are being used more often for a variety of industrial tasks (e.g., milling, polishing, putty applying and laser cutting etc.), with their forms mainly including cooperative multi-robot cells [1], [2] and macro-micro robotic systems (MMRSs) [3], [4]. They excel in dexterity and versatility compared with a single industrial robot. The MMRS shown in Fig. 1(a) was designed to perform polishing tasks on complex curved surfaces (such as wheel hubs) [5]. It consists of a traditional industrial manipulator IRB 4400 (called macro) and a two-rotational and one-translational (2R1T) end-effector (called micro) to perform tasks synergistically.

The macro has a large workspace but low motion resolution and operational flexibility, while the micro can complement it by virtue of its structural characteristics. However, participation of the micro results in redundancy and complexity of the whole system. Therefore, for those tasks with continuous trajectories (such as a polishing process), motion assignment (MA) is a critical concern, i.e., performing trajectory planning for the macro and micro respectively on the premise of satisfying the constraints of a given task path (see Fig. 1(b)). Furthermore, the emergence of redundancy provides channels for performance enhancements. Thus, the corresponding optimization strategy must be considered and embedded into MA.

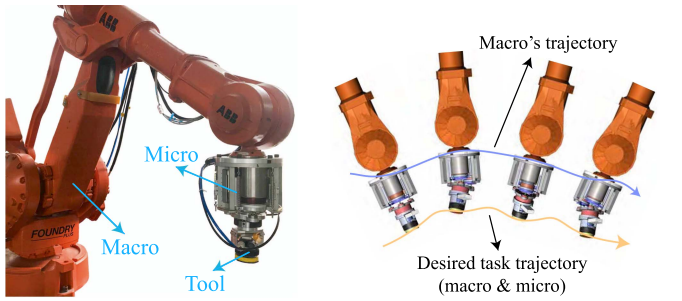


Fig. 1. (a) The prototype of our MMRS: It consists of a industrial robot IRB 4400 (macro), a 2R1T end-effector (micro) and a polishing tool; (b) A description of motion assignment.

A. Related Work

Redundant robots performing trajectory tracking in Cartesian space require well-behaved inverse kinematic schemes. Those iterative schemes based on Jacobian matrices [6] are the most widely accepted, e.g., using the transpose, Moore-Penrose inverse, additional damping [7] or augmentation [8] for iterative solutions. However, most of them are affected by iterative initial values and prone to fall into local optima. Neural networks can also be used to build the inverse kinematic models of redundant robots [9], while their accuracy and generalization performance require corresponding verification.

Considering the requirements of relevant performances, such as manipulability enhancement, obstacle avoidance or joint limit avoidance, the gradient projection method [10] is sometimes adopted. It is implemented based on the null space of the Jacobian matrix, while dealing with multiple complex constraints is a challenge. From the perspective of performance optimization

Manuscript received 29 April 2022; accepted 26 August 2022. Date of publication 7 September 2022; date of current version 14 September 2022. This letter was recommended for publication by Associate Editor D. Zanotto and Editor C. Gosselin upon evaluation of the reviewers' comments. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1308900 and in part by NSFC-Shenzhen Robotic Fundamental Research Center Project under Grant U1813223. *(Corresponding author: Chin-Yin Chen.)*

Yaohua Zhou is with the Zhejiang Key Laboratory of Robotics and Intelligent Manufacturing Equipment Technology, Ningbo Institute of Materials Technology and Engineering, Chinese Academy of Sciences, Ningbo 315201, China, and also with the University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: zhouyaohua@nimte.ac.cn).

Chin-Yin Chen and Guilin Yang are with the Zhejiang Key Laboratory of Robotics and Intelligent Manufacturing Equipment Technology, Ningbo Institute of Materials Technology and Engineering, Chinese Academy of Sciences, Ningbo 315201, China (e-mail: chenchinyin@nimte.ac.cn; glyang@nimte.ac.cn).

Yaonan Li is with the Shenzhen Academy of Robotics, Shenzhen 518000, China (e-mail: ynli@szarobots.com).

Digital Object Identifier 10.1109/LRA.2022.3204867

and constraint analysis of redundant robots, the methods based on quadratic programming (QP) may be more effective. Due to the high efficiency of parallel computing, recurrent neural networks are usually used to solve QP-based redundancy solutions, e.g., joint velocity optimization with model uncertainties [11], joint torque optimization with position-force constraints [12] and flexible obstacle avoidance [13]. However, convexification transformations are sometimes inevitable. Besides, the sampling-based methods [14], [15] are most frequently applied to mobile robots (or mobile manipulators), and also suitable for those industrial occasions based on offline programming [5]. The reasons are that these methods can effectively avoid falling into local optima and handle multiple constraints [14]. Therefore, in this work, a sampling-based method is adopted to solve the motion planning problem of MMRS.

In addition, researchers are gradually willing to consider multiple performance indices to improve the working conditions of redundant robots. This process is often performed by introducing a multi-objective cost function (MOCF). Cho *et al.* [16] summarized normalization-based formulation techniques for MOCFs and discussed the necessity and complexity of trade-off analyses. In the aspect of robotic motion planning, it is a more general way to use a weighted sum (directly or indirectly) of performance indices [17], [18], [19]. For normalization, Pardi *et al.* [17] unified the time dimension and Picard *et al.* [18] employed exponential and hyperbolic sine functions. Although each performance index can be adjusted to a roughly similar level by mapping, the normalization problem of units has not been well solved.

B. Overview

Based on the foregoing literature reviews, the MA problem of MMRS should consist of three aspects: integration of multiple performance indices, avoidance of local optima and effective treatment of constraints. Therefore, two kinds of performance index maps are introduced in this letter to achieve the normalization and integration of multiple performance indices. Meanwhile, a corresponding sampling-based MA strategy is proposed, which can avoid falling into local optima and satisfy given constraints. The main contributions of this work are listed as follows: i) By analyzing the kinematic characteristics of MMRS, the dimension of optimization variables is reduced and the MA problem is simplified. ii) Multiple trajectory constraints are unified to the displacement level, simplifying the establishment of performance index maps and the sampling process. iii) The proposed performance index integration method considers relevant distribution patterns and overcomes the incompatibility problem among units. iv) The proposed MA strategy can avoid local optima to some extent and achieve better performance enhancements at a lower time cost.

The rest of this letter is organized as follows. Section II describes the relevant configuration of MMRS. Section III elaborates on the proposed method, including the establishment of constraint space, integration of performance indices and the MA strategy. Section IV displays and discusses the simulation results. Finally, Section V presents the conclusions and future work.

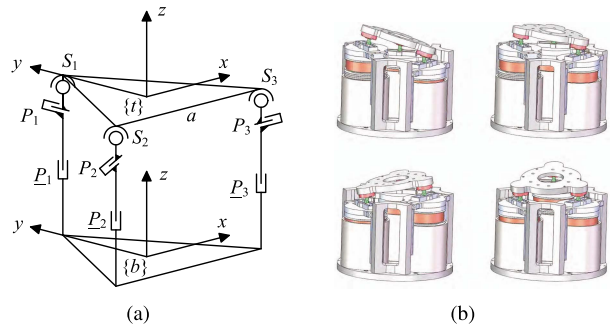


Fig. 2. Relevant configuration of the micro: (a) A mechanism diagram [5]; (b) Some displays of 2R1T motions.

II. Relevant Configuration

A. The Micro

Due to generality, the introduction of the macro is omitted here, and only the micro is presented. The micro is a three-prismatic-prismatic-spherical (3-PPS) parallel mechanism, as shown in Fig. 2(a), which consists of a base platform (with a frame $\{b\}$), three identical PPS chains (arranged symmetrically) and a moving platform (with a frame $\{t\}$). For motions, it can rotate around x-axis and y-axis (the limit is about 18°) and translate along z-axis (the limit is about 30 mm), as shown in Fig. 2(b). And this one degree-of-freedom (DOF) translation can be utilized to implement force control for polishing tasks. Besides, specific mechanism characteristics and forward/inverse kinematics can be found in [5].

B. Kinematics of MMRS

Assume that $\{B\}$ and $\{T\}$ are the base frame and tool frame of the macro respectively and $\{TCP\}$ is the polishing tool frame. Denote $\boldsymbol{q} = [\theta_1, \theta_2, \ldots, \theta_6]^{\mathrm{T}}$ and $\boldsymbol{p} = [\vartheta_1, \vartheta_2, \vartheta_3]^{\mathrm{T}}$ as joint displacements of the macro and micro respectively, and forward kinematics of MMRS can be described as

$$ {}^{B}_{TCP}\boldsymbol{T} = FK_M(\boldsymbol{q}) \cdot FK_m(\boldsymbol{p}) \cdot {}^{t}_{TCP}\boldsymbol{T}, \tag{1} $$

where $FK_M(\boldsymbol{q})$ and $FK_m(\boldsymbol{p})$ represent forward kinematics of the macro and micro respectively, and ${}^{B}_{TCP}\boldsymbol{T}$ denotes the homogeneous transformation matrix from $\{TCP\}$ to $\{B\}$.

For inverse kinematics, MMRS consists of two robotic subsystems, both of which have analytical and unique inverse kinematics (filter inverse kinematic solutions by the Pieper principle for the macro and refer to [5] for the micro). Thus, MA in Cartesian space can be directly mapped to joint space. Since the macro is a complete six-dimensional Cartesian space robot, $\boldsymbol{p}$ can be treated as an independent variable, which simplifies the MA problem. Therefore, the subsequent analyses focus on the micro.

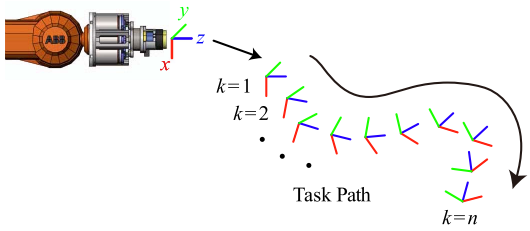


Fig. 3. The task framework of MMRS.

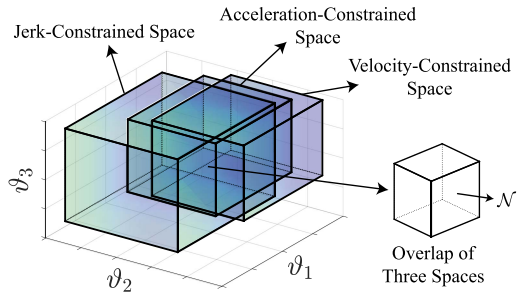


Fig. 4. A visual description of the trajectory constraints on the micro.

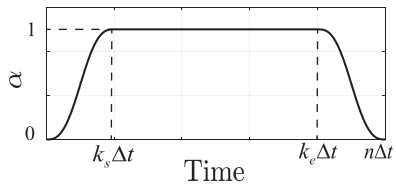


Fig. 5. A feasible variation curve for α. It can guarantee the smoothness of a trajectory at the start/end.

III. Proposed Method

Trajectory tracking in Cartesian space is a general operation mode of an industrial robotic system. As shown in Fig. 3, assume that the task path consists of a series of frames (denoted as ${}^{B}_{TCP}\boldsymbol{T}_{task,k}(k=1\sim n)$) with time nodes. The following works are introduced on this basis.

A. Constraints on Trajectories

Firstly, constraints on joint limits are considered. Let $\mathcal{L}_q$ denote the macro and $\mathcal{L}_p$ denote the micro. Then in this part, the smoothness of trajectories is mainly concerned.

Compared with the condition without MA, participation of the micro can impact the macro's trajectory. Therefore, additional constraints must be imposed on the micro to match the operating conditions of MMRS, which can be implemented by setting thresholds of velocity, acceleration and jerk. These constraints are normalized to the displacement level to adapt to the position control of industrial robots.

Recursive constraints are presented to match the subsequent MA strategy. At each time node $t=k\Delta t$ (Δt is a time step), Taylor expansions of $\boldsymbol{p}_{k+1}$, $\boldsymbol{p}_{k-1}$ and $\boldsymbol{p}_{k-2}$ can be expressed as

$$\begin{cases}\boldsymbol{p}_{k+1}=\boldsymbol{p}_k+\dot{\boldsymbol{p}}_k\Delta t+\frac{1}{2}\ddot{\boldsymbol{p}}_k\Delta t^2+\frac{1}{6}\dddot{\boldsymbol{p}}_k\Delta t^3+O\left(\Delta t^4\right)\\ \boldsymbol{p}_{k-1}=\boldsymbol{p}_k-\dot{\boldsymbol{p}}_k\Delta t+\frac{1}{2}\ddot{\boldsymbol{p}}_k\Delta t^2-\frac{1}{6}\dddot{\boldsymbol{p}}_k\Delta t^3+O\left(\Delta t^4\right)\\ \boldsymbol{p}_{k-2}=\boldsymbol{p}_k-2\dot{\boldsymbol{p}}_k\Delta t+2\ddot{\boldsymbol{p}}_k\Delta t^2-\frac{4}{3}\dddot{\boldsymbol{p}}_k\Delta t^3+O\left(\Delta t^4\right)\end{cases} \tag{2}$$

The velocity, acceleration and jerk at time $t=k\Delta t$ can be deduced from (2), thus

$$\begin{cases}\dot{\boldsymbol{p}}_k=\dfrac{2\boldsymbol{p}_{k+1}+3\boldsymbol{p}_k-6\boldsymbol{p}_{k-1}+\boldsymbol{p}_{k-2}}{6\Delta t}+O\left(\Delta t^3\right)\\ \ddot{\boldsymbol{p}}_k=\dfrac{\boldsymbol{p}_{k+1}-2\boldsymbol{p}_k+\boldsymbol{p}_{k-1}}{\Delta t^2}+O\left(\Delta t^2\right)\\ \dddot{\boldsymbol{p}}_k=\dfrac{\boldsymbol{p}_{k+1}-3\boldsymbol{p}_k+3\boldsymbol{p}_{k-1}-\boldsymbol{p}_{k-2}}{\Delta t^3}+O\left(\Delta t\right)\end{cases} \tag{3}$$

Apply limits on velocity, acceleration and jerk, and remove the truncation errors. Then

$$\begin{cases}\left|\boldsymbol{p}_{k+1}+\frac{3}{2}\boldsymbol{p}_k-3\boldsymbol{p}_{k-1}+\frac{1}{2}\boldsymbol{p}_{k-2}\right|\le 3\boldsymbol{v}_{\max}\Delta t\\ \left|\boldsymbol{p}_{k+1}-2\boldsymbol{p}_k+\boldsymbol{p}_{k-1}\right|\le\boldsymbol{a}_{\max}\Delta t^2\\ \left|\boldsymbol{p}_{k+1}-3\boldsymbol{p}_k+3\boldsymbol{p}_{k-1}-\boldsymbol{p}_{k-2}\right|\le\boldsymbol{j}_{\max}\Delta t^3\end{cases} \tag{4}$$

where $\boldsymbol{v}_{\max},\boldsymbol{a}_{\max},\boldsymbol{j}_{\max}\in\mathbb{R}^{3\times1}$ represent artificially-set maximum limits on velocity, acceleration and jerk respectively. They can be set less than the rated values to match a polishing task. In this way, trajectories of the macro can also be indirectly limited. On the whole, (4) is used to obtain a value set of $\boldsymbol{p}_{k+1}$, i.e., a feasible neighborhood of $\boldsymbol{p}_k$ (denoted as $\mathcal{N}_k$). For interpretation, geometric visualization of $\mathcal{N}$ is revealed. As shown in Fig. 4, three inequalities in (4) can be regarded as three cuboids with different side lengths and centers, and their overlap is also a cuboid, i.e., the set $\mathcal{N}$.

Furthermore, two additional improvements are implemented: i) For soft start/stop, a time-varying scaling factor α is introduced to adjust the above three limits dynamically:

$$\boldsymbol{v}_{\max},\boldsymbol{a}_{\max},\boldsymbol{j}_{\max}\leftarrow\alpha\boldsymbol{v}_{\max},\alpha\boldsymbol{a}_{\max},\alpha\boldsymbol{j}_{\max} \tag{5}$$

where α (bounded by $0\sim1$) can vary based on an S-shaped velocity curve shown in Fig. 5, where $k_s\Delta t$ and $k_e\Delta t$ are two turning points. For the existence of α, the trajectories can be smoother. ii) It can be predicted that the micro may repeatedly approach or leave the joint limits during the task. To guarantee the continuity of jerk and produce soft operation effects near the joint limits, seven-degree polynomials are employed to smooth these turning points after trajectories are generated.

B. Integration of Performance Indices

In a manufacturing process, multiple performance indices may need to be improved. For this, a normalization method is proposed for integrating various performance indices, which is depend on establishing two kinds of performance index maps. It is noteworthy that in this work, performance indices of the macro are primarily concerned since the macro exerts the predominant function and the dexterity of the micro needs to be leveraged.

1) Constraint-Based Performance Index Map: At time $t=k\Delta t$, by sampling $\mathcal{N}_k$, a series of $\boldsymbol{q}_{k+1}$ can be thereby obtained

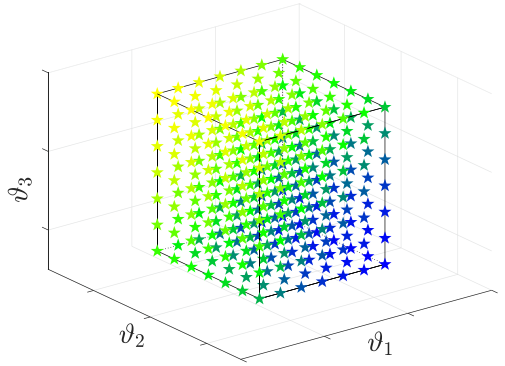


Fig. 6. A visual description of CBPIM (from small to large according to the performance index: blue, green, yellow). The black frame is the set $\mathcal{N}$.

through inverse kinematics.[1] In this way, for any performance index (denoted as $PI(\boldsymbol{q})$), its constraint-based performance index map (CBPIM) can be built, e.g., as shown in Fig. 6. By sorting these sampling points (from small to large according to the performance index), a sequence can be obtained. Let γ denote the sequence number of each sampling point and $PI_c(\gamma)$ denote the sorted performance index function, and the relative ranking of a performance index can be expressed as

$$RPI_c(\boldsymbol{q}) = \frac{1}{N_c} \arg\min_{\gamma} \left| PI(\boldsymbol{q}) - PI_c(\gamma) \right|, \tag{6}$$

where N_c denotes the number of sampling points and $RPI_c(\boldsymbol{q}) \in (0, 1]$ denotes the relative ranking. It is worth noting that (6) obtains normalized variables concerning the performance index distribution, instead of the maximum or minimum [14]. Therefore, it can be applied to various performance distribution patterns.

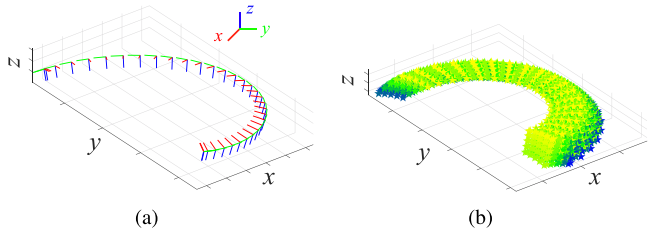
Denote $PI_i(\boldsymbol{q})(i = 1 \sim l)$ as the required performance indices, thus $RPI_{c,i}(\boldsymbol{q})$ can be obtained by the above-mentioned method. After normalizing these performance indices, an MOCF $f_c(\boldsymbol{q})$ can be reasonably constructed using their weighted sum:

$$f_c(\boldsymbol{q}) = \sum_{i=1}^{l} \left(w_i \cdot RPI_{c,i}(\boldsymbol{q}) \right), \tag{7}$$

where w_i is a weight coefficient satisfying $\sum_{i=1}^{l} w_i = 1$, as thus $f_c(\boldsymbol{q}) \in (0, 1]$. Although w_i needs to be artificially set to tell the significance, the incompatibility problem among units is effectively addressed.

2) Task-Based Performance Index Map: To evaluate the overall performance of a trajectory, global performance index distribution must be considered. The performance distribution near the task path is focused in this work, rather than the whole workspace, to exclude other irrelevant positions. The workflow of constructing a task-based performance index map (TBPIM) is as follows: i) Sample the joint space of the micro. ii) Solve joint trajectories of the macro based on those sampling points

[1]In this work, we only take joint displacements of the macro as an example since the performance indices to be optimized in this letter are only related to them. However, when using backward difference formulas, the proposed method is also applicable for performance indices related to joint velocity or acceleration.



(a) (b)

Fig. 7. (a) An assumed task path. (b) A visual description of TBPIM. These sampling points describe the positions of $\{T\}$ (from small to large according to the performance index: blue, green, yellow).

respectively in the absence of MA, which means that the micro keeps stationary and the task is completed by the macro alone. iii) Obtain a distribution map for each performance index. An example is given in Fig. 7(a) and (b), which display an assumed task path and the corresponding TBPIM, respectively.

Similar to (6), by sorting TBPIM (from small to large), a task-based global relative ranking of each performance index (denoted as $RPI_{t,i}(\boldsymbol{q})$) can also be obtained. Define a global cost function as

$$\mathcal{F} = \frac{1}{n\Delta t} \int_0^{n\Delta t} f_t(\boldsymbol{q}) \, \mathrm{d}t, \tag{8}$$

where $f_t(\boldsymbol{q}) = \sum_{i=1}^{l} (w_i \cdot RPI_{t,i}(\boldsymbol{q}))$ represents the integration of the global relative rankings of performance indices. Thus, $\mathcal{F}$ (also in the range of $(0, 1]$) can be regarded as an evaluation of the average relative ranking in TBPIM of a task path. It is the main index used for comparison in the subsequent simulations (Section IV).

C. Selection of Performance Indices

In this part, the following three specific performance indices are considered for the improvement of polishing tasks: *1)* general compliance index, *2)* normal compliance index and *3)* manipulability index.

1) General Compliance Index: The compliance matrix in Cartesian space of the macro can be described as [20]

$$\boldsymbol{C}_r(\boldsymbol{q}) = \boldsymbol{J}_M(\boldsymbol{q}) \boldsymbol{K}_\theta^{-1} \boldsymbol{J}_M(\boldsymbol{q})^{\mathrm{T}}, \tag{9}$$

where $\boldsymbol{J}_M(\boldsymbol{q}) \in \mathbb{R}^{6\times6}$ is the body Jacobian matrix and $\boldsymbol{K}_\theta \in \mathbb{R}^{6\times6}$ is a diagonal matrix characterizing joint stiffness. With (9), the general compliance index can be described as

$$\mathcal{C}_g(\boldsymbol{q}) = \det(\boldsymbol{C}_r(\boldsymbol{q})). \tag{10}$$

(10) describes the overall compliance capacity in Cartesian space. For polishing tasks, operating in a pose with a low general compliance index helps to alleviate deformation and chattering of the macro, reducing its impact on the micro.

2) Normal Compliance Index: The normal compliance index characterizes the compliance of a robot along z-axis of its tool frame (the frame $\{T\}$ in this work) [20], which can be calculated by the following formula [21]:

$$\mathcal{C}_z(\boldsymbol{q}) = \frac{1}{\sqrt{\boldsymbol{r}_z^{\mathrm{T}} \left(\boldsymbol{C}_r(\boldsymbol{q})^{\mathrm{T}} \boldsymbol{C}_r(\boldsymbol{q}) \right)^{-1} \boldsymbol{r}_z}}, \tag{11}$$

where $\boldsymbol{r}_z = [0,0,0,0,0,1]^{\mathrm{T}}$ represents the unit translation along z-axis. The geometric interpretation of $\mathcal{C}_z(\boldsymbol{q})$ is the projection length of the compliance ellipsoid on z-axis [21]. We take $\mathcal{C}_z(\boldsymbol{q})$ into consideration since it is always in the force control direction of the micro, and a low normal compliance index can lead to the stability of force control.

3) Manipulability Index: The manipulability index also needs to be considered since polishing is a dynamic process with continuous trajectories, which differs from drilling. High manipulability can alleviate singularity problems to some extent. Here the manipulability index proposed by Yoshikawa [22] is adopted:

$$\mathcal{W}(\boldsymbol{q}) = \sqrt{\det(\boldsymbol{J}_M(\boldsymbol{q})\boldsymbol{J}_M(\boldsymbol{q})^{\mathrm{T}})}. \tag{12}$$

4) Remarks and Notations: Since i) lower compliance leads to a more singular configuration [23] (which means that $\mathcal{C}_g$ is completely contradictory to $\mathcal{W}$); ii) polishing quality predominantly depends on force control stability of the micro, a compromise is adopted here, i.e., inserting the manipulability index $\mathcal{W}$ into inequality constraints to avoid singularity to some extent (define the threshold of the manipulability index as $\mathcal{W}_t$). Thus denote $\mathcal{C}_g$ as PI_1 and $\mathcal{C}_z$ as PI_2. In addition, considering that $\mathcal{C}_g$ and $\mathcal{C}_z$ possess different units (the former combines "m/N" and "rad/N · m," while the latter is only "m/N"), the proposed integration method of performance indices should be effective when embedding them into an MOCF.

D. MA Strategy

In this part, a sampling-based method is proposed to address the MA problem of MMRS.

At time step k, combining constraints on the task path, joint limits and manipulability index, the following constraint optimization problem is considered:

$$\begin{aligned} &\underset{\boldsymbol{p}}{\text{minimize}}\ f_c(\boldsymbol{q}) \\ &s.t. \begin{cases} {}^{B}_{TCP}\boldsymbol{T}_{task,k+1} = FK_M(\boldsymbol{q}) \cdot FK_m(\boldsymbol{p}) \cdot {}^{t}_{TCP}\boldsymbol{T} \\ \boldsymbol{q} \in \mathcal{L}_q \ \ \boldsymbol{p} \in \mathcal{L}_p \cap \mathcal{N}_k \\ \mathcal{W}(\boldsymbol{q}) \geq \mathcal{W}_t \end{cases} \end{aligned} \tag{13}$$

It is noteworthy that (13) is a three-dimensional problem with a single independent variable $\boldsymbol{p}$, for once the joint displacements of the micro are obtained, the corresponding ones of the macro can be uniquely solved.

Those gradient-based iterative methods may not be convenient to address this time-varying optimization problem, which is because: i) It is prone to fall into local optima since the problem (13) is nonconvex. ii) $f_c(\boldsymbol{q})$ is not an explicit analytic function for $\boldsymbol{p}$, which results in the use of approximate gradients and Hessian matrices, rather than the exact ones.

Here, a sampling-based algorithm is adopted to address the MA problem, since it is universal for all forms of performance indices and can deal with the constraints flexibly. Furthermore, the local optima problem caused by gradient-based nonlinear programming methods can be addressed [14].

Algorithm 1: MA Strategy.

Input:
Geometric parameters;
Task path ${}^{B}_{TCP}T_{task,1}, {}^{B}_{TCP}T_{task,2}, \cdots, {}^{B}_{TCP}T_{task,n}$;
Performance functions $PI_1(\boldsymbol{q}), PI_2(\boldsymbol{q}), \mathcal{W}(\boldsymbol{q})$;
Variables related to constraints $\boldsymbol{v}_{\max}, \boldsymbol{a}_{\max}, \boldsymbol{j}_{\max}$, $\alpha, \mathcal{L}_q, \mathcal{L}_p, \mathcal{W}_t$;
Output:
Joint trajectories $\boldsymbol{q}_k, \boldsymbol{p}_k$ $(k = 1 \sim n)$;

1: **for** $i = 1$ *to* 2 **do**
2: Construct and sort TBPIM for $PI_i(\boldsymbol{q})$;
3: **end for**
4: **for** $k = 3$ *to* $n-1$ **do**
5: Build $\mathcal{N}_k$ and make uniform partitions;
6: **for** $i = 1$ *to* 2 **do**
7: Construct and sort CBPIM for $PI_i(\boldsymbol{q})$;
8: **end for**
9: **for** *each point that satisfies constraints* **do**
10: Solve kinematics and compute $f_c(\boldsymbol{q})$;
11: **if** $f_c(\boldsymbol{q}) < f_{c,optimal}(\boldsymbol{q})$ **then**
12: $\boldsymbol{q}_{k+1} \leftarrow \boldsymbol{q}$;
13: $\boldsymbol{p}_{k+1} \leftarrow \boldsymbol{p}$;
14: **end if**
15: **end for**
16: **end for**
17: Return the corresponding joint trajectories and performance index curves.

Algorithm 1 shows the overall framework of the proposed MA strategy. In Algorithm 1, a task path needs to be given first by an actual manufacturing task. TBPIM needs to be constructed and sorted for each performance index (line 1-3). At time step k, $\mathcal{N}_k$ needs to be built using (4) and (5) (line 5). With $\mathcal{N}_k$, CBPIM can be constructed for each performance index and the corresponding sorted sequences can be stored (line 6-8). By traversing the whole $\mathcal{N}_k$ and eliminating all infeasible points, the optimal $\boldsymbol{q}_k$ and $\boldsymbol{p}_k$ are recorded (line 9-15). Repeat the above steps, and the joint trajectories can be generated recursively.

It can be observed that the proposed strategy reflects generality. According to the framework of Algorithm 1, we can deduce that it can also be applied to MMRS whose macro possesses the complete task degree of freedom. Meanwhile, the performance index functions can be switched flexibly to match a specific manufacturing process. Besides, the performance indices of the micro can also be attached, which can be implemented effortlessly by establishing additional CBPIM and TBPIM. Therefore, the polishing process can be viewed as a case for algorithmic analysis.

IV. Simulations and Results

For generality, three task paths are selected, including a "cosine" curve (Task A), spoke curve (Task B) and Archimedes spiral curve (Task C), as shown in Figs. 8(a)–(c). Here, the polishing angle is set to 0°. Thus the z-axes of these task frames are consistently in the normal direction of the surface.

TABLE I
SIMULATION RESULTS

Task	Task A			Task B			Task C		
Method	PM	GBIM	GA	PM	GBIM	GA	PM	GBIM	GA
Global Cost Function $\mathcal{F}$	0.1058	0.1228	0.1084	0.0916	0.1001	0.0928	0.1513	0.1660	0.1541
Improvement rate of $\mathcal{F}$ (PM vs others)	/	13.84%	2.40%	/	8.49%	1.29%	/	8.86%	1.82%
CPU time (s)	29.2	37.5	138.4	66.8	80.5	316.6	30.2	37.8	142.4

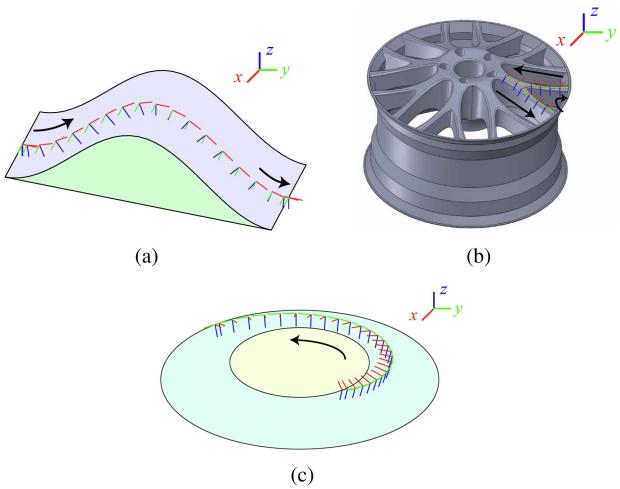


Fig. 8. Three pre-defined task paths: (a) Task A: a surface whose cross section is a "cosine" curve; (b) Task B: a spoke of a wheel hub; (c) Task C: an Archimedean spiral on the surface of a circular truncated cone.

To illustrate the effectiveness, the proposed method is compared with the other two methods (all of which are applied to the above three tasks):

- PM (the proposed method): It performed based on the framework of Algorithm 1.
- GBIM (gradient-based iterative method) [24]: It's a Quasi-Newton method using interior barrier functions, where gradients and Hessian matrices need to be approximately estimated since $f_c(\boldsymbol{q})$ is not an explicit analytic function. The convergence criterion was set to $\|\Delta \boldsymbol{p}\|_\infty \leqslant 0.004$ mm.
- GA (genetic algorithm) [25]: It ran on the MATLAB Toolbox. The population size and convergence criterion (average relative change in the best fitness value) were set to 25 and 0.005 respectively. Other parameters were selected as the default values: crossover probability 0.8, mutation probability 0.01 and migration probability 0.2.

Other main parameters and relevant descriptions are listed as follows: i) $\Delta t = 0.01$ s and $k_s \Delta t = k_e \Delta t = 1$ s. ii) Considering the smoothness, the trajectory limits were set to $\boldsymbol{v}_{\max} = [10\,,\,10\,,\,10]^{\mathrm{T}}$mm/s, $\boldsymbol{a}_{\max} = [1.5\,,\,1.5\,,\,1.5]^{\mathrm{T}} \times 10^3$ mm/s ^{2} and $\boldsymbol{j}_{\max} = [1.8\,,\,1.8\,,\,1.8]^{\mathrm{T}} \times 10^5$ mm/s^3. iii) Based on experimental results of stiffness identification for IRB 4400, the stiffness of the macro in this study is given as $\boldsymbol{K}_\theta = \mathrm{diag}\,(501130\,,\,699886\,,\,424088\,,\,15744\,,\,32234\,,\,13330)$ N·m/rad. iv) The normal compliance index needs more attention since it directly affects force control of the micro. Here $w_1 = 0.3$ and $w_2 = 0.7$. v) By observing the sorted TBPIM of manipulability index, the threshold $\mathcal{W}_t$ was set to 0.2. vi) The initial pose of the micro was set to half of the maximum displacements, i.e., $\boldsymbol{p}_1 = [15\,,\,15\,,\,15]^{\mathrm{T}}$mm.

All numerical simulations have been performed on MATLAB with a desktop computer with a CPU AMD Ryzen 5 3600 (3.60 GHz) and 8 GB RAM.

Table I shows the results of the global cost function $\mathcal{F}$ and CPU time in three task scenarios. Here, $\mathcal{F}$ denotes the average relative ranking of the compliance performance of a trajectory (see (8)). Smaller $\mathcal{F}$ means higher ranking, i.e., a better performance. The mathematical expression of the improvement rate is $(\mathcal{F}_{\mathrm{others}} - \mathcal{F}_{\mathrm{PM}})/\mathcal{F}_{\mathrm{others}}$ ("others" means GBIM or GA). Thus, a positive improvement rate indicates that PM is in a superior position.

From Table I, it can be found that whether for Task A, B or C, PM is superior to GBIM and slightly superior to GA (see the improvement rates of $\mathcal{F}$). This result indicates that when PM is applied, the macro can maintain lower compliance (higher stiffness) to accomplish the same manufacturing task. The main reason for the difference is that the objective function f_c in (13) has multiple local minima due to non-convexity. This is especially true for a 9-DOF redundant robot (the 6-DOF macro and 3-DOF micro) with a complex kinematic model. As is known to all, GBIM is prone to fall into local minima, while GA alleviates this defect to some extent due to its mutation, migration and other behaviors. However, PM is superior to them due to the relative completeness of sampling. The details are analyzed in the subsequent curves.

In terms of CPU time, PM is superior to GBIM and GA under the conditions of those preset parameters. CBPIMs were established consistently in these three methods to implement the normalization of performance indices. For PM, CBPIM merely needs to be judged and compared in each time step. GBIM requires the estimation of gradients and Hessian matrices, whose imprecision affects the iteration efficiency. In addition, additional barrier functions need to be established to handle the inequality constraints on joint displacements and manipulability. GA takes a long time to perform a series of genetic behaviors. Their diverse calculation principles lead to the different time consumption results in Table I. Furthermore, it can be predicted that although the performances of GBIM and GA can be improved by adjusting parameters (such as changing the convergence criteria), they accordingly cost more CPU time.

Next, the variation curves are presented. For analysis, Task C is taken as an example because its overall performance is similar to Task A and B. Figs. 9(a)–(c) and (d)–(f) give the

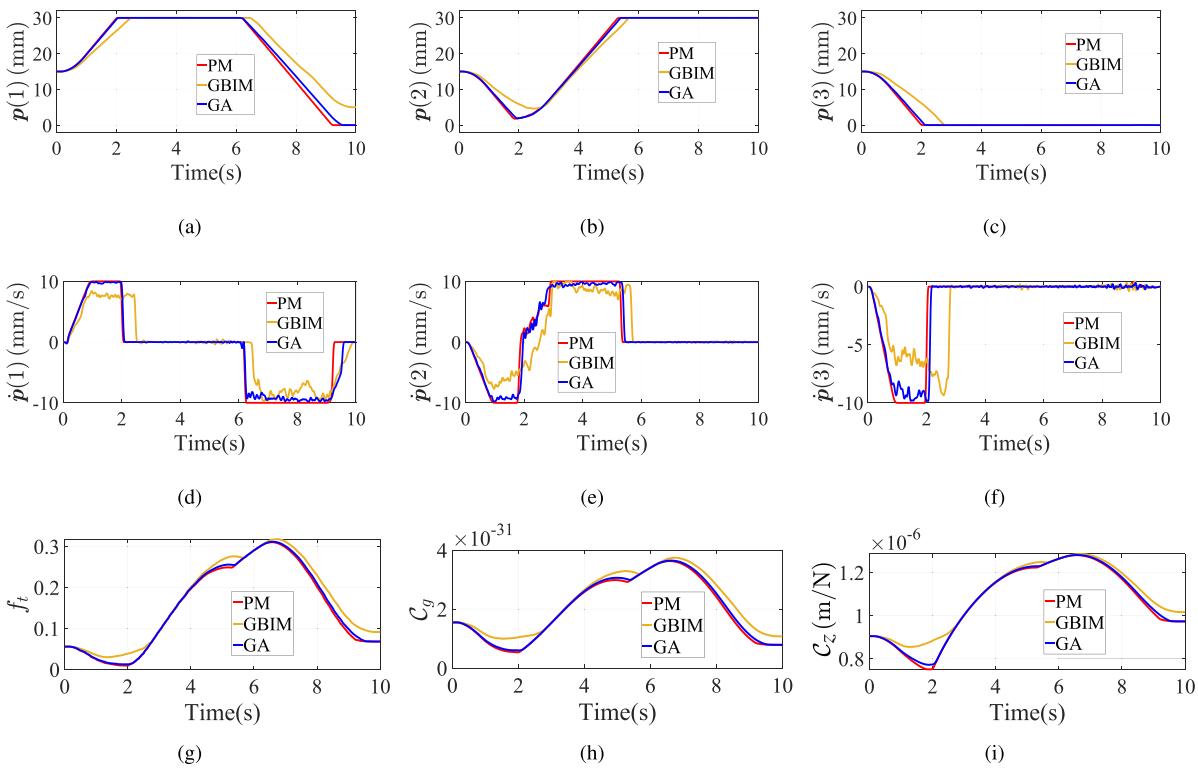


Fig. 9. Specific simulation results of Task C: (a)–(c) Displacement curves of three active joints of the micro. (d)–(f) velocity curves of three active joints of the micro. (g)–(i) Results of f_t, $\mathcal{C}_g$, and $\mathcal{C}_z$.

displacement and velocity curves of the three active joints of the micro, respectively. These three methods can satisfy the joint limit constraint of 0 to 30 mm and velocity constraint of 10 mm/s. The variation curves of f_t, $\mathcal{C}_g$ and $\mathcal{C}_z$ are shown in Fig. 9(g)–(i). Smaller f_t means higher ranking, i.e., lower compliance performance, which is consistent with the results in Table I ($\mathcal{F}$ signifies the average level of f_t). In addition, since f_t is the integration of general compliance index $\mathcal{C}_g$ and normal compliance index $\mathcal{C}_z$, their variation trends are almost the same.

From joint displacement curves (Fig. 9(a)–(c)), it can be found that PM is ahead of GA and GBIM (more evident for the latter), which is consistent with the velocity curves. This is the result of local convergence characteristics, which leads to the occurrence of two phenomena in Fig. 9: i) Observe the following three time periods: from about 1 s to 2.5 s corresponding to joint 1, 2 and 3, from about 4.5 s to 5.5 s corresponding to joint 2 and from about 6 s to 9.5 s corresponding to joint 1. When the joint limits are not reached, GA and GBIM cannot encourage the three active joints to move at the maximum velocity of 10 mm/s. However, from the perspective of results, running at the maximum velocity (i.e., PM) can generally obtain lower compliance performance. ii) There are some fluctuations in joint velocity curves when applying GBIM or GA. This is because each time step corresponds to different initial conditions and task constraints, resulting in different local convergence characteristics, which are manifested by various displacement increments, i.e., velocity fluctuations. However, due to the relative completeness of sampling, PM usually maintains the maximum velocity, so the fluctuations are slighter.

There are also some time periods during the task execution process where the compliance performances of these three methods have little difference (see Fig. 9(g)). From about 2.8 s to 4 s, both joint 1 and 3 reached the joint limits and stopped, and the velocities of joint 2 were almost the same. Similarly, from about 5.8 s to 6.2 s, all three joints stopped as they reached the joint limits. These phenomena caused by joint limit constraints are the sources of the weak differences. However, it can be predicted that the advantages of PM would become more pronounced if the joint limits were relaxed, because the proportion of time spent within the joint limits would increase, corresponding to the difference in compliance performance remaining longer.

In addition to the algorithmic perspective, further improvement of the trajectory performance from a task perspective is concerned next. The effect of the initial pose on the whole trajectory is worth discussing since polishing is a continuous manufacturing process, which is different from drilling. Taking Task C as an example, PM was applied to six different initial poses (Case <1> to <6>) to obtain their curves of f_t, as shown in Fig. 10. It can be observed that PM has convergence characteristics for different initial poses (as do joint displacements, although not given here). This is because in each iteration, it finds the optimum through sampling and traversing, just as those gradient-based methods, which employ negative gradient directions. And the latter has convergence

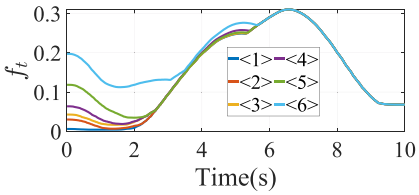


Fig. 10. Descriptions of convergence: Case <1> to <6> have different initial poses.

characteristics [26]. Furthermore, it can be concluded that the initial pose with better performance tends to converge earlier and leads to better overall performance. Therefore, in actual manufacturing tasks, it is recommended to select a better initial pose to avoid spending a lot of time applying the MA strategy to all feasible ones.

V. Conclusions and Future Work

This letter proposes an MA strategy for MMRS. It is used to achieve performance enhancements while solving joint trajectories. Feasible sets is obtained by establishing the micro's trajectory constraints. CBPIM and TBPIM are introduced to achieve normalization and integration of multiple performance indices, which avoids incompatibility among units. The latter is also used to evaluate the overall performance of a trajectory. MA is implemented by traversing the feasible set and operating iteratively. Finally, the comparative simulations prove that the proposed method can avoid local optima and achieve performance enhancements at a lower time cost.

From an algorithmic perspective, some optimal control schemes (such as pseudospectral methods) may be able to obtain further performance enhancement. This is because the trajectory is not generated iteratively based on time steps, and there is no need to consider the effect of the initial pose. However, time consumption and optimality conditions need to be considered accordingly. From a task perspective, optimizing the workpiece pose can achieve a further performance improvement [1], [2], which is equivalent to running the manipulator in a high stiffness region. These ideas will be developed in future works.

References

[1] G. Nicola, N. Pedrocchi, S. Mutti, P. Magnoni, and M. Beschi, "Optimal task positioning in multi-robot cells, using nested meta-heuristic swarm algorithms," *Procedia CIRP*, vol. 72, pp. 386–391, 2018.

[2] S. Mutti, G. Nicola, M. Beschi, N. Pedrocchi, and L. M. Tosatti, "Towards optimal task positioning in multi-robot cells, using nested meta-heuristic swarm algorithms," *Robot. Comput.- Integr. Manuf.*, vol. 71, 2021, Art. no. 102131.

[3] V. L. Nguyen, C.-H. Kuo, and P. T. Lin, "Compliance error compensation of a robot end-effector with joint stiffness uncertainties for milling: An analytical model," *Mechanism Mach. Theory*, vol. 170, 2022, Art. no. 104717.

[4] Z. Ma et al., "Control and modeling of an end-effector in a macro-mini manipulator system for industrial applications," in *Proc. IEEE Int. Conf. Adv. Intell. Mechatronics*, 2017, pp. 676–681.

[5] Y. Zhou et al., "A motion assignment strategy based on macro-micro robotic system for enhancement of kinematic performance," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, 2021, pp. 174–179.

[6] A. Colomé and C. Torras, "Redundant inverse kinematics: Experimental comparative review and two enhancements," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 5333–5340.

[7] T. Sugihara, "Solvability-unconcerned inverse kinematics by the levenberg–marquardt method," *IEEE Trans. Robot.*, vol. 27, no. 5, pp. 984–991, Oct. 2011.

[8] F. Flacco, A. De Luca, and O. Khatib, "Control of redundant robots under hard joint constraints: Saturation in the null space," *IEEE Trans. Robot.*, vol. 31, no. 3, pp. 637–654, Jun. 2015.

[9] R. Gao, "Inverse kinematics solution of robotics based on neural network algorithms," *J. Ambient Intell. Humanized Comput.*, vol. 11, no. 12, pp. 6199–6209, 2020.

[10] J. Wan et al., "A weighted gradient projection method for inverse kinematics of redundant manipulators considering multiple performance criteria," *Strojniski Vestnik/J. Mech. Eng.*, vol. 64, pp. 475–488, 2018.

[11] Z. Xu, S. Li, X. Zhou, W. Yan, T. Cheng, and D. Huang, "Dynamic neural networks based kinematic control for redundant manipulators with model uncertainties," *Neurocomputing*, vol. 329, pp. 255–266, 2019.

[12] Z. Xu, S. Li, X. Zhou, S. Zhou, T. Cheng, and Y. Guan, "Dynamic neural networks for motion-force control of redundant manipulators: An optimization perspective," *IEEE Trans. Ind. Electron.*, vol. 68, no. 2, pp. 1525–1536, Feb. 2021.

[13] Z. Xu, X. Zhou, H. Wu, X. Li, and S. Li, "Motion planning of manipulators for simultaneous obstacle avoidance and target tracking: An RNN approach with guaranteed performance," *IEEE Trans. Ind. Electron.*, vol. 69, no. 4, pp. 3887–3897, Apr. 2022.

[14] H. Zhang, Q. Sheng, Y. Sun, X. Sheng, Z. Xiong, and X. Zhu, "A novel coordinated motion planner based on capability map for autonomous mobile manipulator," *Robot. Auton. Syst.*, vol. 129, 2020, Art. no. 103554.

[15] S. Ren, Y. Xie, X. Yang, J. Xu, G. Wang, and K. Chen, "A method for optimizing the base position of mobile painting manipulators," *IEEE Trans. Automat. Sci. Eng.*, vol. 14, no. 1, pp. 370–375, Jan. 2017.

[16] J.-H. Cho, Y. Wang, R. Chen, K. S. Chan, and A. Swami, "A survey on modeling and optimizing multi-objective systems," *IEEE Commun. Surv. Tut.*, vol. 19, no. 3, pp. 1867–1901, Jul.–Sep. 2017.

[17] T. Pardi, V. Maddali, V. Ortenzi, R. Stolkin, and N. Marturi, "Path planning for mobile manipulator robots under non-holonomic and task constraints," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 6749–6756.

[18] G. Picard, R. Lenain, Y. Mezouar, B. Thuilot, and J. Laneurit, "Multi-trajectory approach for a generic coordination paradigm of wheeled mobile manipulators," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 2329–2336, Apr. 2022.

[19] M. Kang, H. Shin, D. Kim, and S.-E. Yoon, "Torm: Fast and accurate trajectory optimization of redundant manipulator given an end-effector path," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 9417–9424.

[20] R. Pan et al., "Research on an evaluation model for the working stiffness of a robot-assisted bonnet polishing system," *J. Manuf. Processes*, vol. 65, pp. 134–143, 2021.

[21] J. Jiao, W. Tian, W. Liao, L. Zhang, and Y. Bu, "Processing configuration off-line optimization for functionally redundant robotic drilling tasks," *Robot. Auton. Syst.*, vol. 110, pp. 112–123, 2018.

[22] T. Yoshikawa, "Manipulability of robotic mechanisms," *Int. J. Robot. Res.*, vol. 4, no. 2, pp. 3–9, 1985.

[23] G. Li, W. Zhu, H. Dong, and Y. Ke, "Stiffness-oriented performance indices defined on two-dimensional manifold for 6-dof industrial robot," *Robot. Comput.- Integr. Manuf.*, vol. 68, 2021, Art. no. 102076.

[24] L. Lu et al., "Joint-smooth toolpath planning by optimized differential vector for robot surface machining considering the tool orientation constraints," *IEEE/ASME Trans. Mechatronics*, vol. 27, no. 4, pp. 2301–2311, Aug. 2022.

[25] F. Liang, G. Yan, and F. Fang, "Global time-optimal b-spline feedrate scheduling for a two-turret multi-axis nc machine tool based on optimization with genetic algorithm," *Robot. Comput.- Integr. Manuf.*, vol. 75, 2022, Art. no. 102308.

[26] J. Xiang, C. Zhong, and W. Wei, "General-weighted least-norm control for redundant manipulators," *IEEE Trans. Robot.*, vol. 26, no. 4, pp. 660–669, Aug. 2010.