

Optimal Task-Space Tracking with Minimum Manipulator Reconfiguration

Tong Yang¹ *Student Member, IEEE*, Jaime Valls Miro² *Member, IEEE*,
 Yue Wang^{1*} *Member, IEEE* and Rong Xiong¹ *Member, IEEE*

Abstract—An optimal solution to the task-space tracking problem using a non-redundant manipulator is proposed. This is a recurring occurrence in automated manufacturing settings, e.g. welding, deburring, painting, or quality control inspections. Given a pre-defined path for the end-effector to follow, there may not exist a joint-space continuous solution for task-space tracking when the non-linear manipulator kinematics and collision avoidance with obstacles in the workcell are considered. This introduces undesirable manipulator reconfigurations where the end-effector is required to deviate temporarily from the pre-defined path. The unwanted motion results in pausing task-space tracking, often incurring not only ineffective time and energy demands but potentially compromising the quality of the task at hand due to the additional discontinuities. An algorithm is proposed that provides a globally optimal perspective to the choice of suitable joint-space connected segments so that the minimum number of manipulator reconfigurations during task-space tracking is guaranteed. By carefully selecting the inverse kinematic solutions, all sequences ensuring minimum reconfigurability are proven collected by Dynamic Programming. Moreover, a faster greedy strategy is suggested to increase the computational efficiency of the tracker whilst still preserving global optimality and completeness. The effectiveness of the proposed algorithm is validated against traditional sampling-based solvers in simulation and illustrated on challenging real-world tracking experimentation with a Universal Robotics manipulator and a curved-surface object, depicted also in an accompanying video. An open-source implementation has also been provided for the benefit of the robotics community.

I. INTRODUCTION

The task-space non-revisiting tracking problem (TNTP) of a manipulator [1] is a fundamental module that appears often in real-world applications, e.g. welding [2], inspection [3] [4] or other general maintenance duties [4]. In these applications, the task-space path for a manipulator equipped with the appropriate tool is generally pre-defined [5], and the problem is to generate a valid manipulator joint-space trajectory ensuring the end-effector (EE) visits each point on the task-space path in order, and exactly one time [6]. However, a continuous singularity-free joint-space trajectory may be impeded by the manipulator kinematic constraints, and the motion made even more restricted by the obstacles present in the environment. As such the manipulator may be forced

¹ Tong Yang, Yue Wang, and Rong Xiong are with the State Key Laboratory of Industrial Control and Technology, Zhejiang University, P.R. China.

² Jaime Valls Miro is with the Robotis Institute at the University of Technology Sydney (UTS:RI), Sydney, Australia.

This work was partially supported by National Key R&D Program of China (2018AAA0102700), and in part by the Science and Technology Project of Zhejiang Province (Grant No.2019C01043).

* Corresponding author: rxiong@zju.edu.cn

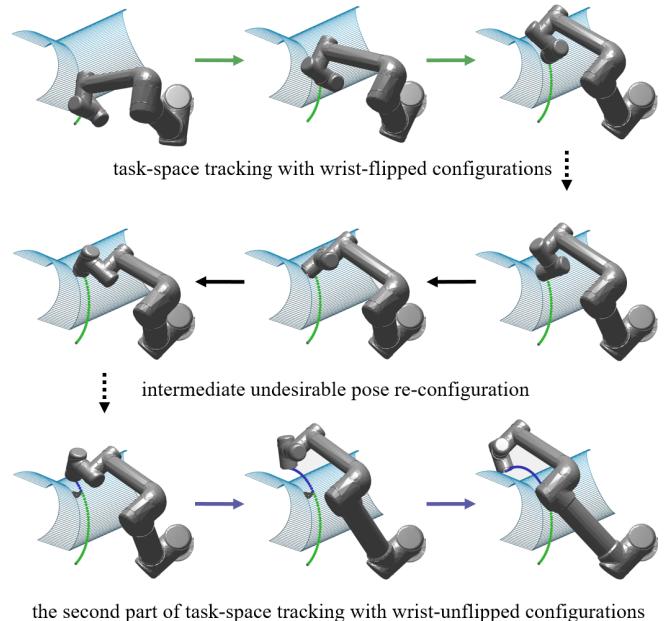


Fig. 1. Illustration of a welding task using a non-redundant manipulator. The intersecting curve of two cylinders, as well as the EE pose during the task is assumed well-defined. At the end of the first segment of continuous tracking (depicted in green), the wrist would hit the forearm if the manipulator moved further along the path. thus, in order to continue tracking the desired path, the manipulator must adopt a different configuration pose, in this case the wrist-unflipped pose is shown for illustration.

to undertake joint-space reconfigurations resulting in the EE departing from the pre-assigned path. The reconfiguration detour often induces not only unwelcome time and energy costs from an efficiency standpoint, but also imposes undesirable discontinuities that may have severe effects in the end results of the task at hand, e.g. during welding or painting. The problem is further compounded by factors such as (a) clutter in the operating envelope, (b) complex traversal paths, and (c) lack of redundancy in the manipulator’s kinematic chain. In all these common eventualities, joint-space motion is more likely truncated, thus rarely producing fully continuous joint-space trajectories for a pretended task-space path. More critically perhaps, there is no known mechanism to guarantee path lift-off optimality during the planning phase.

An illustration of the TNTP problem applied to a figurative welding task is depicted in Fig. 1. Let the task-space path be the intersected curve between the two cylinders. To trace the path section closer to the manipulator base (EE path shown in green), the manipulator adopts the wrist-flipped

configurations. However, there is limited reachability for the manipulator under this configuration in the far side of the path (EE tracking shown in blue in the bottom row, traversed later), as such the manipulator must readjust its pose and choose to adopt a wrist-unflipped configuration. During the rearrangement, an undesirable motion of the EE path (shown in black) is introduced, together with the inevitable suspension of the path tracking at the point of take-off and re-landing. Finally, the manipulator is able to finish the tracking of the desired path with one pose reconfiguration (the latter part of EE tracking is depicted in blue).

A key observation in this setting is the natural discontinuity between different inverse kinematics (IK) solutions of the same EE pose when discarding singular configurations: between an elbow-up configuration and an elbow-down configuration there must be a transition through a singular elbow-straight configuration, the neighbourhood of which is generally disregarded during planning [7], particularly in industrial settings given a manipulator's controller ability to deal effectively with perturbations along singular dimensions [8]. As such, only the pairwise connectedness of valid IK solutions for consecutive EE waypoints needs consideration, and the problem of minimising the number of manipulator reconfigurations can be transformed into an efficient model to select connected pairs of IK solutions (as will be described in further detail in Section III.)

In this paper, we discuss the optimal solution to the TNTP of non-redundant manipulators. Optimality in this context translates to the minimum number of EE deviations from the pre-defined path. The problem constitutes a generalisation of the classic task-space tracking problem seeking a joint-space continuous trajectory, essentially a zero-reconfiguration solution to the TNTP, if it exists. Whilst for the cases where no zero-reconfiguration solution prevails, the assignment can be regarded as “failed” by existing (sampling) algorithms. However, the outcome via the proposed TNTP optimal scheme can now guarantee a solvable min-reconfiguration solution regardless. The contributions of this paper can be summarised as:

- 1) A generalisation of the manipulator task-space tracking task into a TNTP with minimum reconfigurations, which is shown solvable.
- 2) A computationally tractable solver.
- 3) The open-sourcing¹ of the algorithm.

The remainder of this paper is organised as follows². Section II reviews and contextualised the problem within the existing literature. Section III shows that the optimal non-redundant TNTP solution can be transformed into an optimal assignment of joint-space IK solutions. Section IV proposes a solver of the non-redundant TNTP, including a detailed algorithmic diagram, guaranteeing completeness and the uncovering of all the optimal solutions. Experimental

¹An opensource implementation is provided here: https://github.com/ZJUTongYang/min_reconfig_taskspace_tracking

²A video illustrating the concepts hereby described can be found here: https://youtu.be/HHqGBk9_3x8

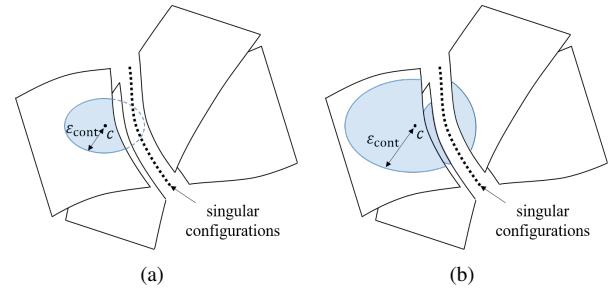


Fig. 2. Demonstration of the joint-space structure near singularities. The adoption of a manipulability measure implicitly forms a gap between disjointed sets of non-singular configurations. (a) A suitable choice of joint-space connectivity measure $\varepsilon_{\text{cont}}$ will provide correct judgment of connectedness between configurations (b) An unsuitable choice of $\varepsilon_{\text{cont}}$, where configurations from different sets might be mistakenly regarded as continuous.

results from simulations and on an actual non-redundant manipulator are collected in Section V, with final concluding remarks gathered in Section VI.

II. RELATED WORK

The task-space tracking problem of a robot manipulator is an established research topic in the literature. Given a task-space path, the valid non-singular manipulator inverse kinematic (IK) solutions [9] have been observed to form constrained subsets of the manipulator joint-space. When the manipulator transits between disjointed sets, the joint-space trajectory connecting configurations from disjointed sets will transition through at a singular state, in all likelihood making the end-effector veer off the pre-defined path, thus incurring an undesirable *pose reconfiguration*.

Existing algorithms focus on a local trajectory generation for task-space tracking, i.e., given the manipulator's current configuration with the EE docked on the pre-defined path at a given orientation, generate a control strategy for each manipulator joint such that the EE moves along the path. Redundancy of the manipulator has generally been assumed [10] so that there exists an ample set of joint-space continuous configurations for continuous task-space tracking, among which valid singular configurations also exist to bridge non-singular configurations [11]. Since the algorithms only need to locally generate an instant motion, they are more effectively referred to as “task-space controllers” [12]. When no local continuous tracking motion can be found, a joint-space path planner will be adopted to find an unrestricted motion towards a configuration ensuring the EE visiting the next task-space point.

However, the structure of valid IK configurations becomes further scattered when the manipulator is non-redundant [13], with obstacles in the environment compounding this condition [14], since each connected set of valid IK solutions has only the same dimension as its task-space preimage (e.g., a curve in 6D space as is the case here). Furthermore, no valid singularity exists for possible utilisation: a non-redundant manipulator in a singular state means it has lost the ability to dispense with perturbations imposed on the EE. As a result, a relatively sizeable task-space path is unlikely to

TABLE I
LIST OF NOTATIONS

Symbol	Meaning
N	number of task-space waypoints
γ	task-space path
$c(i, a_i)$	the a_i -th configuration to visit pose γ
$D(i, a_i)$	the valid robot state corresponding to $c(i, a_i)$
$S(j_1, \dots, j_N)$	a sequence of configurations for all task-space poses
$g(S(\dots))$	the number of reconfigurations in the given sequence

find a continuous joint-space trajectory via traditional task-space controllers, and the manipulator will have to resort to adopting frequent pose reconfigurations during tracking.

More recent existing works [15] [16] have proposed cellular decomposition and graph theory in solving the optimal surface coverage task, also explicitly considering the joint-space connectivity of IK solutions to visit different task-space points. The nature of the surface cell data makes these proposed solvers exponential. In contrast, the work proposed in this manuscript focuses on the more persistent tracking problem, where the computational efficiency for possible real-time applications needs to be carefully considered. Given the reduced dimensionality of the problem, a dynamic programming (DP) approach with an efficient greedy strategy can thus be adapted to reveal a globally optimal and complete solution.

III. PROBLEM MODELLING

In this section, the non-redundant manipulator TNTP is modelled. The commonly utilised “planning-controlling” scheme is adopted to tackle the problem. During the planning stage, and subject to a given task-space path, a joint-space collision-free path is constructed, where consecutive configurations are “close” enough such that their connectedness can be guaranteed. Subsequently, during the control stage, joint-space waypoints are interpolated and assigned suitable reaching times and joint velocities. The focus of this work is on the planning stage, where the manipulator IK relation is not one-to-one but one-to-many [9]. As such, selecting one from all valid IK solutions for each task-space waypoint becomes the fundamental problem. And the objective is to do it in a way that reconfigurations are minimised, and guaranteed at the planning stage. The discussion about control is assumed and outside the scope of the work. A table of notations is provided in Table I.

A. Task-Space Discretisation and Valid Manipulator States

Let the non-redundant TNTP problem be discussed in the ordinary 3D tracking scenario, where the manipulator is 6 DoF. The task-space path to be tracked is parameterised by a function curve $\gamma(t), t \in [0, 1]$ where each point $\gamma(t)$ is a 3D pose of the manipulator EE. If there are multiple intermediate stations for the EE to visit in order, we use a cubic interpolation to generate a continuous curve. Then, a suitable number (say N) of points are sampled from the curve, and we get a series of 3D EE poses $\gamma_i = \gamma(\frac{i-1}{N-1}), 1 \leq i \leq N$. For each point γ_i , knowing the manipulator kinematics, the manipulability constraints [17], and the obstacles in

the surrounding environment, a set of valid IK solutions are collected, denoted by $\{c(i, a_i)\}_{1 \leq a_i \leq A_i}$ where A_i denotes the number of valid IK solutions for visiting point γ_i . The manipulability constraint filters all singularities, which also excludes a neighbouring set of nonsingular configurations of the singularities and thus forms a non-zero width gap between disjointed sets of non-singular configurations. See Fig. 2 for an illustration. If two non-singular configurations are joint-space connectable only by visiting singularities, as long as we set a small distance measure ϵ_{cont} as shown in Fig. 2(a), they will not be mistakenly regarded as connectable configurations. However, note that the width of the gap is implicit, so there does not exist an explicit formula for ϵ_{cont} . How to choose a suitable ϵ_{cont} is to be presented in the next paragraph.

For each valid manipulator configuration $c(i, a_i)$ we assign a state $D(i, a_i)$ as follows: For the first EE pose γ_1 , let $\{D(1, a_1)\}_{1 \leq a_1 \leq A_1}$ be distinct numbers, such as $1, \dots, A_1$. For the i -th ($2 \leq i \leq N$) EE pose, we match $\{c(i, a_i)\}_{1 \leq a_i \leq A_i}$ to the IK solutions of previous point $\{c(i-1, a_{i-1})\}_{1 \leq a_{i-1} \leq A_{i-1}}$ whose state $\{D(i-1, a_{i-1})\}_{1 \leq a_{i-1} \leq A_{i-1}}$ have been assigned.

- 1) If $c(i, a_i)$ is joint-space continuous to a previous configuration $c(i-1, a_{i-1})$, then $D(i, a_i) = D(i-1, a_{i-1})$.
- 2) If $c(i, a_i)$ is not connectable to any configuration in $\{c(i-1, a_{i-1})\}_{1 \leq a_{i-1} \leq A_{i-1}}$, $D(i, a_i)$ is assigned a new state different from all existing states.
- 3) If $c(i, a_i)$ is connectable to multiple IK solutions of the previous point, this indicates that different IK solutions of the same task-space pose are connectable without visiting singularities. This is a wrong judgment because of inappropriate waypoints selection and a too coarse joint-space connectivity measure. See Fig. 2(b) for an illustration. Then the task-space should be resampled with a higher resolution, the connectivity measure should be set smaller, and all above-mentioned processes need to be re-done.

B. The Task-Space Non-Revisiting Problem (TNTP)

After all configurations are assigned a state, each task-space point has a set of possible states, $D(i) \triangleq \{D(i, a_i)\}_{1 \leq a_i \leq A_i}$. Note that here we have assumed the non-nullity of $D(i)$, as if a task-space point is unreachable by any manipulator configuration, the TNTP problem is essentially divided into two sub-TNTP problems with the truncated task-space path segments. So the TNTP problem is finally transformed to the optimal design of state for each task-space point. A valid joint-space trajectory is a sequence of joint-space configurations, which can be represented by a vector of states,

$$S(j_1, j_2, \dots, j_N) \triangleq [D(1, j_1), \dots, D(N, j_N)], 1 \leq j_i \leq A_i \quad (1)$$

and the cost is defined by the number of switches of states,

$$g(S(j_1, \dots, j_N)) \triangleq \sum_{i=1}^{N-1} \#\{D(i, j_i) \neq D(i+1, j_{i+1})\} \quad (2)$$

The optimal solution to the TNTP problem is to find all admissible sequences of configurations such that the cost is

the globally minimum,

$$\begin{aligned} S^*(j_1, j_2, \dots, j_N) &= \underset{[j_1, \dots, j_N]}{\operatorname{argmin}} g(S(j_1, \dots, j_N)) \\ \text{s.t. } &1 \leq j_i \leq A_i, i = 1, \dots, N \end{aligned} \quad (3)$$

Algorithm 1 Non-redundant TNTP Solver

Input: Task-space path γ , obstacles in environment $\{O\}$, manipulator kinematics and collision model, manipulability constraint $\varepsilon_{\text{sing}}$, configuration connectedness measure $\varepsilon_{\text{cont}}$

Output: All optimal joint-space trajectories

```

1: //Select task-space points for IK calculation
2:  $\{\gamma_i\}_{1 \leq i \leq N} \leftarrow \gamma(\frac{i-1}{N-1}), 1 \leq i \leq N$ 
3: //Calculate valid IK solutions for each point
4: for  $1 \leq i \leq N$  do
5:    $\{c(i, a_i)\}_{1 \leq a_i \leq A_i} \leftarrow \text{calculateValidIK}(\gamma_i)$ 
6:    $D(i, a_i) = -1, 1 \leq a_i \leq A_i$ 
7: end for
8:  $\{D(i, a_i)\}_{\substack{1 \leq i \leq N \\ 1 \leq a_i \leq A_i}} \leftarrow \text{assignJointState}(\{c(i, a_i)\}_{\substack{1 \leq i \leq N \\ 1 \leq a_i \leq A_i}})$ 
9:  $P = \{[0, \dots, 0]\}, \tilde{P} = \emptyset$  // preserved solutions
10: for  $1 \leq i \leq N$  do
11:   //In the  $i$ -th stage
12:   for each solution  $S(a_1, \dots, a_{i-1}, 0, \dots, 0)$  in  $P$  do
13:     if  $\exists j_i, D(i, j_i) = D(i-1, a_{i-1})$  then
14:       // Greedy Strategy
15:       create  $p = S(a_1, \dots, a_{i-1}, a_{i-1}, 0, \dots, 0)$ 
16:       push  $p$  into  $G(i, a_{i-1})$ 
17:     else
18:       for  $1 \leq j_i \leq A_i$  do
19:         create  $p = S(a_1, \dots, a_{i-1}, j_i, 0, \dots, 0)$ 
20:         push  $p$  into  $G(i, j_i)$ 
21:       end for
22:     end if
23:   end for
24:   for  $1 \leq j_i \leq A_i$  do
25:     preserve only least-cost elements in  $G(i, j_i)$ 
26:     push  $G(i, j_i)$  into  $\tilde{P}$ 
27:   end for
28:    $P \leftarrow \tilde{P}$ 
29: end for
30: //All optimal solutions have been collected in  $P$ 
31: for each element in  $P$  do
32:   collect  $\{c(i, a_i)\}_{1 \leq i \leq N}$  based on  $S(a_1, \dots, a_N)$ 
33:   for  $1 \leq i \leq N-1$  do
34:     if  $c(i, a_i)$  and  $c(i+1, a_{i+1})$  not continuous then
35:       do bi-RRT from  $c(i, a_i)$  to  $c(i+1, a_{i+1})$ 
36:       insert the undesirable motion between  $c(i, a_i)$ 
and  $c(i+1, a_{i+1})$ 
37:     end if
38:   end for
39: end for
40: return  $P$ 

```

IV. TNTP OPTIMAL SOLVER

In this section, the non-redundant TNTP problem is effectively solved, and all optimal solutions are proven collectible.

A. Principle of Optimality

We observe that if the state of an intermediate point is assigned, then the calculation of cost becomes two independent parts. Denote the sub-TNTP problem in a similar form with less number of variables $n < N$,

$$S(j_p, \dots, j_q) \triangleq [D(p, j_p), \dots, D(q, j_q)], 1 \leq p < q \leq N \quad (4)$$

with the cost defined similar as Eqn. (2). Let point i be assigned with state $S(i, a_i)$, then

$$\begin{aligned} &\underset{[j_1, \dots, j_{i-1}, j_{i+1}, \dots, j_N]}{\operatorname{argmin}} g(S(j_1, \dots, j_{i-1}, a_i, j_{i+1}, \dots, j_N)) \\ &= \underset{[j_1, \dots, j_{i-1}]}{\operatorname{argmin}} g(S(j_1, \dots, j_{i-1}, a_i)) \\ &\quad + \underset{[j_{i+1}, \dots, j_N]}{\operatorname{argmin}} g(S(a_i, j_{i+1}, \dots, j_N)) - 1 \end{aligned} \quad (5)$$

Here the last term -1 appears because when the manipulator finishes covering the last point of the first sub-TNTP with configuration $D(i, a_i)$, it has actually entered the second sub-TNTP with configuration $D(i, a_i)$ without EE lift-off. Then we can see that an optimal solution to the non-redundant TNTP must contain an optimal solution to the sub-TNTP, which essentially satisfies the principle of optimality for dynamic programming (DP) [18]. This motivates us to encode a partly-solved solution by the state of their frontier point: Let the state of points be assigned from point 1 to point N , we denote the *frontier point* by the last assigned point. We group the problem states with the same cost and the same decision on the frontier point,

$$G(i, a_i) = \{S(j_1, \dots, j_{i-1}, a_i) \mid 1 \leq j_k \leq A_k, 1 \leq k \leq i-1\} \quad (6)$$

In the first stage, all possible states of point 1 are collected, with sets $G(1, 1), \dots, G(1, A_1)$ being obviously constructed with only one element in each set. In the $i (\geq 2)$ -th stage, the state of point $1 \sim (i-1)$ has been assigned and different solutions are collected. After the state of point i is assigned and the cost is updated, all solutions are re-grouped by the state of point i . Only the least-cost solutions are still preserved for the subsequent iteration.

B. Greedy Strategy

Typically, a greedy search strategy performs significantly faster than dynamic programming since no back-tracking process is required. However, it might not guarantee global optimality (minimal cost). In this section we show that greedily choosing the continuous configurations preserves global optimality and completeness in collecting all optimal solutions. The global optimality is verified by observing that in the i -th stage, let there exist a state of point i which is

equal to a_{i-1} ,

$$\begin{aligned}
& \underset{\substack{[j_i, \dots, j_N]}}{\operatorname{argmin}} g(S(a_1, \dots, a_{i-1}, j_i, \dots, j_N)) \\
&= g(S(a_1, \dots, a_{i-1})) - 1 + \underset{\substack{[j_i, \dots, j_N]}}{\operatorname{argmin}} g(S(a_{i-1}, j_i, \dots, j_N)) \\
&= g(S(a_1, \dots, a_{i-1})) - 1 \\
&+ \begin{cases} \underset{\substack{[j_{i+1}, \dots, j_N]}}{\operatorname{argmin}} g(S(a_{i-1}, a_{i-1}, j_{i+1}, \dots, j_N)) \\ \underset{\substack{[j_{i+1}, \dots, j_N]}}{\operatorname{argmin}} g(S(a_{i-1}, a_i, \dots, j_N)) + 1, \text{ enforce } a_i \neq a_{i-1} \end{cases} \\
&= g(S(a_1, \dots, a_{i-1})) - 1 \\
&+ \begin{cases} \underset{\substack{[j_{i+2}, \dots, j_N]}}{\operatorname{argmin}} g(S(a_{i-1}, a_{i-1}, a_{i-1}, \dots, j_N)), \text{ if } a_{i+1} = a_{i-1} \\ \underset{\substack{[j_{i+2}, \dots, j_N]}}{\operatorname{argmin}} g(S(a_{i-1}, a_{i-1}, a_{i+1}, \dots, j_N)) + 1, \text{ else} \\ \underset{\substack{[j_{i+1}, \dots, j_N]}}{\operatorname{argmin}} g(S(a_{i-1}, a_i, \dots, j_N)) + 1, \text{ enforce } a_i \neq a_{i-1} \end{cases} \quad (7)
\end{aligned}$$

We know that if we assign $j_i = a_{i-1}$, the cost will be always no higher than assigning other states $j_i \neq a_{i-1}$ to point i , regardless of whether at the next stage point $(i+1)$ may have the same state as point i . Hence, when the next unassigned point has the same possible state as the frontier point, we can discard all other possible states and directly assign this one.

Next, we show that no optimal solution is missed by the greedy choice of states. If the greedy strategy is not complete, that means an optimal solution contains one of the discarded states by the greedy selection of state, which will be proven retrievable by the collected optimal solutions. Continuing the deduction from Eqn. (7), we have

$$\cdots \text{(Eqn. (7))} \\ = g(S(a_1, \dots, a_{i-1})) - 1 \\ + \begin{cases} \underset{[j_{i+2}, \dots, j_N]}{\operatorname{argmin}} g(S(a_{i-1}, a_{i-1}, a_{i-1}, \dots, j_N)), \text{ if } a_{i+1} = a_{i-1} \\ \underset{[j_{i+2}, \dots, j_N]}{\operatorname{argmin}} g(S(a_{i-1}, a_{i-1}, a_{i+1}, \dots, j_N)) + 1, \text{ else} \end{cases} \\ + \begin{cases} \underset{[j_{i+2}, \dots, j_N]}{\operatorname{argmin}} g(S(a_{i-1}, a_i, a_i, \dots, j_N)) + 1, \text{ if } a_{i+1} = a_i \\ \underset{[j_{i+2}, \dots, j_N]}{\operatorname{argmin}} g(S(a_{i-1}, a_i, a_{i+1}, \dots, j_N)) + 2, \text{ else} \end{cases} \quad (8)$$

Note that the term written in red has a higher cost than the term written in blue, so a discarded state of the current point contributes to an optimal solution only when there exists the same state for the next point, and the algorithm assigns it to the next point at the next stage. However, the solutions are equivalent: when a point can be continuously visited together with either its previous point or its subsequent point, the two solutions are both optimal and can continuously transform from one to another by locally adjusting the point at which the EE departs the task-space path. See Fig. 3 for an illustration. Hence, all optimal solutions are proven collectible by a dynamic programming approach with the greedy utilisation of continuous manipulator configurations.

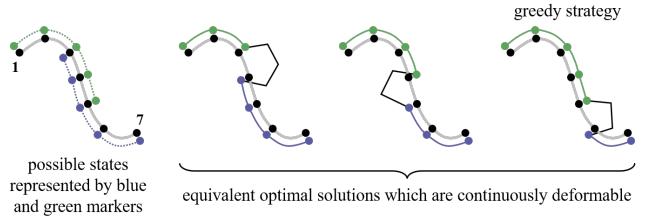


Fig. 3. Illustration of equivalent task-space tracking solutions. Given the 7 discrete points sampled from a task-space curve, two sets of joint-space connected IK solutions are represented by green and blue dots. All solutions on the right consist of a continuous tracking motion with configurations represented by green, an undesirable motion denoted by a black broken line, and a continuous tracking motion represented by blue, and they are all optimal solutions. The greedy strategy will find the last solution as it will lazily choose the configurations in green until it is not trackable.

V. EXPERIMENTS

The proposed algorithm generates all optimal joint-space trajectories ensuring the task-space tracking of a given EE path and a minimum number of manipulator pose re- configurations. Simulated and real-world experiments are presented in this section. In section V-A, the detailed explanation about a case study is presented, with valid manipulator IK configurations illustrated, showing the necessity of manipulator pose re-configuration. To the authors' knowledge, there has not been an efficient algorithm for task-space tracking when the manipulator cannot follow the pre-defined path, as such all discussions were taken for redundant manipulators. So in section V-B we compare the proposed algorithm with the commonly used approach of randomly selecting the next configuration when continuous tracking is impossible. In section V-C, the proposed algorithm is tested on a real-world Universal Robots UR5 manipulator. An implementation in MATLAB has been open sourced: https://github.com/ZJUTongYang/min_reconfig_taskspace_tracking.

A. Case Study

A task-space tracking process is shown here as a case study of the proposed algorithm. The task-space path is generated by interpolating four EE waypoints. Some valid IK solutions are collected in Fig. 4 for better illustration of this case study. For one of the optimal tracking solutions see Fig. 5. Obstacles in the environment are a ground plane and a sphere. Because the starting point of the task-space curve is near the ground, only one IK solution is valid (marked by green dots) so the manipulator would adopt it. However, after the manipulator is about to reach the third intermediate goal, the forearm will hit the spherical obstacle, so one pose reconfiguration is required. The last configuration of the first segment of tracking is shown in Fig. 5(a). By changing to a configuration like shown in Fig. 5(b), the forearm is lifted a little, above the wrist, which makes the EE visit the third intermediate goal. However, the forearm will still hit the spherical obstacle if it continues tracking, where another pose reconfiguration is required. See Fig. 5(c), by adjusting the configuration from shoulder-right to shoulder-left, the manipulator is able to finish the tracking task without

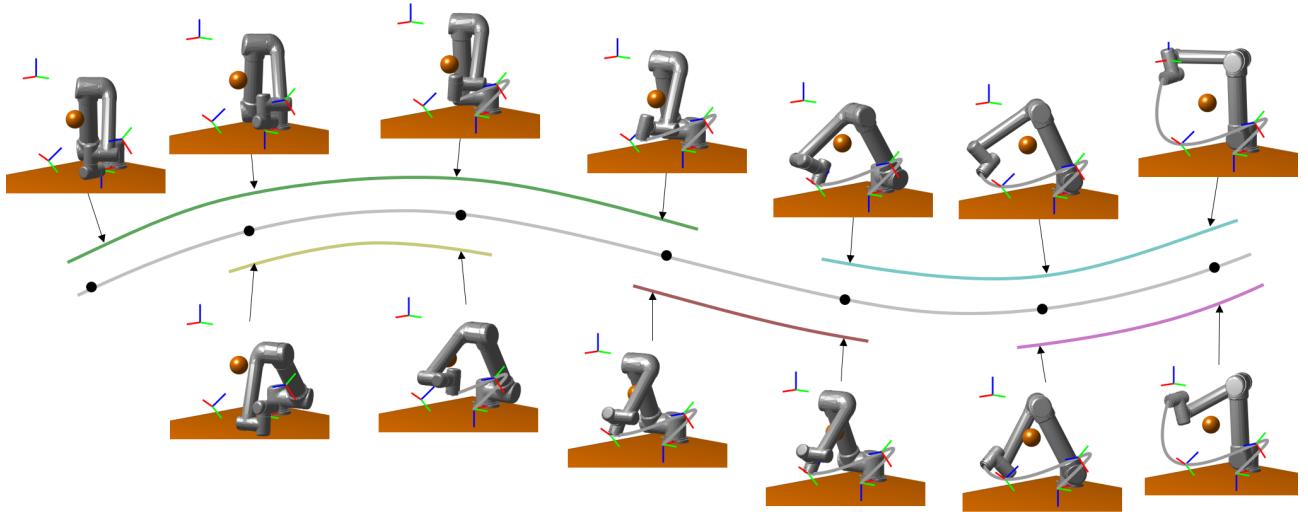


Fig. 4. Illustration of valid IK solutions along the task-space path. The coloured curves show the joint-space connectivity between configurations.

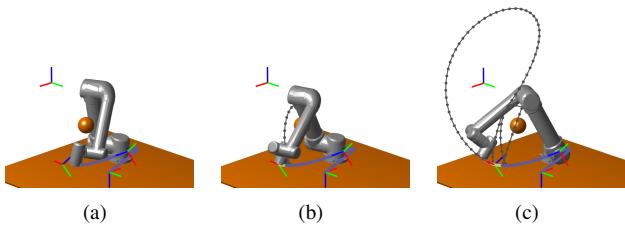


Fig. 5. Illustration of a task-space tracking where two manipulator reconfigurations are required. The three segments of the continuous task-space tracking process (EE poses) are marked by blue, golden, and red colour in order. The gray curves show the trace of EE during reconfiguration.

colliding the spherical obstacle. The reader is referred to our supplementary video to observe the full animation of this case study.

B. Comparison with Sampling-based Planners

All existing solvers for task-space tracking are locally calculating a joint-space admissible trajectory, and when there is locally no valid trajectory to continue tracking, a joint-space path planner (such as the RRT [9] used in this work, as often employed in manipulator planning software such as the widely used MoveIt!³ by the ROS research community, and the OPML⁴ it is based on underneath) is carried out, as such the selected IK solution for visiting the next task-space point is randomly chosen. For fair evaluation, we assume the random selection of the IK solution for visiting the first task-space point, and the algorithms will greedily choose the continuous IK solutions. When the manipulator has to perform a pose reconfiguration motion, the starting pose of the next segment of task-space tracking is again randomly assigned.

An illustration of the test cases is shown in Fig. 6, whilst some relevant statistics are collected in Table II. “Num.” refers to the number of manipulator pose reconfigurations generated by the algorithms. For a fair evaluation of the performance of sampling-based planners in these settings,

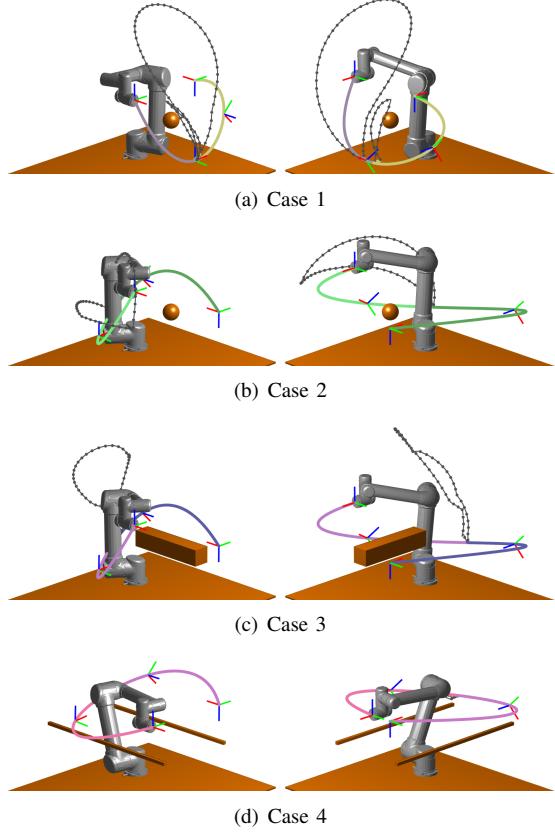


Fig. 6. Illustration of simulated comparison tests. The trajectories shown in the figures are the optimal solutions generated by the proposed algorithm.

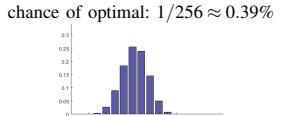
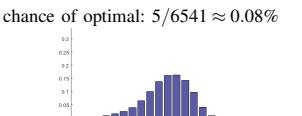
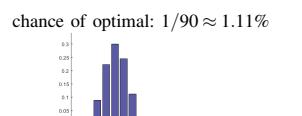
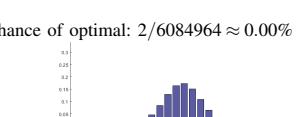
all possible assignments of IK configurations are enumerated, and the mean number of pose reconfigurations for all possible situations is collected as “Mean.”. The last column, “Distribution of Reconf”, reflects the exact histogram (frequency distribution) of the number of reconfigurations in the resulting paths generated by sampling-based planners.

Further metrics are also reported in the table for completeness. While the objective of this work is minimum lift-off task-space tracking trajectories, other criteria would also be of relevance when considering the tracking problem. There

³<https://moveit.ros.org/>

⁴<http://ompl.kavrakilab.org/>

TABLE II
SIMULATION RESULTS VERSUS SAMPLING-BASED PLANNERS

Case	Method	Num. ¹	Mean. ²	Time ³	Bounding Box	Efficiency (T) ⁴	Efficiency (J) ⁵	Distribution of Reconf.
case 1	ours (1/1 ⁶)	2	6.25	42.58s (0.85m×1.00m×1.05m)	0.89m³	1.48m/ 5.04m	6.30rad ⁶ / 17.03rad⁵/19.18rad⁶	chance of optimal: 1/256 ≈ 0.39% 
	random 1	7		117.70s (1.25m×1.15m×1.05m)	1.51m ³	1.46m/14.29m	6.14rad ⁶ /47.08rad ⁵ /51.53rad ⁶	
	random 2	8		122.68s (1.00m×1.25m×1.05m)	1.31m ³	1.46m/13.62m	6.12rad ⁶ /49.07rad ⁵ /54.86rad ⁶	
case 2	ours (2/5)	1	9.96	44.06s (1.15m×1.35m×0.85m)	1.32m ³	2.66m/5.07m	11.38rad ⁶ /17.62rad ⁵ /20.60rad ⁶	chance of optimal: 5/6541 ≈ 0.08% 
				39.06s (1.15m×1.45m×1.00m)	1.67m ³	2.65m/4.52m	11.47rad ⁶ / 15.62rad⁵/17.49rad⁶	
	random 3	3		53.13s (1.05m×1.25m×0.60m)	0.79m³	2.63m/ 3.71m	12.30rad ⁶ /21.25rad ⁵ /25.35rad ⁶	
	random 4	14		176.37s (1.15m×1.50m×1.05m)	1.81m ³	2.47m/17.69m	11.69rad ⁶ /70.55rad ⁵ /74.57rad ⁶	
case 3	ours (1/1)	1	5.10	47.14s (1.10m×1.35m×1.00m)	1.49m³	2.66m/ 4.79m	11.91rad ⁶ / 18.86rad⁵/21.20rad⁶	chance of optimal: 1/90 ≈ 1.11% 
	random 5	7		120.30s (1.20m×1.50m×1.00m)	1.80m ³	2.58m/10.29m	11.01rad ⁶ /48.12rad ⁵ /53.73rad ⁶	
	random 6	4		77.15s (1.20m×1.25m×1.00m)	1.50m ³	2.63m/5.93m	11.33rad ⁶ /30.86rad ⁵ /34.24rad ⁶	
case 4	ours (2/2)	1	13.74	38.05s (1.10m×1.40m×0.70m)	1.08m³	2.54m/ 2.80m	11.52rad ⁶ / 15.22rad⁵/18.15rad⁶	chance of optimal: 2/6084964 ≈ 0.00% 
				39.04s (1.20m×1.40m×1.00m)	1.68m ³	2.54m/4.34m	12.60rad ⁶ /15.62rad ⁵ / 17.82rad⁶	
	random 7	7		147.49s (1.90m×1.55m×1.05m)	3.09m ³	2.49m/21.76m	11.35rad ⁶ /58.99rad ⁵ /66.86rad ⁶	
	random 8	10		215.92s (1.90m×1.65m×1.10m)	3.45m ³	2.47m/31.47m	11.31rad ⁶ /86.37rad ⁵ /94.85rad ⁶	

¹ Number of reconfigurations.

² Mean number of reconfigurations of all greedy solutions.

³ Execution time. Constant joint velocity is assumed, hence shortest-length joint-space is the fastest motion. Ranges of execution time (including undesirable motions) are approximately (sampled for case 2 and 4 given the large population): (Case 1) 42.58s~157.67s, (Case 2) 39.06s~199.31s, (Case 3) 47.14s~120.30s, (Case 4) 38.05s~466.76s.

⁴ Efficiency (T): task-space tracking path length / end-effector path length.

⁵ Efficiency (J): joint-space tracking path length / the first 5-dim joint-space path length / joint-space path length.

⁶ Not all optimal solutions are shown for brevity in each case. A / B: optimal solutions detailed / total number of optimal solutions.

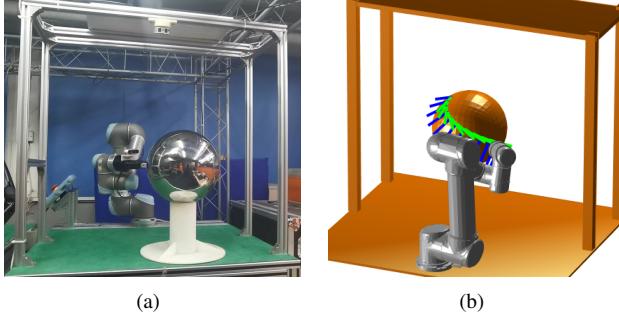


Fig. 7. The real-world scenario and its corresponding simulated environment.

may be motions that contain slightly more reconfigurations but shorter travelling distance, such as the “random 3” result. Ultimately, a trade-off can be derived subject to other criteria, but that falls out of scope for this work.

Taking case 1 as an example, there are 256 different greedy assignments for the segments of continuous task-space motion, among which only one solution identifies with the optimal solution (with 2 reconfigurations). A cuboid workcell admitting this motion is of size 0.89m^3 (“Bounding Box”), the length of task-space motion is 5.04m, and the joint-space travelling distance is 19.18rad^6 . The average number of reconfigurations is shown to be 6.25. When looking at the binned histogram, there is around a 70% chance the sampling-based planner will generate a solution with 5 ~ 7 reconfigurations. The reader is referred to the supplementary video for further visualisations and comparatives during the

resulting planning motions. Time-wise, the optimal tracker (“ours”) is invariably faster in all cases presented.

Note that it is not the focus of this work to exhaustively compare the performance of different joint-space planners that may be able to produce marginally better reconfiguration trajectory in relation to EE lift-off and task space continuity. It is apparent that without a mechanism to select globally optimal IK solutions to visit at the next iteration along the path, solutions based on traditional sampling-based planners will inevitably lead to unnecessary manipulator reconfigurations regardless. In fact, examples in this work have shown a nearly 99% certainty to obtain a non-optimal choice with traditional planners (1 – the probability of randomly attaining the optimal solution, shown in the last column in Table II).

C. Real-world Illustration

The proposed algorithm is also evaluated in a real-world scenario. The real-world environment is modelled into MATLAB as shown in Fig. 7. Let the task-space path be a surface curve on the sphere. The x-axis of the EE is parallel to the surface normal vector, and the y-axis is parallel to the tangent of the surface curve, then z-axis is well-defined following the right-hand coordinates. As such the task is non-redundant for a 6-Dof manipulator. The gripper imitates a non-zero length tool whose collision model is set as a cylinder (visualisation omitted in Fig. 7(b) simulation). Fig. 8 captures the key moments, the reader is referred to the video for full visualisation. To pursue the optimal solution, the manipulator is instructed to start tracking with the configuration shown

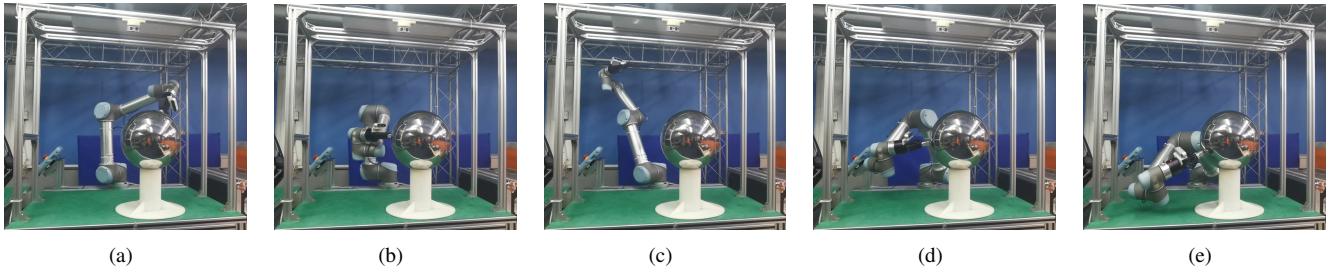


Fig. 8. Video stills of the manipulator configurations during the real-world tests (refer to video for full sequence)

in Fig. 8(a), and continuously track the task-space path until pose Fig. 8(b), where the wrist is about to hit the forearm. One pose reconfiguration from a shoulder-right configuration (Fig. 8(b)) to a shoulder-left configuration (Fig. 8(d)) becomes necessary, the interim lift-off motion captured by Fig. 8(c). The total execution time is 52s, where 18s, 29s, and 5s are used for the manipulator motion in the first segment, the undesirable reconfiguration, and the second and final segment of the tracking, respectively. The manipulator is thus guaranteed to be able to complete tracking of the desired path with only 1 (optimal) pose reconfiguration.

VI. CONCLUSION

A novel mechanism to generate an optimal joint-space trajectory for the task-space non-revisiting tracking problem (TNTP) has been proposed in this work. The optimality is translated to the minimum number of segmentations of the pre-defined path caused by nonlinear manipulator kinematics and collision with obstacles in the environment. Manipulator reconfigurations are shown to be necessary for concatenating consecutive segments of continuous task-space tracking. These undesirable deviations generally result in additional time and energy penalties. When compared to existing task-space tracking solutions, the proposed algorithm provides a globally optimal perspective to the choice of suitable manipulator inverse kinematics, maximising the joint-space connectedness during the tracking task. All optimal solutions (sequences of IK solutions ensuring minimal reconfigurations) are proven collected via a dynamic programming solver, where a proposed greedy speeding-up strategy is shown to be without loss of global optimality and completeness. Simulated comparisons and real-world illustration has proven the validity of the proposed algorithm, with substantial reconfigurability and resulting trajectory execution time improvements. These have been supplemented by a detailed video and an open source implementation in MATLAB for the research community benefit.

On a side note we like to observe that the proposed algorithm calculates a “joint-space” cost for a given task-space path. It is argued that a metric based on the simple summations of joint-space distance between consecutive valid IK configurations is not always an appropriate cost measurement, as reconfigurations can significantly influence tracking performance, and cannot be overlooked in applications where accuracy might be critical. In contrast, the proposed algorithm, with a minimum number of manipulator reconfigurations being explicitly considered, builds a more

realistic estimation of task-space tracking cost, which might be inspirational to the community for other works.

REFERENCES

- [1] X. Liang, X. Huang, M. Wang, and X. Zeng, “Adaptive task-space tracking control of robots without task-space-and joint-space-velocity measurements,” *IEEE Transactions on robotics*, vol. 26, no. 4, pp. 733–742, 2010.
- [2] J. Norberto Pires, A. Loureiro, T. Godinho, P. Ferreira, B. Fernando, and J. Morgado, “Welding robots,” *IEEE Robotics Automation Magazine*, vol. 10, no. 2, pp. 45–55, 2003.
- [3] D. Nakhaeinia, R. Fareh, P. Payeur, and R. Laganière, “Trajectory planning for surface following with a manipulator under rgb-d visual guidance,” in *2013 IEEE international symposium on safety, security, and rescue robotics (SSRR)*, pp. 1–6, IEEE, 2013.
- [4] E. Simetti, S. Galeano, and G. Casalino, “Underwater vehicle manipulator systems: Control methodologies for inspection and maintenance tasks,” in *OCEANS 2016-Shanghai*, pp. 1–7, IEEE, 2016.
- [5] C.-S. Chen and S.-K. Chen, “Synchronization of tool tip trajectory and attitude based on the surface characteristics of workpiece for 6-dof robot manipulator,” *Robotics and Computer-Integrated Manufacturing*, vol. 59, pp. 13–27, 2019.
- [6] M. Janiak, “From motion planning through waypoints to open-loop trajectory tracking algorithm,” in *2017 11Th International Workshop on Robot Motion and Control (Romoco)*, pp. 142–147, IEEE, 2017.
- [7] R. V. Mayorga and A. K. Wong, “A singularities avoidance approach for the optimal local path generation of redundant manipulators,” in *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*, pp. 49–54, IEEE, 1988.
- [8] W. Xu, J. Zhang, B. Liang, and B. Li, “Singularity analysis and avoidance for robot manipulators with nonspherical wrists,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 1, pp. 277–290, 2015.
- [9] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.
- [10] O. Egeland, “Task-space tracking with redundant manipulators,” *IEEE Journal on Robotics and Automation*, vol. 3, no. 5, pp. 471–475, 1987.
- [11] J. M. Porta, L. Jaijlet, and O. Bohigas, “Randomized path planning on manifolds based on higher-dimensional continuation,” *International Journal of Robotics Research*, vol. 31, no. 2, pp. 201–215, 2012.
- [12] B. Xian, M. S. de Queiroz, D. Dawson, and I. Walker, “Task-space tracking control of robot manipulators via quaternion feedback,” *IEEE Transactions on Robotics and Automation*, vol. 20, no. 1, pp. 160–167, 2004.
- [13] R. Mayorga and A. Wong, “A singularities avoidance method for the trajectory planning of redundant and nonredundant robot manipulators,” in *Proceedings of the 1987 IEEE International Conference on Robotics and Automation*, vol. 4, pp. 1707–1712, IEEE, 1987.
- [14] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” in *Autonomous robot vehicles*, pp. 396–404, Springer, 1986.
- [15] T. Yang, J. V. Miro, Q. Lai, Y. Wang, and R. Xiong, “Cellular decomposition for non-repetitive coverage task with minimum discontinuities,” *IEEE/ASME Transactions on Mechatronics*, 2020.
- [16] T. Yang, J. V. Miro, Y. Wang, and R. Xiong, “Non-revisiting Coverage Task with Minimal Discontinuities for Non-redundant Manipulators,” in *Proceedings of Robotics: Science and Systems*, (Corvallis, Oregon, USA), July 2020.
- [17] T. Yoshikawa, “Translational and rotational manipulability of robotic manipulators,” *American Control Conference*, vol. 27, pp. 228–233, 1990.
- [18] R. Bellman, *Dynamic programming*. Courier Corporation, 2013.