*Research Article*

INTERNATIONAL JOURNAL OF ADVANCED ROBOTIC SYSTEMS

*International Journal of Advanced Robotic Systems*
March-April 2019: 1–14
© The Author(s) 2019
DOI: 10.1177/1729881419836858
journals.sagepub.com/home/arx
SAGE

# Inverse kinematics-based motion planning for dual-arm robot with orientation constraints

**Jiangping Wang, Shirong Liu, Botao Zhang and Changbin Yu**

**Abstract**

This article proposes an efficient and probabilistic complete planning algorithm to address motion planning problem involving orientation constraints for decoupled dual-arm robots. The algorithm is to combine sampling-based planning method with analytical inverse kinematic calculation, which randomly samples constraint-satisfying configurations on the constraint manifold using the analytical inverse kinematic solver and incrementally connects them to the motion paths in joint space. As the analytical inverse kinematic solver is applied to calculate constraint-satisfying joint configurations, the proposed algorithm is characterized by its efficiency and accuracy. We have demonstrated the effectiveness of our approach on the Willow Garage's PR2 simulation platform by generating trajectory across a wide range of orientation-constrained scenarios for dual-arm manipulation.

**Keywords**

Motion planning, orientation constraints, inverse kinematics, decoupled dual-arm robots

Date received: 20 November 2018; accepted: 16 February 2019

Topic: Robot Manipulation and Control
Topic Editor: Andrey V Savkin
Associate Editor: Jose de Jesus Rubio

## Introduction

To safely perform the manipulation tasks, motion planning algorithms that can generate collision-free motions from an initial to a final configuration in cluttered environments are needed. However, the motion planning problem is known to be PSPACE-hard,[1] which means that the computational complexity will increase dramatically with the increase of a robot's degrees of freedom (DOF). Sampling-based algorithms[2–5] in general are considered the dominant approach in dealing with the motion planning problem of high-DOF robotic manipulators because of efficiency, conveniently handling kinodynamic constraints,[6] and probabilistic completeness. Recently, sampling-based methods[7–24] have also been extended to deal with manipulation planning problem under pose or kinematic closure constraints. However, these planners usually are difficult to solve the dual-arm motion planning involving orientation constraints of the robot's end-effector. Examples of such constraint tasks are ubiquitous in our daily life, for example, moving cups full of water from one place to another while keeping them from spilling all along the motion; manipulating a tray with food or glasses on it without tipping throughout the motion path.

In this article, we focus on the dual-arm motion planning problem with end-effector orientation constraints, that is computing a collision-free path for both arms between an initial state and a goal state while maintaining a specified

School of Automation, Hangzhou Dianzi University, Hangzhou, Zhejiang, China

**Corresponding author:**
Shirong Liu, School of Automation, Hangzhou Dianzi University, Xiasha High Education Zone, Hangzhou, Zhejiang 310018, China.
Email: liushirong@hdu.edu.cn

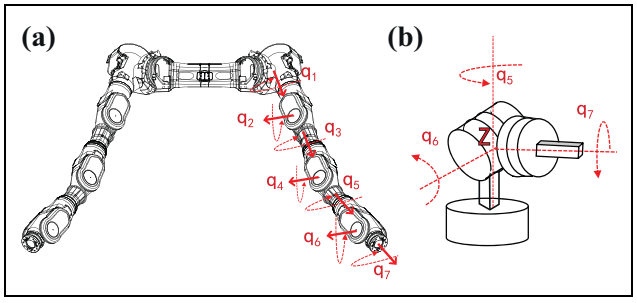


**Figure 1.** (a) A prototype of dual-arm robot with two decoupled manipulators. The left arm configuration consists of the main arm joints ($q_1$, $q_2$, $q_3$, and $q_4$) and the wrist joints ($q_5$, $q_6$, and $q_7$). Right arm has the similar structure. (b) Spherical wrist, three revolute axes intersect at a single point.

orientation throughout the path. In the process of dual-arm cooperation,[25] there may be also closure constraint between the two arms, for example, when two arms simultaneously grasp and manipulate a single object. In general, these problems can be interpreted as finding constraint-satisfying configurations and connecting them to generate valid paths without being trapped in local minima.[7] The main difficulty is that the introduction of constraint satisfaction problem (CSP)[26] poses a significant challenge to the solution of constraint-satisfying configurations, which is an NP-Complete problem. On the one hand, the constraint manifolds (the set of all constraint-satisfied configurations) are zero-measure manifolds[8] embedded in the ambient configuration space (C-space). Therefore, the probability that a randomly sampling configuration in C-space lies on the desired constraint manifolds is extremely low and usually null. On the other hand, due to the nonlinear relationship between configuration parameters derived from orientation and closure constraints, it is complicated to calculate the analytical inverse kinematics (IK) solutions satisfied the constraint conditions.

In this article, we propose an efficient and probabilistically complete planning algorithm called orientation-constrained rapidly exploring random trees (OC-RRT) to address motion planning problem involving orientation constraints for decoupled dual-arm robots. The proposed algorithm is based on a novel combined computing framework of IK solutions and sampling-based planning approach. Although the sampling-based planning algorithm in this work takes the RRT[2] as an example, it is also suitable for other sampling-based planners such as Probabilistic RoadMaps (PRMs)[3] and RRT variants.[4,5] As is the case with human arms, the decoupled dual-arm robots with two decoupled manipulators (Figure 1(a)) are designed to allow the decoupling between position and orientation of the robot wrist,[27–29] which have been widely adopted, such as the DLR's Rollin Justin, Yaskawa Motoman's SDA10D, Willow Garage's PR2, and Rethink's Baxter robot. As the last three joints of the decoupled manipulator have concurrent axes, the position of the wrist represented by $Z$ in Figure 1(b) is determined only by the main arm joints while the orientation can be controlled just by the wrist joints. This special property (decoupling of the positioning and the orientation subtasks) makes it possible to decouple the planning for the main arm joints from the wrist joints.[30] Hence, the proposed algorithm can directly calculate the constraint-satisfying joint configurations relied on the ability of solving the analytical IK. A block diagram describing the workflow of the global system architecture is shown in Figure 2. Note that as the task constraints considered in this article are collision-free and kinematic constraints only, the other constraints, such as force and dynamic constraints, are not taken into account.

The main contributions of this article can be summarized as follows. (1) an IK-based motion planning algorithm is presented for decoupled dual-arm manipulation with orientation constraints, which directly calculates the constraint-satisfying configuration by analytical IK instead of the previously proposed Jacobian pseudo-inverse projection methods. (2) As the adoption of the analytical IK solver, our planner is characterized by efficiency, accuracy, and without iteration modification. (3) We distinguish this work from the closed-chain motion planning and motion planning with orientation constraint, because considering closure constraint and orientation constraint simultaneously is more difficult than just respecting a single constraint. (4) The other constraints such as the task CSP of trajectory between nodes, probabilistic completeness, avoid singularities, and the limits of joint angle are also considered in this work.

The rest of this article is organized as follows. In the second section, a survey of related work of dual-arm motion planning involving orientation constraints is given. Some preliminaries including the classification of dual-arm operation, the representation of end-effector's pose constraint, and the description of constrained motion planning are provided in the third section. After that, the detail of OC-RRT planner is presented in the fourth section. In the fifth section, a wide range of OC scenarios are used to validate the prominent performance of the OC-RRT planner. Finally, conclusions of this study are presented in the sixth section.

## Related work

Before going into detail of the OC-RRT algorithm, a brief summary of historical approaches to the closed-chain or OC motion planning problem are presented.

Projection strategy[7,9,31] and randomized gradient descent (RGD)[10,11,32] are very common methods that have a wide range of applications. In dealing with pose or closure constraints, the projection strategies iteratively projecting randomly sampled configurations onto the constraint manifolds have been proved to be feasible. LaValle et al.[10] presented the first method to handle the closure constraints of closed-chain robots using a RGD algorithm. An improved

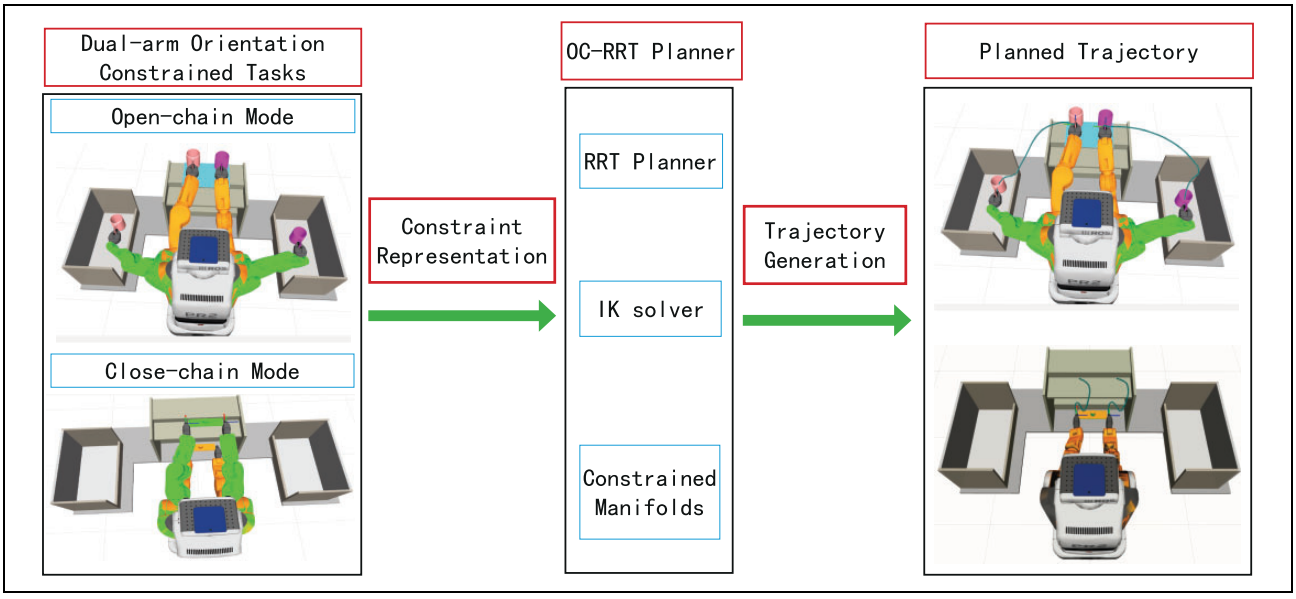


**Figure 2.** The block diagram shows the workflow of the global system architecture, which gives the planning process.

RGD algorithm was designed by Yakey[11] for more general closure constraints. Some other researchers adopted a more effective decomposition approach[12,13] to deal with the closure constraints, which combines IK computation with PRM techniques. To further reduce the computing time of handling closure constraints, random loop generator (RLG)[14,15] was proposed to increase the probability of randomly generating valid configurations. Gharbi et al.[16] used the singular configurations to connect different self-motion manifolds. Position/orientation constraints and joint velocity constraints between cooperative robots[17] were mainly investigated. However, the above methods are only suitable for closed-chain path planning problems.

For the pose constraints on the end-effector, Stilman[7] compared three projection algorithms for pose-constrained tasks and the final results indicated that the Jacobian pseudo-inverse projection was typically faster and more invariant than the RGD algorithm. Berenson et al.[18] used the Task Space Regions (TSRs) to represent a unified framework of pose-related constraints and proposed the CBiRRT planner[19] to solve general end-effector constraints. To improve the efficiency of the projection operation, recently, planners such as Tangent Space RRT (TS-RRT),[20] Atlas-RRT,[21] Tangent Bundle RRT (TB-RRT),[22] and Atlas + X planner[23] sampled new joint configurations within the tangent spaces, which are nearby the constraint manifold. Kingston et al.[24] divided these previous methodologies into five categories: (1) relaxation, (2) projection, (3) tangent space sampling, (4) incremental Atlas construction, and (5) reparameterization. But the approach of decoupling between translational and rotational motions has not been studied yet.

Researchers have proposed several kinematics-based planners capable of planning for regrasping tasks, but they only focus on how to choose the feasible grasping configurations rather than the whole path constraint. Bertram et al.[33] calculated several joint configurations of grasping using the IK solver and then set them as goals for the randomized planner. Berenson et al.[34] used workspace goal regions (WGRs) and two probabilistically complete planners (RRT-JT and IKBiRRT) to deal with end-effector pose constraints. Related IK-based approaches (RRT-JT and IK-RRT)[35] for dual-arm manipulation planning and regrasping tasks were presented by Vahrenkamp et al.[35] and Xian et al.[36] connected nearby C-space through the IK-switch to address complex closed-chain manipulation tasks. Such methods are mainly designed for the selection of goal or switching joint configurations in the path planning, which are infeasible to maintain orientation and closure constraints throughout the whole path. There has been research on dual-arm manipulation planning with orientation constraints[37] using the graph heuristic search techniques. Also, some approaches tried to achieve the approximation of the constraint manifolds by off-line computation,[38] model learning,[39] or demonstrations learning[40,41] but only for certain scenarios.

We compared several classical methods as shown in Table 1. The previous research mainly focuses on closure or pose constraint problem, to the best of our knowledge, the multiple CSP of considering closure and orientation constraint simultaneously has been rarely addressed by randomized path planning approaches.

## Preliminaries

In this section, some mathematical notations and preliminaries of our planning algorithm are given. The preliminaries consists of three parts: classification of dual-arm operations according to the type of dual-arm manipulation tasks,

**Table 1.** Comparison of several constrained motion planners.

| Planners | Tasks | Advantages | Drawbacks |
|---|---|---|---|
| RGD-RRT | Pose or closure constraints | Minimizes constraint error by randomized gradient descent. | Suitable for simple configurations; requires many iterations; inefficient. |
| RLG-PRM | Closure constraints | Satisfies the closure constraint by the analytic IK solver with high efficiency. | Need to select appropriate redundancy angle; weak completeness guarantee. |
| CBiRRT | General end-effector pose constraints | Able to plan with a variety of constraints; probabilistic completeness. | Computation of Jacobian pseudo-inverses; many iterations. |
| TS-RRT | Pose constraints | Less consuming time, less number of iterations and fewer extended nodes overall than the CBiRRT planner. | Hard to guarantee the validity of the lazy projection procedure; computation of Jacobian pseudo-inverses. |
| Atlas-RRT | Singularity-free, pose or kinodynamic constraints | Creates tangent polytopes for more uniform sampling; projects at each step for more accurate sampling and interpolation. | Additional computation of tangent polytopes; computation of Jacobian pseudo-inverses. |
| IKBiRRT | Regrasping tasks | Finds grasping configurations by analytic IK solver with high efficiency. | Just specifies the goal configuration not for all nodes of the random tree. |

IK: inverse kinematic; RGD: randomized gradient descent; PRM: probabalistic RoadMap; RLG: random loop generator; RRT: rapidly-exploring random trees; TS: Tangent Space.

representation of pose constraints between the two arms, and description of motion planning on constraint manifold.

## Classification of dual-arm manipulation

A summary of dual-arm manipulation conducted by Smith was discussed,[42] where dual-arm operation was split into modes of non-coordinated manipulation (each arm performs a different task) and coordinated manipulation (both arms implement different parts of the same task). The coordinated manipulation mode[43] was further divided into bimanual manipulation, where two arms are engaged in manipulation of a single object within the shared space and goal-coordinated manipulation, where two arms operate separately without kinematic constraints but both are solving the same task.

When there is no orientation constraint imposed on the end-effector of the manipulators, the dual-arm manipulation can often be realized by well-investigated motion planners for closed kinematic chains or two single-arm robots (please refer to "Related work" section for general reviews). However, the introduction of orientation constraint makes dual-arm manipulation more complicated because of the difficulty of parameterizing lower dimensional orientation constraint manifolds and solving analytic IK solutions for the desired orientation. Thus, the dual-arm manipulation with orientation constraints is an important and difficult part of the dual-arm manipulation, which can be categorized into open-chain manipulation mode and close-chain manipulation mode.

As only kinematic constraints are considered in this article, dual-arm manipulation can be grouped into *open-chain manipulation* and *close-chain manipulation* according to whether they form a closed-chain mechanism. In the mode of open-chain manipulation, the two robot arms operate separately with loose or without closed kinematic chains. Compared with open-chain manipulation, close-chain manipulation has more stringent constraints on the closed kinematic chains, which makes it more complicated. As for dual-arm manipulation, the cases of non-coordinated manipulation can be considered as the open-chain manipulation and the bimanual manipulation can be referred to as the close-chain manipulation. In the cases of goal-coordinated manipulation, both open-chain manipulation and close-chain manipulation exist in the process of operation.

## Constraints description

The orientation and kinematic closure constraints of dual-arm robot most commonly take the form of the position and/or orientation of the robot's end-effector. In this article, homogeneous transformation of frame $\{b\}$ relative to frame $\{a\}$ is used to describe position and orientation of end-effector, which consists of a $3 \times 3$ rotation matrix ${}^a_bR$ and a $3 \times 1$ translation vector ${}^a_bp$

$$ {}^a_bT = \begin{bmatrix} {}^a_bR & {}^a_bp \\ 0 & 1 \end{bmatrix} \tag{1} $$

To express pose constraint (denoted by $C^r$) of the task frame relative to the world frame intuitively, a six-dimensional vector $[x, y, z, \alpha, \beta, \gamma]^{\mathrm{T}} \in \mathbb{R}^6$ is adopted here. The three-dimensional vector $[x, y, z]^{\mathrm{T}}$ indicates translation and three-dimensional Roll-Pitch-Yaw (RPY) Euler angles $\Phi = [\alpha, \beta, \gamma]^{\mathrm{T}}$ about fixed axes indicates rotation

$$ C^r = \begin{bmatrix} x \\ y \\ z \\ \alpha \\ \beta \\ \gamma \end{bmatrix}, \text{where} \begin{cases} x \in (x_{\min}, x_{\max}) \\ y \in (y_{\min}, y_{\max}) \\ z \in (z_{\min}, z_{\max}) \\ \alpha \in (\alpha_{\min}, \alpha_{\max}) \\ \beta \in (\beta_{\min}, \beta_{\max}) \\ \gamma \in (\gamma_{\min}, \gamma_{\max}) \end{cases} \tag{2} $$

As there is no explicit kinematic closure constraint in open-chain manipulation mode, the dual-arm system can be considered as two independent single-arm systems with predefined orientation constraints on each arm. That is, Euler angles $\Phi = [\alpha, \beta, \gamma]^{\mathrm{T}}$ are set to fixed values while the position variables $[x, y, z]^{\mathrm{T}}$ are random values within the operating space for each arm. In terms of close-chain manipulation mode, two arms are engaged in manipulation of one common object with fixed orientation of both right and left arm. In such cases, each arm not only has specified orientation all along the entire path but also has kinematic position constraint between the two arms. That means Euler angles $\Phi = [\alpha, \beta, \gamma]^{\mathrm{T}}$ of the two arms are set to the same fixed values and translation vector of end-effector $[x, y, z]^{\mathrm{T}}$ between the two arms maintains a fixed value.

### Constraint manifold

It has long been recognized that the notion of C-space ($\mathcal{Q}$) allows the complicated motion planning problems to be easily described in a unified way. The advantage of using the $\mathcal{Q}$ is that motion planning for the robot with complex geometric shape is equivalent to motion planning for a point in $\mathcal{Q}$. For example, a $n$-DOF robot's configuration $q = (q_1, q_2, \ldots, q_n)$ can be represented as a point in $\mathcal{Q}$ and a homogeneous transformation of the robot can be converted into a path in $\mathcal{Q}$, where $q_1, q_2, \ldots, q_n$ are the joint angles of the robotic arm. The set of points in $\mathcal{Q}$ that correspond to valid configurations of the robot (the configurations do not cause collision with any obstacles or itself in workspace) is defined as free space $\mathcal{Q}_{free} \subseteq \mathcal{Q}$. Given an initial configuration $\boldsymbol{q}_{init} \in \mathcal{Q}_{free}$ and a goal configuration $\boldsymbol{q}_{goal} \in \mathcal{Q}_{free}$, the classical version of the motion planning problem can be defined as finding a continuous path $\tau$ such that

$$\tau : [0, 1] \rightarrow \mathcal{Q}_{free}$$
$$\tau(0) = \boldsymbol{q}_{init}; \tau(1) = \boldsymbol{q}_{goal}$$

In general, sampling-based planners are typically efficient in planning motions for high-dimensional systems and provide probabilistic completeness guarantees. Instead of computing $\mathcal{Q}_{free}$ exactly, these planners incrementally sample collision-free configurations and connect them to construct the approximation of the $\mathcal{Q}_{free}$ until a solution is eventually found. For a more in-depth review, the randomly sampling configuration consists of multiple independent random variables $q_1, q_2, \ldots, q_n$ as there is no strict constraint on the pose of the robot's end-effector.

However, for kinematic constrained motion planning, these random variables $q_1, q_2, q_n$ are usually correlated (nonlinear constraints imposed on some or all of these joint variables) because the configuration need to satisfy specific nonlinear equality constraints $f(\boldsymbol{q}) = C^r$ according to task requirements. Such equality constraints have the potential to reduce the dimensionality of $\mathcal{Q}$ and thus a lower dimensional manifold is delineated. This lower dimensional constraint manifold embedded in the ambient $\mathcal{Q}$ is also known as constraint manifold $\mathcal{M}_{\mathcal{C}} \subset \mathcal{Q}$, where all the configurations must meet the nonlinear equality constraints. Supposing there are $k$ ($n > k > 0$) equality constraints imposed, the $(n-k)$-dimensional implicit $\mathcal{M}_{\mathcal{C}} \subset \mathbb{R}^{n-k}$ within the $n$-dimensional $\mathcal{Q} \subset \mathbb{R}^n$ can be defined as

$$\mathcal{M}_{\mathcal{C}} = \{\boldsymbol{q} = (q_1, q_2, \ldots, \boldsymbol{q}_n) \in \mathcal{Q} | f(\boldsymbol{q}) = C^r\} \quad (4)$$

As the structure of the constraint manifold region is not known a priori, motion planners need to be able to either directly calculate analytical IK solutions that inherently satisfy the special equality constraints or iteratively project the invalid configurations onto the constraint manifold. Obviously, computing analytical IK solutions can be much more rapid than the projection method, but the process of IK is fairly complicated or even null. Fortunately, the analytical IK solutions that satisfy the pose constraints can be solved for decoupled manipulators (see "Generating IK solutions" section).

## OC-RRT planner

To solve the constrained motion planning problem, algorithms must conduct a search on constraint manifold $\mathcal{M}_{\mathcal{C}}$. In this article, an analytical IK solver is adopted to directly generate constraint-satisfying joint configurations that lie on $\mathcal{M}_{\mathcal{C}}$ for decoupled dual-arm robots.

### Generating IK solutions

Decoupled manipulators are frequently equipped with a humanoid spherical wrist from the viewpoint of the anthropomorphic arm structure, which allows decoupling motion of the position problem from the orientation problem. This makes it possible to generate configurations that lie on constraint manifold directly. A general decoupled redundant manipulator with seven revolute joints is employed in this work, whose world frame $\{w\}$ is fixed to the base frame $\{0\}$. We assume that the robot's end-effector is mounted on the wrist fixedly, which can effectively transform constraints of the end-effector frame $\{e\}$ into the wrist frame $\{7\}$

$$^{0}_{7}T = {}^{0}_{\mathrm{e}}T \cdot ({}^{7}_{e}T)^{-1} \quad (5)$$

where ${}^{7}_{e}T$ will be a constant matrix once the end-effector is fixed and ${}^{0}_{e}T$ is predefined according to the task constraint.

As the last three joint axes of the decoupled manipulator intersect at the wrist point, ${}^{0}_{7}p$ is equal to ${}^{0}_{5}p$, thereby

$$^{0}_{7}p = {}^{0}_{1}p + {}^{0}_{1}R \cdot {}^{1}_{2}p + {}^{0}_{2}R \cdot {}^{2}_{3}p + {}^{0}_{3}R \cdot {}^{3}_{4}p + {}^{0}_{4}R \cdot {}^{4}_{5}p \quad (6)$$

where ${}^{0}_{7}p = {}^{0}_{\mathrm{e}}R \cdot {}^{e}_{7}p + {}^{0}_{\mathrm{e}}p$ according to equation (5). Apparently, the position of wrist is independent of the wrist angles $\boldsymbol{q}_{wrist}$, which consists of the last three joint angles

$\boldsymbol{q}_{wrist} = (q_5, q_6, q_7)$. Hence, only main arm angles $\boldsymbol{q}_{marm} = (q_1, q_2, q_3, q_4)$ are considered for the analysis of equation (6) and analytical IK solutions of $\boldsymbol{q}_{marm}$ can be solved through equation (6) by parametrizing a redundancy. When the first angle $q_1$ is chosen as the redundant angle $\boldsymbol{q}_{redund}$, equation (6) can be readily rewritten in the form

$$ {}^1_0R \cdot ({}^0_7p - {}^0_1p) = {}^1_2p + {}^1_2R \cdot {}^2_3p + {}^1_3R \cdot {}^3_4p + {}^1_4R \cdot {}^4_5p \quad (7) $$

Thus, we have three scalar equations and three unknowns, the analytical IK solutions of $\boldsymbol{q}_{marm}$ can be solved.

Likewise, the orientation of wrist can be controlled by $\boldsymbol{q}_{wrist}$ independently from $\boldsymbol{q}_{marm}$. Analytical IK solution(s) of $\boldsymbol{q}_{wrist}$ can be solved through equation (8)

$$ ({}^0_4R)^{-1} \cdot {}^0_eR \cdot ({}^7_eR)^{-1} = {}^4_5R \cdot {}^5_6R \cdot {}^6_7R \quad (8) $$

Please refer to Angeles[44] for more detailed explanation.

## Sampling constraint-satisfying configurations

Although the study of motion planners with orientation or kinematic closure constraints (see "Related work" section) has been proved feasible using Jacobian pseudo-inverse projection techniques, these methods have many technical challenges, such as avoiding joint limits and singularity, iterations, and computational efficiency. What is more, dual-arm manipulation with orientation constraint not only has specified orientation all along the entire path, but also need to maintain a relatively fixed position constraint between the two arms' end-effector in close-chain manipulation mode. The multiple constraints on end-effector make the problem more complicated than only imposed orientation or kinematic closure constraint.

To overcome these difficulties, an analytical IK solver is applied to calculate constraint-satisfying configurations directly, which has preferable efficiency and satisfactory accuracy. The dual-arm system is decomposed into the "active arm" (named arm1) and the "passive arm" (named arm2). Due to the predefined orientation constraint of tasks, the active arm needs to maintain fixed Euler angles $\Phi$ all along the whole motion, while the position values are random values within the operating space. The same situation applies to passive arm in open-chain manipulation mode. The orientation constraint manifold $\mathcal{M}_{\mathcal{C}}^{\mathcal{O}}$ of active arm, for example, is expressed as

$$ \mathcal{M}_{\mathcal{C}}^{\mathcal{O}} = \{\boldsymbol{q}_{arm1} = (q_1, \ldots, q_7) \in \mathcal{Q} | \, f(\boldsymbol{q}_{arm1}) = \Phi\} \quad (9) $$

where $\boldsymbol{q}_{arm1}$ is the active arm configuration consisting of seven joint angles. Function $f()$ is the forward kinematics mapping from C-space to Cartesian space.

As for mode of close-chain manipulation, we suppose that a grasping object by the two hands is rigid and the length of the rigid object determines the relatively fixed position between the two arms. As aforementioned, the configuration of active arm is assigned to maintain a predefined orientation in the entire movement and then the configuration of passive chain is computed to meet the closure constraint by IK solver. Because of the kinematic redundancy of the passive arm, there are an infinite number of IK solutions[45] corresponding to the desired end-effector pose obtained from the forward kinematics of the active arm. The set of all these IK solutions in joint space is referred to as self-motion manifold $\mathcal{M}_{\mathcal{C}}^{\mathcal{S}}$, which can be written as

$$ \mathcal{M}_{\mathcal{C}}^{\mathcal{S}} = \{\boldsymbol{q}_{arm2} = (q_8, \ldots, \boldsymbol{q}_{14}) \in \mathcal{Q} | \, f(\boldsymbol{q}_{arm2}) = p_{arm2}\} \quad (10) $$

where $\boldsymbol{q}_{arm2}$ is the passive arm configuration consisting of seven joint angles and $p_{arm2}$ is the end-effector pose of passive arm calculated by the forward kinematics of active arm.

As the kinematics analysis of decoupled manipulators mentioned before, position exploration can be done through random sampling of the main arm angles, while the orientation of wrist can be controlled by the wrist angles. The pseudocode procedure of generating a constraint-satisfying configuration is described in Algorithm 1. For the active arm (arm1), the random sampling function *RandomConfig*() repeatedly samples new random states of main arm angles $\boldsymbol{q}_{marm1}^{rand}$ and analytic IK of wrist angles *Wrist_Inv_Kin*() tries to calculate the wrist angles $\boldsymbol{q}_{wrist1}^{rand}$ for the given Euler angles $\Phi$. Obviously, the random state of active arm $\boldsymbol{q}_{arm1}^{rand}$ consisting of $\boldsymbol{q}_{marm1}^{rand}$ and $\boldsymbol{q}_{wrist1}^{rand}$ lies on the orientation constraint manifold. In terms of the passive arm (arm2), the similar steps mentioned above are extended to the passive arm (line 6 and 7) and also the similar results can be obtained when operating in open-chain manipulation mode. Otherwise, in close-chain manipulation mode, the forward kinematic function $FK()$

**Algorithm 1.** *OCSampleConfig*($\Phi$).

```
1:  while Inv_Kin_Success do
2:      q_marm1^rand ← RandomConfig(q_marm1);
3:      q_wrist1^rand ← Wrist_Inv_Kin(q_marm1^rand, Φ);
4:      q_arm1^rand ← (q_marm1^rand, q_wrist1^rand);
5:      if Open_Chain_True then
6:          q_marm2^rand ← RandomConfig(q_marm2);
7:          q_wrist2^rand ← Wrist_Inv_Kin(q_marm2^rand, Φ);
8:      else if Close_Chain_True then
9:          p_arm2 ← FK(q_arm1^rand);
10:         q_redund^rand ← RandomConfig(q_redund);
11:         q_marm2^rand ← Arm_Inv_Kin(q_redund^rand, p_arm2);
12:         q_wrist2^rand ← Wrist_Inv_Kin(q_marm2^rand, Φ);
13:     end if
14:     q_arm2^rand ← (q_marm2^rand, q_wrist2^rand);
15: end while
16: return q_rand ← (q_arm1^rand, q_arm2^rand);
```

**Algorithm 2.** OC-RRT.

**Require:** $V \leftarrow \{\boldsymbol{q}_{init}\}; E \leftarrow \emptyset;$
**Ensure:** $T \leftarrow (V, E);$
1: **for** $i = 1 \rightarrow n$ **do**
2: $\quad \boldsymbol{q}_{rand} \leftarrow OCSampleConfig(\Phi);$
3: $\quad \boldsymbol{q}_{nearest} \leftarrow Nearest(T = (V, E), \boldsymbol{q}_{rand});$
4: $\quad \boldsymbol{q}_{new} \leftarrow OCSteer(\boldsymbol{q}_{nearest}, \boldsymbol{q}_{rand}, \Phi);$
5: $\quad$ **if** $CollisionFree(\boldsymbol{q}_{nearest}, \boldsymbol{q}_{new})$ **then**
6: $\quad\quad V \leftarrow V \bigcup \{\boldsymbol{q}_{new}\}; E \leftarrow E \bigcup \{\boldsymbol{q}_{nearest}, \boldsymbol{q}_{new}\};$
7: $\quad$ **end if**
8: **end for**
9: **return** $T \leftarrow (V, E);$

is used to figure out the pose of the passive arm's end-effector $p_{arm2}$ by $\boldsymbol{q}_{arm1}^{rand}$ and the length of the rigid object. Then analytical IK solutions of main arm angles $\boldsymbol{q}_{marm2}^{rand}$ and wrist angles $\boldsymbol{q}_{wrist2}^{rand}$ can be solved by a random redundant angle $\boldsymbol{q}_{redund}^{rand}$ through equations (6) and (3), respectively. Obviously, the state of passive arm $\boldsymbol{q}_{arm2}^{rand}$ consisting of $\boldsymbol{q}_{marm2}^{rand}$ and $\boldsymbol{q}_{wrist2}^{rand}$ lies on the self-motion manifold. The above code loops until IK is solved successfully ($Inv\_Kin\_Success = true$), that is, $\boldsymbol{q}_{rand}$ meets the restrictive requirements of joint angles range and avoiding singularity.

## OC-RRT algorithm

Sampling-based planners, such as RRT-Connect (a bidirectional search version of RRT)[4], RRT* (an asymptotically optimal RRT),[5] anytime RRT*,[46] and RABIT* (regionally accelerated batch informed trees),[47] are state-of-the-art techniques for solving the motion planning problem, which use a variety of distributions for sampling the C-space they search. Instead of running the planner on the C-space, the OC-RRT algorithm works on the constraint manifold to meet the OC requirement of task directly. As shown in Algorithm 2, the OC-RRT algorithm strives to search the constraint manifold for a feasible path by growing a space-filling tree $T$. The tree is initially rooted at the start point $\boldsymbol{q}_{init}$, and a random node $\mathrm{q}_{rand}$ is generated by $OCSampleConfig()$ at each incremental expansion of the tree-grow. Then the nearest node (parent node) $\boldsymbol{q}_{nearest}$ is selected from all the nodes on $T$ and an attempt $OCSteer()$ is made to generate the new node (child node) $\boldsymbol{q}_{new}$ by moving a step size $\varepsilon$ from $\boldsymbol{q}_{nearest}$ toward $\boldsymbol{q}_{rand}$. If the path between the $\boldsymbol{q}_{nearest}$ and $\boldsymbol{q}_{new}$ is collision-free, then $\boldsymbol{q}_{new}$ is added to $V$ (the vertices of $T$) and the edge $(\boldsymbol{q}_{nearest}, \boldsymbol{q}_{new})$ is added to $E$ (the edges of $T$). The extending tree $T$ never halts until a solution has been found or some failure rules are satisfied.

To depict the functionality of the presented OC-RRT, it is reasonable to compare with the RRT algorithm[3] and point out the relevant modifications. The main differences between OC-RRT and RRT are as following:

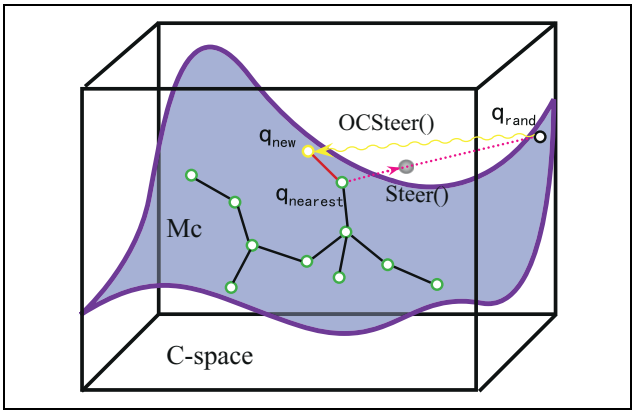


**Figure 3.** New node (gray) generated by Steer() is outside of constraint manifold (blue). Hence, OCSteer() is proposed to generate new node (yellow) lying on the constraint manifold.

1. Sampling: compared with RRT algorithm, OC-RRT uses $OCSampleConfig()$ (Algorithm 1) to generate random sampling node $\boldsymbol{q}_{rand}$ uniformly on constraint manifold rather than C-space. By this way, all the possible constraint-satisfying configuration on the constraint manifold can be found.
2. Steering: In the RRT algorithm, the function $Steer(\boldsymbol{q}_{nearest}, \boldsymbol{q}_{rand})$ generates a new node by moving a step size $\varepsilon$ in the straight line from the nearest node $\boldsymbol{q}_{nearest}$ to the randomly sampled node $\boldsymbol{q}_{rand}$. It is obvious that $\boldsymbol{q}_{new}$ may escape from the constraint manifold by applying the $Steer()$ to the OC-RRT algorithm. Hence, $OCSteer()$ is proposed to generate new node $\boldsymbol{q}_{new}$ lying on constraint manifold (Figure. 3). For the active arm (arm1), $Steer()$ (line 2 of Algorithm 3) generates the main arm angles of the new node $\boldsymbol{q}_{marm1}^{new}$ by moving a step size $\varepsilon$ away from the main arm angles of the nearest node $\boldsymbol{q}_{marm1}^{nearest}$ toward the direction of the main arm angles of randomly sampled node $\boldsymbol{q}_{marm1}^{rand}$. Then, analytic IK of wrist angles $Wrist\_Inv\_Kin()$ tries to calculate the wrist angles of the new node $\boldsymbol{q}_{wrist1}^{new}$ for the given Euler angles $\Phi$. Obviously, the new state of active arm $\boldsymbol{q}_{arm1}^{new}$ consisting of $\boldsymbol{q}_{marm1}^{new}$ and $\boldsymbol{q}_{wrist1}^{new}$ lies on the orientation constraint manifold. In addition, as a small change in the joint angles results in a slight change in the pose of the end-effector and in turn it is also true, the final motion of the active arm's end-effector does not jump between adjacent nodes of the random trees when the step size $\varepsilon$ is small enough. That means the edge between $\boldsymbol{q}_{arm1}^{nearest}$ and $\boldsymbol{q}_{arm1}^{new}$ also lies on the orientation constraint manifold when $\varepsilon$ is small.

In terms of the passive arm, similar steps mentioned above are extended to the passive arm and also the similar results can be obtained when operating in open-chain manipulation mode. Otherwise, the forward kinematic function $FK()$ is used to figure out the position of the

**Algorithm 3.** *OCSteer*($q_{nearest}$, $q_{rand}$, $\Phi$).

**while** $Inv\_Kin\_Success$ **do**
  $\boldsymbol{q}_{marm1}^{new} \leftarrow Steer(\boldsymbol{q}_{marm1}^{nearest}, \boldsymbol{q}_{marm1}^{rand})$;
  $\boldsymbol{q}_{wrist1}^{new} \leftarrow Wrist\_Inv\_Kin(\boldsymbol{q}_{marm1}^{new}, \Phi)$;
  $\boldsymbol{q}_{arm1}^{new} \leftarrow (\boldsymbol{q}_{marm1}^{new}, \boldsymbol{q}_{wrist1}^{new})$;
  **if** $Open\_Chain\_True$ **then**
    $\boldsymbol{q}_{marm2}^{new} \leftarrow Steer(\boldsymbol{q}_{marm2}^{nearest}, \boldsymbol{q}_{marm2}^{rand})$;
    $\boldsymbol{q}_{wrist2}^{new} \leftarrow Wrist\_Inv\_Kin(\boldsymbol{q}_{marm2}^{new}, \Phi)$;
  **else if** $Close\_Chain\_True$ **then**
    $p_{arm2} \leftarrow FK(\boldsymbol{q}_{arm1}^{new})$;
    $q_{redund}^{new} \leftarrow Steer(q_{redund}^{nearest}, q_{redund}^{rand})$;
    $\boldsymbol{q}_{marm2}^{new} \leftarrow Arm\_Inv\_Kin(q_{redund}^{new}, p_{arm2})$;
    $\boldsymbol{q}_{wrist2}^{new} \leftarrow Wrist\_Inv\_Kin(\boldsymbol{q}_{marm2}^{new}, \Phi)$;
  **end if**
  $\boldsymbol{q}_{arm2}^{new} \leftarrow (\boldsymbol{q}_{marm2}^{new}, \boldsymbol{q}_{wrist2}^{new})$;
**end while**
**return** $\boldsymbol{q}_{new} \leftarrow (\boldsymbol{q}_{arm1}^{new}, \boldsymbol{q}_{arm2}^{new})$;

passive arm's end-effector $p_{arm2}$ by $\boldsymbol{q}_{arm1}^{new}$, and the length of the rigid object in close-chain manipulation mode. $Steer(\boldsymbol{q}_{redund}^{nearest}, \boldsymbol{q}_{redund}^{rand})$ generates the new redundant joint $\boldsymbol{q}_{redund}^{new}$ and then analytical IK solutions of main arm angles $\boldsymbol{q}_{marm2}^{new}$ and wrist angles $\boldsymbol{q}_{wrist2}^{new}$ can be solved through equation (6) and (3), respectively. Obviously, the new node $\boldsymbol{q}_{new}$ is a constraint-satisfying configuration, and the final motion of the robot's end-effectors does not jump between adjacent nodes of the random trees when the $\varepsilon$ is small. The above code loops until IK is solved successfully ($Inv\_Kin\_Success = true$), that is, $\boldsymbol{q}_{new}$ meets the restrictive requirements of joint angles range and avoiding singularity.

Probabilistic completeness guarantee of the OC-RRT algorithm follows from the property: given infinite time, every possible constraint-satisfying configuration on the constraint manifolds will be added to the space-filling tree. What is more, the trajectory of joint angle and the final motion of the robot's end-effector do not jump between adjacent nodes of the random trees when the step size $\varepsilon$ is small enough ($\varepsilon \in (0.01, 0.05)$ in our simulation of "Simulation experiments" section). Thus, the trajectory between the $\boldsymbol{q}_{nearest}$ and $\boldsymbol{q}_{new}$ can also respect the constraints of the task, which is verified to be effective by the simulation results of "Simulation experiments" section.

## Simulation experiments

To validate the performance of the proposed OC-RRT algorithm, four scenarios (named Industrial I, Industrial II, Tabletop, and Passageway) were implemented on the Willow Garage's PR2 robot in MoveIt![48] simulator. PR2 is designed with two 7-DOF decoupled manipulators, which allows us to decouple planning for the main arm joints (the first four joints) from the wrist joints (the last three joints) as shown in Figure 4. The shoulder pan joint ($q_8$) of right arm was selected as the redundant joint for the analytical IK solution(s) of the right arm's main arm configuration, the joint limits and singularities were also taken into account in the process of solving the IK solutions. For decoupled manipulators with spherical wrist, it is possible to split the singularity computation into two separate problems: (1) computation of main arm singularities resulting from the motion of the main arm joints and (2) computation of wrist singularities resulting from the motion of the wrist joints. Main arm singularities only happen when $\sin(q_4) = 0$, while wrist singularities only happen when $\sin(q_6) = 0$. To ensure that the trajectory between nodes meets the task constraints while ensuring the planning speed, the step size $\varepsilon$ was set to a random value between 0.01 and 0.05. There is no doubt that the smaller the step size $\varepsilon$, the better performance our approach will be, but the planning time will increase. Thus, how to choose the optimal $\varepsilon$ is another problem that is not discussed here.

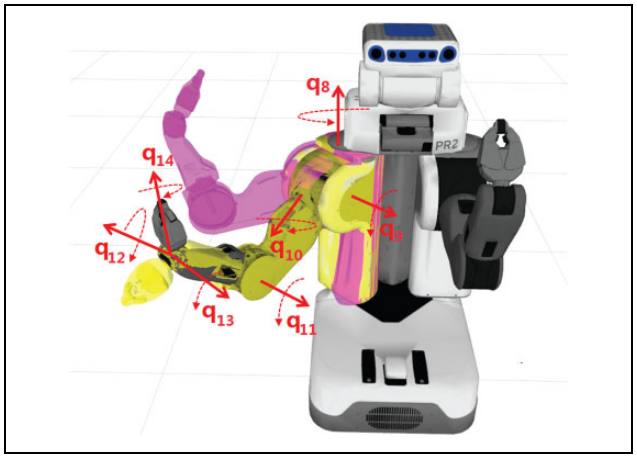


**Figure 4.** The right arm configuration of the PR2 robot consists of the main arm joints ($q_8$, $q_9$, $q_{10}$, and $q_{11}$) and the wrist joints ($q_{12}$, $q_{13}$, and $q_{14}$). Changes in the position of the PR2 robot's wrist, for example, from current state in gray to a new state in pink, are only relevant for the main arm joints, which have nothing to do with the wrist joints. The orientation of the PR2 robot's wrist can be controlled only by the wrist joints, for instance, from the gray state to the yellow state. The left arm has the similar properties.

For the sake of comparison, the CBiRRT proposed by Berenson et al.[19] was implemented for dealing with the same task. As the CBiRRT planner and our OC-RRT planner are randomized, we carried out 20 times repetitive experiments for each simulated scenarios. The two planners are coded in the open-source Open Motion Planning Library (OMPL),[49] and the laptop used to install the Robot Operating System (ROS) has 6 GB of RAM and an Intel i3-6100 CPU quadcore processor running at 2.5 GHZ.

### *Evaluation index*

To evaluate the performance capabilities of the proposed algorithm, several indicators were introduced: (1) planning time, (2) number of nodes, (3) success rate, and (4) root-

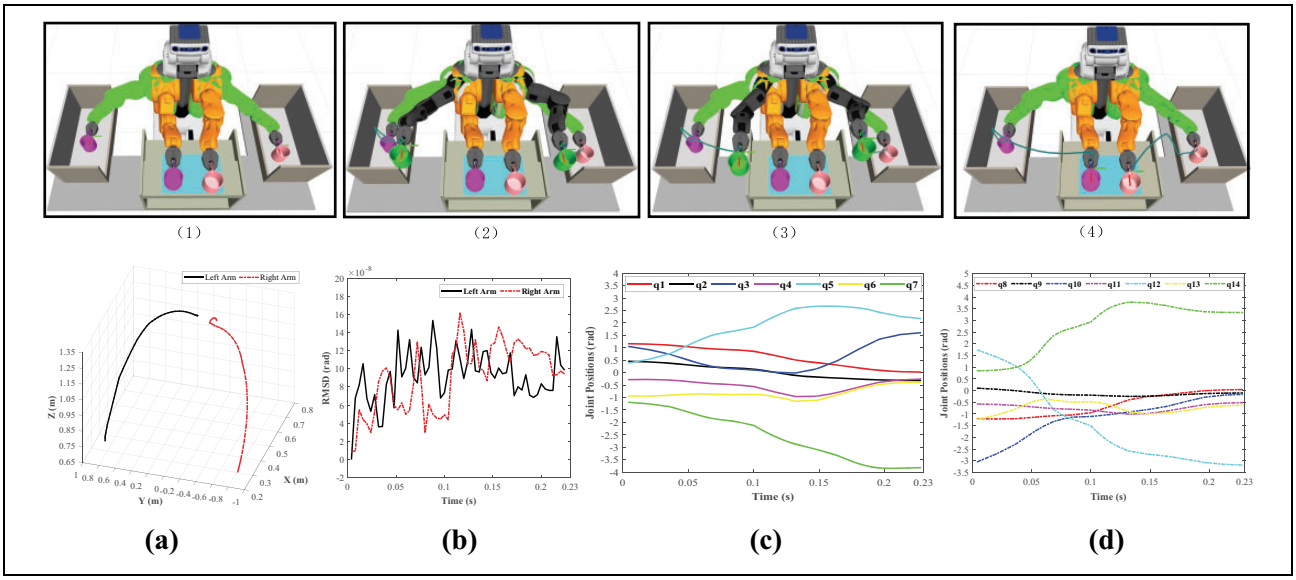


**Figure 5.** Industrial I scenario: (1)–(4) snapshots of moving two cups full of water from start pose (green) to goal pose (yellow) by using the OC-RRT planner. Constraints: upright orientation and collision-free constraint. Blue: trajectories of two arms' end-effector. Green: start configuration. Yellow: goal configuration. (a) Trajectories of two arms' end-effector. (b) RMSD of orientation. (c) Joint position trajectories of left arm. (d) Joint position trajectories of right arm. OC-RRT: orientation-constrained rapidly exploring random trees; RMSD: root-mean-square deviation.

mean-square deviations (RMSDs) of position or orientation for path accuracy. The RMSD or RMSE[50,51] is a well-known evaluating index, especially in probability statistics. The orientation RMSD (in radians) is defined as

$$O\_RMSD = \sqrt{\frac{\sum_{k=1}^{n}\left((\Delta\alpha_k)^2 + (\Delta\beta_k)^2 + (\Delta\gamma_k)^2\right)}{3 \times n}} \tag{11}$$

and the position RMSD (in meters) is defined as

$$P\_RMSD = \sqrt{\frac{\sum_{k=1}^{n}\left((\Delta p_k^x)^2 + (\Delta p_k^y)^2 + (\Delta p_k^z)^2\right)}{3 \times n}} \tag{12}$$

where $n$ is the number of sampling nodes; $\Delta\alpha, \Delta\beta, \Delta\gamma$ are the difference between the measured value (derived from forward kinematics) and the desired Euler angles of end-effector; $\Delta p^x, \Delta p^y, \Delta p^z$ are the difference between the measured value (derived from forward kinematics) and the desired value of two end-effectors' distance.

## Experimental results in open-chain mode

As shown in Figure 5, the simulated PR2 robot has to vertically move two cups full of water from the initial (green) to the target configuration (yellow) in the cluttered environment (Industrial I). According to requirement of the task, not only the start pose and the target pose but also the entire planned trajectory needs to satisfy the end-effector's orientation constraint with Euler angles $\alpha = \beta = \gamma = 0$ radians.

Figure 5(1)–(4) showed a typical result of the snapshots from the execution of the OC-RRT planner, where the planned collision-free path (shown in blue) is smooth. The accompanying video showed the motion of the robot for this task. Figure 5(a) presented the path of the two arms' end-effector. The corresponding orientation RMSD of the two arms' end-effectors was fixed to zero with high accuracy as shown in Figure 5(b) and the trajectories of two arms' each joint moved smoothly as shown in Figure 5(c) (left arm) and (d) (right arm).

## Experimental results in close-chain mode

Due to the orientation and closure constraints of the arms, the distance between two grasping points of the arms and the cluttered environments caused a very tight cooperative workspace for dual-arm manipulation. The multiple simultaneous constraints make the problem particularly complicated. We evaluated the performance of the proposed planner by presenting numerical results in three different scenarios[52] (Industrial II, Table top and Narrow Passageway) as shown in Figures 6 to 8. The simulated PR2 robot was required to vertically manipulate a tray with a cup on it from the start configuration (green) to the goal configuration (yellow) in the cluttered environments, while maintaining a relatively fixed position constraint between the two arms' end-effector.

Upper part of each figure (Figures 6 to 8) presented the typical result of the snapshots from the execution of the OC-RRT planner for the three different scenarios, where the planner can compute a collision-free path successfully

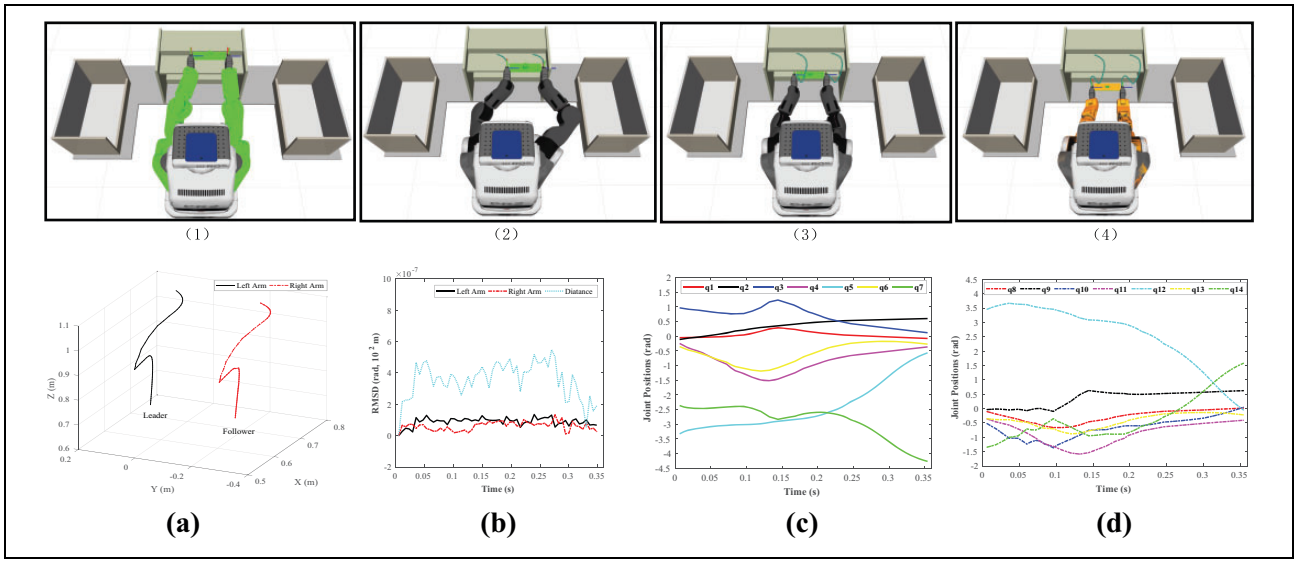


**Figure 6.** Industrial II scenario: (1)–(4) snapshots of moving a tray with a cup on it from start pose (green) to goal pose (yellow). (a) Trajectories of two arms. (b) RMSDs of orientation and position. (c) Joint position of left arm. (d) Joint position of right arm. RMSD: root-mean-square deviation.

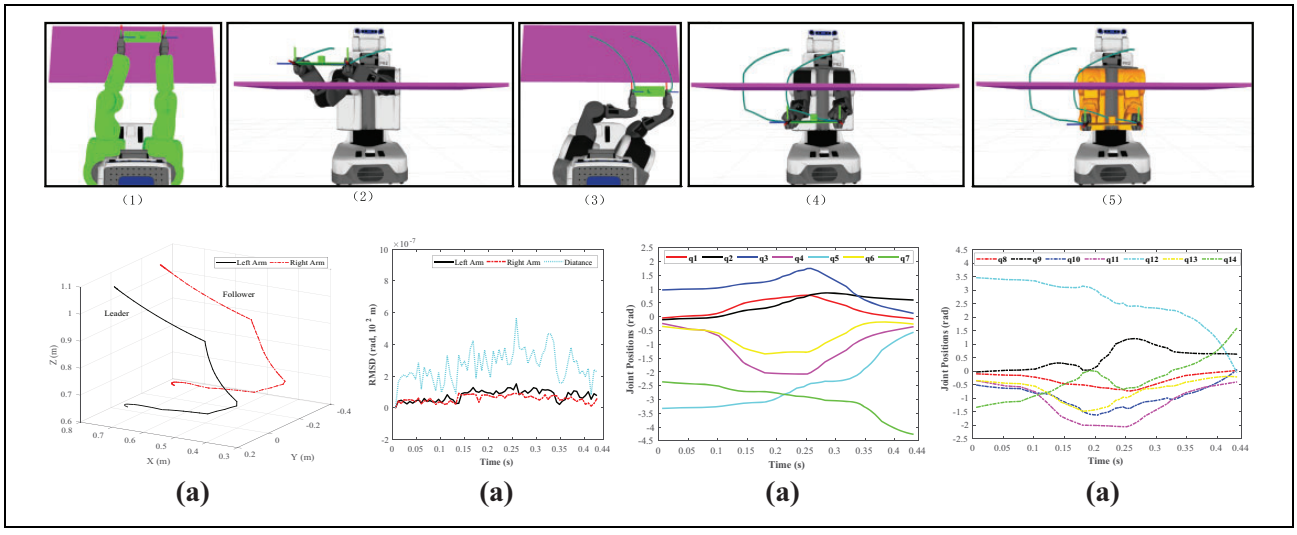


**Figure 7.** Tabletop scenario: (1)–(5) snapshots of moving a tray with a cup on it from start pose to goal pose. (a) Trajectories of two arms. (b) RMSDs of orientation and position. (c) Joint position of left arm. (d) Joint position of right arm. RMSD: root-mean-square deviation.

while satisfying the orientation and closure constraints. The motion of the robot for the different scenarios is fully visible in the accompanying video. The subgraph (a) of each figure showed the entire trajectory of the two arms, where the left arm is identified as the active arm (Leader) and right arm as the passive (Follower). We can visually see that the planned trajectory is smooth and meets the kinematic closure constraints. The position and orientation RMSD of the two arms' end-effectors were fixed to zero with high accuracy as shown in subgraph (b) of each figure. The corresponding joint trajectories of the two arms were shown in subgraph (c) (left arm) and (d) (right arm) of each figure.

## Discussion

The comparative results of simulation between the proposed OC-RRT and the CBiRRT[19] in four different scenarios are given in Table 2. According to the results in Table 2, the OC-RRT and the CBiRRT both can successfully complete the motion planning of dual-arm with the same constraints on PR2 simulation platform, but the OC-RRT algorithm can plan constraint-satisfying trajectory with efficiency and accuracy.

In open-chain manipulation mode, the proposed OC-RRT algorithm is superior to the CBiRRT planner in terms of the indicators of Planning Time, Number of Nodes, and

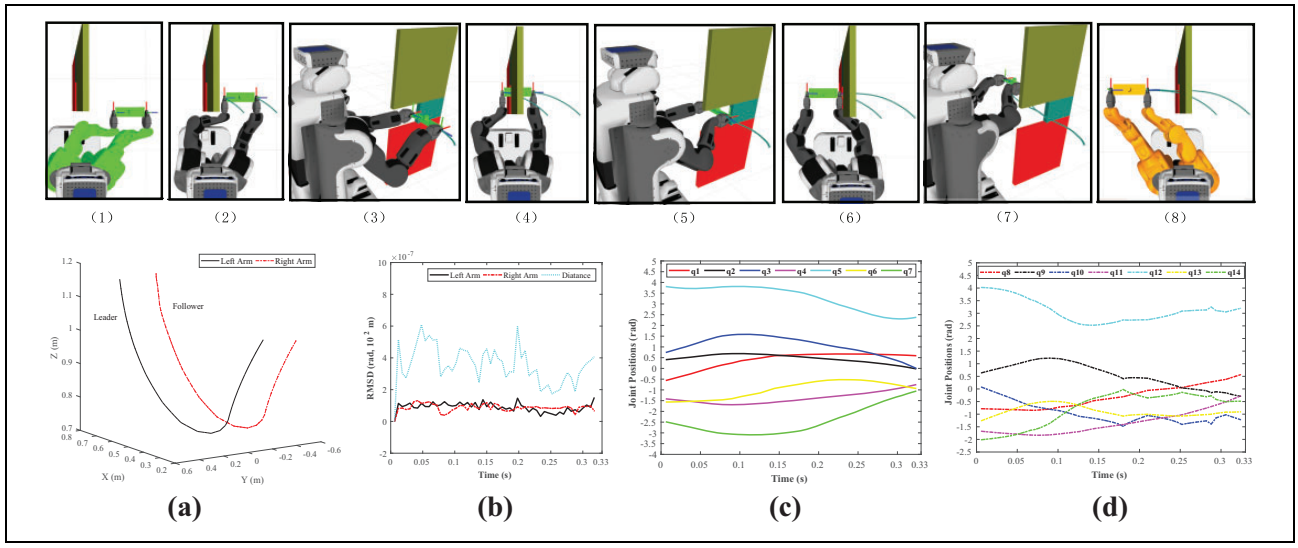


**Figure 8.** Narrow passageway scenario: (1)–(8) snapshots of moving a tray with a cup on it from start pose to goal pose. (a) Trajectories of two arms. (b) RMSDs of orientation and position. (c) Joint position of left arm. (d) Joint position of right arm. RMSD: root-mean-square deviation.

**Table 2.** Simulation results (efficiency and accuracy) from 20 times trials for the four different scenarios.

| Scenario | | Industrial I | | Industrial II | | Tabletop | | Passageway | |
|---|---|---|---|---|---|---|---|---|---|
| Planner | | CBiRRT | OC-RRT | CBiRRT | OC-RRT | CBiRRT | OC-RRT | CBiRRT | OC-RRT |
| Planning time (s) | Mean | 0.73 | 0.26 | 1.73 | 0.31 | 1.36 | 0.39 | 7.21 | 0.43 |
| | Min | 0.49 | 0.11 | 0.86 | 0.17 | 0.51 | 0.22 | 3.59 | 0.29 |
| | Max | 1.53 | 0.39 | 4.07 | 0.43 | 5.10 | 0.58 | 9.74 | 0.82 |
| Number of nodes | Mean | 331 | 128 | 585 | 142 | 463 | 189 | 928 | 257 |
| | Min | 118 | 52 | 397 | 76 | 309 | 91 | 691 | 139 |
| | Max | 474 | 163 | 764 | 194 | 952 | 252 | 1472 | 594 |
| ORMSD ($10^{-7}$ rad) | Mean | 5.9 | 0.95 | 5.4 | 0.85 | 8.1 | 0.7 | 11.7 | 1.2 |
| | Min | 2.7 | 0.5 | 3.7 | 0.3 | 3.1 | 0.52 | 6.3 | 0.2 |
| | Max | 9.2 | 1.3 | 7.1 | 1.4 | 11.8 | 1.7 | 15.5 | 2.9 |
| PRMSD ($10^{-5}$ m) | Mean | (No) | (No) | 12.8 | 3.1 | 10.8 | 1.9 | 15.0 | 3.7 |
| | Min | (No) | (No) | 9.3 | 0.5 | 8.7 | 0.7 | 8.5 | 1.3 |
| | Max | (No) | (No) | 16.9 | 7.6 | 15.1 | 6.9 | 19.4 | 8.4 |
| Success rate (%) | | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |

OC-RRT: orientation-constrained rapidly exploring random trees; RMSD: root-mean-square deviation.

ORMSD. All these superiorities can be attributed to that the OC-RRT planner uses the analytical IK technique to sample constraint-satisfying configurations directly without iteration modification. Besides, the extremely tiny ORMSD can not only illustrate that the nodes generated by the OC-RRT planner meet the orientation constraint accurately but also verify that the trajectories between nodes respect the orientation constraint of the task.

Similarly, the proposed OC-RRT planner is more accurate and efficient than the CBiRRT planner in terms of all indicators in close-chain manipulation mode. In addition, the extremely tiny RMSD of orientation and position can not only illustrate that the nodes generated by the OC-RRT planner meet the orientation and kinematic closure constraints of the task with high accuracy but also verify that the trajectories between nodes respect these constraints.

## Conclusion

In this article, we have presented an efficient and probabilistic complete planning algorithm-combined sampling-based technique with analytical IK solver to address the dual-arm motion planning problem involving orientation constraints for decoupled dual-arm robots. The general problem is interpreted as finding constraint-satisfying configurations and connecting them to generate valid paths in joint space without being trapped in local minima. The key idea of our approach is to efficiently explore the constraint-satisfying configurations by directly sampling them on constraint manifold through analytic IK. Besides, we also limit the joint space displacements between nearest node and new node by setting a small step size $\varepsilon$, so that the trajectories between nodes on the space-filling tree can satisfy

the task constraints. Without losing the major advantage of the RRT algorithm, our method can explore all the possible constraint-satisfying configurations when given infinite time and can avoid falling into local minima by uniform sampling strategy. As the closed-form IK solver is applied to calculate constraint-satisfying configurations, our approach is significantly efficient and accurate. Four typical scenarios were used to demonstrate the effectiveness of the OC-RRT algorithm on the Willow Garage's PR2 simulation platform in ROS MoveIt!.

In our recent work, we have successfully validated the proposed OC-RRT algorithm on the Baxter robot built by Rethink Robotics. The proposed algorithm has also been extended to other sampling-based motion planners such as PRM planner and RRT-Connect planner. Especially when our method was applied to the RRT-connect algorithm, the planning time and path were obviously improved. To make the proposed algorithm framework more versatile, future research will focus on solving general pose constraint tasks for decoupled manipulators. Additionally, considering the influence of the contact force between two arms in the close-chain mode, another future research is how to integrate motion planner and forces/position controller to deal with closure constrained motion planning for dual-manipulators.

## Authors' note

Changbin Yu is also affiliated to School of Engineering, Westlake University, Hangzhou, Zhejiang, China.

## Declaration of conflicting interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

## Funding

The authors disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported in part by the NSFC (Grant no. 61503108), NSFC-DFG (Grant no. 61761136005), 111 Project (Grant no. D17019), and the Key Research and Development Program of Zhejiang Province, China (Grant no. 2019C04018).

## ORCID iD

Jiangping Wang https://orcid.org/0000-0001-5960-1853
Shirong Liu https://orcid.org/0000-0002-8357-3116
Botao Zhang https://orcid.org/0000-0002-7826-3121

## References

1. Reif JH. Complexity of the mover's problem and generalizations. In: *20th Annual symposium on foundations of computer science*, San Juan, Puerto Rico, 29–31 October 1979, pp. 421–427. Piscataway: IEEE.
2. Lavalle SM and Kuffner JJ. Rapidly-exploring random trees: progress and prospects. In: Donald BR, Lynch K, and Rus D (eds) *Algorithmic and computational robotics new directions, 2000*. Wellesley: A.K. Peters, 2001, pp. 293–308.
3. Kavraki LE, Svestka P, Latombe JC, et al. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans Robot Autom* 1996; 12(4): 566–580.
4. Kuffner JJ and Lavalle SM. RRT-connect: an efficient approach to single-query path planning. In: *IEEE International conference on robotics and automation*, San Francisco, CA, USA, 24–28 April 2002, pp. 995–1001.
5. Karaman S and Frazzoli E. Sampling-based algorithms for optimal motion planning. *Int J Robot Res* 2011; 30(7): 846–894.
6. Yang L, Xiao J, Qi J, et al. Gart: an environment-guided path planner for robots in crowded environments under kinodynamic constraints. *Int J Adv Robot Syst* 2016; 13(6): 1–18.
7. Stilman M. Task constrained motion planning in robot joint space. In: *Proceedings of the 2007 IEEE/RSJ international conference on intelligent robots and systems*, San Diego, CA, USA, 29 October–2 November 2007, pp. 3074–3081. Piscataway: IEEE.
8. Jaillet L and Porta JM. Path planning under kinematic constraints by rapidly exploring manifolds. *IEEE T Robot* 2013; 29(1): 105–117.
9. Sucan IA and Kavraki LE. On the performance of random linear projections for sampling-based motion planning. In: *Proceedings of the 2009 IEEE/RSJ international conference on intelligent robots and systems*, St. Louis, MO, USA, 10–15 October 2009, pp. 2434–2439. Piscataway: IEEE.
10. Lavalle SM, Yakey JH, and Kavraki LE. A probabilistic roadmap approach for systems with closed kinematic chains. In: *Proceedings of the 1999 IEEE international conference on robotics and automation*, Detroit, MI, USA, 10–15 May 1999, pp. 1671–1676. Piscataway: IEEE.
11. Yakey JH, Lavalle SM, and Kavraki LE. Randomized path planning for linkages with closed kinematic chains. *IEEE T Robot Autom* 2001; 17(6): 951–958.
12. Han L and Amato N. A kinematics-based probabilistic roadmap method for closed chain systems. In: Donald BR, Lynch K, and Rus D (eds) *Algorithmic and Computational Robotics: New Directions*. Wellesley: A.K. Peters, 2001, pp. 233–246.
13. Xie D and Amato NM. A kinematics-based probabilistic roadmap method for high DOF closed chain systems. In: *Proceedings of the 2004 IEEE international conference on robotics and automation*, New Orleans, LA, USA, 26 April–1 May 2004, pp. 473–478. Piscataway: IEEE.
14. Cortes J, Simeon T, and Laumond JP. A random loop generator for planning the motions of closed kinematic chains using PRM methods. In: *Proceedings of the 2002 IEEE international conference on robotics and automation*, Washington, DC, USA, 11–15 May 2002, pp. 2141–2146. Piscataway: IEEE.
15. Cortes J and Simeon T. Sampling-based motion planning under kinematic loop-closure constraints. In: Erdmann M, Overmars M, Hsu D, et al. (eds) *Algorithmic foundations of robotics VI*. Berlin: Springer, 2004, pp. 75–90.
16. Gharbi M, Cortes J, and Simeon T. A sampling-based path planner for dual-arm manipulation. In: *Proceedings of the 2008 IEEE/ASME international conference on advanced*

*intelligent mechatronics*, Xian, China, 2–5 July 2008, pp. 383–388. Piscataway: IEEE.
17. Gan Y, Dai X, and Li J. Cooperative path planning and constraints analysis for master-slave constraints analysis for master-slave industrial robots. *Int J Adv Robot Syst* 2012; 9(88): 1–13.
18. Berenson D, Srinivasa S, and Kuffner JJ. Task space regions: a framework for pose-constrained manipulation planning. *Int J Robot Res* 2011; 30(12): 1435–1460.
19. Berenson D, Srinivasa SS, Ferguson D, et al. Manipulation planning on constraint manifolds. In: *Proceedings of the 2009 IEEE international conference on robotics and automation*, Kobe, Japan, 12–17 May 2009, pp. 625–632. Piscataway: IEEE.
20. Suh C, Um TT, Kim B, et al. Tangent space RRT: a randomized planning algorithm on constraint manifolds. In: *Proceedings of the 2011 IEEE international conference on robotics and automation*, Shanghai, China, 9–13 May 2011, pp. 4968–4973. Piscataway: IEEE.
21. Porta JM, Jaillet L, and Bohigas O. Randomized path planning on manifolds based on higher-dimensional continuation. *Int J Robot Res* 2012; 31(2): 201–215.
22. Kim B, Um TT, Suh C, et al. Tangent bundle RRT: a randomized algorithm for constrained motion planning. *Robotica* 2016; 34(1): 202–225.
23. Voss C, Moll M, and Kavraki LE. Atlas + X: sampling-based planners on constraint manifolds. *Technical Report, Department of Computer Science*, Rice University, Houston, 2017.
24. Kingston Z, Moll M, and Kavraki LE. Sampling-based methods for motion planning with constraints. In: *Annual review of control, robotics, and autonomous systems* 2018; 1(1): 159–185.
25. Liu T, Lei Y, Han L, et al. Coordinated resolved motion control of dual-arm manipulators with closed chain. *Int J Adv Robot Syst* 2016; 13(3): 1–14.
26. Dechter R, Kask K, Bin E, et al. Generating random solutions for constraint satisfaction problems. In: *Proceedings of the eighteenth national conference on artificial intelligence*, Edmonton, Alberta, Canada, 28 July–1 August 2002, pp. 15–21. AAAI.
27. Paul RP and Zhang H. Computationally efficient kinematics for manipulators with spherical wrists based on the homogeneous transformation representation. *Int J Robot Res* 1986; 5(5): 32–44.
28. Guinot JC and Bidaud P. Analysis of a robot wrist device for mechanical decoupling of end-effector position and orientation. In: Morecki A, Bianchi G, and Kedzior K (eds) *Robot manipulators and control systems 6*. Berlin: Springer, 1987, pp. 68–77.
29. Sciavicco L and Siciliano B. *Modelling and control of robot manipulators*. Berlin: Springer Science & Business Media, 2010.
30. Gochev K, Narayanan V, Cohen B, et al. Motion planning for robotic manipulators with independent wrist joints. In: *Proceedings of the 2014 IEEE international conference on robotics and automation*, Hong Kong, China, 31 May–7 June 2014, pp. 461–468. Piscataway: IEEE.
31. Rubio JJ. Discrete time control based in neural networks for pendulums. *Appl Soft Comput* 2018; 68: 821–832.
32. Rubio JJ, Garcia E, Aquino G, et al. Learning of operator hand movements via least angle regression to be teached in a manipulator. *Evol Syst* 2018. DOI: 10.1007/s12530-018-9224 -1.
33. Bertram D, Kuffner J, Dillmann R, et al. An integrated approach to inverse kinematics and path planning for redundant manipulators. In: *Proceedings of the 2006 IEEE international conference on robotics and automation*, Orlando, FL, USA, 15–19 May 2006, pp. 1874–1879. Piscataway: IEEE.
34. Berenson D, Srinivasa SS, Ferguson D, et al. Manipulation planning with workspace goal regions. In: *Proceedings of the 2009 IEEE international conference on robotics and automation*, Kobe, Japan, 12–17 May 2009, pp. 618–624. Piscataway: IEEE.
35. Vahrenkamp N, Berenson D, Asfour T, et al. Humanoid Motion planning for dual-arm manipulation and re-grasping tasks. In: *Proceedings of the 2009 IEEE/RSJ international conference on intelligent robots and systems*, St. Louis, MO, USA, 10–15 October 2009, pp. 2464–2470. Piscataway: IEEE.
36. Xian Z, Lertkultanon P, and Pham QC. Closed-chain manipulation of large objects by multi-arm robotic systems. *IEEE Robot Autom Lett* 2017; 2(4): 1832–1839.
37. Cohen B, Chitta S, and Likhachev M. Search-based planning for dual-arm manipulation with upright orientation constraints. In: *Proceedings of the 2012 IEEE international conference on robotics and automation*, Saint Paul, MN, USA, 14–18 May 2012, pp. 3784–3790. Piscataway: IEEE.
38. Sucan IA and Chitta S. Motion planning with constraints using configuration space approximations. In: *Proceedings of the 2012 IEEE/RSJ international conference on intelligent robots and systems*, Vilamoura, Portugal, 7–12 October 2012, pp. 1904–1910. Piscataway: IEEE.
39. Wei Z, Chen W, Wang H, et al. Manipulator motion planning using flexible obstacle avoidance based on model learning. *Int J Adv Robot Syst* 2017; 14(3): 1–12.
40. Li C and Berenson D. Learning object orientation constraints and guiding constraints for narrow passages from one demonstration. In: *Springer proceedings in advanced robotics* (eds Kulic D, Nakamura Y, Khatib O, et al.), 2016, pp. 197–210. Berlin: Springer.
41. Ekvall S and Kragic D. Robot learning from demonstration: a task-level planning approach. *Int J Adv Robot Syst* 2008; 5(3): 223–234.
42. Smith C, Karayiannidis Y, Nalpantidis L, et al. Dual arm manipulation—a survey. *Robot Auton Syst* 2012; 60(10): 1340–1353.
43. Tomi M, Jovanovi K, Chevallereau C, et al. Toward optimal mapping of human dual-arm motion to humanoid motion for tasks involving contact with the environment. *Int J Adv Robot Syst* 2018; 15(1): 1–9.

44. Angeles J. *Fundamentals of robotic mechanical systems: theory, methods, and algorithms*. Berlin: Springer, 2002.
45. Freddi A, Longhi S, Monteri A, et al. Redundancy analysis of cooperative dual-arm manipulators. *Int J Adv Robot Syst* 2016; 13(5): 1–14.
46. Karaman S, Walter MR, Perez A, et al. Anytime motion planning using the RRT*. In: *Proceedings of the 2011 IEEE international conference on robotics and automation*, Shanghai, China, 9–13 May 2011, pp. 9–13. Piscataway: IEEE.
47. Choudhury S, Srinivasa S, and Scherer S. Regionally accelerated batch informed trees (RABIT*): a framework to integrate local information into optimal path planning. In: *Proceedings of the 2016 IEEE international conference on robotics and automation*, Stockholm, Sweden, 16–21 May 2016, pp. 4207–4214. Piscataway: IEEE.
48. Chitta S, Sucan I, and Cousins S. Moveit! [ros topics]. *IEEE Robot Autom Mag* 2012; 19(1): 18–19.
49. Sucan IA, Moll M, and Kavraki LE. The open motion planning library. *IEEE Robot Autom Mag* 2012; 19(4): 72–82.
50. Rubio JJ. Modified optimal control with a backpropagation network for robotic arms. *IET Control Theory A* 2012; 6(14): 2216–2225.
51. Lu Q and Han QL. Mobile robot networks for environmental monitoring: a cooperative receding horizon temporal logic control approach. *IEEE T Cybernetics* 2019; 99(2): 698–711.
52. Cohen B, Chitta S, and Likhachev M. Single- and dual-arm motion planning with heuristic search. *Int J Robot Res* 2013; 33(2): 305–320.