2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

# Accelerating Bi-Directional Sampling-Based Search for Motion Planning of Non-Holonomic Mobile Manipulators

Shantanu Thakar Pradeep Rajendran Hyojeong Kim Ariyan M. Kabir Satyandra K. Gupta

***Abstract*— Determining a feasible path for nonholonomic mobile manipulators operating in congested environments is challenging. Sampling-based methods, especially bi-directional tree search-based approaches, are amongst the most promising candidates for quickly finding feasible paths. However, sampling uniformly when using these methods may result in high computation time. This paper introduces two techniques to accelerate the motion planning of such robots. The first one is coordinated focusing of samples for the manipulator and the mobile base based on the information from robot surroundings. The second one is a heuristic for making connections between the two search trees, which is challenging owing to the nonholonomic constraints on the mobile base. Incorporating these two techniques into the bi-directional RRT framework results in about 5x faster and 10x more successful computation of paths as compared to the baseline method.**

## I. Introduction

Mobile manipulators are useful for transporting and delivering objects [1]–[7]. In such transportation tasks, the manipulator can be at a "home" position on the mobile base (MB) and the need for its planning arises only when the MB arrives at the target location. However, in cases where large objects are to be transported, there may be a need to simultaneously move the manipulator and the MB, especially in cases where there is a narrow door or a passage. An example can be seen in Fig. 1(a), where the mobile manipulator needs to transport a long rod while avoiding contact with the machinery through a door.

Motion planning for nonholonomic mobile manipulators is challenging due to the high number of degrees of freedom (DOF) and the motion constraints on the MB. Sampling-based methods [8], [9] provide an effective way for motion planning of such robots. Bidirectional RRT like methods [10] provide a promising approach for motion planning of such systems. However, when the robot is carrying objects through congested environments, as shown in Fig. 1(a), narrow passages may form in the high dimensional configuration space of the robot, and it may take a long time to compute a feasible path. In factory settings with time-critical operations, this can be detrimental to the productivity. Hence, there is a need for specialized techniques to speed up motion planning for such systems. One such technique is to focus the sampling in appropriate regions of the configuration space. To achieve this focusing, we should search in regions of the configuration space which are relevant to the problem and avoid searching in regions that are not. To select relevant regions in the configuration space, we use the information provided by the description of the workspace, which is the environment surrounding the robot.

The focusing technique considered in this paper for the motion planning of mobile manipulators is shown in Fig. 1(b). Here, there are disks in the workspace along the floor from the start to the goal location of the MB. Also, there are spheres in the workspace from the starting end-effector (EE) location of the robot till the goal EE location. Focused sampling in the disks for the MB can be used to accelerate the search towards goal MB location. Similarly, sampling for the EE poses in the spheres and mapping the samples to the configuration space can also be used to accelerate the search. However, the focused sampling for the MB and the manipulator must be coordinated appropriately, so that together they are effective.

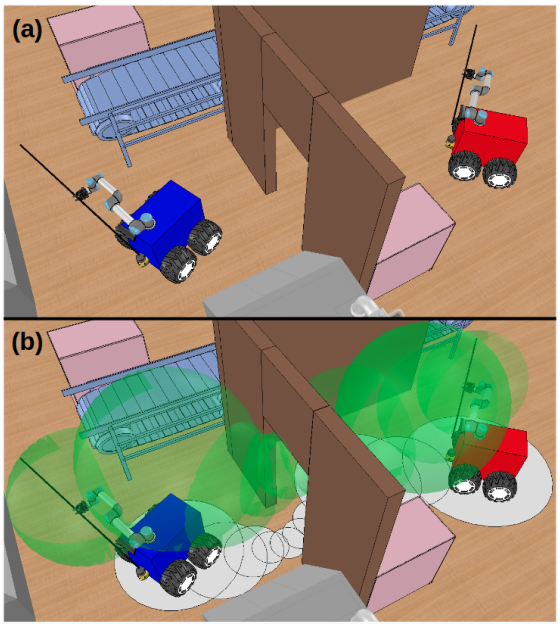


Fig. 1: (a) A nonholonomic mobile manipulator carrying a long rod in a congested factory environment (start: red, goal: blue) (b) Workspace focusing with green spheres for the manipulator and with white disks for the mobile base

We perform search by growing two rapidly exploring random trees [10] in the high dimensional configuration space of the mobile manipulator, one from the start and the other from the goal configuration, with samples being generated inside the focus regions. These two trees need to be connected in order to find a feasible path from the start to the goal. However, connecting the two trees can be challenging due to the boundary value problem that needs to be solved for a feasible path between the connecting configurations of the respective trees [8]. For randomly selected nodes (one from each tree), this boundary value problem in most cases will not have a feasible solution. Moreover, attempting to solve a boundary value problem

is computationally expensive and hence should be done between pairs of nodes that seem promising. Furthermore, even if there is a successful connection between the MB configurations of two nodes, the manipulator configurations of those two nodes may not connect because those two configurations may be very different. Hence, there is a need for a heuristic, which helps in determining the nodes of the two trees, between which connection attempts should be made. This can help in reducing the number of unsuccessful connections attempts and hence reduce computation time.

In this paper, we present a method for motion planning of nonholonomic mobile manipulators carrying large objects. Our work makes the following contributions: (a) a method to coordinate the focused sampling for the MB and the manipulator configurations, and (b) a heuristic to determine which nodes from the corresponding trees are to be selected for a connection attempt due to the nonholonomy of the MB. Finding a feasible solution quickly is challenging and of paramount importance in the context of mobile manipulation in congested environments. Therefore, our objective is to determine feasible solutions quickly.

## II. Related Work

Sampling-based motion planners like RRT, PRM and their variants [10]–[13], are very useful in a wide variety of motion planning problems with high dimensional configuration spaces. They are computationally fast as compared to search based planners. Moreover, these planners in addition to [10], [14], [15] are probabilistic complete. Planners like in [16] focus the sampling in appropriate areas of the configuration space using workspace information to significantly speed up the computation of paths for high DOF systems. In [17]–[19], workspace and configuration space sampling are scheduled appropriately in order to have enough variations in the configurations of the tree such that the benefits of focusing as well as complete random sampling can be reaped.

Work has done on task-constrained mobile manipulation planning [13], which is an extension of Informed-RRT* [14]. Since the MB in these is holonomic, determining the connection between start and goal trees is straightforward. These, along with BIT* [15], [20] can be extended to the planning of nonholonomic mobile manipulators. In these methods, the focus region is refined, as informed by each successive solution. As progressively better solutions lead to shrinking focus regions, optimal paths are found quickly. These methods perform exceptionally well for problems with the following two (not limited to) features: (a) determining the initial feasible path is easy and quick and (b) the length of the optimal path is relatively long i.e there is no maze-like structure in the configuration space. Firstly, since the focusing is done based on a feasible path, it is necessary to determine a feasible initial path quickly. Hence, the performance of these algorithms is limited by the technique used to determine the initial path. Secondly, the focus region is based on the length of a feasible path. If the length of the optimal path is large (typically when there are narrow tunnels in the configuration space), the focus region will be comparable to the entire configuration space, resulting in a minimal focusing effect. The planning for nonholonomic mobile manipulator carrying large objects in cluttered environments do not satisfy either of these conditions and hence using these methods may not be ideal.

Multi-modal and hierarchical planners like [21], [22] are very effective at generating feasible motions for mobile manipulators in complex environments under uncertainty. Planners like TGGS [23] use separate roadmaps for the base and the arm for planning. Such methods cannot be directly used when the manipulator and the MB need to coordinate their motions. Sampling can be focused by incorporating artificial potential field (APF) [24], [25] into RRT based methods like in [26], [27]. Here random samples are moved towards the goal by performing gradient descent on the potential field. The resultant configuration is then used as the random sample for RRT*. Such methods can be used for a high DOF system like a mobile manipulator in cluttered environments; however, defining the APF appropriately is challenging. Sampling-based motion planning for nonholonomic mobile manipulators has been studied in [28], where the robot has to track the end-effector trajectory. However, end-effector trajectories may not be available.

## III. Problem Statement

Let the workspace of the robot be denoted as $\mathcal{W} \subset \mathbb{R}^3$. Let $\mathbf{q} = \{x, y, \phi, \theta_1, \theta_2, \ldots, \theta_n\}$ be the configuration of mobile manipulator with a manipulator with $n$ joints. $(x, y, \phi)$ is the pose of the MB (MB), i.e the location and the orientation. Let the geometry of the mobile manipulator at a configuration $\mathbf{q}$ be represented as a set of rigid bodies $\mathcal{M}(\mathbf{q}) \subset \mathcal{W}$. Let $\mathcal{O} \subset \mathcal{W}$ be the set of workspace obstacles. Let $\mathcal{C}$ be the configuration space of the mobile manipulator containing all the valid configurations of the robot. The set of joint configurations that lead to collision is denoted by $\mathcal{C}_{obs} = \{\mathbf{q} \in \mathcal{C} : \mathcal{M}(\mathbf{q}) \cap \mathcal{O} \neq \emptyset\}$. The set of joint configurations that are collision-free is $\mathcal{C}_{free} = \{\mathbf{q} \in \mathcal{C} \backslash \mathcal{C}_{obs}\}$. Let $\mathbf{T}$ be the homogeneous transformation matrix representing the pose of the End-Effector (EE) in $\mathcal{W}$. For a given joint configuration $\mathbf{q}$, we can find $\mathbf{T}$ by applying Forward Kinematics (FK). We use the dot notation to extract quantities (e.g. $\mathbf{T}.p$ denotes the position component and $\mathbf{T}.R$ the rotation matrix). A tree node $\mathbf{n}$ consists of a configuration $\mathbf{q}$, the EE transform frame $\mathbf{T}(\mathbf{q})$. The terms nodes and configurations are used interchangeably and we make the distinction where appropriate. $\mathbf{q}.q_b$ denotes the pose of the 3-D MB and $\mathbf{q}.q_m$ denotes the $n$-D configuration of the manipulator from the node $\mathbf{q}$. $\mathcal{T}_s$ denotes the tree of nodes rooted at the start node and $\mathcal{T}_g$ for the tree rooted at the goal node.

We are interested in the path planning problem for a non-holonomic mobile manipulator. Given the geometric and kinematic robot model, workspace obstacles $\mathcal{O}$, the start and the end configurations $\mathbf{q}_s, \mathbf{q}_g$, the objective is to find a collision-free path made of configurations $Q = \{\mathbf{q}_k\}_{k=0}^{K}$ where $\mathbf{q}_0 = \mathbf{q}_s, \mathbf{q}_K = \mathbf{q}_g$ and the non-holonomic constraint is satisfied between every two consecutive $\mathbf{q}_k.q_b$.

## IV. Determining Focus Regions

In this section, we explain the procedure to determine the focus regions for the MB and the EE, within which sampling happens during run-time for the bi-directional RRT [10].

### A. Focus Region for the mobile base (MB)

The focus region for sampling MB locations is determined by generating disks on the $x-y$ plane from the start to the goal MB locations. The first disk is generated with the start location of the MB $(\mathbf{q}_s.q_b.x, \mathbf{q}_s.q_b.y)$ as the center. The distance to the nearest obstacle minus the incircle radius of the MB is taken as the radius. $k$ points are sampled randomly on the circumference for the first disk. The point closest to the goal location is chosen, and a disk is created with that point as the center and radius is determined as before. This procedure is followed until the goal location $(\mathbf{q}_g.q_b.x, \mathbf{q}_g.q_b.y)$ is inside a disk. This results in a sequence of free-space disks $\{\mathcal{D}_i\}$ from the start to the goal MB locations. This helps in focusing the sampling on being within those passages that are wide enough for the mobile manipulator to pass. Since this strategy to generate these free-area disks is greedy, we may not generate them along the shortest possible path. However, the greedy approach is fast and aligns well with our objective to determine a feasible path quickly.

### B. Focus Region for Manipulator EE

To focus the sampling for the EE, we determine a collision-free focus region in the workspace from the starting EE position to the goal EE position. We define a free space sphere $\mathcal{S}$ to be a sphere in $\mathcal{W}$ with no obstacle inside. We find a sequence of connected free space spheres $\{\mathcal{S}_i\}$ from the start EE position $(\mathbf{T}_s.p)$ to the goal EE position $(\mathbf{T}_g.p)$. The first free space sphere $\mathcal{S}_1$ is centered at $(\mathbf{T}_s.p)$. We compute the radius by finding the distance to the nearest obstacle using [29]. Then, we sample $k$ points on the surface of $\mathcal{S}_1$. We add these points to a *priority queue* based on their distances to $\mathbf{T}_g.p$. We then *pop* the $m$ closest points to the goal EE location, and of those chose closest to one of the free space disk centers in $(x, y)$. Then we make the next free space sphere $\mathcal{S}_2$. We continue this until the $\mathbf{T}_g.p$ is within a sphere. This gives a sequence of free space spheres for the EE through the workspace $\{\mathcal{S}_i\}$.

The sequence of free-space disks and spheres can be seen in Fig. 2. It can be seen that if we do not consider the disks to determine the EE free space spheres, they may pass through the narrow gap, resulting in inconsistency in the MB and the EE paths. The procedures for generating $\{\mathcal{D}_i\}$ and $\{\mathcal{S}_i\}$ is almost identical and is described in Algo. 1, (for $\{\mathcal{D}_i\}$ $m = 1$ and exclude line 9). In the algorithm, we use ball to refer to a disk (2-ball) or a sphere (3-ball) where $\mathbf{P}$ is 2D or 3D location of the center, respectively.

## V. Constructing Trees

Construction of trees for bi-directional RRT has a few standard steps like finding the nearest neighbor, extend, connect, and swapping trees. In this section, we explain how each of these for a nonholonomic mobile manipulator is implemented to construct search trees. The random sampling

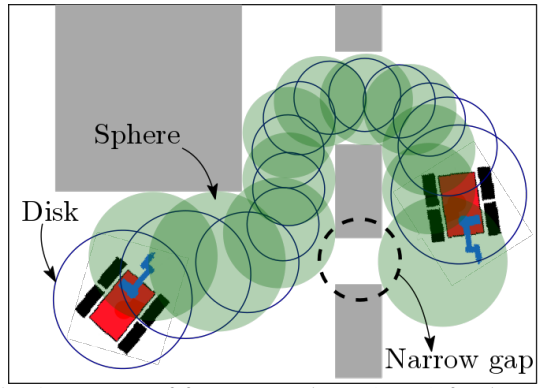


Fig. 2: A sequence of free space spheres (green) for the EE from the start to the end along the MB free space disks (blue)

**Algorithm 1** Generate free-space balls

1: **function** FS_Balls($\mathbf{P}_s, \mathbf{P}_g$)
2: $V \leftarrow \Phi$, $E \leftarrow \Phi$, PQ $\leftarrow \Phi$
3: $r_1 \leftarrow$ DistanceToObstacle($\mathbf{P}_s$)
4: $Q_1 \leftarrow$ CreateBall($\mathbf{P}_s, r_1, \Phi$)
5: $H(Q_1) \leftarrow ||\mathbf{P}_g - \mathbf{P}_s|| - r_1$
6: INSERT(PQ, $H(Q_1), Q_1$)
7: **while** $PQ \neq \Phi$ **do**
8: $Qs \leftarrow$ POP($Q, m$)
9: $Q \equiv (\mathbf{P}, r, Q_{parent}) \leftarrow$ ClosestToFreeAreaDisks($Qs$)
10: $V \leftarrow V \cup Q$
11: $E \leftarrow E \cup (Q_{parent}, Q)$
12: **if** $||\mathbf{P}_g - \mathbf{P}|| < r$ **then**
13: **return** $(V, E)$
14: **end if**
15: $P \leftarrow \mathcal{U}(Q, k)$
16: **for** $p_i \in P$ **do**
17: **if** $p_i$ is outside all other balls **then**
18: $r' \leftarrow$ DistanceToObstacle($p_i$)
19: $Q' \leftarrow$ CreateBall($p_i, r', Q$)
20: INSERT(PQ, $||\mathbf{P}_g - p_i|| - r', Q'$)
21: **end if**
22: **end for**
23: **end while**
24: **end function**

is done inside either of the focus regions. A random sample is either the complete $(3+n)$ dimension configuration $\mathbf{q}_{rand}$ (i.e in $\mathcal{C}$) where the MB position is inside one of the disks $\{\mathcal{D}_i\}$ and its orientation and the $n$ dimensions of the manipulator are uniformly random. Or, the random sample is a homogeneous transformation matrix $\mathbf{T}_{rand}$, with $\mathbf{T}_{rand}.p$ being the EE position inside one of the spheres $\{\mathcal{S}_i\}$ and $\mathbf{T}_{rand}.R$ being a random rotation matrix. By abuse of notation, this sampling is said to be in $\mathcal{W}$.

### A. Finding Nearest Neighbors

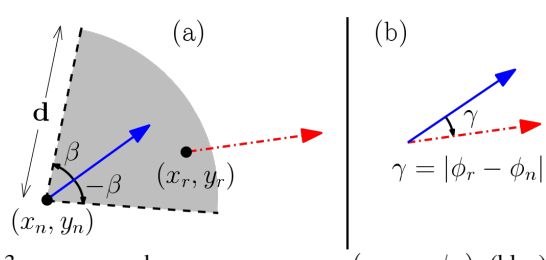


Fig. 3: $\mathbf{q}_{near}$ where, $\mathbf{q}_{near}.q_b = (x_n, y_n, \phi_n)$ (blue) is a candidate nearest neighbor configuration for the randomly sampled configuration $\mathbf{q}_{rand}$ where, $\mathbf{q}_{rand}.q_b = (x_r, y_r, \phi_r)$ (red).

The procedure to determine the nearest neighbor when sampling is done in $\mathcal{C}$ is different as compared to when

sampling is done in $\mathcal{W}$. To determine the nearest neighbor in $\mathcal{C}$, we need to determine the nearest neighbor for the MB configuration first. Due to the non-holonomic nature of the MB, the 'nearness' of manipulator configurations does not matter unless the MB poses are near. The Fig. 3(a) shows the criterion for choosing the nearest poses for the MB. There is a need to take the orientation of the MB of the sampled configuration ($\mathbf{q}_{rand}$) into account. The criterion is that $(x_r, y_r)$ should lie within the gray arc (within $\mathbf{d}$) described by $\beta$. Further, $\mathbf{sgn}(\gamma) = \mathbf{sgn}(\beta)$ and $|\gamma| \leq \alpha$, where $\gamma = |\phi_r - \phi_n|$ is the difference in the orientations of the MB at $\mathbf{q}_{rand}$ and $\mathbf{q}_{near}$. We determine $K$ nearest nodes only based on the above criterion for nearness and of those, we choose the one with the least Euclidean distance between the manipulator configurations $\mathbf{q}_{near}.q_m$ and $\mathbf{q}_{rand}.q_m$. Here, $\gamma$, $\beta$, $\alpha$ , $K$ and $\mathbf{d}$ are parameters that need to be set. The nearest neighbor, when sampling is done in $\mathcal{W}$ is the node in the tree with the closest EE location to $\mathbf{T}_{rand}.p$.

### B. Extend towards the Random Sample

Once the $\mathbf{q}_{near}$ is chosen according the the conditions in Sec. V-A, we extend the MB configuration as shown in Fig. 4. We rotate the MB at $(x_n, y_n)$ to point towards $(x_r, y_r)$ and move along the line joining $(x_n, y_n)$ and $(x_r, y_r)$ by a distance $\mathbf{d}_e$ to reach the point $(x_e, y_e)$. The extended MB configuration is $\mathbf{q}_e.q_b = (x_e, y_e, \phi_e)$, where $\phi_e = \arctan(y_r - y_n, x_r - x_n)$. Further, we interpolate the manipulator configuration towards $\mathbf{q}_{rand}.q_m$ from $\mathbf{q}_{near}.q_m$ and check for collision at intermediate steps. The extended configuration is $\mathbf{q}_e$.

When the random sampling is in $\mathcal{W}$, we drive the mobile manipulator at the near configuration $\mathbf{q}_{near}$ towards $\mathbf{T}_{rand}$ using Jacobian pseudo-inverse method [30]. We use the analytical Jacobian $\mathbf{J}$ of the EE as a function of the robot configuration $\mathbf{q}$ as $\mathbf{J}\dot{\mathbf{q}} = \dot{\mathbf{x}}$, where $\mathbf{x}$ is the $6 \times 1$ EE pose. As the mobile manipulator is a $3 + n$ DOF robot, $\mathbf{J}$ is a $6 \times (3 + n)$ matrix. Hence, we can write $\dot{\mathbf{q}} = \mathbf{J}^+\dot{\mathbf{x}} + \mathbf{N}\dot{\psi}$ Where, $\mathbf{J}^+$ is the Moore-Penrose inverse, $\mathbf{J}^+ = \mathbf{J}^T(\mathbf{J}\mathbf{J}^T)^{-1}$. $\mathbf{N}$ is a matrix representing the projection into the null space of $\mathbf{J}$ and $\dot{\psi}$ is an arbitrary $(3 + n)$-dimensional velocity vector. Here, $\mathbf{N}$ projects the velocity $\dot{\mathbf{q}}$ into the null-space of $\mathbf{J}$ and the corresponding motion does not affect the EE tracking task. The second term in the RHS can be used to obtain different joint velocities $\dot{\mathbf{q}}$ with the same EE velocity $\dot{\mathbf{x}}$. This can be exploited to make sure that the non-holonomic constraints of the MB are satisfied. In this regard, we represent the non-holonomic constraints of the MB i.e $\dot{x}\sin\phi - \dot{y}\cos\phi = 0$ as $\mathbf{J}_h\dot{\mathbf{q}} = \dot{\mathbf{x}}_h$, where $\mathbf{J}_h$ is a $1 \times$ (3+n) vector with all the terms zero except the first and the second terms which are $\sin\phi$ and $-\cos\phi$ respectively.

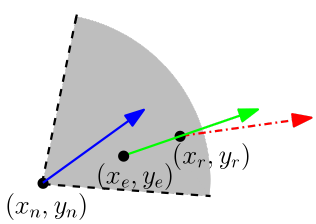

Fig. 4: Extending from $\mathbf{q}_{near}.q_b = (x_n, y_n, \phi_n)$ (blue) to $\mathbf{q}_{rand}.q_b = (x_r, y_r, \phi_r)$ (red), lands up at $\mathbf{q}_e.q_b = (x_e, y_e, \phi_e)$ (green) at a distance $\mathbf{d}_e$ from $(x_n, y_n)$

$\dot{\mathbf{x}}_h$ is $\mathbf{0}$. Combining these, we get

$$\dot{\psi} = (\mathbf{J}_h\mathbf{N})^+(\dot{\mathbf{x}}_h - \mathbf{J}_h\mathbf{J}^+\dot{\mathbf{x}}) \tag{1}$$

$$\dot{\mathbf{q}} = \mathbf{J}^+\dot{\mathbf{x}} + \mathbf{N}(\mathbf{J}_h\mathbf{N})^+(\dot{\mathbf{x}}_h - \mathbf{J}_h\mathbf{J}^+\dot{\mathbf{x}}) \tag{2}$$

$$\Delta\mathbf{q} \approx \mathbf{J}^+\Delta\mathbf{x} - \mathbf{N}(\mathbf{J}_h\mathbf{N})^+\mathbf{J}_h\mathbf{J}^+\Delta\mathbf{x} \tag{3}$$

We determine the Jacobian $\mathbf{J}$ at $\mathbf{q}_{near}$. $\Delta\mathbf{x}$ is the difference between the EE pose at $\mathbf{q}_{rand}$ and $\mathbf{q}_{near}$. We perform collision checks along $\Delta\mathbf{q}$ from $\mathbf{q}_{near}$ in steps to reach $\mathbf{q}_e$ which in the best case is $\mathbf{q}_{near} + \Delta\mathbf{q}$.

### C. Heuristics for Connecting Trees

The start and goal trees grow in the configuration space with sampling either in configuration space or workspace. We attempt a connection between the two trees when they have equal number of nodes. We select the newest node $\mathbf{q}_{new}$ (or $\mathbf{q}_e$) of one of the trees and use the strategy described in Sec. V-A to determine its nearest neighbor ($\mathbf{q}_o$) in the other tree. We then attempt a connection between the two nodes using a strategy similar to the one described in Sec. V-B. We rotate the MB at its place at $(\mathbf{q}_o.q_b.x, \mathbf{q}_o.q_b.y)$ to point towards $(\mathbf{q}_e.q_b.x, \mathbf{q}_e.q_b.y)$ and then drive towards it. After reaching $(\mathbf{q}_e.q_b.x, \mathbf{q}_e.q_b.y)$, we rotate it at its place to make its orientation $\mathbf{q}_e.q_b.\phi$. We then interpolate the manipulator configuration from $\mathbf{q}_o.q_m$ to $\mathbf{q}_e.q_m$ along this MB path. A connection is successful if this connecting path of the mobile manipulator is collision-free and satisfies the joint limits. The heuristic refers to the strategy used to determine candidate nearest neighbors in the other tree or the same tree. Here we ignore those nodes which do not satisfy the conditions described in Sec. V-A when determining the nearest nodes.

## VI. The HS-Bi-RRT Algorithm

In this section, we formalize the algorithm based on focused sampling for the MB and EE. The over-all algorithm, Hybrid Sampling-based Bi-directional RRT (HS-Bi-RRT), is shown in Algo 2. We initialize the two trees $(\mathcal{T}_s, \mathcal{T}_g)$ with the start and the goal configurations as well as the sphere and disk sequences (lines 2-5). These are reversed for the goal tree. The current tree is initialized as the start tree and the other tree is initialized as the goal tree. For each tree, there is a current sphere and a current disk in which sampling will be done (lines 7-10). Both these are represented by 'ball' in the Algo. 2 from lines 12 onward, as we will perform similar operations on both. $\mathcal{R}_s$ is a sampling ratio which tells the probability of sampling in the configuration space as compared to the workspace. In the function NextRandSample we sample in both with equal probability, i.e $\mathcal{R}_s$ is 0.5. When the sampling is to be done in $\mathcal{C}$, we generate a random sample in the current disk via a normal distribution with a mean at the center and standard deviation $\sigma$ for the MB location. The MB orientation and the manipulator configuration are sampled at random. Further, 10% of the sampling in $\mathcal{C}$ is done in the entire configuration space with no focusing for MB, to maintain the completeness of the algorithm. When sampling is to be done in $\mathcal{W}$, we similarly sample the EE location in the current sphere with a normal distribution with a mean at the center and standard deviation $\sigma$. This standard deviation is specific for the current

ball (i.e the current sphere and the current disk) and the current tree. The idea behind using the normal distribution for sampling is that there is relatively more free space for the robot to move near the center. $l$ is the label for whether the sampling is done in $\mathcal{C}$ or $\mathcal{W}$. Depending on $l$ we find the nearest nodes and extend as explained in Secs. V-A & V-B.

**Algorithm 2** Hybrid Sampling based Bi-directional RRT

1: **function** HS-BI-RRT($\mathbf{q}_s, \mathbf{q}_g, \{\mathcal{D}_{n_1}\}, \{\mathcal{S}_{n_2}\}, \mathcal{R}_s$ )
2: $\mathcal{T}_s \leftarrow \{\text{NODE}(\mathbf{T}_s, \mathbf{q}_s, \mathcal{S}_1, \mathcal{D}_1)\}$,
3: $\mathcal{T}_g \leftarrow \{\text{NODE}(\mathbf{T}_g, \mathbf{q}_g, \mathcal{S}_{n_2}, \mathcal{D}_{n_1})\}$,
4: $\mathcal{T}_s$.SPHERES $\leftarrow \{\mathcal{S}_{n_2}\}$, $\mathcal{T}_g$.SPHERES $\leftarrow$ REVERSE$\{\mathcal{S}_{n_2}\}$
5: $\mathcal{T}_s$.DISK $\leftarrow \{\mathcal{D}_{n_1}\}$, $\mathcal{T}_g$.DISK $\leftarrow$ REVERSE$\{\mathcal{D}_{n_1}\}$
6: $\mathcal{T}_\mathbf{c} \leftarrow \mathcal{T}_s$, $\mathcal{T}_\mathbf{o} \leftarrow \mathcal{T}_g$
7: $\mathcal{T}_\mathbf{c}$.CURRENT_SPHERE $\leftarrow \mathcal{T}_s$.SPHERES$\{1\}$
8: $\mathcal{T}_\mathbf{o}$.CURRENT_SPHERE $\leftarrow \mathcal{T}_g$.SPHERES$\{1\}$
9: $\mathcal{T}_\mathbf{c}$.CURRENT_DISK $\leftarrow \mathcal{T}_s$.DISKS$\{1\}$
10: $\mathcal{T}_\mathbf{o}$.CURRENT_DISK $\leftarrow \mathcal{T}_g$.DISKS$\{1\}$
11: **while** $t < t_{max}$ **do**
12: $(\mathbf{S}_{rand}, l) \leftarrow$ NEXTRANDSAMPLE($\mathcal{T}_\mathbf{c}$.CURRENT_BALL, $\mathbf{T}_g, \mathcal{R}_s$)
13: $\mathbf{q}_{near} \leftarrow$ NEAREST NODE($\mathcal{T}_\mathbf{c}, \mathbf{S}_{rand}, l$)
14: $(\mathbf{T}_{new}, \mathbf{q}_{new}) \leftarrow$ EXTEND($\mathbf{q}_{near}, \mathbf{S}_{rand}, l$)
15: **if** $\mathbf{q}_{new} \neq \Phi$ **then**
16: $\mathcal{T}_\mathbf{c} \leftarrow$ UPDATE TREE($\mathbf{q}_{new}$)
17: $\mathcal{T}_\mathbf{c}$.CURRENT_BALL.$\sigma = (1-\lambda) \cdot \mathcal{T}_\mathbf{c}$.CURRENT_BALL.$\sigma$
18: **else**
19: $\mathcal{T}_\mathbf{c}$.CURRENT_BALL.$\sigma = (1+\lambda) \cdot \mathcal{T}_\mathbf{c}$.CURRENT_BALL.$\sigma$
20: **end if**
21: **if** $\mathcal{T}_\mathbf{c}$.CURRENT_BALL.$\sigma < \xi''$ **then**
22: $\mathcal{T}_\mathbf{c}$.CURRENT_BALL.$\sigma = \xi$
23: $\mathcal{T}_\mathbf{c}$.CURRENT_BALL $\leftarrow \mathcal{T}_\mathbf{c}$.CURRENT_BALL.**child**
24: **end if**
25: **if** $\mathcal{T}_\mathbf{c}$.CURRENT_BALL.$\sigma > \xi'$ **then**
26: $\mathcal{T}_\mathbf{c}$.CURRENT_BALL.$\sigma = \xi$
27: $\mathcal{T}_\mathbf{c}$.CURRENT_BALL $\leftarrow \mathcal{T}_\mathbf{c}$.CURRENT_BALL.**parent**
28: **end if**
29: **if** TOTALNODES($\mathcal{T}_\mathbf{c}$) > TOTALNODES($\mathcal{T}_\mathbf{o}$) **then**
30: SWAP($\mathcal{T}_\mathbf{c}, \mathcal{T}_\mathbf{o}$)
31: **else**
32: $(\{\Theta_{\mathbf{connects}}\}, \text{FLAG}) \leftarrow$ CONNECT($\mathcal{T}_\mathbf{c}, \mathcal{T}_\mathbf{o}$)
33: **if** FLAG **is** SUCCESS **then**
34: **return** $\mathbf{Path}(\mathcal{T}_\mathbf{c}, \mathcal{T}_\mathbf{o}, \{\Theta_{\mathbf{connects}}\})$
35: **end if**
36: **end if**
37: **end while**
38: **end function**

When we have a new node $\mathbf{q}_{new}$ (or $\mathbf{q}_e$) after sampling within a sphere or disk with the corresponding $\sigma$, it suggests that this sphere or disk is promising for further sampling. We then reduce the $\sigma$ of this disk or sphere so as to exploit it by further focusing the sampling in it, like in line 17. However, if there is a failure in generating a $\mathbf{q}_{new}$, it conveys that we need to explore more in order to grow the tree instead of focusing more. Hence, we increase the standard deviation $\sigma$ of that disk or sphere for sampling (line 19). The increase and decrease of the standard deviation are parameterized by $\lambda$ (lines 17 & 19). This $\lambda$ is different for a sphere and a disk and needs to be tuned appropriately.

There has to be a limit on the exploration and exploitation being done. If we exploit till $\sigma$ is too small, then we may end up in very similar locations for the random samples. Once exploitation is sufficiently done, i.e the reduction of the $\sigma$ of the current sphere or the current disk less than a set value ($\xi'$), we reset the $\sigma$ to the original value $\xi$ and move ahead to the next sphere or disk in the sequence. (lines 21-24). Further, if $\sigma$ is too large, we may end up not focusing at all. That also suggests that there have been repeated failures in generating a new configuration. Hence, there is a need to retract and go to the previous sphere or disk in the sequence. Further, the standard deviation needs to be reset for the sphere or disk (lines 25-28). If the current ball is the first or the last, we reset the $\sigma$ and do not move to a neighboring sphere or disk.

To make sure that both the start tree and the goal tree grow equally, we swap them when the current tree has a greater number of nodes [17] (lines 29-30). Further, when they have an equal number of nodes, we make a connection attempt using the procedure explained in Sec. V-C (line 32). The algorithm returns a feasible path once a successful connection is found (line 34) within the timeout.

## VII. Results

### A. Experimental Setup

We implemented the planner HS-Bi-RRT in MATLAB on a Dell workstation with Intel Xeon 3.5 GHz and 32 GB RAM. We benchmarked our method with existing algorithms and variations of our method. WD-Bi-RRT refers to when $\mathcal{R}_s$ is zero, and the sampling for the MB happens inside the workspace disk only, whereas the sampling for the manipulator is done at random in its configuration space. WS-Bi-RRT is a variant of our algorithm when $\mathcal{R}_s$ is 1, i.e. sampling happens only in the workspace spheres for the EE location. All of these use the heuristic for connecting the two trees. Bi-RRT refers to the standard RRT-connect [10] with sampling in the entire configuration space along. Whereas, Bi-RRT+ is Bi-RRT with the heuristic for connection.

We measure the time and the average path cost, which is the distance traveled by the object being carried. We have tested our method on a diverse set of 20 scenarios with a failure time-out of 1000 seconds. The diversity in scenarios is introduced by having a differential drive mobile manipulator to carry different large objects, which are planar (like a broom, rod, plate, wheel) and 3D (trash bin, chair) in different types of challenging scenarios. We have also extended the algorithm to a bimanual mobile manipulator carrying two objects in its two arms. The 20 scenarios are a combination of the robots carrying these objects in various congested environments. A representative subset of the test scenarios is shown in Fig. 5. These scenarios contain easy and challenging cases, to show the relative performances of the competing methods. The objects being carried are large, and it can be observed that we cannot have the manipulator in a "home" position and carry them without coordinating the motions of the MB and the manipulator. The parameters are as follows: $\alpha = \frac{\pi}{8}$, $|\beta| = \frac{\pi}{8}$, $\lambda = 0.2$, $\xi = 0.05$, $\xi' = 0.75$, $\xi'' = 3$. These were determined empirically by testing in numerous scenarios.

### B. Discussion

In Tab. I, we show the computation time and path cost for the five cases. The path cost signifies how good the initial solution is. A better and faster initial solution, when coupled with an any-time algorithm like BIT* may produce the optimal solution quicker. When tested over 20 scenarios, we observe that HS-Bi-RRT is, on average $\sim 3$ times faster than

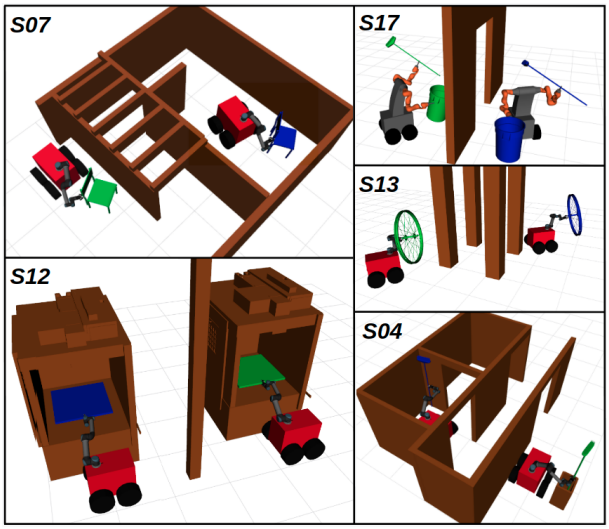


Fig. 5: The 5 of the 20 test scenes for which the numerical results are presented in Tab. I. Green: Start, Blue: Goal configurations

| Scene | Average Computation Time (s) | | | | Average Path Cost (m) | | | |
|---|---|---|---|---|---|---|---|---|
| | HS-Bi-RRT | WD-Bi-RRT | WS-Bi-RRT | Bi-RRT+ | HS-Bi-RRT | WD-Bi-RRT | WS-Bi-RRT | Bi-RRT+ |
| *S04* | 37.5 | 163.4 | 832.0 | 191.9 | 20.0 | 23.9 | 16.6 | 25.0 |
| *S07* | 32.9 | 175.8 | 727.7 | 200.4 | 26.0 | 29.1 | 20.4 | 49.2 |
| *S12* | 20.9 | 75.7 | 51.0 | 101.5 | 18.7 | 60.3 | 10.2 | 72.6 |
| *S13* | 19.1 | 33.0 | 131.2 | 29.6 | 19.7 | 26.2 | 12.8 | 35.6 |
| *S17* | 77.3 | 302.4 | 802.1 | 399.7 | 76.2 | 89.1 | 51.1 | 99.7 |

TABLE I: The average computation time and average path length of the EE for he 5 test cases. The average is over 20 runs. Computation time includes the time required for generating free-space balls.

the next best method for that scene. We use the heuristic in all the competing methods. If we select nodes for connection based on direct Euclidean distance in $\mathcal{C}$ like in [13] instead of using the heuristic, there is on average about 73% failure rate for HS-Bi-RRT, 81% for WD-Bi-RRT, 71% for WS-Bi-RRT and 90% for Bi-RRT. Further, when there is a success, the computation time is $\sim 9$ times more on average. This shows the importance of using the heuristic for connections.

In scene *S04* the robot carries a long broom from a box to the room via a narrow passage. In *S07* the robot transports a chair from one room to another. In both, there is a single narrow passage to be traversed to reach the goal for the MB. It can be observed in Tab. I that for these cases, HS-Bi-RRT, which has focusing for the MB as well as the manipulator, produces a path significantly faster than the other three. WD-Bi-RRT, which has focusing for the MB, produces a solution faster than Bi-RRT+. It was observed that the time taken by WS-Bi-RRT is high due to the infeasibility to determine successful connections between the two trees. This happens due to biasing towards very different configurations for the manipulator, which do not connect in such narrow passages. However, WS-Bi-RRT produces the best quality solution. This can be attributed to the greedy sampling for the EE pose towards the goal. In these scenes, each method was successful in producing a solution for all the 20 attempts.

In scene *S12*, the robot has to transport a large metallic sheet from one machine to another while avoiding a pillar on the way. The pillar may result in the workspace spheres taking a different homotopy class as compared to the workspace disks. The MB cannot pass between the pillar and the machine as the gap is too small. This can result in configurations in the two trees that may never connect due to collision with the pillar. This is where the advantage of sampling for the manipulator in the entire configuration space is evident. However, it should also be observed in Tab. I that HS-Bi-RRT still produces a better solution quicker than WD-Bi-RRT on average. This is due to the fact that focusing for the manipulator, even if misguided, helps when coupled with complete configuration space sampling. Here, WS-Bi-RRT is successful in producing a solution within 1000 seconds only $\sim 60\%$ of the times. However, when WS-Bi-RRT does produce a solution, it does so faster than WD-Bi-RRT and Bi-RRT+ and with a shorter path length.

Scene *S13* shows the cases when there are multiple homotopy classes for the MB. Here, the time taken by HS-Bi-RRT is lower yet comparable to that of WD-Bi-RRT and Bi-RRT+. This is due to the fact that there is enough free space for them to grow and find connections. In fact, Bi-RRT+ is faster than WD-Bi-RRT as it can find a longer path in the free space quickly. In such cases with multiple homotopy classes, the quality of solution by HS-Bi-RRT is highly dependent on the initial sequence of workspace disks. It may not find the correct homotopy class in each run. We can easily extend our algorithm to manually select a homotopy class to bias the sampling of the MB configurations.

In *S17*, we show that our algorithm can be extended to a nonholonomic bi-manual mobile manipulator for transporting two different objects in the two arms and passing through a narrow door. The algorithm is modified to have two different workspace sphere sequences for the two EEs. Here, it can be observed that results follow the same trends as in the *S04* and *S07*. Moreover, WS-Bi-RRT fails to produce a result in 7 out of 20 runs. In this example, determining a feasible connection between the two trees is even more challenging due to the presence of 2 manipulators.

The video of simulations of the robot trajectory and physical experiments can be found in the following link: https://youtu.be/UfxfBq3bjHw

## VIII. Conclusions and Future Work

In this paper, we have addressed the motion planning for nonholonomic mobile manipulators using a sampling-based method described by our algorithm HS-Bi-RRT. In such problems, determining a feasible solution is a challenge due to narrow passages in the configuration space and the nonholonomic constraints of the MB. Focusing the sampling in different regions of the configuration space can significantly reduce the computation time. For a mobile manipulator, this focusing must be done for the MB and the manipulator in appropriate regions to accelerate the growth of the trees towards each other. The primary contribution of this paper is a method to coordinate the focusing for the MB and the manipulator in the workspace and grow the trees in appropriate directions. Moreover, we have presented a heuristic for determining connections between the MB first and then find connections for the manipulator. It was observed that without this heuristic, each of the algorithms

have a high failure rate to even compute a path. Furthermore, we also sample in the entire configuration space to make sure that the algorithm is probabilistic complete. We have shown that these together significantly reduce the computation time as compared to existing methods in diverse scenarios. In this work, we have not addressed cases where the workspace disks are misguided, for example, having a short door so that the mobile manipulator cannot pass, but the mobile base alone can pass under. Since our method is probabilistically complete, it will find a path in this case as well, but the computation time will be significantly higher. In the future, we also plan to use the techniques in [31] for closed-loop implementation of the trajectories generated by our method.

**Acknowledgment:** This work is supported in part by National Science Foundation Grants #1634431 and #1925084. Opinions expressed are those of the authors and do not necessarily reflect the views of the sponsor.

## REFERENCES

[1] S. Thakar, L. Fang, B. Shah, and S. K. Gupta, "Towards time-optimal trajectory planning for pick-and-transport operation with a mobile manipulator," in *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2018, pp. 981–987.

[2] A. M. Kabir, A. Kanyuck, R. K. Malhan, A. V. Shembekar, S. Thakar, B. C. Shah, and S. K. Gupta, "Generation of synchronized configuration space trajectories of multi-robot systems," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8683–8690.

[3] S. Bøgh, M. Hvilshøj, M. Kristiansen, and O. Madsen, "Identifying and evaluating suitable tasks for autonomous industrial mobile manipulators (aimm)," *The International Journal of Advanced Manufacturing Technology*, vol. 61, no. 5-8, pp. 713–726, 2012.

[4] S. Thakar, A. Kabir, P. Bhatt, R. Malhan, P. Rajendran, B. Shah, and S. K. Gupta, "Task assignment and motion planning for bimanual mobile manipulation," in *IEEE International Conference on Automation Science and Engineering (CASE)*, Vancouver, Canada, August 2019.

[5] A. M. Kabir, S. Thakar, P. M. Bhatt, R. K. Malhan, P. Rajendran, B. C. Shah, and S. K. Gupta, "Incorporating motion planning feasibility considerations during task-agent assignment to perform complex tasks using mobile-manipulators," in *IEEE International Conference on Robotics and Automation*, Paris, France, June 2020.

[6] S. Thakar, P. Rajendran, V. Annem, A. Kabir, and S. K. Gupta, "Accounting for part pose estimation uncertainties during trajectory generation for part pick-up using mobile manipulators," in *IEEE International Conference on Robotics and Automation (ICRA)*, Montreal, Canada, May 2019, pp. 1329–1336.

[7] V. Annem, P. Rajendran, S. Thakar, and S. K. Gupta, "Towards remote teleoperation of a semi-autonomous mobile manipulator system in machine tending tasks," in *ASME Manufacturing Science and Engineering Conference (MSEC)*, Erie, USA, June 2019.

[8] S. M. LaValle and J. J. Kuffner Jr, "Randomized kinodynamic planning," *The international journal of robotics research*, vol. 20, no. 5, pp. 378–400, 2001.

[9] A. Shkolnik, M. Walter, and R. Tedrake, "Reachability-guided sampling for planning under differential constraints," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 2859–2865.

[10] J. J. Kuffner and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 2. IEEE, 2000, pp. 995–1001.

[11] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[12] S. M. Lavalle, "Planning Algorithms," *Journal of Chemical Information and Modeling*, vol. 53, no. 9, pp. 1689–1699, 2013.

[13] F. Burget, M. Bennewitz, and W. Burgard, "Bi 2 rrt*: An efficient sampling-based path planning framework for task-constrained mobile manipulation," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 3714–3721.

[14] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed rrt*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 2997–3004.

[15] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Batch informed trees (bit*): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 3067–3074.

[16] M. Rickert, A. Sieverling, and O. Brock, "Balancing exploration and exploitation in sampling-based motion planning," *IEEE Transactions on Robotics*, vol. 30, no. 6, pp. 1305–1317, 2014.

[17] P. Rajendran, S. Thakar, A. Kabir, B. Shah, and S. K. Gupta, "Context-dependent search for generating paths for redundant manipulators in cluttered environments," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, Macau, China, November 2019.

[18] P. Rajendran, S. Thakar, and S. K. Gupta, "User-guided path planning for redundant manipulators in highly constrained work environments," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2019, pp. 1212–1217.

[19] S. Thakar, P. Rajendran, A. M. Kabir, and S. K. Gupta, "Manipulator motion planning for part pick-up and transport operations from a moving base," *IEEE Transactions on Automation Science and Engineering*, 2020.

[20] S. Choudhury, J. D. Gammell, T. D. Barfoot, S. S. Srinivasa, and S. Scherer, "Regionally accelerated batch informed trees (rabit*): A framework to integrate local information into optimal path planning," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 4207–4214.

[21] V. Pilania and K. Gupta, "A hierarchical and adaptive mobile manipulator planner with base pose uncertainty," *Autonomous Robots*, vol. 39, no. 1, pp. 65–85, 2015.

[22] V. Pilania and K. Gupta, "Mobile manipulator planning under uncertainty in unknown environments," *The International Journal of Robotics Research*, vol. 37, no. 2-3, pp. 316–339, 2018.

[23] C. Bowen and R. Alterovitz, "Accelerating motion planning for learned mobile manipulation tasks using task-guided gibbs sampling," in *Robotics Research*. Springer, 2020, pp. 251–267.

[24] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Autonomous robot vehicles*. Springer, 1986, pp. 396–404.

[25] H. G. Tanner and K. J. Kyriakopoulos, "Nonholonomic motion planning for mobile manipulators," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 2. IEEE, 2000, pp. 1233–1238.

[26] A. H. Qureshi and Y. Ayaz, "Potential functions based sampling heuristic for optimal path planning," *Autonomous Robots*, vol. 40, no. 6, pp. 1079–1093, 2016.

[27] Z. Tahir, A. H. Qureshi, Y. Ayaz, and R. Nawaz, "Potentially guided bidirectionalized rrt* for fast optimal path planning in cluttered environments," *Robotics and Autonomous Systems*, vol. 108, pp. 13–27, 2018.

[28] G. Oriolo and C. Mongillo, "Motion planning for mobile manipulators along given end-effector paths," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE, 2005, pp. 2154–2160.

[29] P. F. Felzenszwalb and D. P. Huttenlocher, "Distance transforms of sampled functions," *Theory of computing*, vol. 8, no. 1, pp. 415–428, 2012.

[30] F. Caccavale and B. Siciliano, "Quaternion-based kinematic control of redundant spacecraft/manipulator systems," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, vol. 1. IEEE, 2001, pp. 435–440.

[31] R. Colombo, F. Gennari, V. Annem, P. Rajendran, S. Thakar, L. Bascetta, and S. K. Gupta, "Parameterized model predictive control of a nonholonomic mobile manipulator: A terminal constraint-free approach," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2019, pp. 1437–1442.