

A Novel Method for Trajectory Planning of Cooperative Mobile Manipulators

Hossein Bolandi, Amir Farhad Ehyaei

College of Electrical Engineering, Iran University of Sciences and Technology, Namak, Tehran, Iran

ABSTRACT

We have designed a two-stage scheme to consider the trajectory planning problem of two mobile manipulators for cooperative transportation of a rigid body in the presence of static obstacles. In the first stage, with regard to the static obstacles, we develop a method that searches the workspace for the shortest possible path between the start and goal configurations, by constructing a graph on a portion of the configuration space that satisfies the collision and closure constraints. The final stage is to calculate a sequence of time-optimal trajectories to go between the consecutive points of the path, with regard to the nonholonomic constraints and the maximum allowed joint accelerations. This approach allows geometric constraints such as joint limits and closed-chain constraints, along with differential constraints such as nonholonomic velocity constraints and acceleration limits, to be incorporated into the planning scheme. The simulation results illustrate the effectiveness of the proposed method.

Key words: Closed kinematic chains, collision avoidance, cooperative manipulation, high-dimensional systems, mobile manipulators, nonholonomic motion planning, probabilistic complete planner

INTRODUCTION

Cooperative mobile manipulators have received extensive attention in recent years due to their wide range of applications including transporting long and heavy materials.^[1-16] While cooperative mobile manipulators can offer advantages over a single mobile manipulator in terms of their capability to carry out difficult tasks, motion planning of these systems is complicated by the need to maintain a closed-chain structure, the so-called closure constraint. A cooperative mobile manipulator system, in addition to closed-chain constraints, is subject to kinodynamic and nonholonomic constraints in the form of differential equations,^[17] along with collision constraints, which make motion planning even more difficult. However, these constraints are necessary to find a reliable and efficient solution, often called a trajectory.

Many previous studies have investigated the trajectory generation for cooperative mobile manipulators, under various conditions.^[1-16] Two approaches for a solution to this problem can be distinguished,^[18,19] namely: The decoupled and direct approaches. The decoupled approach, involves first searching for a path in the configuration space and then finding a time-optimal time scaling for the path, subject to the actuator limits.^[18] This has the desirable benefit of decomposing the complexity of motion planning problems in two steps, as mentioned earlier. However, the path from the first stage might not be transformable into

an executable trajectory and the cost associated with the final trajectory could be expensive. The main reason for these problems is that the approach ignores the differential constraints of the action model in the first stage.^[19] To overcome these deficiencies, a direct approach has been developed, in which, differential constraints are considered in the planning process. Most of the literatures available on the direct approach, compute exact trajectories using the optimal control theory and nonlinear optimization methods for some specific low-dimensional problems.^[1-11] The main drawback of these methods is that the number of variables and the complexity of the formulation rapidly increase for problems with high degrees of freedom. To this end, in recent years, probabilistic complete methods have been extensively studied within the direct framework.^[13-16]

In contrast to other available methods that construct a global representation of the configuration space, these methods discretize their representation of free configuration space to handle high-dimensional configuration spaces. Due to the effectiveness of these methods in cooperative manipulation systems,^[12,20,21] some recent efforts have been directed to generalize them for cooperative mobile manipulators.^[13-16] For instance, in^[13] a single query method based on the Rapidly-exploring Random Tree algorithm (RRT) has been presented. The method constructs a randomized tree between the start and goal configurations and searches

Address for correspondence:

Hossein Bolandi, College of Electrical Engineering, Iran University of Science and Technology, Narmak, P.O.Box 16844, Tehran, Iran
E-mail: h_boland@iust.ac.ir

for an optimal path during the construction phase. A disadvantage of this method is that the generated tree is only valid for certain start and goal configurations; hence, for a new query, another tree will have to be constructed, which restricts the method to a limited range of applications.^[22] Therefore,^[14] proposes a multiple query two-stage method, in which first a graph, namely, the 'Roadmap' is constructed, based on the Probabilistic Roadmap Method (PRM) by neglecting the presence of obstacles and assuming a fixed location for the bases of mobile manipulators. Then, it populates the environment with copies of the kinematic roadmap in random locations and connects the collision-free configurations of the same closure type to build the final roadmap. The method, because of its higher speed in the query phase and ability to change its goal configuration in an online manner, is more executable in real-time practical situations. However, in this method, the probability of satisfying loop closure equations in a randomly sampled configuration is near zero and this fact lowers the performance of the algorithm. To solve this problem in^[15,16] a simple and general geometric-guided sampling algorithm called Random Loop Generator (RLG) has been proposed, which notably increases the probability of obtaining real solutions when solving the loop closure equations. The main deficiency of the method presented for randomized motion planning of cooperative mobile manipulators is that they ignore differential constraints in the motion planning process.^[23] However, these constraints will be considered to calculate a reliable and efficient solution.

Therefore, in this study we propose a novel two stage scheme that considers the trajectory planning problem of two mobile manipulators for cooperative transportation of a rigid body. The environment is supposed to be a 3-D space that includes static obstacles. The method finds an optimal trajectory in which such constraints as nonholonomic and closed-chain constraints, along with the joint and acceleration limits can be easily be dealt with in the presence of static obstacles. The method utilizes the advantages of direct and decoupled approaches along with the power of probabilistic methods in handling high dimensional configuration spaces to have a reliable and fast trajectory planning algorithm.

The article is organized as follows: Section 2 introduces the model of the cooperative system. In section 3, we design the details of our new method for trajectory planning of the cooperative mobile manipulators under differential constraints in the presence of static obstacles. Finally, we present simulation results in section 4, to show the effectiveness of the algorithm, and some concluding remarks in section 5.

MODEL OF COOPERATIVE SYSTEM

Figure 1 shows our selected model, including a cooperative system of two-wheeled mobile manipulators

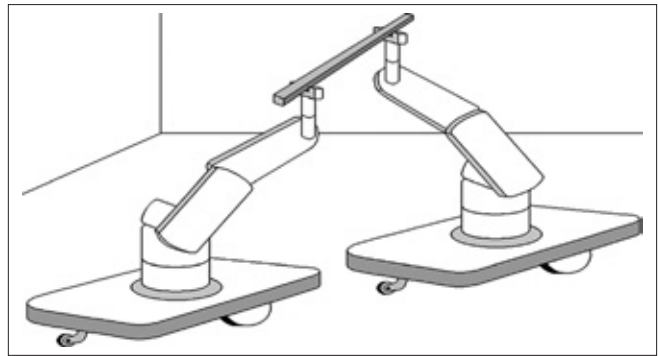


Figure 1: Cooperative transportation by a dual mobile manipulator system

transporting a common payload. Each mobile manipulator module consists of a wheeled mobile robot with a 5-DOF mounted revolute-joint manipulator. Two rotational joints have been considered at the bottoms and the tips of both the manipulators, with the aim of making the system applicable to more practical situations. In the sequel, it is assumed that the motion of vehicles is restricted to the horizontal plane and both the end effectors catch the object tightly.

The problem can now be stated as follows: Given a group of two nonholonomic mobile manipulators grasping a rigid body, we should find a trajectory to steer the system in a cooperative manner between two configurations, in an environment with static obstacles such that the acceleration of each variable in the configuration space remains within certain bounds.

It should be noted that the combination of different types of constraints (including holonomic, nonholonomic, and dynamic constraints) in such a system makes the motion planning problem complicated and it requires careful evaluation to realize the payload manipulation task efficiently. Therefore, in the remaining section we use the above-mentioned model to generate the constraint equations of the system.

Closed-chain Constraints

When a collection of links is arranged so that it forms a loop, the configuration space becomes much more complicated because the joint angles must be chosen in a way that the loops remain closed. This leads to constraints in which some links must maintain specified positions relative to each other. To derive these constraints, we consider the cooperative system in more detail, as shown in Figure 2. Nomenclatures in this figure are defined as follows:

- l_i ($i = 1, \dots, 6$): Length of i^{th} link in the closed kinematic chain
- θ_i ($i = 1, \dots, 6$): The i^{th} joint angle rotating in the vertical plane in the closed kinematic chain
- θ_{pi} ($i = 1, \dots, 4$): The i^{th} joint angle rotating in the horizontal plane in the closed kinematic chain
- l_{obj} : Length of object between two end effectors

$l_{p_{i1}}, l_{p_{i2}}$ ($i = 1, \dots, 4$): Length of links attached to the i^{th} joint rotating in the horizontal plane

From this figure and the assumptions mentioned before, one can conclude that the mobile manipulator system is subject to holonomic constraints, expressed as:

$$\vec{f}(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_{p_1}, \theta_{p_2}, \theta_{p_3}, \theta_{p_4}, \theta_{b_1}, \theta_{b_2}, d_{12}, \alpha_{12}) = \vec{0} \quad (1)$$

where θ_{b_1} and θ_{b_2} represent the orientation of the base structures (detailed derivation of this equation can be found in Appendix A).

However, in order to lower the computational cost of our method, we have designed the desired trajectory in a manner that both the mobile bases have a fixed position and orientation relative to each other. To this end, we define the following conditions:

$$\begin{cases} d_{12} = g_b \\ \alpha_{12} = 0 \\ \theta_{b_1} = \theta_{b_2} - \pi = \theta_b \end{cases} \quad (2)$$

where g_b is a constant positive value. We also make the mounted manipulators to cooperate in a single plane. By using (2), this leads to the following desired trajectories:

$$\theta_{p_1} = \theta_{p_2} = \theta_{p_3} = \theta_{p_4} = 0 \quad (3)$$

Therefore, by substituting (2) and (3) in (1) and after further manipulation one gets:

$$\theta_1 + \theta_2 + \theta_3 = \theta_4 + \theta_5 + \theta_6$$

$$l_3 s_{123} + l_2 s_{12} + l_1 s_1 - l_{obj} c_{456} = l_4 s_{456} + l_5 s_{56} + l_6 s_6 \quad (4)$$

$$l_3 c_{123} + l_2 c_{12} + l_1 c_1 + l_{obj} s_{456} = l_4 c_{456} + l_5 c_{56} + l_6 c_6 + g_b$$

in which, we define the following parameters:

$$\begin{aligned} c_{ijk} &= \cos(\theta_i + \theta_j + \theta_k), & c_{ij} &= \cos(\theta_i + \theta_j), & c_i &= \cos(\theta_i) \\ s_{ijk} &= \sin(\theta_i + \theta_j + \theta_k), & s_{ij} &= \sin(\theta_i + \theta_j), & s_i &= \sin(\theta_i) \\ l_3 &= l_{p_{21}} + l_{p_{22}}, & l_4 &= l_{p_{31}} + l_{p_{32}} \end{aligned} \quad (5)$$

Differential Constraints

Differential constraints exist in a model of every nonholonomic motion planning problem and restrict admissible velocities and accelerations. Here, we consider an upper limit on the acceleration of all joint space variables. In addition, the mobile platform used in this article is a kind of car-like mobile robot as shown in Figure 3; therefore, we suppose that the wheels are rolling without skidding and slipping.

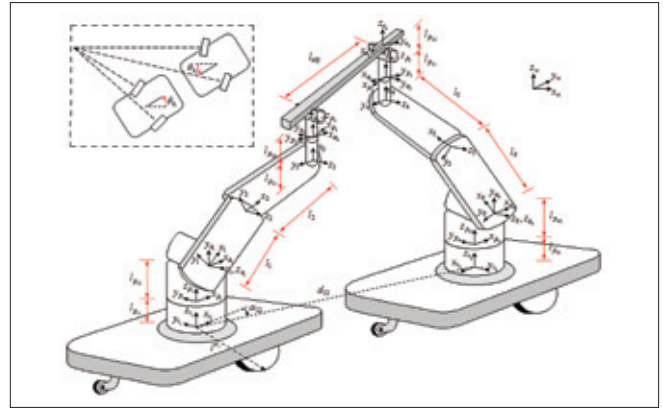


Figure 2: Cooperative system with its attached coordinate system

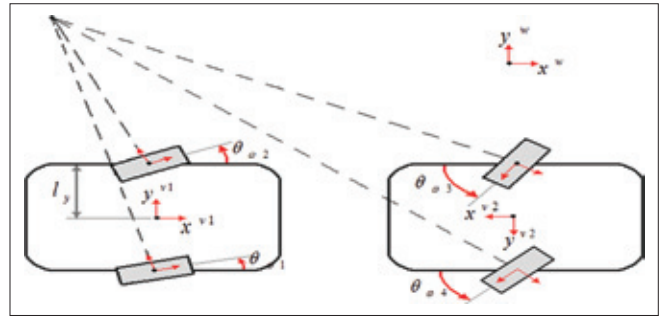


Figure 3: Base structure subsystem of the cooperative system

To derive the equations due to these constraints, we need to compute the velocity of each wheel in its attached coordinate system [Figure 3].

In this regard, we can state that:^[17,24]

$$v^{\omega_j} = \dot{x}_{b_i} x^w + \dot{y}_{b_i} y^w + \dot{\theta}_{b_i} (z^v \times l^E_j) \quad (6)$$

where x_{b_i} and y_{b_i} denote the position and θ_{b_i} represents the orientation of the i^{th} base structure ($i = 1, 2$) and l^{ω_j} is the position vector from the origin of $\{v_i\}$ to the contact point of the wheel with the ground [Figure 2]. Hence, the velocity of each wheel can be written as:

$$v^{\omega_j} = \begin{bmatrix} \dot{x}_{b_i} \cos(\theta_{b_i} + \theta_{\omega_j}) + \dot{y}_{b_i} \sin(\theta_{b_i} + \theta_{\omega_j}) - \dot{\theta}_{b_i} (l_y^{\omega_j} \cos \theta_{\omega_j}) \\ \dot{y}_{b_i} \cos(\theta_{b_i} + \theta_{\omega_j}) - \dot{x}_{b_i} \sin(\theta_{b_i} + \theta_{\omega_j}) + \dot{\theta}_{b_i} (l_y^{\omega_j} \sin \theta_{\omega_j}) \\ 0 \end{bmatrix} \quad (7)$$

and θ_{ω_j} is the steering angle of the j^{th} wheel ($j = 1, \dots, 4$) and also:

$$l_y^{\omega_j} = (-1)^j l_y \quad (8)$$

The non-skidding condition implies that the second term of the velocity vector in (7) vanishes. Therefore, for each mobile manipulator module these constraints can be taken

into account as:

$$-\dot{x}_b \sin(\theta_b + \theta_{\omega_j}) + \dot{y}_b \cos(\theta_b + \theta_{\omega_j}) + \dot{\theta}_b (l_y^{\omega_j} \sin \theta_{\omega_j}) = 0 \quad (9)$$

Consequently, the steering angle of the wheels is uniquely determined through the following equations:

$$\tan(\theta_{\omega_j}) = \begin{cases} \frac{\dot{y}_b \cos \theta_b - \dot{x}_b \sin \theta_b}{\dot{x}_b \cos \theta_b + \dot{y}_b \sin \theta_b - \dot{\theta}_b l_y^{\omega_j}} & j = 1, 2 \\ \frac{\dot{y}_b \cos \theta_b - \dot{x}_b \sin \theta_b + \dot{g}_b \theta_b}{\dot{x}_b \cos \theta_b + \dot{y}_b \sin \theta_b + \dot{\theta}_b l_y^{\omega_j}} & j = 3, 4 \end{cases} \quad (10)$$

and $\dot{\theta}_{\omega_j}$ becomes:

$$\theta_{\omega_j} = \begin{cases} \frac{1}{1 + \tan^2(\theta_{\omega_j})} \frac{d}{dt} \left(\frac{\dot{y}_b \cos \theta_b - \dot{x}_b \sin \theta_b}{\dot{x}_b \cos \theta_b + \dot{y}_b \sin \theta_b - \dot{\theta}_b l_y^{\omega_j}} \right) & j = 1, 2 \\ \frac{1}{1 + \tan^2(\theta_{\omega_j})} \frac{d}{dt} \left(\frac{\dot{y}_b \cos \theta_b - \dot{x}_b \sin \theta_b + \dot{g}_b \theta_b}{\dot{x}_b \cos \theta_b + \dot{y}_b \sin \theta_b + \dot{\theta}_b l_y^{\omega_j}} \right) & j = 3, 4 \end{cases} \quad (11)$$

Furthermore, by defining r^{ω_j} as the radius of the j^{th} wheel, the non-slipping constraints can be written as:^[17,24]

$$v_i^{\omega_j} = \omega_i^{\omega_j} \times r^{\omega_j} z^{\omega_j} \quad (12)$$

Now, we can obtain the angular velocities corresponding to the driving torques as follows:

$$\dot{\Psi}_j = \begin{cases} \frac{1}{r^{\omega_j}} \left(\dot{x}_b \cos(\theta_b + \theta_{\omega_j}) + \dot{y}_b \sin(\theta_b + \theta_{\omega_j}) - l_y^{\omega_j} \dot{\theta}_b \cos \theta_{\omega_j} \right) & j = 1, 2 \\ -\frac{1}{r^{\omega_j}} \left(\dot{x}_b \cos(\theta_b + \theta_{\omega_j}) + \dot{y}_b \sin(\theta_b + \theta_{\omega_j}) + \dot{\theta}_b g \sin \theta_{\omega_j} + l_y^{\omega_j} \dot{\theta}_b \cos \theta_{\omega_j} \right) & j = 3, 4 \end{cases} \quad (13)$$

DESIGN OF THE TRAJECTORY UNDER CLOSED-CHAIN AND DIFFERENTIAL CONSTRAINTS

The general methodology of the PRMs is to construct a graph (roadmap) during a preprocessing stage that represents the connectivity of the robot's free configuration space and then query the roadmap using an optimal graph-search algorithm (e.g., A^*), in order to find the shortest possible path between start and goal configurations. However, most of the motion planning techniques for closed-chain mechanisms could not directly account for the differential constraints, which could render the planned trajectory infeasible. Here, we have to find a trajectory between intermediate points, generated

during the path planning process, considering differential constraints.

Roadmap Construction

Some efforts have been recently made to apply the PRM method to closed chain systems.^[12-16,20,21] The main drawback of these approaches is that too many samples may have to be tested before finding a feasible configuration and too much computing time is spent in solving closure equations leading to imaginary values. To solve this problem, we utilize the PRM approach, with a geometrically guided sampling method called the Random Loop Generator (RLG),^[15] which notably increases the probability of obtaining real solutions when solving the loop closure equations [Algorithm 1 in Table 1].

In addition, in the large workspace of a mobile manipulator system, PRM methods require several hours of computation time to generate a well-connected roadmap.

Thus, we fix the position and orientation of both the manipulator bases first, and construct a roadmap (fixed-base roadmap), which contains n different self-collision-free closure configurations. Then we populate the environment in m random locations with copies of the kinematic roadmap and connect the configurations of the same closure type.

The basic principle of the RLG method, as stated in algorithm 1, is grounded on separating the configuration variables into two sets: Active (q_a) and passive (q_p).

There are some limitations in defining active and passive variables,^[15] for example:

1. The joint space variables in q_a and q_p correspond to the consecutive joints in the mechanism.
2. The number of passive variables is equal to the object's degrees of freedom.

If multiple choices exist for these variables that satisfy the above-mentioned limitations, all of them can be used in the method. In the remaining part of this article, we refer to active and passive variables as follows:

Table 1: RLG method for guided random sampling in the roadmap construction phase (Algorithm 1)

1. Specify active and passive subchains as q_a and q_p
2. for each active variable
3. compute on interval that closed - chain constraint equations have a solution in it.
4. Choose the active variable randomly from the computed interval
5. end for
6. solve the closed - chain constraint equations for passive variables after substituting active variables.
7. if there is a solution
8. use q_a and q_p to construct the closed - chain random configuration (repeat until attaining a real solution)

$$q_a = [\theta_1, \theta_2, \theta_6]^T$$

$$q_p = [\theta_3, \theta_4, \theta_5]^T$$

The planner directly acts on the active variables while the passive ones are obtained by solving loop-closure equations. An important part of algorithm 1 is to compute an interval for each of the active variables, increasing the probability of having real solutions for loop closure equations. Therefore, we should find a subset of values for each active variable that makes its workspace reachable by the remaining chain of the system. To this end, we require to illustrate the workspace of the closed chain system, which is usually a very complicated task. Thus, we have to use an approximate approach. In this regard, we simplify the model of our system as shown in Figure 4.

Therefore, we can express the reachable workspace for each active variable as the intersection area illustrated in Figure 5. The external and internal radii, r_{ext} and r_{int} , in the figure correspond to the maximum and minimum extensions of the chain respectively and are approximated as follows:

$$\hat{r}_{ext} = \sum_{i=1}^k L_i$$

$$\hat{r}_{int} = \begin{cases} 2L_{max} - \hat{r}_{ext} & 2L_{max} > \hat{r}_{ext} \\ 0 & o.w. \end{cases}$$

where L_i and L_{max} are the lengths of the i^{th} and the longest link in the chain respectively. The proof of this equation can be found in Appendix B. Then, after choosing each active variable in its computed interval, we solve the closed-chain constraint equations for passive variables. Now, the configuration vector of the fixed-base system, is written as:

$$\theta = [q_a^T, q_p^T]^T$$

Moreover, in order to consider the base structure mobility, utilizing algorithm 2 [Table 2], we populate the entire workspace with randomly selected parts of the initial roadmap. Let us choose the base configuration, g_{ob} , as a random vector including the position and orientation of the first robot, relative to the world frame:

$$g_{ob} = (x_b, y_b, \theta_b)$$

Then, we sample a node, θ , randomly from the initial roadmap and check the combined vector of (g_{ob}, θ) for collision. If the node is collision-free, we add it to the new roadmap. This routine continues for all neighbors of θ and is repeated for m different positions, to cover the entire workspace.

We collected all roadmap nodes with the same closed configuration in a set and used the PRM connection method

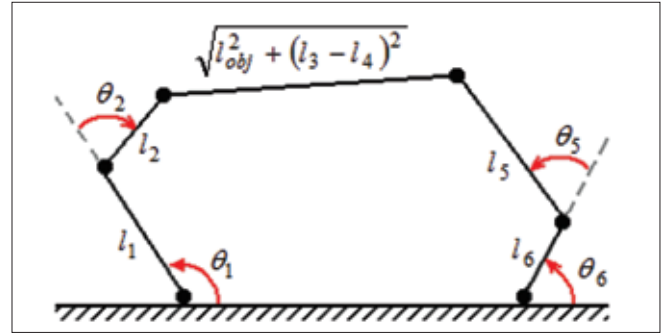


Figure 4: A simplified model of the fixed base system to be used in RLG algorithm

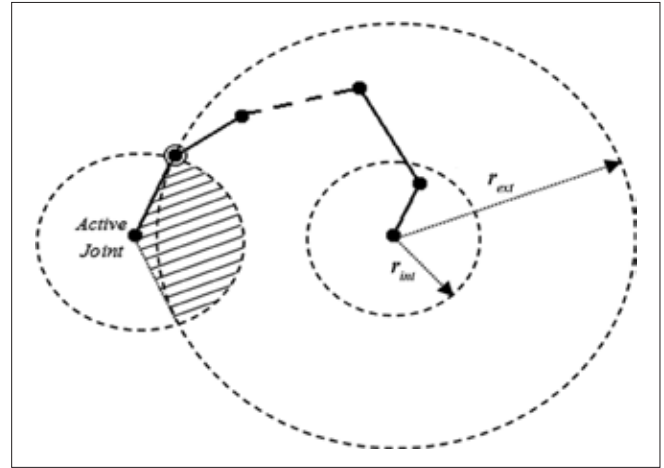


Figure 5: Subset of values for an active joint to be reachable by the remaining chain of the system

Table 2: Populating the environment with copies of the fixed-base roadmap (Algorithm 2)

1. Generate random base configuration g_{ob}
2. choose random vertex θ repeatedly from the fixed – base roadmap until attaining a collision – free configuration (g_{ob}, θ)
3. if there exist a collision – free configuration
4. retain (g_{ob}, θ) as a roadmap vertex
5. For each neighbor of θ , say $\tilde{\theta}$, in the fixed – base roadmap
6. if $(g_{ob}, \tilde{\theta})$ is collision – free
7. retain $(g_{ob}, \tilde{\theta})$ as a roadmap vertex
8. retrieve the path $\theta(t)$ connecting θ and $\tilde{\theta}$ from the fixed – based roadmap
9. if $(g_{ob}, \theta(t))$ is collision – free for all intermediate configurations along the path,
10. add an edge between (g_{ob}, θ) and $(g_{ob}, \tilde{\theta})$ (repeat as desired)

to connect the nodes in the set, as illustrated in Figure 6. Finally, we searched the graph for the shortest possible path between the start and goal configurations, with an optimal graph search algorithm.

Avoiding Collision with Fixed Obstacles

Sampling-based planners must perform many collision checks in order to build a roadmap and spend most of their running time performing such checks. Therefore,

their collision detection method must be very fast, without missing any collision or incorrectly detecting collisions, even when the workspace has complex geometry.

We used two types of collision checks in our motion planner: Static checks, which were used to test whether a sampled configuration in the roadmap was in the free space, and dynamic checks to test its local paths, which were continuous sets of configurations. In the static methods a common method was to break complex objects (robot link, obstacle, etc.) down, using a bounding volume hierarchy (BVH), which was a hierarchy of BVs (e.g., spheres) that approximated the geometry of the object at successive levels of detail.^[25]

BVH model of the cooperative system

The choice of the type of bounding volume for a given application is a trade-off between the ‘tightness of fit’ and the speed of operations between two such volumes. Therefore, to build a BVH for the cooperative mobile manipulator system we use bounding spheres, which are very quick to test for collision with each other in conjunction with a more precise, but also more expensive type of bounding volume according to Figure 7.

Static collision checking

The basic idea behind this method is, to check two objects for collisions, their BVHs are searched from top down, making it possible to quickly discard large subsets of the objects contained in disjointed BVs. In other words, if the BVs at the top level are in collision then their children are also checked for collision. Otherwise, the algorithm will not search any of the children.

Dynamic collision checking

The classical approach to perform dynamic checks is to sample each path at some fixed resolution and statically check each sampled configuration for collision. This approach is approximate and can miss collisions. Therefore, we use a newly presented dynamic checker^[26] that exactly determines whether a path lies in free space, by choosing an adaptive sampling resolution along the local path. This checker automatically decides whether a path segment between two collision-free configurations needs to be bisected further.

Let $A_i(q)$ denote an object A_i from a collection of rigid objects (including each of the mobile manipulators, payload, obstacles, etc.) at configuration q . We define $n_{ij}(q)$ to be any non-trivial lower bound on the Euclidian distance between $A_i(q)$ and $A_j(q)$. Let $\lambda_i(q_a, q_b)$ be an upper bound on the lengths of the curves traced by all points in A_i between configurations q_a and q_b along a path segment. A sufficient condition for two objects A_i and A_j not to collide along a path in the configuration space is:^[26]

$$\lambda_i(q_a, q_b) + \lambda_j(q_a, q_b) < n_{ij}(q_a) + n_{ij}(q_b) \tag{18}$$

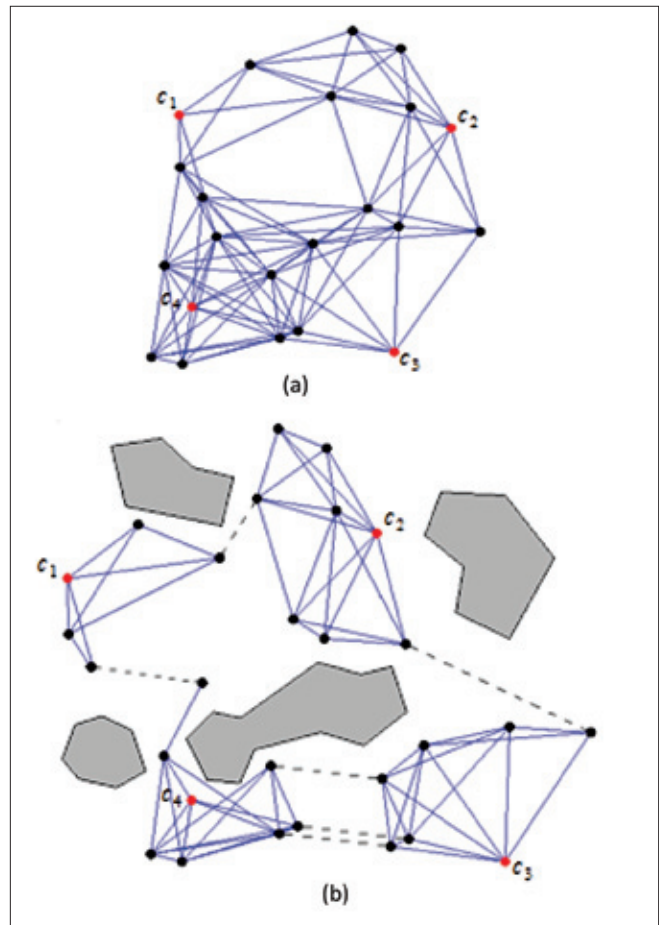


Figure 6: (a) Initial roadmap without base structure mobility; (b) Distributing the initial roadmap to consider the base structure mobility

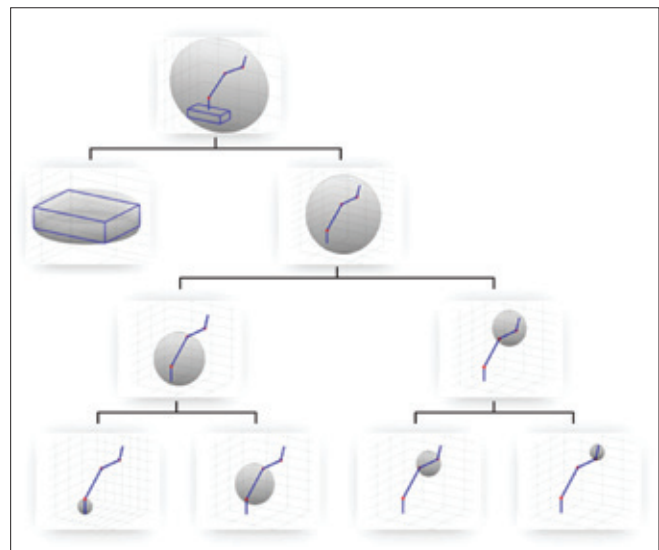


Figure 7: A BVH model for the mobile manipulator system

If this inequality is verified for all pairs of objects A_i and A_j , then the path segment is collision-free, otherwise, it must be bisected. Furthermore, to compute a lower bound on the distance between two objects we work according to the

classical BVH collision checker. However, instead of testing if two BVs intersect, we compute the distance between them and find the smallest distance found.

Design of the Trajectory

A challenging problem during the planning process is that the differential constraints shall be satisfied without missing closed-chain constraints. Here, we present an extension of the RLG algorithm to solve this trajectory planning problem. First, we describe the trajectory of each active variable between the sequential pairs of way points using the third order polynomial equations, which are expressed in a normalized interval of time as follows:

$$q_{ik}(s) = a_{0ik} + a_{1ik}s + a_{2ik}s^2 + a_{3ik}s^3, \quad (19)$$

$$i \in \{x_b, y_b, \theta_b, \theta_1, \theta_2, \theta_6\}$$

$$k = 1, 2, \dots, n_{seg}, 0 \leq s \leq 1$$

where, n_{seg} is the number of path segments, and:

$$s = \frac{\Delta t}{t_{fik}} \quad (20)$$

where, t_{fik} is the time required to traverse the k^{th} segment of the path. Now, we use the position and velocity values at the endpoints of the path segment to calculate the coefficients of q_{ik} as follows:

$$a_{0ik} = q_{ik}(0), a_{1ik} = \left. \frac{dq_{ik}}{ds} \right|_{s=0}$$

$$a_{2ik} = 3(q_{ik}(1) - q_{ik}(0)) - \left(2 \left. \frac{dq_{ik}}{ds} \right|_{s=0} + \left. \frac{dq_{ik}}{ds} \right|_{s=1} \right) \quad (21)$$

$$a_{3ik} = \left. \frac{dq_{ik}}{ds} \right|_{s=0} + \left. \frac{dq_{ik}}{ds} \right|_{s=1} - 2(q_{ik}(1) - q_{ik}(0))$$

In this regard, we use a simple numerical approach to find the velocity vector in the intermediate points of the path: If the slope sign of the straight lines between each point and the previous and next ones change, the velocity is equal to zero, otherwise it is computed as the average of the two slopes. Next, utilizing loop closure equations we extract a closed form to find passive variables from (4):

$$\begin{cases} \sin(\xi + \theta_4) = \frac{(l_3 - l_4)^2 + l_{obj}^2 + l_5^2 - \beta^2 - \gamma^2}{2l_5 \sqrt{l_{obj}^2 + (l_3 - l_4)^2}} \\ \theta_3 = \text{atan2}(\beta, -\gamma) - \text{atan2}(d, d') \\ \theta_5 = \theta_1 + \theta_2 + \theta_3 - \theta_4 - \theta_6 \end{cases} \quad (22)$$

where:

$$\begin{cases} \xi = \arctan\left(\frac{l_3 - l_4}{l_{obj}}\right) \\ \beta = g + l_6 \cos(\theta_6) - l_1 \cos(\theta_1) - l_2 \cos(\theta_1 + \theta_2) \\ \gamma = l_6 \sin(\theta_6) - l_1 \sin(\theta_1) - l_2 \sin(\theta_1 + \theta_2) \\ d = (l_3 - l_4) \cos(\theta_1 + \theta_2) + l_{obj} \sin(\theta_1 + \theta_2) \\ \quad - l_5 \cos(\theta_1 + \theta_2 - \theta_4) \\ d' = -(l_3 - l_4) \sin(\theta_1 + \theta_2) + l_{obj} \cos(\theta_1 + \theta_2) \\ \quad + l_5 \sin(\theta_1 + \theta_2 - \theta_4) \end{cases} \quad (23)$$

Now, we calculate the minimum time to go from one intermediate point to the next with regard to the maximum allowed accelerations. Toward this end, the following theorem establishes a lower limit for the trajectories, in the form of cubic polynomials such as those in (19).

Theorem 1. Consider the cubic polynomial given by the following equation:

$$q_{ik}(s) = a_{0ik} + a_{1ik}s + a_{2ik}s^2 + a_{3ik}s^3, \quad 0 \leq s \leq 1 \quad (24)$$

with $s = \frac{\Delta t}{t_{fik}}$ and $\left| \frac{d^2 q_{ik}}{dt^2} \right| \leq \alpha_{ik}$. Then the lower bound on t_{fik}

that guaranties the acceleration limits of q_{ik} is:

$$t_{fik} \geq \sqrt{\frac{2 \max(|a_{2ik}|, |a_{2ik} + 3a_{3ik}|)}{\alpha_{ik}}} \quad (25)$$

Proof. As the second derivative of a cubic polynomial is a line, its maximum value occurs in one of the corresponding endpoints:

$$\left| \frac{d^2 q_{ik}}{ds^2} \right| \leq \max\left(\left| \frac{d^2 q_{ik}}{ds^2} \right|_{s=0}, \left| \frac{d^2 q_{ik}}{ds^2} \right|_{s=1} \right) \quad (26)$$

In other words:

$$\left| \frac{d^2 q_{ik}}{ds^2} \right| \leq \max(|2a_{2ik}|, |2a_{2ik} + 6a_{3ik}|) \quad (27)$$

By using $\frac{d^2 q_{ik}}{dt^2} = \frac{1}{t_{fik}^2} \frac{d^2 q_{ik}}{ds^2}$ we have:

$$\left| \frac{d^2 q_{ik}}{dt^2} \right| \leq \frac{2}{t_{fik}^2} \max(|a_{2ik}|, |a_{2ik} + 3a_{3ik}|) \quad (28)$$

Furthermore, in view of the constraint $\left| \frac{d^2 q_{ik}}{dt^2} \right| \leq \alpha_{ik}$, the following condition shall be satisfied:

$$\frac{2}{t_{fik}^2} \max(|a_{2ik}|, |a_{2ik} + 3a_{3ik}|) \leq \alpha_{ik} \quad (29)$$

and (25) is concluded.

However, due to the nonlinearity and complexity of the equations for the wheel velocities and the passive variables, solving the acceleration condition for the exact lower bound on t_{fik} can be prohibitively expensive and furthermore pose numerical problems. Therefore, utilizing equations (11), (13), and (22) we compute a linear approximation of the acceleration and write the following condition:

$$t_{fik}^2 \geq \beta_{ins} \frac{\left. \frac{dq_{jk}}{ds} \right|_{s=1} - \left. \frac{dq_{jk}}{ds} \right|_{s=0}}{\alpha_{jk}}, \quad (30)$$

$$j \in \{\theta_3, \theta_4, \theta_5, \theta_{\omega_{1,2}}, \psi_{1,2}, \theta_{\omega_{3,4}}, \psi_{3,4}\}, \quad k = 1, 2, \dots, n_{seg}$$

where α_{jk} is the maximum allowed acceleration and $\beta_{ins} > 1$ is an insurance factor that compensates for the approximation error and maintains the actuator acceleration in the safe bound. Obviously, using path-smoothing algorithms and increasing the number of path segments lowers the corresponding deviation from the exact value of the acceleration. Consequently, the total lower bound on the time to move between adjacent points in each section of the trajectory would be the maximum of the lower bounds computed in (25) and (30):

$$t_{fk} = \max_{i,j} (t_{fik}, t_{fjk}) \quad (31)$$

However, to find these lower bounds, because of the lack of time response information, we utilized an approximation of $\frac{dq_{ik}}{ds}$ in each intermediate point of the path, in place of the actual velocities, and assumed that:

$$\left. \frac{dq_{ik}}{ds} \right|_{s=1} = \left. \frac{dq_{i(k+1)}}{ds} \right|_{s=0} \quad (32)$$

Hence, from (20) we can write:

$$t_{fk} \left. \frac{dq_{ik}}{dt} \right|_{t=t_{fk}} = t_{f(k+1)} \left. \frac{dq_{i(k+1)}}{dt} \right|_{t=0} \quad (33)$$

which shows that with this assumption there are discontinuous velocities when we move between adjacent segments of the path. Therefore, we apply the bounds computed in (31) to find an approximated velocity in each intermediate point through the previously mentioned numerical approach. Then, we can write:

$$\begin{aligned} a_{0ik} &= q_{ik}(0), \quad a_{1ik} = t_{fk} \left. \frac{dq_{ik}}{dt} \right|_{t=0} \\ a_{2ik} &= 3 \left(q_{ik}(t_{fk}) - q_{ik}(0) \right) - t_{fk} \left(2 \left. \frac{dq_{ik}}{dt} \right|_{t=0} + \left. \frac{dq_{ik}}{dt} \right|_{t=t_{fk}} \right) \\ a_{3ik} &= t_{fk} \left(\left. \frac{dq_{ik}}{dt} \right|_{t=0} + \left. \frac{dq_{ik}}{dt} \right|_{t=t_{fk}} \right) - 2 \left(q_{ik}(t_{fk}) - q_{ik}(0) \right) \end{aligned} \quad (34)$$

Finally, substituting (34) into (25) and using (30) and (31) we can compute new values for t_{fk} .

SIMULATION RESULTS

Let us consider two same mobile manipulators, as shown in Figure 2. To verify the effectiveness of our method we will conduct simulations based on the following assumptions:

- Each mobile manipulator is subjected to the differential constraints mentioned in (9) and (13) and its goal is to cooperate with the other to satisfy the closed-chain constraints in (1)
- The transporting object is a rigid body that cannot be deformed
- The start and goal configurations are set to:

$$\begin{aligned} q_{start} &= \left[0, 0, 0, \frac{\pi}{2}, 0, 0, 0, 0, \frac{\pi}{2} \right]^T \\ q_{goal} &= \left[10, 10, \frac{\pi}{4}, 0, \frac{\pi}{2}, 0, 0, \frac{\pi}{2}, 0 \right]^T \end{aligned} \quad (34)$$

- The environment is populated with six static obstacles in the form of spheres with specified position and radius, as shown in Figure 8b and c
- Some important parameters used in the planner are chosen as per Table 3:

Figure 8 shows a 3-D visualization for the trajectory of the object with and without obstacles. As shown here, there is a smooth trajectory between the start and goal configurations in both cases.

Furthermore, accelerations of the computed trajectories for joint space variables are shown in Figure 9, and it can be seen that they are bounded within the maximum allowed accelerations (1 rad/s^2).

From (19-21) the trajectory of each active variable between the sequential pairs of way points is described by the third order polynomial equations. Therefore, in Figure 9 the accelerations of the active joints are in the form of ramp signals; however, this is not the case for the passive variables. The quality of the designed trajectory is shown in Figure 10. This figure illustrates the deviation of the joint space variables from their final values.

Table 3: Simulation parameters

Parameter	Value
n	300
m	200
g_b, l_{obj}	1.2 (m)
l_1, l_6	0.8 (m)
l_2, l_5	0.5 (m)
l_3, l_4	0.3 (m)

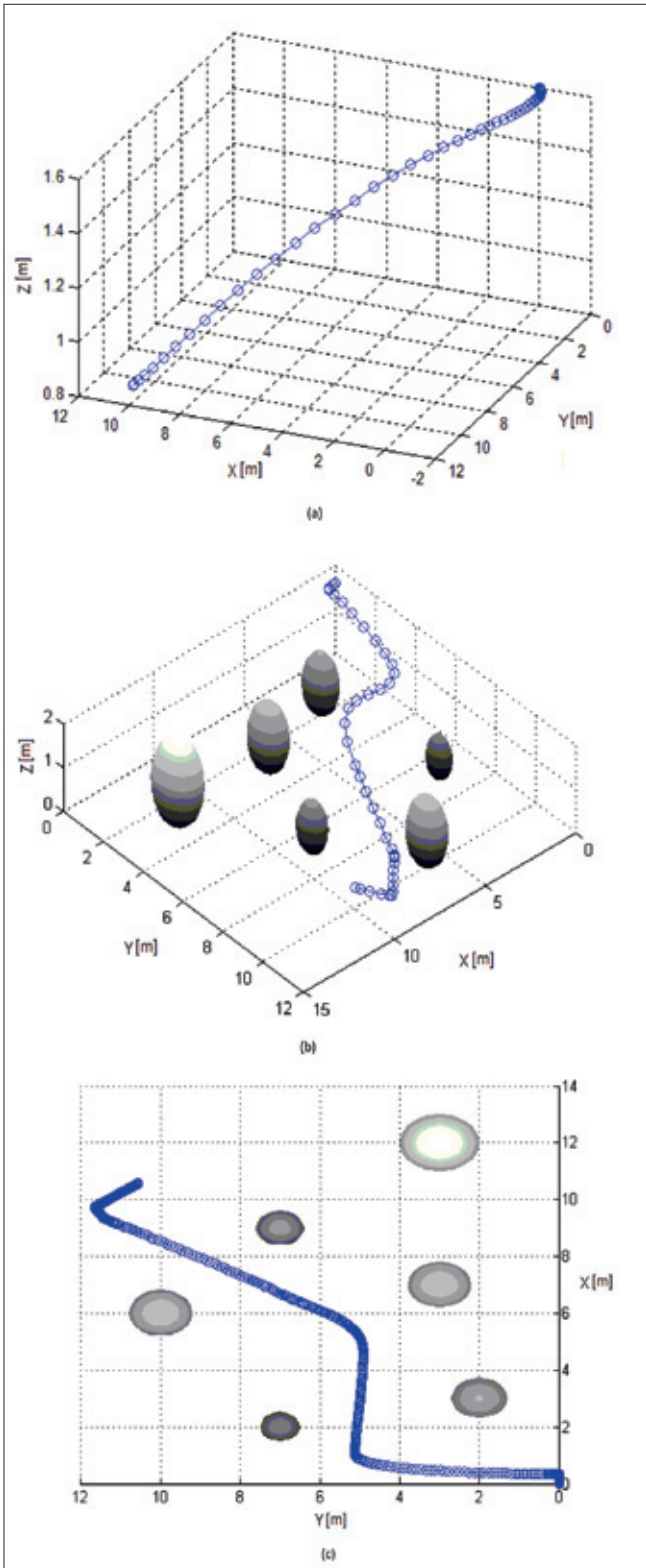


Figure 8: Object's trajectory between the predefined start and goal configurations: (a) without obstacle, (b) with obstacles, (c) with obstacles (top view)

It can be seen that the errors converge to zero as time goes on. Figure 11 illustrates the time required to reach the goal configuration.

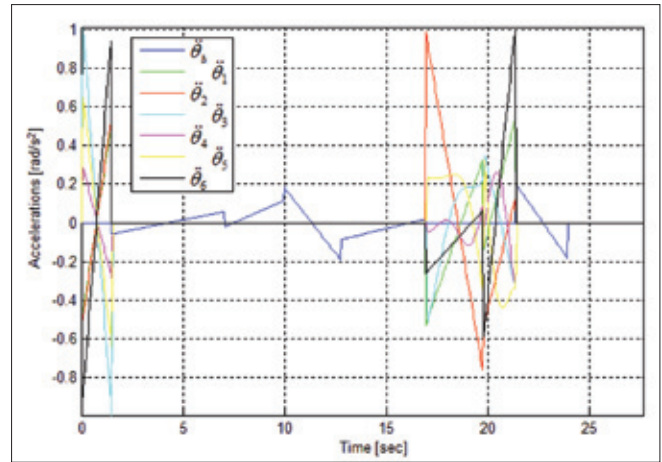


Figure 9: Joint accelerations of the manipulators

CONCLUSIONS

A novel trajectory planning method has been designed for dual mobile manipulators grasping a common object. Because of its structure, our method can handle high dimensional configuration spaces efficiently.

It also utilizes the advantage of decoupled methods to decompose the complexity of the trajectory planning problem in two steps, which increases the simplicity of the planning process. The main advantage of this method is that such constraints as nonholonomic and closed-chain constraints, along with the joint and acceleration limits, can be easily dealt with in the presence of static obstacles. In comparison with the other approaches discussed in literature, the advantages of the proposed method have been shown in Table 4.

Furthermore, the results of computer simulations confirmed the effectiveness of the method. A good idea to improve the algorithm proposed in this study is to develop a method to keep the system from colliding with the moving obstacles. Also, investigating the trajectory planning of more than two cooperating mobile manipulators to perform the carrying task seems to be an interesting route to follow.

APPENDIX

Detail Derivation of Equation (1)

Utilizing the coordinate systems in Figure 2 along with the concept of transformation matrices, we can write:

$$\begin{aligned}
 {}^{v_1}T_{p_1} {}^{p_1}T_{a_1} {}^{a_1}T_1 {}^1T_2 {}^2T_3 {}^3T_{a_2} {}^{a_2}T_{p_2} {}^{p_2}T_e = \\
 {}^{v_1}T_{v_2} {}^{v_2}T_{p_4} {}^{p_4}T_{a_4} {}^{a_4}T_6 {}^6T_5 {}^5T_4 {}^4T_{a_3} {}^{a_3}T_{p_3} {}^{p_3}T_e
 \end{aligned}
 \tag{A.1}$$

in which ${}^A_B T$ is a matrix that transforms the coordinate

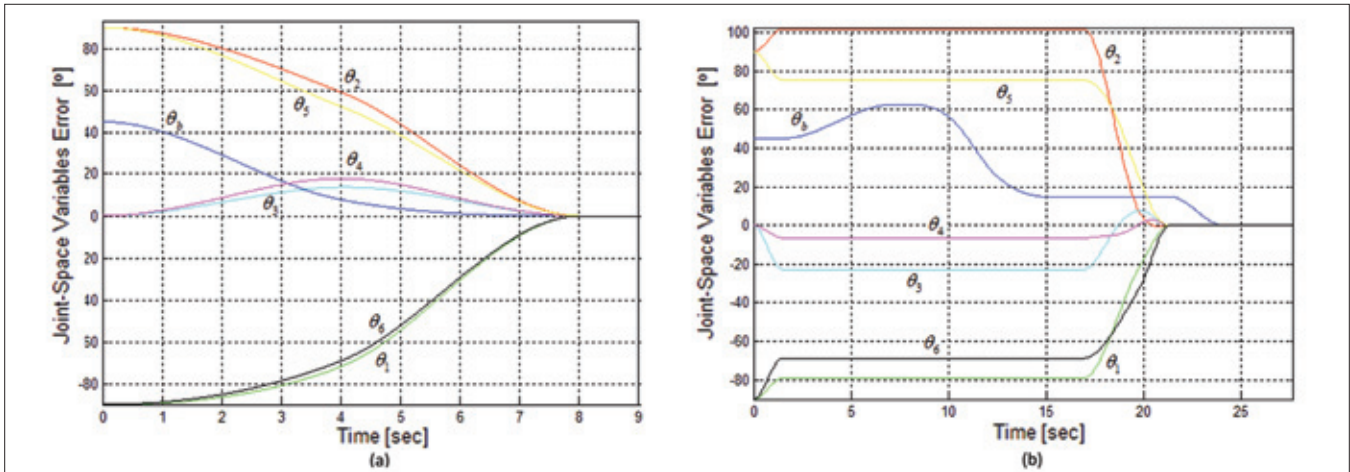


Figure 10: Errors of the joint space variables: (a) without obstacle, (b) with obstacles

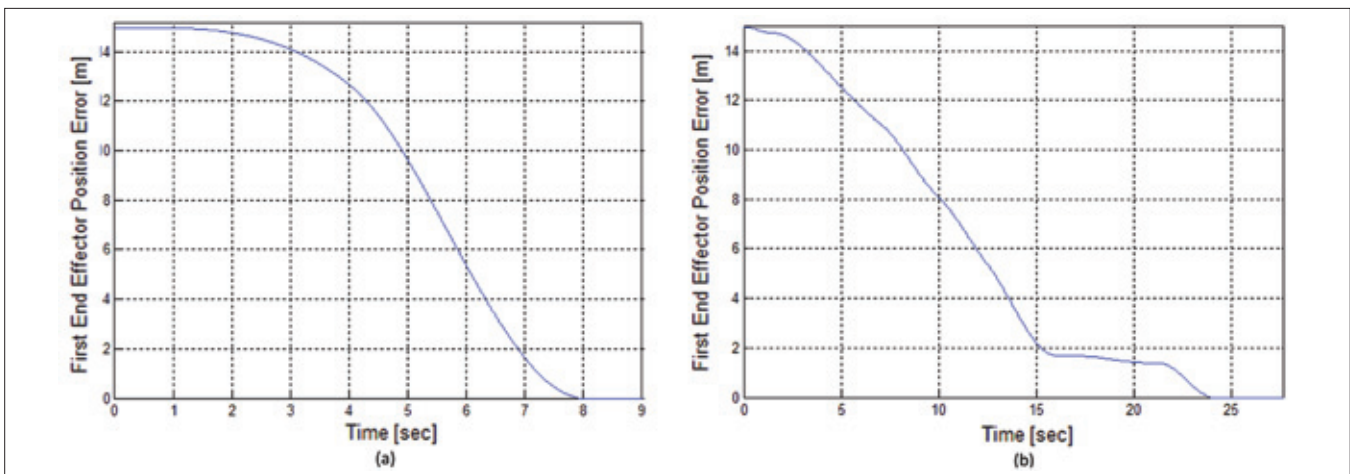


Figure 11: The distance between current and goal positions of the first end effector: (a) without any obstacle, (b) with static obstacles

Table 4: Comparison of recent approaches on motion planning for cooperative mobile manipulators

Method	Criteria			
	Computational complexity	Considering differential constraints in the trajectory	Guarantees to find a solution if it exists	Applicable to high dimensional systems
Complete methods				
Optimal control based 1,11	Very high	Yes	Yes	No
Numerical optimization 2,3	High	Yes	No	No
Artificial potential fields 6	Very high	Yes	Yes	No
Randomized or probabilistic complete methods				
Single query methods 13	Moderate	No	Yes	Yes
Multiple query methods 13-16	Low	No	Yes	Yes
Our method	Low	Yes	Yes	Yes

system $\{B\}$ into $\{A\}$, and the different coordinate systems in the above equation are defined as follows:

Therefore, from (A.1), after further manipulation, the validity of (1) can be verified.

Proof of Equation (15)

The reachable workspace of any articulated mechanism

can be bounded by two concentric spheres, as shown in Figure 5. The radii of these spheres correspond to the length of the mechanism in the case of minimum and maximum extensions. This length is defined by the distance between the origins of the base-frame and the end-frame as:

$$r = \left| \sum_{i=1}^k L_i a_i \right| \tag{B.1}$$

$${}_{p_1}^v T = \begin{bmatrix} \cos(\theta_{p_1}) & -\sin(\theta_{p_1}) & 0 & 0 \\ \sin(\theta_{p_1}) & \cos(\theta_{p_1}) & 0 & 0 \\ 0 & 0 & 1 & l_{p_{11}} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}_{a_1}^{p_1} T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & l_{p_{12}} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}_{a_1}^{a_1} T = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}_{2}^1 T = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & l_1 \\ \sin(\theta_2) & \cos(\theta_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}_{3}^2 T = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & l_2 \\ \sin(\theta_3) & \cos(\theta_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}_{a_2}^3 T = \begin{bmatrix} 0 & 0 & 1 & l_{p_{21}} \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}_{p_2}^{a_2} T = \begin{bmatrix} \cos(\theta_{p_2}) & -\sin(\theta_{p_2}) & 0 & 0 \\ \sin(\theta_{p_2}) & \cos(\theta_{p_2}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}_{e_2}^{p_2} T = \begin{bmatrix} -1 & 0 & 0 & l_{obj} \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & l_{p_{22}} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}_{e_2}^{p_3} T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & l_{p_{32}} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}_{v_2}^{v_1} T = \begin{bmatrix} \cos(\theta_{B_2} - \theta_{B_1}) & -\sin(\theta_{B_2} - \theta_{B_1}) & 0 & d_{12} \cos(\alpha_{12}) \\ \sin(\theta_{B_2} - \theta_{B_1}) & \cos(\theta_{B_2} - \theta_{B_1}) & 0 & d_{12} \sin(\alpha_{12}) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}_{p_4}^{v_2} T = \begin{bmatrix} \cos(\theta_{p_4}) & -\sin(\theta_{p_4}) & 0 & 0 \\ \sin(\theta_{p_4}) & \cos(\theta_{p_4}) & 0 & 0 \\ 0 & 0 & 1 & l_{p_{41}} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}_{a_4}^{p_4} T = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & l_{p_{42}} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}_{6}^{a_4} T = \begin{bmatrix} \cos(\theta_6) & -\sin(\theta_6) & 0 & 0 \\ \sin(\theta_6) & \cos(\theta_6) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}_{5}^6 T = \begin{bmatrix} \cos(\theta_5) & -\sin(\theta_5) & 0 & l_6 \\ \sin(\theta_5) & \cos(\theta_5) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}_{4}^5 T = \begin{bmatrix} \cos(\theta_4) & -\sin(\theta_4) & 0 & l_5 \\ \sin(\theta_4) & \cos(\theta_4) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}_{a_3}^4 T = \begin{bmatrix} 0 & 0 & 1 & l_{p_{31}} \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}_{p_3}^{a_3} T = \begin{bmatrix} \cos(\theta_{p_3}) & -\sin(\theta_{p_3}) & 0 & 0 \\ \sin(\theta_{p_3}) & \cos(\theta_{p_3}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where L_i ($i = 1, \dots, k$) is the length of i^{th} link and $a_i \in [-1, 1]$ is a factor that maps these lengths in the direction of the vector, connecting the origins of the base-frame and the end-frame. Thus, by substituting $a_i = -1$ or $a_i = 1$, $\forall i \in \{1, \dots, k\}$ in (B.1) we can compute a maximum value for r as follows:

$$r_{\max} = \sum_{i=1}^k L_i \quad (\text{B.2})$$

Now, suppose that the j^{th} link of the system has the longest length, L_{\max} ; then, from (B.1) one gets:

$$r = |L_1 a_1 + L_2 a_2 + L_3 a_3 + n + L_{\max} a_j + n + L_k a_k| \quad (\text{B.3})$$

Therefore, we can write the following equation to compute a minimum value for r :

$$r_{\min} = \begin{cases} -L_1 - L_2 - n - L_{j-1} + L_{\max} - L_{j+1}n - L_k & \text{if } L_{\max} > L_1 + L_2 + n + L_{j-1} + L_{j+1}n + L_k \\ 0 & \text{if } L_{\max} \leq L_1 + L_2 + n + L_{j-1} + L_{j+1}n + L_k \end{cases} \quad (\text{B.4})$$

and from (B.2) we can conclude that:

$$r_{\min} = \begin{cases} 2L_{\max} - r_{\max} & \text{if } 2L_{\max} > r_{\max} \\ 0 & \text{if } 2L_{\max} \leq r_{\max} \end{cases} \quad (\text{B.5})$$

which shows the validity of (15).

REFERENCES

1. J. P. Desai, C. C. Wang, M. Zefran, and V. Kumar. Motion planning for multiple mobile manipulators. In *Proc. IEEE International Conference on Robotics and Automation*, pp. 2073-8, 1996.
2. J. P. Desai, and V. Kumar. Motion planning for cooperating mobile manipulators. *J Robot Syst* 16, pp. 557-79, 1999.
3. S. Furuno, M. Yamamoto, and A. Mohri. Trajectory planning of cooperative multiple mobile manipulators. In *Proc. 2003 IEEE International Conference on Intelligent Robots and Systems*. vol. 1, pp. 136-41, 2003.
4. Y. Yamamoto, and S. Fukuda. Trajectory planning of multiple mobile manipulators with collision avoidance capability. In *Proc. 2002 IEEE International Conference on Robotics and Automation*. vol. 4, pp. 3565-70, 2002.
5. J. Albaric, and R. Zapata. Motion planning of cooperative nonholonomic mobile manipulators. In *Proc. 2002 IEEE International Conference on Systems, Man, and Cybernetics*. vol. 6, pp. 329-34, 2002.
6. H. G. Tanner, S. G. Loizou, and K. J. Kyriakopoulos. Nonholonomic navigation and control of cooperating mobile manipulators. *IEEE Trans Rob Autom* 19, pp. 53-64, 2003.
7. J. Vannoy, and X. Jing. Real-time motion planning of multiple mobile manipulators with a common task objective in shared work environments. In *Proc. 2007 IEEE International Conference on Robotics and Automation*, pp. 20-6, 2007.
8. M. Abou-Samah, C. P. Tang, R. M. Bhatt, and V. Krovi. A kinematically compatible framework for cooperative payload transport by nonholonomic mobile manipulators. *Auton Robots* 21, pp. 227-42, 2006.
9. R. M. Bhatt. Towards modular cooperation between multiple nonholonomic mobile manipulators, Ph.D. Thesis. Buffalo: State University of New York; 2007.
10. C. P. Tang, R. Bhatt, and V. Krovi. Decentralized kinematic control of payload transport by a system of mobile manipulators. In *Proc. 2004 IEEE International Conference on Robotics and Automation*, pp. 2462-7, 2004.
11. J. P. Desai, and V. Kumar. Nonholonomic motion planning for multiple mobile manipulators. In *Proc. IEEE International Conference on Robotics and Automation*, pp. 3409-14, 1997.
12. Y. L. Fu, Q. H. Yan, and Y. L. Ma. Trajectory planning of mobile manipulators in constrained workspace. *J Harbin Institute Technol (New Series)* 15, pp. 521-4, 2008.
13. J. Cortes. Motion planning algorithms for general closed-chain mechanisms, Ph.D. Thesis. Toulouse, France: Institute National Polytechnique de Toulouse, 2003.
14. L. Han, and N. Amato. A kinematics-based probabilistic roadmap method for closed chain systems. In *Proc. 2000 International Workshop on Algorithmic Foundations of Robotics (WAFR)*, pp. 233-45, 2000.
15. J. Cortes, T. Simeon, and J. P. Laumond. A random loop generator for planning the motions of closed kinematics chains using PRM methods. In *Proc. 2002 IEEE International Conference on Robotics and Automation*, pp. 2141-6, 2002.
16. J. Cortes, and T. Simeon. Sampling-based motion planning under kinematic loop-closure constraints. In *Proc. 6th International Workshop on Algorithmic Foundations of Robotics*, 2004.
17. H. G. Tanner, K. J. Kyriakopoulos, and N. I. Krikellis. Modeling of multiple mobile manipulators handling a common deformable object. *J Robot Syst* 15, pp. 599-623, 1998.
18. E. Todt, G. Rush, and R. Suarez. Analysis and classification of multiple robot coordination methods. In *Proc. 2000 IEEE International Conference on Robotics and Automation*, pp. 3158-63, 2000.
19. H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, et al. *Principles of robot motion: Theory, Algorithms, and Implementation*. United States: MIT Press; 2005.
20. S. M. LaValle, J. H. Yakey, and L. E. Kavraki. A probabilistic roadmap approach for systems with closed kinematic chains. In *Proc. 1999 IEEE International Conference on Robotics and Automation*, pp. 1671-6, 1999.
21. J. H. Yakey, S. M. LaValle, and L.E. Kavraki. Randomized path planning for linkages with closed kinematic chains. *IEEE Trans Rob Autom* 17, pp. 951-8, 2001.
22. J. P. Laumond. *Robot motion planning and control*. Springer, 1998.
23. S. M. Lavalley. *Planning algorithms*. Cambridge: Cambridge University Press; 2006.
24. H. G. Tanner, and K. J. Kyriakopoulos. Mobile manipulator modeling with Kane's approach. *Robotica* 19, pp. 675-90, 2003.
25. G. Bergen. *Collision detection in interactive 3D environments*. Amsterdam: Elsevier; 2004.
26. F. Schwarzer, M. Saha, and J. C. Latombe. Adaptive dynamic collision checking for single and multiple articulated robots in complex environments. *IEEE Trans Robot* 21, pp. 338-53, 2005.

How to cite this article: Bolandi H, Ehyaei AF. A novel method for trajectory planning of cooperative mobile manipulators. *J Med Sign Sens* 2011;1:24-35

Source of Support: Nil, **Conflict of Interest:** None declared

BIOGRAPHIES



Hossein Bolandi received his D.Sc. degree in electrical engineering from George Washington University, Washington, D.C., in 1990.

Since 1990, he has been with the College of Electrical Engineering, Iran University of Science and Technology, Tehran, Iran, where he is an associate professor. His research interests are in attitude determination and control subsystems of satellites, robotics, and adaptive control.

Dr. Bolandi is the author of 25 journal articles and 105 articles in the international and Iranian conference proceedings.



Amir Farhad Ehyaei received the B.Sc. degree in control engineering from the Sharif University of Technology, Tehran, Iran, in 2001. He received the M.Sc. degree in control engineering in 2003, from the Iran University of Science and Technology,

Tehran, Iran, where he is currently a Ph.D. candidate. His main fields of interest include navigation and control of multi-robot systems.

Since 2004, he has worked with several companies in the field of Instrumentation and Process control including Power Engineering Consultants of the Ministry of Energy (MOSHANIR) and Iran International Engineering Company (IRITEC).