IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received January 14, 2020, accepted January 27, 2020, date of publication January 30, 2020, date of current version February 25, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2970540

Motion Planning With Pose Constraints Based on Direct Projection for Anthropomorphic Manipulators

JIANGPING WANG, (Student Member, IEEE), SHIRONG LIU, (Member, IEEE), BOTAO ZHANG, AND QIANG LU, (Member, IEEE)
School of Automation, Hangzhou Dianzi University, Hangzhou 310018, China

Corresponding author: Shirong Liu (liushirong@hdu.edu.cn)

This work was supported in part by the Key Research and Development Program of Zhejiang Province, China, under Grant 2019C04018, in part by the NSFC under Grant 61503108, and in part by the 111 Project under Grant D17019.

ABSTRACT This paper presents an efficient planning algorithm, called Direct-Projection Rapidly-exploring Random Tree (DP-RRT), to address the motion planning with end-effector pose constraints for anthropomorphic manipulators. The key of this planning problem is to find constraint-satisfying configurations on the constraint manifolds and connect them to generate a collision-free and smooth path. In the previous works, the configurations that satisfy pose constraints are generally calculated by the numerical iteration methods based on Jacobian projection techniques. However, such approaches have many technical challenges, such as joint limits and singularity, many numerical iterations and much computing time. In this work, we propose a **Direct Projection** method based on the analytic inverse kinematics (IK) that can directly project configurations onto the constraint manifolds instead of the numerical iteration methods. The proposed DP-RRT algorithm combines the Direct Projection method with the Rapidly-exploring Random Tree (RRT) algorithm, where the RRT algorithm is employed to explore the ambient space by growing tree branches, and the Direct Projection method is used to project the tree branches onto the constraint manifolds for constructing a constraint-satisfying path. As the analytic IK solver is used to calculate the constraint-satisfying configurations, the DP-RRT algorithm is characterized by high efficiency and no numerical iteration. Besides, avoiding joint limits and singularity, as well as the smoothness of the end-effector and the joints trajectory are also considered. The effectiveness of the proposed algorithm is demonstrated on the Willow Garage's PR2 simulation platform in a wide range of pose-constrained cases.

INDEX TERMS Motion planning, direct projection, pose constraints, analytic inverse kinematics, anthropomorphic manipulator.

I. INTRODUCTION

The manipulator motion planning is one of the challenging problems in the field of robot manipulators, which involves finding a collision-free path under certain constraints from an initial to a goal configuration among a collection of obstacles. This problem is known to be a PSPACE-hard problem [1] and its complexity increases with the number of the robot's degrees of freedom (DOFs) and the task constraints. Sampling-based motion planners [2]–[4] are generally regarded as the state-of-the-art method for such high-dimensional planning problems due to the outstanding advantages in efficiency, easy implementation and the ability to handle a multitude of different constraints [5]. But these planners are difficult to solve the motion planning with pose (position and/or orientation) constraints imposed on the robot's end-effector [6], [7], which are ubiquitous in industry and our daily life, e.g., moving a cup full of water from one place to another while keeping it vertically all along the motion; opening a door or rotating a valve.

There are two main factors prohibiting the adoption of sampling-based planning methods for these problems. Firstly, it is difficult to directly compute feasible configurations that lie on the desired constraint manifolds [8], [9]. For the task-constrained motion planning, one always wants to find a direct way that can quickly generate configurations

The associate editor coordinating the review of this manuscript and approving it for publication was Liang Hu.

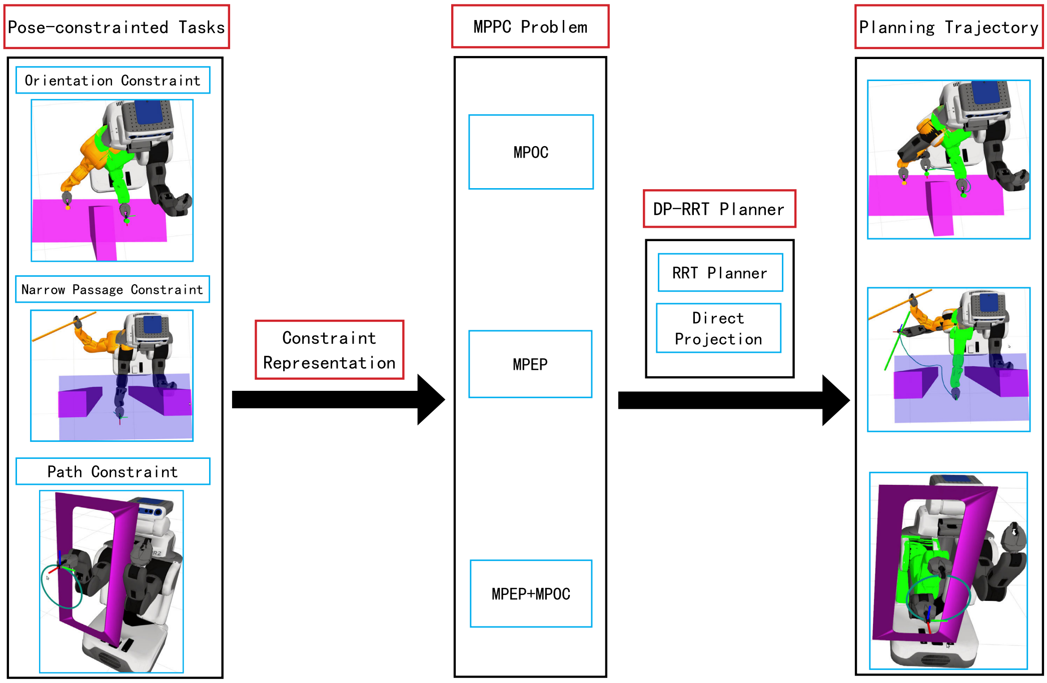


FIGURE 1. An overview of the DP-RRT algorithm for the motion planning with pose constraints (MPPC) problem implemented on the PR2 simulation robot.

meeting the task constraints. However, position and orientation of the robot's end-effector usually are coupled, which complicates the solution to the analytic inverse kinematics (IK) of the desired position or orientation. Secondly, the probability that a randomly sampling configuration in configuration space (C-space) lies on the desired constraint manifolds is extremely low and usually null. The constraint manifold is the space composed of all constraint-satisfying configurations, which is a zero-measure space of the ambient space or C-space. Intuitively, the desired constraint manifolds have small volumes in C-space and thus the probability of sampling from the constraint manifolds is very low. In the previous works [6]–[19], the numerical iteration methods based on Jacobian projection techniques have been generally used to generate configurations that satisfy pose constraints. However, such approaches have many technical challenges, such as avoiding joint limits and singularity, requiring many numerical iterations and much computing time.

In this paper, we focus on the motion planning with pose constraints (MPPC) of anthropomorphic manipulators by a Direct Projection method that can directly find the constraint-satisfying configurations without the numerical iteration. By combining the Direct Projection method with the Rapidly-exploring Random Tree (RRT) [3] algorithm, an sampling-based planning algorithm termed Direct-Projection Rapidly-exploring Random Tree (DP-RRT) is presented for planning on the constraint manifolds. In detail, we first decompose the MPPC problem into two independent subproblems of the lower-dimensional subspace, namely motion planning along end-effector paths (MPEP) [10] and motion planning with orientation constraints (MPOC) [11], [12]. Then the Direct Projection method based on the analytic IK of the anthropomorphic manipulator is proposed to project configurations of the ambient C-space onto the constraint manifolds. Finally, the DP-RRT algorithm uses the RRT algorithm to explore the ambient C-space by growing tree branches and employs the Direct Projection method to project the tree branches onto the constraint manifolds for constructing a constraint-satisfying path. An overview of the DP-RRT algorithm for the MPPC problem is presented in Figure 1.

Contributions: The proposed DP-RRT algorithm can be applied to solve the general MPPC problems for anthropomorphic robots, which are generally solved by numerical iteration methods based on Jacobian projection techniques. Our approach is different from the previously proposed numerical iteration methods, where the Direct Projection method is adopted to project configurations onto the constraint manifolds. The planning time of the DP-RRT algorithm is significantly reduced, because the analytic IK solver is applied to compute the constraint-satisfying configurations directly.

II. RELATED WORK

To address the motion planning with pose-constrained tasks, the iterative projection strategies (Figure 2), which iteratively project unsatisfying samples in C-space closer to the constraint manifolds by a projection operator, have been verified to be effective. The iterative projection technique was originally proposed for the kinematic closure constraints of closed-chain robots by the randomized gradient

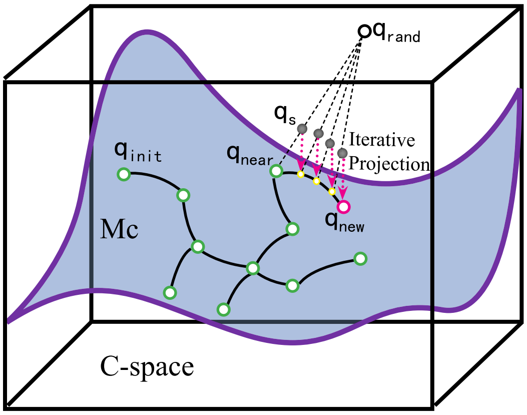


FIGURE 2. In the iterative projection method, the seed configurations q_s represented by gray dots are pushed onto the constraint manifolds (blue space) iteratively and then connected to form a branch.

descent (RGD) algorithm [13], [14]. The Tangent Space Sampling and the First-Order Retraction [15] were proposed to solve the task constraints imposed on the manipulator's end-effector, which are more invariant and effective than the RGD algorithm. More recently, Berenson et al. [16] unified the representation of pose-related constraints by the Task Space Regions (TSRs) and presented the Constrained Bi-directional Rapidly-exploring Random Tree (CBiRRT) [9] to solve the general end-effector constraints. In order to further improve the efficiency of the projection operation, planners such as Tangent Space RRT [8], Atlas RRT [17] and Tangent Bundle RRT [18] were proposed to iteratively project samples onto the approximation of constraint manifolds: tangent spaces or tangent polytopes. Kingston et al. [6] summarized the previous works and presented a framework that can decouple the constraints from sampling-based planners. Although iterative projection strategies based on Jacobian pseudoinverse are feasible, these methods have many technical challenges, such as avoiding joint limits and singularity, requiring many numerical iterations and thus taking a lot of computing time.

As for the MPEP problem of redundant anthropomorphic arms, we need to find a sequence of collision-free joint motions so that the robot's end-effector can track the desired Cartesian path. Since there are infinitely many IK solutions (self motion) for a redundant manipulator due to its kinematic redundancy, the robot can be with different joint movements to move along the same path. This provides means for avoiding obstacle under the condition of meeting the primary task. Thus, the MPEP can be considered as a special case of the redundancy resolution [19] based on the kinematic control of the manipulator, where the redundancy resolution enables the robot to achieve appropriate configurations under the constraints of velocity [20]–[23], acceleration [24], [25] or avoiding collisions. However, these approaches would suffer from the singularity and much iteration time problem, as they work at the velocity or acceleration level with the numerical iteration method. An alternative approach is the analytical inverse kinematics-based motion planning methods [10], [26] at the position level, which can only track a position path with improved efficiency. However, it is not suitable for the manipulator to track a path with both position and orientation constraints because of the nonlinear coupling between them.

III. CONSTRAINT REPRESENTATION AND DIRECT PROJECTION

In this section, a generalized representation of pose constraints will be provided firstly, which can be divided into position constraint and orientation constraint. Then the constraint manifolds will be introduced to describe the general MPPC problem. Finally, the Direct Projection method based on the analytical IK solver of 7-DOF anthropomorphic manipulators will be given to show how to project the configurations onto the constraint manifolds directly.

A. POSE CONSTRAINT REPRESENTATION

Pose constraints usually take the form of the position and/or orientation constraints of the robot's end-effector. In this work, a 6-dimensional vector function $[x(t), y(t), z(t), \alpha(t), \beta(t), \gamma(t)]^T$ is adopted to specify the end-effector pose constraints, where we assume that the vector function only depends on a single parameter t. The vector function consists of a translation component $P(t) = [x(t), y(t), z(t)]^T$ and a 3-dimensional rotation component described by Roll-Pitch-Yaw (RPY) Euler angles $\phi(t) = [\alpha(t), \beta(t), \gamma(t)]^T$.

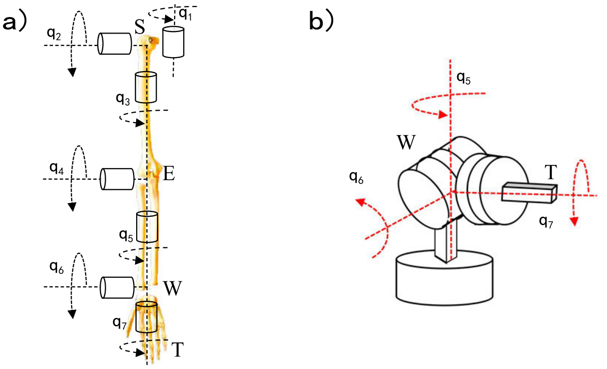
$$C^r = \begin{bmatrix} x(t) \\ y(t) \\ z(t) \\ \alpha(t) \\ \beta(t) \\ \gamma(t) \end{bmatrix}, \quad where \begin{cases} x(t) \in (x_{min}, x_{max}); \\ y(t) \in (y_{min}, y_{max}); \\ z(t) \in (z_{min}, z_{max}); \\ \alpha(t) \in (\alpha_{min}, \alpha_{max}); \\ \beta(t) \in (\beta_{min}, \beta_{max}); \\ \gamma(t) \in (\gamma_{min}, \gamma_{max}). \end{cases} \tag{1}$$

The first three rows of C^r limit the positions of the robot's end-effector in the direction of x, y and z axes (in meters) and the last three rows limit the rotations about these axes (in radians).

B. CONSTRAINT MANIFOLDS

The C-space (denoted by $\mathcal{Q}$) consisting of all possible configurations is a key concept in motion planning. The advantage of using C-space is that it allows us to describe the robots of complex geometric shape with a single point in the space $\mathcal{Q}$. For example, the configuration of a robot arm with n joints can be represented by an n-dimensional vector containing n joint positions, $\boldsymbol{q} = (q_1, q_2, \ldots, q_n)$. Therefore, the robot motion planning is equivalent to the path planning of a single point in $\mathcal{Q}$. The free C-space $\mathcal{Q}_{free} \subseteq \mathcal{Q}$ consists of all the collision-free configurations in $\mathcal{Q}$, where the robot does not collide with obstacles or itself. The basic motion planning problem is to find a continuous path τ in $\mathcal{Q}_{free}$ from an initial state $\boldsymbol{q}_{init} \in \mathcal{Q}_{free}$ to a goal state $\boldsymbol{q}_{goal} \in \mathcal{Q}_{free}$, such that

$$\begin{aligned} \tau(0, 1) &\rightarrow \mathcal{Q}_{\text{free}} \\ \tau(0) &= \boldsymbol{q}_{\text{init}}; \tau(1) = \boldsymbol{q}_{\text{goal}} \end{aligned} \tag{2}$$



c)

Joints	α	a	θ	d
1	-π/2	a_1	q_1	d_1
2	π/2	0	q_2	0
3	-π/2	0	q_3	d_3
4	π/2	0	q_4	0
5	-π/2	0	q_5	d_5
6	π/2	0	q_6	0
7	0	0	q_7	d_7

FIGURE 3. (*a*) **A general kinematic model of the 7-DOF anthropomorphic manipulator in analogy with a human arm. The configuration of the anthropomorphic manipulator consists of the main-arm joints (q_1, q_2, q_3 and q_4) and the wrist joints (q_5, q_6 and q_7). The symbol of S, E, W, T means shoulder, elbow, wrist, end-effector respectively.** (*b*) **Spherical wrist, three revolute axes intersect at a single point *W*.** (*c*) **D-H parameters of the anthropomorphic manipulator.**

The above motion planning only focuses on the collision avoidance, without considering the pose constraints imposed on the robot's end-effector. It means that the manipulator's configuration parameters $q_1, q_2, \ldots, q_n$ are independent to each other. However, for the MPPC problem, these parameters $q_1, q_2, \ldots, q_n$ are correlated due to the nonlinear equality constraints $F(\boldsymbol{q}) = C^r$ imposed on the robot. Supposing there are k $(n > k > 0)$ equality constraints and thus a $(n-k)$-dimensional sub-manifold can be delineated in C-space $\mathcal{Q}$. The lower-dimension sub-manifold is also known as constraint manifolds $\mathcal{M}_\mathcal{C} \subset \mathcal{Q}$, where all the configurations must meet the constraints $F(\boldsymbol{q}) = C^r$.

$$\mathcal{M}_\mathcal{C} = \{\boldsymbol{q} = (q_1, q_2, \ldots, q_n) \in \mathcal{Q} \mid F(\boldsymbol{q}) = C^r\}. \tag{3}$$

As mentioned, the region of constraint manifolds is not known a priori. To generate the constraint-satisfying configurations, the planning algorithms need to either iteratively project the C-space configurations onto the constraint manifolds or directly compute analytical IK solutions that inherently satisfy the equality constraints (Direct Projection). However, computing the constraint-satisfying configurations by the analytical IK method is difficult and even infeasible due to the coupling of the end-effector position and orientation. Fortunately, for the anthropomorphic manipulators, it is possible to generate such constraint-satisfying configurations by the Direct Projection method.

C. DIRECT PROJECTION

To solve the MPPC problem, we need to project the C-space configurations onto the constraint manifolds $\mathcal{M}_\mathcal{C}$. This paper presents a Direct Projection method based on the analytic IK that can directly project the C-space configurations onto $\mathcal{M}_\mathcal{C}$ for anthropomorphic manipulators. As is the case with human arms, anthropomorphic manipulators (Figure 3(*a*)) consists of three revolute joints in the shoulder, a revolute joint in the elbow and three revolute joints in the humanoid spherical wrist (Figure 3(*b*)). The key feature of the spherical wrist is that it decouples the position from the orientation of end-effector motion [27], [28]. That is, the wrist position and orientation can be independently controlled by the main-arm joints $\boldsymbol{q}_{arm}$ (the first four joints) and the wrist joints $\boldsymbol{q}_{wrist}$ (the last three joints) respectively. This special property makes it possible to derive IK solutions in an analytic way [29] by *inverse position kinematics* and *inverse orientation kinematics*.

As shown in Figure 3(*c*), the Denavit-Hartenberg (D-H) [30] parameters are given to formulate the kinematic mode of a general 7-DOF anthropomorphic manipulator, where d_1, d_3, d_5 and d_7 are the length of shoulder, upper arm, forearm and end-effector respectively, and a_1 is the shoulder offset. Suppose that a desired pose of the end effector frame {7} relative to the base frame {0} is expressed as:

$${}^0_7T = \begin{bmatrix} {}^0_7R & {}^0_7P \\ 0 & 1 \end{bmatrix}, \tag{4}$$

where ${}^0_7T$ is the homogeneous transformation matrix consisting of a 3 × 3 rotation matrix ${}^0_7R$ for orientation and a 3 × 1 translation vector ${}^0_7P$ for position. As the robot's end-effector is mounted on the spherical wrist fixedly, we can calculate the position of the wrist ${}^0_6P$ relative to base frame {0} when the end-effector position and orientation are known.

$${}^0_6P = {}^0_7P - {}^0_7R \cdot {}^6_7P, \tag{5}$$

where ${}^6_7P = (0, 0, d_7)^T$ is a constant vector.

Since the last three joint axes of the anthropomorphic manipulator intersect at the wrist, the wrist position ${}^0_6P$ is determined only by the main-arm joints $\boldsymbol{q}_{arm} = (q_1, q_2, q_3, q_4)$. Obviously, there are four unknowns for three scalar equations. Considering that the main-arm joint angles can be easily solved if the position of shoulder is specified, hence we choose the first angle q_1 as the redundant variable. The position of shoulder relative to the base frame {0} can be computed as ${}^0_1P{=}[a_1\ c_1, a_1\ s_1, d_1]^T$. Thus, the points of shoulder (S), elbow (E) and wrist (W) can form a triangle, and two sets of elbow joint angle q_4 can be calculated by the triangle cosine theorem

$$q_4 = \pm(\pi - \mathrm{acos2}(d_3^2 + d_5^2 - d_{16}^2, 2d_3d_5)), \tag{6}$$

where $d_{16} = \|{}^{0}_{6}P - {}^{0}_{1}P\|$ is the length of the triangle hypotenuse.

According to (5), the position of the wrist ${}^{0}_{6}P$ can be calculated by the known end-effector pose ${}^{0}_{7}T$. Therefore, the following formula can be further obtained:

$${}^{0}_{5}P = {}^{0}_{1}P + {}^{0}_{1}R \cdot {}^{1}_{2}P + {}^{0}_{1}R \cdot {}^{1}_{2}R \cdot {}^{2}_{3}P + {}^{0}_{1}R \cdot {}^{1}_{3}R \cdot {}^{3}_{4}P + {}^{0}_{1}R \cdot {}^{1}_{4}R \cdot {}^{4}_{5}P. \tag{7}$$

where ${}^{0}_{5}P = {}^{0}_{6}P$, ${}^{0}_{1}P = (0, 0, d_1)^T$, ${}^{2}_{3}P = (0, 0, d_3)^T$, ${}^{4}_{5}P = (0, 0, d_5)^T$ and ${}^{1}_{2}P = {}^{3}_{4}P = (0, 0, 0)^T$. Furthermore, the relation (7) can be simplified to $({}^{0}_{1}R)^{-1} \cdot ({}^{0}_{5}P - {}^{0}_{1}P) = {}^{1}_{2}R \cdot {}^{2}_{3}P + {}^{1}_{4}R \cdot {}^{4}_{5}P$and then

$$\begin{bmatrix} x_7c_1 + y_7s_1 - a_1 \\ -z_7 + d_1 \\ y_7c_1 - x_7s_1 \end{bmatrix} = \begin{bmatrix} d_5c_2c_3s_4 + d_5s_2c_4 + d_3s_2 \\ d_5s_2c_3s_4 - d_5c_2c_4 - d_3c_2 \\ d_5s_3s_4 \end{bmatrix}. \tag{8}$$

Two sets of solution can be obtained from (8),

$$\begin{aligned} q_3 &= \mathrm{asin2}(y_7\, c_1 - x_7\, s_1, d_5\, s_4) \\ q_2 &= \mathrm{atan2}(k_1k_3 - k_2k_4, k_1k_4 + k_2k_3) \end{aligned} \tag{9}$$

for $q_3 \in [-\pi/2, \pi/2)$, and

$$\begin{aligned} q_3 &= \pi - \mathrm{asin2}(y_7\, c_1 - x_7\, s_1, d_5\, s_4) \\ q_2 &= \mathrm{atan2}(k_1k_3 - k_2k_4, k_1k_4 + k_2k_3) \end{aligned} \tag{10}$$

for $q_3 \in [\pi/2, 3\pi/2)$. The notations c_i and s_i are the abbreviations for $cos(q_i)$ and $sin(q_i)$ respectively, and $k_1 = d_5\, c_4 + d_3$, $k_2 = d_5\, c_3\, s_4$, $k_3 = x_7\, c_1 + y_7\, s_1 - a_1$, $k_4 = z_7 - d_1$.

So far, q_2, q_3, q_4 can be obtained by *inverse position kinematic* when the redundancy angle q_1 is specified. On the other hand, as the orientation of end-effector is determined by $\boldsymbol{q}_{wrist}$, analytical IK solutions of $\boldsymbol{q}_{wrist}$ can be solved through *inverse orientation kinematics* described by ${}^{4}_{5}R \cdot {}^{5}_{6}R \cdot {}^{6}_{7}R = ({}^{0}_{4}R)^{-1} \cdot {}^{0}_{7}R$. And it can be expressed as

$$\begin{bmatrix} c_5c_6c_7 - s_5s_7 & -s_5c_7 - c_5c_6s_7 & c_5s_6 \\ c_5s_7 + s_5c_6c_7 & c_5c_7 - s_5c_6s_7 & s_5s_6 \\ -s_6c_7 & s_6s_7 & c_6 \end{bmatrix} = \left(p_{ij}\right), \tag{11}$$

where p_{ij} are the elements of the $({}^{0}_{4}R)^{-1} \cdot {}^{0}_{7}R$. Thus, the following two sets of solutions can be derived

$$\begin{aligned} q_5 &= \mathrm{atan2}(p_{23}, p_{13}) \\ q_6 &= \mathrm{atan2}(\sqrt{p_{23}^2 + p_{13}^2}, p_{33}) \\ q_7 &= \mathrm{atan2}(p_{32}, -p_{31}) \end{aligned} \tag{12}$$

for $q_6 \in (0, \pi)$, and

$$\begin{aligned} q_5 &= \mathrm{atan2}(-p_{23}, -p_{13}) \\ q_6 &= \mathrm{atan2}(-\sqrt{p_{23}^2 + p_{13}^2}, p_{33}) \\ q_7 &= \mathrm{atan2}(-p_{32}, p_{31}) \end{aligned} \tag{13}$$

for $q_6 \in (-\pi, 0)$. To summarize, once q_1 was chosen as the redundant variable, the remaining joint angles $q_2 \sim q_7$ can be calculated analytically. For a given end-effector pose and an appropriate redundancy angle, up to eight sets of analytical IK solutions can be solved.

Algorithm 1 *DirectProjection*$(\boldsymbol{q}_s = (\boldsymbol{q}_{arm}, \boldsymbol{q}_{wrist}))$

```
 1: if OrientationConstraintTrue then
 2:     $\Phi \leftarrow \phi(t)$;
 3:     $\boldsymbol{q}_{wrist} \leftarrow InverOrienKinem(\Phi, \boldsymbol{q}_{arm})$;
 4: else
 5:     $\Phi \leftarrow F(\boldsymbol{q}_s)$;
 6: end if
 7: if PositionConstraintTrue then
 8:     $\boldsymbol{q}_{arm} \leftarrow InverPositKinem(P(t_i), \Phi, q_1)$;
 9:     $\boldsymbol{q}_{wrist} \leftarrow InverOrienKinem(\Phi, \boldsymbol{q}_{arm})$;
10: end if
11: return $\boldsymbol{q} \leftarrow (\boldsymbol{q}_{arm}, \boldsymbol{q}_{wrist})$;
```

Based on the analytical IK of anthropomorphic manipulators, the pseudo-code procedure of projecting a C-space configuration $\boldsymbol{q}_s$ onto the constraint manifolds is described in Algorithm 1. In detail, for the MPOC problem with a predefined orientation $\phi(t)$ (Lines 1 and 2), the inverse orientation kinematics function *InverOrienKinem*() tries to calculate the wrist angles $\boldsymbol{q}_{wrist}$ for the given Euler angles Φ without changing the position of wrist (Line 3). Obviously, the new configuration $\boldsymbol{q}$ consisting of the original $\boldsymbol{q}_{arm}$ and the calculated $\boldsymbol{q}_{wrist}$ can meet the given orientation constraints (Line 11). From Eq. 12 and Eq. 13, we can see that up to two configurations can be projected onto the orientation constraint manifolds.

As to the position-constrained task MPEP, the robot's end-effector needs to follow the desired path $P(t) \in \mathbb{R}^3$ assigned with a path parameter t. In order to track the path, we discretize $P(t)$ into a series of way-points by a sequence $\{t_1, t_2, \ldots, t_N\}$, and then complete the path planning between the way-points. Thus, the MPEP problem is to find a sequence of joint configurations $\{\boldsymbol{q}(t_1), \boldsymbol{q}(t_2), \ldots, \boldsymbol{q}(t_N)\}$ in C-space that satisfy the forward kinematics $F(\boldsymbol{q}(t_i)) = P(t_i)$, $i = 1, 2 \ldots, N$ and such that no collision occurs along the joint-space path (Figure 4). In order to seek the configuration(s) $\boldsymbol{q}$ such that $F(\boldsymbol{q}) = P(t_i)$, the proposed Direct Projection method projects the configuration $\boldsymbol{q}_s$ of the ambient C-space onto the position constraint manifolds (or null-space manifolds) directly. Firstly, we need to check whether there are orientation constraints. If so, Euler angles Φ is assigned the predefined orientation $\phi(t)$ (Line 2); otherwise it retains the original orientation derived by Forward Kinematic function $F()$ (Line 5). Then, the inverse position kinematics function *InverPositKinem*() calculates the main-arm angles $\boldsymbol{q}_{arm}$ with the given $P(t_i)$ and Φ (Line 8). Finally, *InverOrienKinem*() calculates the $\boldsymbol{q}_{wrist}$ with the given Φ and $\boldsymbol{q}_{arm}$ (Line 9). Obviously, the new configuration $\boldsymbol{q}$ consisting of $\boldsymbol{q}_{arm}$ and $\boldsymbol{q}_{wrist}$ can meet the given position constraints and even orientation constraints (Line 11). As aforementioned, up to eight new configurations can be obtained.

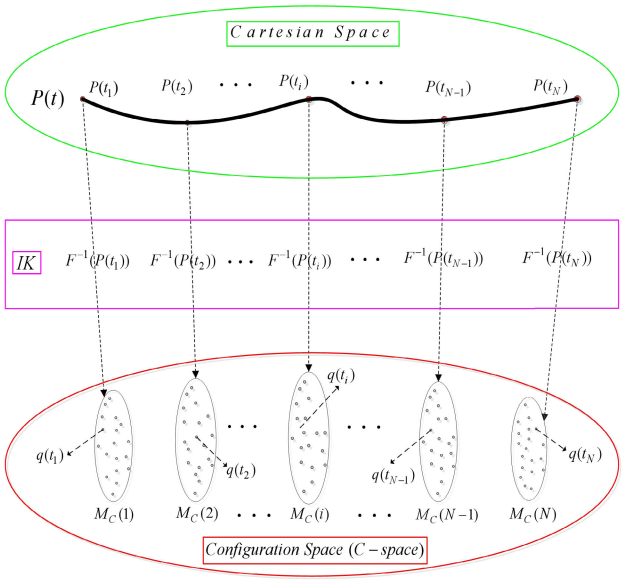


FIGURE 4. ***P*(*t*) is a desired trajectory which has a sequence of discrete way-points. For each way-point, there are infinite IK solutions that can form the position constraint manifolds (or null-space manifolds). For instance, the position constraint manifolds $\mathcal{M}_C(i)$ consists of infinite IK solutions $q(t_i)$.**

Algorithm 2 DP-RRT

```
Input: $V \leftarrow \{q_{init}\}$; $V_{pre\&cur} \leftarrow \{q_{init}\}$; $V_{current} \leftarrow \emptyset$; $E \leftarrow \emptyset$;
Output: $G \leftarrow (V, E)$;
1:  if $OrientationConstraintTrue$ then
2:      $Num = 1$;
3:  else if $PositionConstraintTrue$ then
4:      $Num = N$;
5:  end if
6:  for $i = 1 \rightarrow Num$ do
7:      $V_{current} \leftarrow Initialize(V_{current})$;
8:      for $j = 1 \rightarrow n$ do
9:          $q_{rand} \leftarrow SampleFree(j)$;
10:         $q_{near} \leftarrow Nearest(V_{pre\&cur}, q_{rand})$;
11:         $q_s \leftarrow Steer(q_{near}, q_{rand})$;
12:         $q_{nodes}, \delta(q_{near}, q_{nodes}) \leftarrow DirectProjection(q_s)$;
13:         $q_{new}, \delta(q_{near}, q_{new}) \leftarrow SelectOptimum(q_{nodes})$;
14:         if $CollisionChecking(\delta(q_{near}, q_{new}))$ then
15:             $V \leftarrow V \bigcup \{q_{new}\}$; $E \leftarrow E \bigcup \{\delta(q_{near}, q_{new})\}$;
16:             $V_{pre\&cur} \leftarrow V_{pre\&cur} \bigcup \{q_{new}\}$;
17:             $V_{current} \leftarrow V_{current} \bigcup \{q_{new}\}$;
18:         end if
19:     end for
20:     $V_{pre\&cur} \leftarrow Initialize(V_{pre\&cur})$;
21:     $V_{pre\&cur} \leftarrow V_{current}$;
22: end for
23: return $G \leftarrow (V, E)$;
```

IV. DP-RRT ALGORITHM

To address the MPPC problem for anthropomorphic manipulators, the DP-RRT algorithm is proposed to directly search on the constraint manifolds for finding a collision-free path. The proposed algorithm is based on a combined Direct Projection method and the Rapidly-exploring Random Tree (RRT) algorithm, which uses the RRT algorithm to explore the ambient C-space by growing tree branches, and employs the Direct Projection method to project the tree branches onto the constraint manifolds for constructing a constraint-satisfying path. Besides, avoiding joint limits and singularity, as well as the smoothness of the end-effector and the joints trajectory are also considered in this work.

A. ALGORITHM OVERVIEW

The DP-RRT algorithm, shown in Algorithm 2, starts with a space-filling tree (V, E) rooted at an initial node q_{init}, a vertex set $V_{pre\&cur}$ for storing nodes of the previous and current constraint manifolds, and an empty vertex set $V_{current}$ for storing nodes of the current constraint manifolds. Please note that the vertex sets $V_{pre\&cur}$ and $V_{current}$ are specially designed for cases with a series of constraint manifolds like the MPEP problem.

For the MPOC problem with fixed orientation constraints, the planning process is conducted on the orientation constraint manifolds from start to finish, thus the number of constraint manifolds Num is one (Line 2 of Algorithm 2). While the number of constraint manifolds is equal to the number of way-points N for the MPEP problem (Line 4). Then, the proposed DP-RRT algorithm uses $V_{pre\&cur}$ to expand on the current constraint manifolds iteratively. In each iteration, the new nodes q_{nodes} and the branches $\delta(q_{near}, q_{nodes})$ are directly projected onto the current constraint manifolds by the Direct Projection method. There will be multiple nodes and branches satisfying the constraints, but we only choose the one that satisfies the limits of joint range, singularity and the "shortest trajectory in joint space". Thus the function *SelectOptimum*() is proposed to select the optimal node q_{new} and the branch $\delta(q_{near}, q_{new})$. Finally, in Line 14, a collision detector is employed to check whether q_{new} and $\delta(q_{near}, q_{new})$ are collision-free by incremental checking the projected intermediation nodes. If the path $\delta(q_{near}, q_{new})$ is collision-free, q_{new} will be added to V, $V_{pre\&cur}$ and $V_{current}$, and then $\delta(q_{near}, q_{new})$ will be added to E in Lines 15~17. In Line 19, the above loop program iterates until the goal configuration q_{goal} is reached.

For the MPEP problem, we can complete the motion from the previous constraint manifolds to the current through the above steps. Then $V_{pre\&cur}$ will replace $V_{current}$ as the data structure to search the nearest node of the next loop while $V_{current}$ will be emptied and prepared for adding the new nodes of the new current constraint manifolds. The above loop program is conducted N times as way-points $P(t_i)$ goes by until the goal region is reached.

The complexity of the above DP-RRT algorithm is analyzed by the time complexity and the space complexity. The time complexity can be calculated by the sum of the time that all nodes are added to the tree. As the operations of sampling, steer, direct projection, select optimal node and collision checking can be done in linear time for n iterations (Lines 9-17 of Algorithm 2), the complexity can be represented as $\mathcal{O}(n)$. The complexity of selecting the nearest node depends on the number of vertices already in the tree, thus it can be calculated as $\mathcal{O}(n^2 - n) \approx \mathcal{O}(n^2)$ [31]. It can be seen that the time complexity of the DP-RRT algorithm is bounded by $\mathcal{O}(Num{*}n^2)$, where Num is the number of constraint manifolds. The space complexity is defined as the amount of memory used, which can be represented by the number nodes in the tree. Thus, the space complexity of the DP-RRT algorithm is $\mathcal{O}(Num{*}n)$.

B. GENERIC COMPONENTS

The DP-RRT algorithm is typically composed of the following generic components, including the Direct Projection of the Space-filling Trees, the Selection of the Optimal Node, the Collision Checking and the Vertex Sets.

1) DIRECT PROJECTION OF THE SPACE-FILLING TREES

Direct Projection of the Space-filling Trees is the core of the DP-RRT algorithm, shown in Lines 8∼12 of Algorithm 2. It starts with the RRT algorithm in C-space and uses the Direct Projection method to directly project the new C-space configuration (node) $\boldsymbol{q}_s$ and branch $\boldsymbol{\delta}(\boldsymbol{q}_{near}, \boldsymbol{q}_s)$ onto the constraint manifolds. Specifically, in Lines 9 and 10, a random node $\boldsymbol{q}_{rand}$ is sampled in C-space and its nearest node $\boldsymbol{q}_{near}$ is found in $V_{pre\&cur}$. An attempt $Steer()$ is made to generate a new node $\boldsymbol{q}_s$ and a straight path $\boldsymbol{\delta}(\boldsymbol{q}_{near}, \boldsymbol{q}_s)$ in C-space by moving a step-size $\boldsymbol{\epsilon}$ in the straight line from $\boldsymbol{q}_{near}$ toward $\boldsymbol{q}_{rand}$. It is usually that $\boldsymbol{q}_s$ and $\boldsymbol{\delta}(\boldsymbol{q}_{near}, \boldsymbol{q}_s)$ are not on the constraint manifolds. Then, in Line 12, the C-space $\boldsymbol{q}_s$ and $\boldsymbol{\delta}(\boldsymbol{q}_{near}, \boldsymbol{q}_s)$ are projected onto the constraint manifolds directly by the Direct Projection method (Algorithm 1), thus the new nodes $\boldsymbol{q}_s$ and the branches $\boldsymbol{\delta}(\boldsymbol{q}_{near}, \boldsymbol{q}_{nodes})$ can be generated on the constraint manifolds. It should be noted that as the intermediate nodes of the branch $\boldsymbol{\delta}(\boldsymbol{q}_{near}, \boldsymbol{q}_s)$ are projected onto the constraint manifolds by the Direct Projection method (Figure 5), the branches $\boldsymbol{\delta}(\boldsymbol{q}_{near}, \boldsymbol{q}_{nodes})$ can satisfy the pose constraints with high accuracy. This is quite different from [10], [26], where the branches between the new and nearest configurations are generated by the linear interpolation method.

2) SELECTION OF THE OPTIMAL NODE

Due to multiple solutions of the analytic IK, multiple nodes $\boldsymbol{q}_{nodes}$ on the constraint manifolds will be projected from $\boldsymbol{q}_s$ generated by the RRT algorithm (Line 12 of Algorithm 2). So firstly we need to remove the nodes that near the singularity or outside the joint limits. For the anthropomorphic manipulators with spherical wrists, the singularity can be decoupled as the main-arm singularity and wrist singularity [32]. The main-arm singularity occurs only when $q_4 = 0$ or $q_4 = \pi$ and the wrist singularity happens only when $q_6 = 0$ or $q_6 = \pi$. In Line 13 of Algorithm 2, if all the possible new nodes are removed, $\boldsymbol{q}_{nodes}$ will be an empty set.

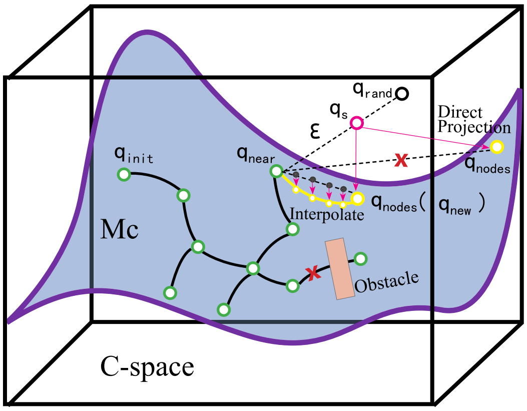


FIGURE 5. A sketch of the DP-RRT algorithm. The whole process generates random branches $\delta(q_{near}, q_s)$ iteratively in C-space and the random branches will be projected onto the constraint manifolds by Direct Projection method (red solid arrow). Then, the optical new node q_{new} and branch (yellow curve) will be retained while other nodes and branches will be eliminated. Finally, collision detection is carried out to determine whether the new node and branch should be added to the space-filling tree.

The other problem is how to select the new node when there are multiple nodes in $\boldsymbol{q}_{nodes}$. According to the differential kinematics of robot manipulators, differential motion of the robot's joints will lead to differential motion of its end-effector. That is to say, the differential motion of the robot's end-effector can be caused by the differential motion of its joints, but it may result from the jumping motion of the joints. Considering that the joint angles between the adjacent nodes should not change dramatically, we use the criterion of "shortest trajectory in joint space" to select the optimal node $\boldsymbol{q}_{new}$ from $\boldsymbol{q}_{nodes}$. Thus, an Euclidean distance of joint angles is adopted as the criteria of choose. As 2π radian yields the same rotation as 0 radian for continuous joint (identification), the Euclidean distance $d(\boldsymbol{q}_{near}, \boldsymbol{q}_{new})$ between vectors $\boldsymbol{q}_{near}$ and $\boldsymbol{q}_{new}$ is defined as:

$$d(\boldsymbol{q}_{near}, \boldsymbol{q}_{new}) = \left\| \sum_{k=1}^{7} \Delta^2(q_{near}^k, q_{new}^k) \right\|, \tag{14}$$

where q_{near}^k, q_{new}^k are the kth element of $\boldsymbol{q}_{near}$ and $\boldsymbol{q}_{new}$ respectively. $\Delta(q_{near}^k, q_{new}^k) = min\{|q_{near}^k - q_{new}^k|, 2\pi - |q_{near}^k - q_{new}^k|\}$ is the $\mathbb{S}^1$ metric. We will discard the IK solutions that lead to joint angles jump through this criteria. Therefore, the joint angles between the nearest and new configurations would not jump and the smoothness of the joint trajectories can be achieved. According to the differential kinematics of robot manipulators, the smoothness of the end-effector trajectory can be ensured.

3) COLLISION CHECKING

In the RRT planner, a collision detector is employed to check whether the straight path between the new node and the nearest node are collision-free by incremental checking at some interval from one end to the other along the path. For the DP-RRT algorithm, the projected path $\boldsymbol{\delta}(\boldsymbol{q}_{near}, \boldsymbol{q}_{new})$ in joint space is not a straight path. Instead of checking the

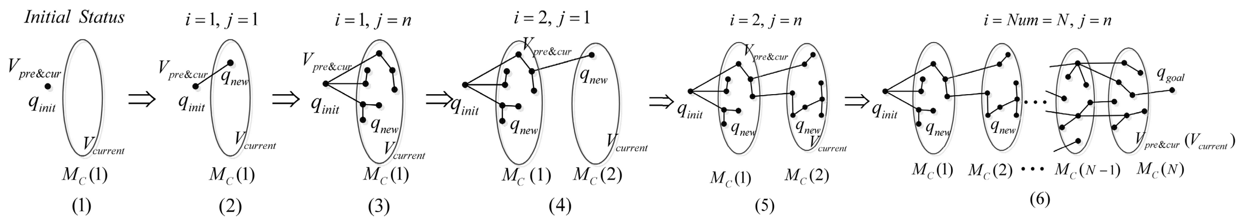


FIGURE 6. The program flow sketch map of the tree expansion on the position constraint manifolds. $V_{pre\&cur}$ is a moving tree that moves sequentially from $\{q_{init}\}$ to $\mathcal{M}_C(N)$. $V_{current}$ is a temporary tree that prepares the new nodes for $V_{pre\&cur}$.

straight path between $\boldsymbol{q}_{near}$ and $\boldsymbol{q}_{new}$, we incrementally check the straight segments between the projected intermediation nodes and the $\delta(\boldsymbol{q}_{near}, \boldsymbol{q}_{new})$ is collision-free only when all the segments are collision-free.

4) VERTEX SETS

There is only one vertex set V in RRT algorithm, which is used to store the vertices of random space-filling trees. However, for the cases of connecting a series of constraint manifolds by sequence in the MPEP problem, the current constraint manifolds $V_{current}$ and the current and previous constraint manifolds $V_{pre\&cur}$ are needed. $V_{current}$ stores the vertices corresponding to the current pose and $V_{pre\&cur}$ stores the vertices corresponding to the current pose and the previous pose. Thus, two different connection modes can be obtained, i.e. the *forward motion* from the previous constraint manifolds to the current ones, and the *self motion* connecting two vertices on the same constraint manifolds.

As shown in Algorithm 2, the initial node is located at the previous constraint manifolds $V_{pre\&cur} = \{\boldsymbol{q}_{init}\}$, and the space to be sampled is the current constraint manifolds $V_{current} = \emptyset$. For the fixed orientation tasks, the planning process is conduct on the orientation constraint manifolds from beginning to end (Lines 6∼22), thus the number of constraint manifolds *Num* is one. For MPEP problem, the number of constraint manifolds is equal to the number of way-points N (Line 4) and the planned path can be the self motion or the forward motion. As depicted in Figure 6 (1)-(3), the DP-RRT algorithm starts from an initial node $\boldsymbol{q}_{init}$ in the previous constraint manifolds and repeatedly generates new nodes $\boldsymbol{q}_{new}$ on the current constraint manifolds $M_c(1)$ until n new nodes have been added to $V_{current}$. Then $V_{pre\&cur}$ will replace $V_{current}$ as the data structure to search the nearest node of the next loop while $V_{current}$ will be emptied and prepared for adding the new nodes of the new current constraint manifolds. This process corresponds to Figure 6 (4), (5) and Lines 20, 21 of Algorithm 2. As presented in Figure 6 (6), the above loop program is conducted N times as way-point $P(t_i)$ goes by until the goal region is reached.

V. PLANNING EXPERIMENTS

To demonstrate the validity of the proposed algorithm, three sets of simulation experiments are performed by the DP-RRT planner on the humanoid PR2 (Willow Garage) simulation robot in the Robot Operating System (ROS) [33].

TABLE 1. Simulation results of case 1 (20 runs).

	DP-RRT	CBiRRT	TS-RRT
Avg.Time (s)	0.446	1.453	0.897
Min.Time (s)	0.117	0.696	0.721
Max.Time (s)	0.927	2.421	1.041
Avg.Nodes	191	531	177
Avg.Error (rad)	$6.58*10^{-7}$	$1.37*10^{-5}$	$8.03*10^{-4}$
success rate	100%	100%	100%

The ROS is a flexible framework for writing robot software and creating complex and robust robot behavior across a wide variety of robotic platforms, which has been widely used and considered as the main development framework in robotics. The CBiRRT planner [9], the RRT-connect planner [34], the Control_Based planner [21], the Task Space RRT (TS-RRT) planner [35] and the Global Redundancy Resolution PRM (GRR-PRM) planner [36] are used as the comparative methods in these simulation experiments, which are typically adopted to deal with such constraint tasks. All the algorithms are run on ROS MoveIt! simulator [37] with an open-source Open Motion Planning Library (OMPL) [38].

A. EXPERIMENTAL PLATFORM

The simulation platform of the PR2 is one of the representative humanoid robots, which has been widely used to verify the function and performance of the anthropomorphic robot [12], [17], [39]. We consider the MPPC problems for the 7-DOF anthropomorphic manipulator of the PR2 platform. The main-arm joints of the PR2 manipulator consist of four joints: shoulder pan joint (q_1), shoulder lift joint (q_2), upper arm roll joint (q_3) and elbow flex joint (q_4), which determine the position movement of the robot's wrist. While the orientation of the robot's wrist can be controlled by the wrist joints consisting of forearm roll joint (q_5), wrist flex joint (q_6) and wrist roll joint (q_7). The shoulder pan joint (q_1) is selected as the redundant joint for the analytical IK solution of the main-arm configuration.

B. EXPERIMENTS AND RESULTS

We evaluate the performance of our algorithm by applying it to three typical pose-constrained cases: end-effector orientation constraint, narrow passage constraint and end-effector path constraint. Since all the planners are randomized, 20 times repetitive experiments are performed for each planner and the statistic data of comparative experiments are given in Table 1 for Case 1, Table 2 for Case 2 and Table 3 for Case 3. The first column in Table 1, 2 and 3

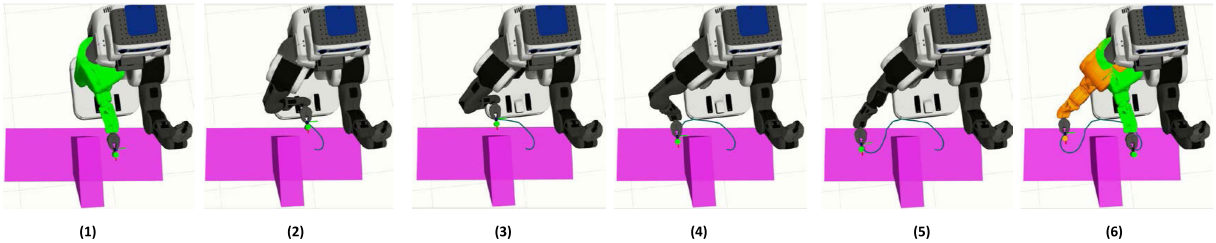


FIGURE 7. The PR2 robot moves a cup from the green pose to the yellow pose in a cluttered environment while subjecting to upright constraints by using the DP-RRT planner. Green: the start configuration. Yellow: the goal configuration. Red: the obstacles (a table, a baffle on the table). Blue: the trajectory of the end-effector.

TABLE 2. Simulation results of Case 2 (20 runs).

	DP-RRT	RRT-connect	TS-RRT
Avg.Time (s)	0.409	0.805	0.611
Min.Time (s)	0.111	0.378	0.310
Max.Time (s)	0.853	1.532	1.105
Avg.Nodes	151	362	160
success rate	100%	100%	100%

TABLE 3. Simulation results of Case 3 (20 runs).

	DP-RRT		Control_Based	GRR-PRM
	N=72	N=180	N=72	N=72
Avg.Time (s)	2.336	8.23	9.1	13.8
Min.Time (s)	1.966	6.02	6.8	10.4
Max.Time (s)	2.562	11.76	12.4	19.8
Avg.Nodes	180	450	287	670
Avg.Error (m)	$1.2*10^{-4}$	$4.9*10^{-6}$	$1.7*10^{-5}$	$2.3*10^{-5}$
success rate	100%	100%	100%	100%

shows several evaluating indicators including average planning time (Avg.Time), minimum planning time (Min.Time), maximum planning time (Max.Time), average number of nodes (Avg.Nodes), path accuracy (Avg.Error) and success rate. The algorithms based on different strategies are presented in the first row.

1) CASE 1: END-EFFECTOR ORIENTATION CONSTRAINT

When manipulating objects, it is often necessary to maintain its orientation fixed. For example, the robot is asked to pick a cup of water from one place to another while keep it from tipping or spilling all along the motion [12], [40]. In this case, orientation need to be specified in the planned trajectory. The experiments in this paper only allow for enforcing an exact value of the Euler angles, i.e., the yaw angle γ, pitch angle β and roll angle α are set to zero. Thus, the constraint can be represented as

$$C^r = \begin{bmatrix} x(t) \in (-\infty, +\infty) \\ y(t) \in (-\infty, +\infty) \\ z(t) \in (-\infty, +\infty) \\ \alpha(t) = 0 \\ \beta(t) = 0 \\ \gamma(t) = 0 \end{bmatrix}. \tag{15}$$

The first simulation experiment is represented in Figure 7, the PR2 robot has to move a cup vertically from the green pose to the yellow pose in a cluttered environment. According to the task, both the starting pose (green pose) and the target pose (yellow pose) must satisfy the orientation constraint with Euler angles $\alpha = \beta = \gamma = 0$ radians. In order to compare the performance of the proposed planner, the CBiRRT planner and the TS-RRT planner are conducted for the same scenario. The CBiRRT planner is a typical projection-based planning method [16] that has been widely used for pose-constrained tasks. While the TS-RRT is a popular task-space planning method [35] that samples and expands nodes in the low-dimensional task space and thus can find paths with fewer nodes than the C-space planners.

According to the averaged results of 20 times repetitive experiments in Table 1, all the planners can complete the task on the PR2 simulation robot with respect to success rate. We can find that the proposed DP-RRT planner is obviously superior to the CBiRRT planner in the indicators of planning time, number of nodes and path accuracy. Besides, it consumes less computing time and gets higher precision than the TS-RRT planner, while there is little difference in the number of extended nodes between the two approaches.

The advantage in planning time can be attributed to that the DP-RRT planner uses the fast analytical IK technique to sample configurations on the orientation constraint manifolds directly rather than the numerical iteration methods. And the high accuracy of the DP-RRT planner owes to the accuracy of its intermediate nodes, which are all obtained by the direct projection method rather than the linear interpolation methods. A typical result of the snapshots from the execution of the DP-RRT planner is shown in Figure 7. The corresponding Euler angles of the robot's end-effector are fixed to zero with high accuracy as shown in Figure 8(a) and the corresponding joint position trajectories are smooth as shown in Figure 8(b).

2) CASE 2: NARROW PASSAGE CONSTRAINT

Narrow passage constraints are common in our daily life and industrial environment, e.g., the PR2 robot is assigned to take a long T-shaped broom through a hallway and then into a room [41]. Sampling-based planners are notoriously challenging at handling such problems efficiently as there are some special restrictions on the orientation and/or position of robot's end-effector within the narrow passage. Indeed, narrow passages represented in C-space is a small region but critical to the connectivity of the free space [17]. Thus, the probability of randomly sampling configurations in the narrow passage space is extremely low by the uniform

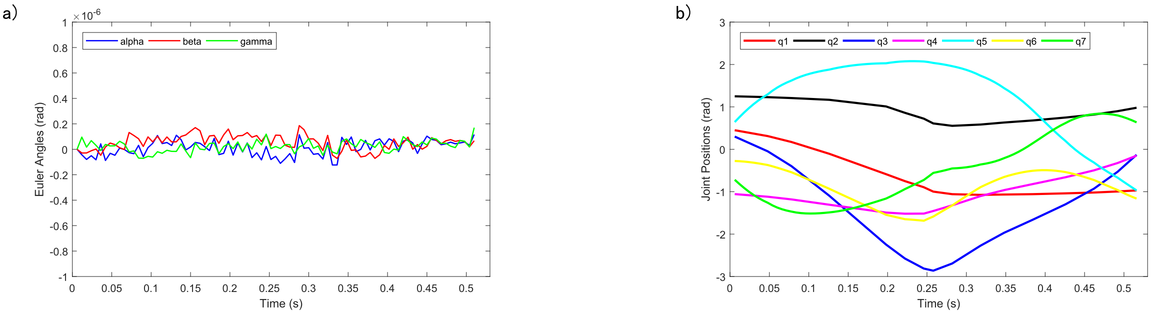


FIGURE 8. (*a*) The trajectory of Euler angles produced by the DP-RRT algorithm are fixed to zero with high accuracy. (*b*) The smooth joint position trajectories can be obtained by the DP-RRT algorithm.

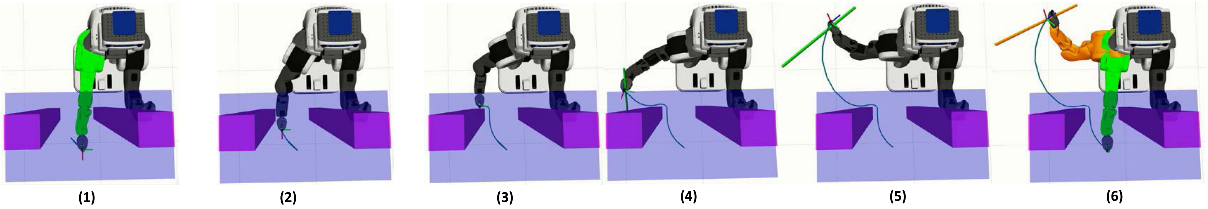


FIGURE 9. The PR2 robot operates a long stick from the green pose to the yellow pose through a narrow passage by the DP-RRT planner. Green: the start configuration. Yellow: the goal configuration. Red: the obstacles. Light Blue: the narrow passage space. Blue: The trajectory of the end-effector. (1), (2) and (3) are snapshots of using the DP-RRT algorithm, where the robot's end-effector is limited in the narrow passage space. (4) is a snapshot of the switching point between the DP-RRT and the RRT planner. (5) and (6) are snapshots of using the RRT algorithm, where the end-effector of robot is outside of the narrow passage space.

sampling strategy. Solutions for the narrow passage problem are often to generate multiple trees simultaneously like RRT-connect planner [34]. Another solution is to boost the sampling density inside narrow passage and improve the connectivity of roadmaps by the task space planner [35]. In other words, the orientation and/or position of the end-effector derived from the random sampling should be constrained within the narrow passage space.

A typical scenario is portrayed in Figure 9, the PR2 manipulator need to move a long stick from the green pose to the yellow pose through a narrow passage formed by two pillars. The narrow passage prevent the stick to go through it, unless an appropriate orientation is set [42]. Here the narrow passage region is a rectangular space in light blue: S_{np}=$\{(x, y, z) \mid x \in [0.2, 0.8],\ y \in [-0.8, 0.4],\ z \in [0, 2]$ (in meters), whose space range depends on the constraint requirements of the task. Euler angles (α, β, γ) of the end-effector are set to zero within the narrow passage space according to the environmental prior information. Thus, the constraint can be represented as

$$C^r = \begin{bmatrix} x(t) \in [0.2, 0.8] \\ y(t) \in [-0.8, 0.4] \\ z(t) \in [0, 2] \\ \alpha(t) = 0 \\ \beta(t) = 0 \\ \gamma(t) = 0 \end{bmatrix}. \tag{16}$$

To evaluate the DP-RRT algorithm, the comparative simulation experiment is conducted by two typical planners: the RRT-connect planner and the TS-RRT planner. To measure the efficiency improvement, we compare the Avg.Time, Min.Time, Max.Time and Avg.Nodes of the planners, as shown in Table 2.

By comparing these date of planning time, the DP-RRT planner for narrow passage constraint task is superior to the RRT-connect planner and the TS-RRT planner obviously. A typical result of the snapshots from the execution of the DP-RRT planner is shown in Figure 9. The corresponding Euler angles of the robot's end-effector and the joint positions are shown in Figure 10(*a*) and Figure 10(*b*) respectively.

3) CASE 3: END-EFFECTOR PATH CONSTRAINT

In many applications, the manipulator is assigned to trace a given end-effector path while avoiding collisions. Door opening is an evident example: the robot need grasp the door handle and rotate a given circular path around the door-hinge to open the door [43]. There are many other applications in industry like painting [23]–[25], welding [44] and so on.

We tested our algorithm on a typical scenario shown in Figure 11. The end-effector of PR2 manipulator is arranged to follow a circle path $P(t)$ with a radius of 0.15 meters through a small window (in red) and returns to its original pose when finished. The Euler angles of the end-effector is set to α=β=γ=0. Thus, the constraint can be represented as

$$C^r = \begin{bmatrix} x(t) = P_x(t_i) \\ y(t) = P_y(t_i) \\ z(t) = P_z(t_i) \\ \alpha(t) = 0 \\ \beta(t) = 0 \\ \gamma(t) = 0 \end{bmatrix}. \tag{17}$$

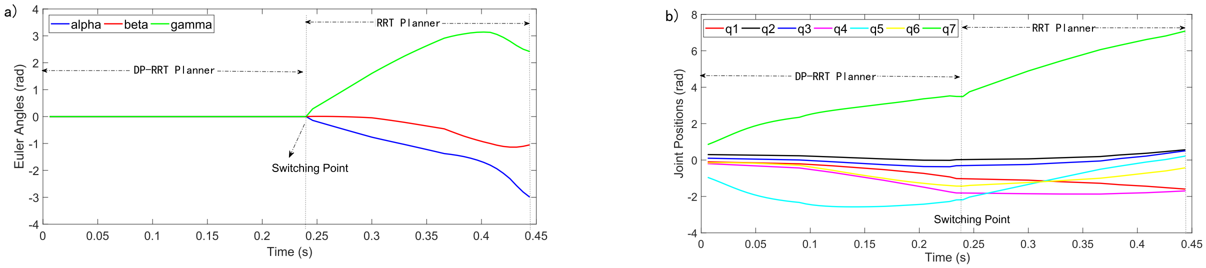


FIGURE 10. (a) The Euler angle trajectory of the whole motion is smooth by using the DP-RRT algorithm within narrow passage space, no jumping occurs even at the switching point. In the beginning, Euler angles of robot's end-effector are set to zero in the narrow passage space by using the DP-RRT algorithm. When out of the space, the RRT planner is adopted. The output joint position trajectories are smooth all along the motion and without jump even at the switching point. (b) The joint position trajectories are smooth throughout the motion and without jumping even at the switching point.

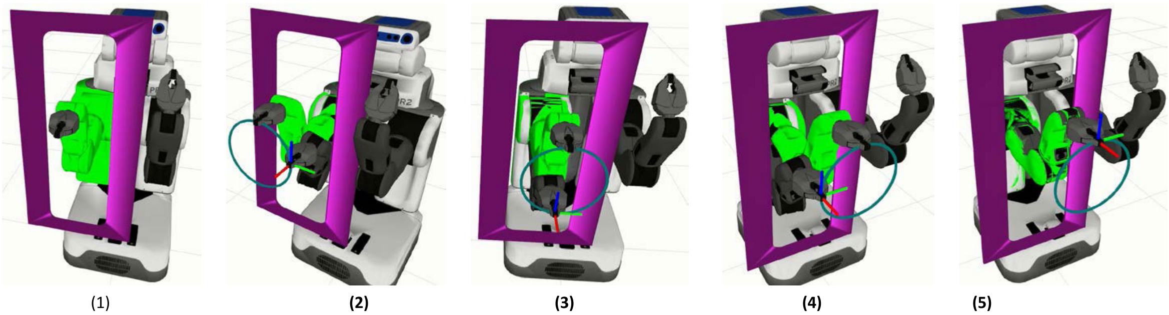


FIGURE 11. The PR2 manipulator is assigned to track a circle path while avoiding collisions by the DP-RRT planner. Red: the obstacle. Blue: the circle path.

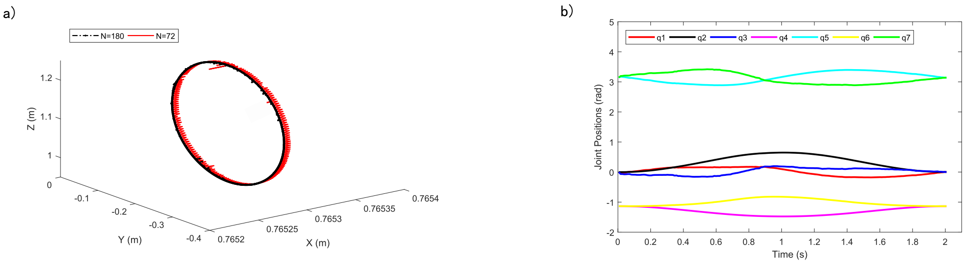


FIGURE 12. (a) The two trajectories of the end-effector are produced by the DP-RRT planner with N=72 and N=180 respectively, where the path accuracy can be greatly improved with the increase of the way-points. (b) The joint position trajectories are smooth throughout the motion by the DP-RRT planner with N=72.

To compare the performance of the proposed planner, the Control_Based approach and the GRR-PRM planner are conducted for the same scenario. The Control_Based approach is a typical kinematic control method [21] that uses the Jacobian pseudoinverse technique to track a given path. And the GRR-PRM planner focuses on finding the global redundancy resolution [36] for tracking the task path.

In this case, the Control_Based approach with $N = 72$, the GRR-PRM planner with $N = 72$ and the DP-RRT planner with $N = 72$ and $N = 180$ are applied and the performance results are illustrated in Table 3. We can find that with the same number of way-points N=72, the DP-RRT planner gives better results in terms of planning time but lose superiority in path precision compared with the Control_Based planner and the GRR-PRM planner. By increasing the way-points (N), the tracking error can be reduced, but the planning time and the number of position constraint manifolds will increase. In order to get higher tracking accuracy, we increase the number of the way-points with $N = 180$ and the results indicate that the DP-RRT planner with $N = 180$ outperforms the Control_Based planner and the GRR-PRM planner in terms of tracking error, while the average running time is about 9 s for both the DP-RRT planner and the Control_Based planner. The higher tracking accuracy can be achieved because the number of intermediate nodes increases significantly with the increase of way-points. And the intermediate nodes are all obtained by the Direct Projection method. A typical result of the snapshots from the execution of the DP-RRT planner with $N = 72$ is shown in Figure 11. The corresponding path trajectory of the robot's end-effector and joint positions are shown in Figure 12(a) and Figure 12(b), respectively.

VI. CONCLUSION

In this paper, we have presented an efficient planning algorithm to address the MPPC problem for anthropomorphic manipulators. The algorithm is to combine the RRT algorithm with the Direct Projection method, which uses the RRT algorithm to explore the ambient space by growing tree branches, and then directly projects the tree branches onto the constraint manifolds for constructing a constraint-satisfying path. As the Direct Projection method based on the analytic IK solver is applied to calculate constraint-satisfying configurations, our approach can be more efficient and accurate than the numerical methods. Three different pose-constrained cases were performed with a simulated PR2 robot in ROS MoveIt! to verify the proposed DP-RRT algorithm compared with several state-of-the-art planners planners. The results show that our approach can provide significant benefits for motion planning with pose constraints. The direct projection method also has been successfully applied to the closed-chain motion planning with orientation constraints for a class of dual-arm robots in our work [45]. It is shown that the proposed method is a widely-fit method for constrained motion planning of the anthropomorphic manipulators.

REFERENCES

[1] J. H. Reif, "Complexity of the Mover's problem and generalizations," in *Proc. 20th Annu. Symp. Found. Comput. Sci. (SFCS)*, Oct. 1979, pp. 421–427.

[2] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Autom.*, vol. 12, no. 4, pp. 566–580, Aug. 1996.

[3] S. M. Lavalle and J. J. Kuffner, "Rapidly-exploring random trees: Progress and prospects," in *Algorithmic and Computational Robotics New Directions*, B. R. Donald, K. Lynch, and D. Rus, Eds. Wellesley, MA, USA: Wellesley Press, 2001, pp. 293–308.

[4] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, Jun. 2011.

[5] H. Zhang, Y. Wang, J. Zheng, and J. Yu, "Path planning of industrial robot based on improved RRT algorithm in complex environments," *IEEE Access*, vol. 6, pp. 53296–53306, 2018.

[6] Z. Kingston, M. Moll, and L. E. Kavraki, "Sampling-based methods for motion planning with constraints," *Annu. Rev. Control Robot. Autom. Syst.*, vol. 1, no. 1, pp. 159–185, May 2018.

[7] Z. Yao and K. Gupta, "Path planning with general end-effector constraints," *Robot. Autom. Syst.*, vol. 55, no. 4, pp. 316–327, Apr. 2007.

[8] C. Suh, T. T. Um, B. Kim, H. Noh, M. Kim, and F. C. Park, "Tangent space RRT: A randomized planning algorithm on constraint manifolds," in *Proc. IEEE Int. Conf. Robot. Autom.*, Shanghai, China, May 2011, pp. 4968–4973.

[9] D. Berenson, S. S. Srinivasa, D. Ferguson, and J. J. Kuffner, "Manipulation planning on constraint manifolds," in *Proc. IEEE Int. Conf. Robot. Autom.*, Kobe, Japan, May 2009, pp. 625–632.

[10] G. Oriolo, M. Ottavi, and M. Vendittelli, "Probabilistic motion planning for redundant robots along given end-effector paths," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Lausanne, Switzerland, Jun. 2003, pp. 1657–1662.

[11] C. Milton and D. Jennings, "Path planning for a manipulator with constraints on orientation," *J. Intell. Robot. Syst.*, vol. 11, nos. 1–2, pp. 67–77, Mar. 1994.

[12] B. Cohen, S. Chitta, and M. Likhachev, "Search-based planning for dual-arm manipulation with upright orientation constraints," in *Proc. IEEE Int. Conf. Robot. Autom.*, Saint Paul, MN, USA, May 2012, pp. 3784–3790.

[13] S. Lavalle, J. Yakey, and L. Kavraki, "A probabilistic roadmap approach for systems with closed kinematic chains," in *Proc. IEEE Int. Conf. Robot. Autom.*, Detroit, MI, USA, May 1999, pp. 1671–1676.

[14] J. Yakey, S. Lavalle, and L. Kavraki, "Randomized path planning for linkages with closed kinematic chains," *IEEE Trans. Robot. Autom.*, vol. 17, no. 6, pp. 951–958, Dec. 2001.

[15] M. Stilman, "Task constrained motion planning in robot joint space," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Diego, CA, USA, Nov. 2007, pp. 3074–3081.

[16] D. Berenson, S. Srinivasa, and J. Kuffner, "Task space regions: A framework for pose-constrained manipulation planning," *Int. J. Robot. Res.*, vol. 30, no. 12, pp. 1435–1460, Oct. 2011.

[17] L. Jaillet and J. M. Porta, "Path planning under kinematic constraints by rapidly exploring manifolds," *IEEE Trans. Robot.*, vol. 29, no. 1, pp. 105–117, Feb. 2013.

[18] B. Kim, T. T. Um, C. Suh, and F. C. Park, "Tangent bundle RRT: A randomized algorithm for constrained motion planning," *Robotica*, vol. 34, no. 1, pp. 202–225, Jan. 2016.

[19] S. Seereeram and J. Wen, "A global approach to path planning for redundant manipulators," *IEEE Trans. Robot. Autom.*, vol. 11, no. 1, pp. 152–160, Feb. 1995.

[20] Z. Jia, S. Chen, X. Qu, P. Zhang, and N. Zhong, "Velocity-level tri-criteria optimization scheme for different complex path tracking of redundant manipulators," *IEEE Access*, vol. 7, pp. 64289–64296, 2019.

[21] G. Oriolo and M. Vendittelli, "A control-based approach to task-constrained motion planning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, St. Louis, MO, USA, Oct. 2009, pp. 297–302.

[22] A. H. Quispe and M. Stilman, "Deterministic motion planning for redundant robots along end-effector paths," in *Proc. 12th IEEE-RAS Int. Conf. Humanoid Robots*, Osaka, Japan, Nov. 2012, pp. 785–790.

[23] G. Oriolo, M. Cefalo, and M. Vendittelli, "Repeatable motion planning for redundant robots over cyclic tasks," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1170–1183, Oct. 2017.

[24] Z. Jia, X. Qu, S. Chen, N. Zhong, P. Zhang, and F. Ouyang, "Acceleration-level multi-criteria optimization for remedying joint-angle drift of redundant manipulators on complex path tracking," *IEEE Access*, vol. 7, pp. 95716–95724, 2019.

[25] D. Guo and Y. Zhang, "Acceleration-level inequality-based man scheme for obstacle avoidance of redundant robot manipulators," *IEEE Trans. Ind. Electron.*, vol. 61, no. 12, pp. 6903–6914, Dec. 2014.

[26] G. Oriolo and C. Mongillo, "Motion planning for mobile manipulators along given end-effector paths," in *Proc. IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, Jan. 2005, pp. 2154–2160.

[27] R. P. Paul and H. Zhang, "Computationally efficient kinematics for manipulators with spherical wrists based on the homogeneous transformation representation," *Int. J. Robot. Res.*, vol. 5, no. 2, pp. 32–44, Jun. 1986.

[28] K. Gochev, V. Narayanan, B. Cohen, A. Safonova, and M. Likhachev, "Motion planning for robotic manipulators with independent wrist joints," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Hong Kong, Jun. 2014, pp. 461–468.

[29] M. Tarokh, K. Keerthi, and M. Lee, "Classification and characterization of inverse kinematics solutions for anthropomorphic manipulators," *Robot. Auto. Syst.*, vol. 58, no. 1, pp. 115–120, Jan. 2010.

[30] H. Asada and J. J. E. Slotine, *Robot Analysis and Control*. New York, NY, USA: Wiley, 1986.

[31] M. Svenstrup, T. Bak, and H. J. Andersen, "Minimising computational complexity of the RRT algorithm a practical approach," in *Proc. IEEE Int. Conf. Robot. Autom.*, Shanghai, China, May 2011, pp. 5602–5607.

[32] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*, New York, NY, USA: Springer, 2008.

[33] M. Quigley, K. Conley, and B. Gerkey, "ROS: An open-source robot operating system," in *Proc. ICRA Workshop Open Source Softw.*, Kobe, Japan, May 2009, pp. 1–6.

[34] J. Kuffner and S. Lavalle, "RRT-connect: An efficient approach to single-query path planning," in *Proc. IEEE Int. Conf. Robot. Automat. Symp. Process.*, San Francisco, CA, USA, Apr. 2002, pp. 995–1001.

[35] A. Shkolnik and R. Tedrake, "Path planning in 1000+ dimensions using a task-space Voronoi bias," in *Proc. IEEE Int. Conf. Robot. Autom.*, Kobe, Japan May 2009, pp. 2061–2067.

[36] K. Hauser and S. Emmons, "Global redundancy resolution via continuous pseudoinversion of the forward kinematic map," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 3, pp. 932–944, Jul. 2018.

[37] S. Chitta, I. Sucan, and S. Cousins, "MoveIt! [ROS topics]," *IEEE Robot. Automat. Mag.*, vol. 19, no. 1, pp. 18–19, Mar. 2012.

[38] I. A. Sucan, M. Moll, and L. E. Kavraki, "The open motion planning library," *IEEE Robot. Automat. Mag.*, vol. 19, no. 4, pp. 72–82, Dec. 2012.

[39] C. Phillips-Grafflin, N. Alunni, H. B. Suay, J. Mainprice, D. Lofaro, D. Berenson, S. Chernova, R. W. Lindeman, and P. Oh, "Toward a user-guided manipulation framework for high-DOF robots with limited communication," *Int. Serv. Robot.*, vol. 7, no. 3, pp. 121–131, Jul. 2014.

[40] T. Kunz and M. Stilman, "Manipulation planning with soft task constraints," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Vilamoura, Portugal, Oct. 2012, pp. 1937–1942.

[41] F. Islam, V. Narayanan, and M. Likhachev, "Dynamic multi-heuristic A," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Seattle, WA, USA, May 2015, pp. 2376–2382.

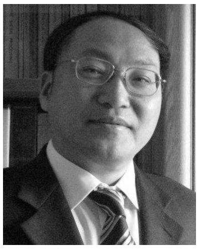
[42] K. Gochev, A. Safonova, and M. Likhachev, "Planning with adaptive dimensionality for mobile manipulation," in *Proc. IEEE Int. Conf. Robot. Autom.*, Saint Paul, MN, USA, May 2012, pp. 2944–2951.

[43] D. Aarno, F. Lingelbach, and D. Kragic, "Constrained path planning and task-consistent path adaptation for mobile manipulators," in *Proc. 12th Int. Conf. Adv. Robot.*, Seattle, WA, USA, Oct. 2006, pp. 268–273.

[44] A. L. Olsen and H. G. Petersen, "Motion planning for gantry mounted manipulators: A ship-welding application example," in *Proc. IEEE Int. Conf. Robot. Autom.*, Roma, Italy, May 2007, pp. 4782–4786.

[45] J. Wang, S. Liu, B. Zhang, and C. Yu, "Inverse kinematics-based motion planning for dual-arm robot with orientation constraints," *Int. J. Adv. Robot. Syst.*, vol. 16, no. 2, Mar. 2019, Art. no. 172988141983685.

JIANGPING WANG (Student Member, IEEE) received the M.S. degree in control engineering from Hangzhou Dianzi University, Hangzhou, China, in 2015, where he is currently pursuing the Ph.D. degree with the School of Automation. His current research interests include motion planning, human–robot interaction, and intelligent robot systems.



SHIRONG LIU (Member, IEEE) received the B.Sc. degree in electrical engineering from Zhejiang University, Hangzhou, China, in 1978, the M.Sc. degree in industrial automation from Chongqing University, Chongqing, China, in 1986, and the Ph.D. degree in system and control from the East China University of Science and Technology, Shanghai, China, in 2000. He is currently a Professor with the School of Automation and the Director of the Institute of Electrical Engineering and Automation, the Head of the Institute of Automation, and the Director of the Intelligent Mobile Robotics Laboratory, Hangzhou Dianzi University, Hangzhou, China. His research interests include intelligent robotics, intelligent control, system modeling, and renewable-energy electric power systems and control.

BOTAO ZHANG received the Ph.D. degree in control engineering from the East China University of Science and Technology, Shanghai, China, in 2012. He is currently an Associate Professor with the School of Automation, Hangzhou Dianzi University, Hangzhou, China. His current research interests include intelligent robot systems and computational intelligence.

QIANG LU (Member, IEEE) received the B.Eng. and Ph.D. degrees in electrical engineering from the East China University of Science and Technology, Shanghai, China, in 2000 and 2007, respectively, and the Ph.D. degree in computer science from Central Queensland University, Rockhampton, QLD, Australia, in 2013. In 2007, he joined Hangzhou Dianzi University, Hangzhou, China, where he is currently a Professor with the School of Automation. His research interests include multirobot systems, nonlinear control, and swarm intelligence.