International Journal of Innovative Computing, Information and Control
Volume **8**, Number **2**, February **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 1077–1102

TRAJECTORY PLANNING OF TWO COOPERATIVE MOBILE MANIPULATORS UNDER CLOSED-CHAIN AND DIFFERENTIAL CONSTRAINTS

Hossein Bolandi and Amir Farhad Ehyaei

College of Electrical Engineering
Iran University of Science and Technology
Narmak, Tehran 16844, Iran
h_bolandi@iust.ac.ir; ehyaei@ee.iust.ac.ir

Received September 2010; revised January 2011

Abstract. *In this paper, we consider a new method for trajectory planning of two mobile manipulators for cooperative transportation of a rigid body. The method consists of constructing a graph on a portion of the configuration space that satisfies collision and closure constraints and searching the graph for the shortest possible path using an optimal graph search algorithm. Then, a sequence of time-optimal trajectories for movement between the consecutive points of the path is calculated. This approach allows for 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. We also propose a heuristic method to keep the system from colliding with moving obstacles by adjusting a time scaling factor based on linear estimation of obstacles' position. Simulation results illustrate the effectiveness of the proposed method.*

Keywords: Cooperative mobile manipulators, Nonholonomic motion planning, Velocity adjustment, Collision avoidance, Moving obstacles

1. **Introduction.** Over the past few decades, various robotic systems with different structural and computational complexities ranging from simple mobile robots [1-4] to cooperative multi-robot systems, including multiple mobile robots [5-8], multiple manipulators [9-11], multi-fingered hands [12-15] and multi-legged vehicles [16,17], have been extensively studied in a variety of contexts focusing on motion planning. Among these efforts, an interesting topic that has recently attracted a considerable amount of research is the motion planning of cooperative mobile manipulators [18-31]. Despite the expanded workspace and increased capability in carrying out more complicated and dexterous tasks, the motion planning of these systems is complicated due to their high degrees of freedom and presence of several kinematic and dynamic constraints, including nonholonomic, closed chain and obstacle avoidance constraints [32,33]. Thus far, two main approaches for motion planning have been proposed: "Cartesian space method" and "joint space methods". Cartesian space planning methods [18-20] first compute a desired trajectory for the manipulated object in Cartesian space and then, with regard to the constraints of the system, compute the corresponding desired trajectory in the joint space for each mobile manipulator system. However, since these methods need to solve the inverse kinematic problem, they may fall into singular states, which lead to increased complexity in planning the trajectory of the object. Therefore, numerous methods in the literature have been devoted to motion planning in the joint space [21-31], mainly categorized as "complete method" and "probabilistically complete methods".

A complete planner [21-27] generates an exact collision free trajectory in a continuous configuration space, if one exists. For example, methods based on optimal control theory [21,22] attempt to find exact trajectories that optimize a cost function by defining necessary conditions on the trajectories. However, a main drawback of these methods is that the number of variables and the complexity of the formulation rapidly increase due to an increase in the degrees of freedom and in the number of obstacles. In addition, the defined necessary conditions are too complex for almost any robot system and cannot be solved analytically. Therefore, numerical methods, such as the "nonlinear optimization method", have been proposed [23,24] to solve these problems by transforming the optimal control problem to a finite dimensional parameter optimization problem. However, in some cases, these methods cause the control parameter space to be littered with many local minima. Hence, these methods require an initial guess and the solution reached depends heavily on that guess. Additionally, while nonlinear optimization methods may result in rapid convergence to a locally optimal trajectory, there are no guarantees that the solver will be able to find any solution [34]. Other heuristic methods [25,26] and a method based on artificial potential fields [27] have also been proposed to solve a part of these problems. However, these methods can only be applied in a limited range of mechanisms with low dimensional configuration spaces because of their high computational complexity.

As such, "probabilistically complete method" have been introduced as a substitute scheme to overcome most of the deficiencies of the previous methods. In spite of complete methods which consider the entire configuration space in finding a suitable path, these methods utilize a discrete representation of the configuration space in the motion planning process by random sampling techniques. Due to the effectiveness of these methods in cooperative manipulation systems, some recent efforts have been directed to generalize them for cooperative mobile manipulators [28-31]. For instance, in [28] a single query method based on the Rapidly-exploring Random Tree algorithm (RRT) has been presented. This method constructs a randomized tree between the start and final configurations and searches for an optimal path during the construction phase. A disadvantage of this method is that the generated tree is only valid for certain configurations and, in case of a new query, another tree must be constructed, limiting the range of applications that can take advantage of this method [32]. Therefore, a multiple query, two-stage method in which a graph, "Roadmap", is first 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, has been proposed [29]. Then, the method populates the environment with copies of the kinematic roadmap in random locations and connects collision-free configurations of the same closure type to build the final roadmap. The method is used more in real-time practical situation because of its higher speed in the query phase and ability to change its goal configuration in an online manner. However, in this method, the probability of satisfying loop closure equations in a randomly sampled configuration is nearly zero and this fact lowers the performance of the algorithm. To solve this problem, a simple and general geometric guided sampling algorithm called Random Loop Generator (RLG) has been proposed [30,31] that notably increases the probability of obtaining real solutions when solving the loop closure equations. However, many deficiencies and problems still need to be considered.

A main deficiency of the methods presented for randomized motion planning of cooperative mobile manipulators is that they ignore differential constraints in the motion planning process [33]. However, these constraints should be considered in order to calculate a reliable and efficient solution. Two approaches have been proposed to consider differential constraints in motion planning: decoupled and direct approaches. The decoupled approach [35,36], 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. This has the desirable benefit of decomposing the complexity of motion planning in two steps. 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 high [42]. These deficiencies motivated the development of the direct approach [37-44]. In this approach, differential constraints are considered in the planning process, resulting in more reliable and efficient solutions. However, the solutions are believed to be more difficult than those of the decoupled approach.

Another significant challenge in randomized motion planning is to prevent the system from colliding with moving obstacles, which considerably increases the complexity of the motion planning process [45]. In this regard, various modified and combined versions of RRT [46-48] and PRM [49,50] methods have been proposed that adapt to dynamic environments, which update only the necessary portions of the corresponding graph that are relevant to the moving obstacles. However, in these methods, when uncertainty in the environment is found, a path is completely replanned using the updated graph and, consequently, the computation time and memory requirements grow exponentially in the dimension of the state space, making this approach impractical for high-dimensional systems.

In order to solve these challenges, we propose a novel two-stage scheme that considers the trajectory planning problem of two mobile manipulators for cooperative transport of a rigid body. The environment is a 3-D space including static and moving obstacles. In addition, we only assume the robots have access to the position information of the obstacles in the environment. An offline method has been presented as the first stage to find an optimal trajectory in which nonholonomic and closed-chain constraints along with the joint and acceleration limits can be easily dealt with in presence of static obstacles. The method utilizes the advantages of the direct and decoupled approaches along with the ability of probabilistic methods to handle high dimensional configuration spaces resulting in a reliable and fast trajectory planning algorithm. Next, according to the prediction of the moving obstacles, the avoidance process is designed based on defining a *velocity adjustment zone* around the robots. When the obstacles move into this zone area, the robots move forward following the designated trajectory and avoid the moving obstacles by increasing or decreasing their motion velocity. This strategy has the desirable benefit of avoiding replanning, which decreases computational complexity and makes our method applicable in real-time implementation.

The paper is organized as follows: Section 2 introduces the model of the cooperative system. The fundamentals of the existing algorithms utilized in this paper and their extension to the proposed model of the mobile manipulator system are provided in Section 3. In Section 4, the details of our new method for trajectory planning of the cooperative mobile manipulators under differential constraints in the presence of fixed and moving obstacles are discussed. Finally, we present simulation results in Section 5 to show the effectiveness of the algorithm, and provide concluding remarks in Section 6.

2. Model of the Cooperative System.

Figure 1 shows our selected model including a cooperative system of two-wheeled mobile manipulators transporting a common payload. Each mobile manipulator module consists of a wheeled mobile robot with a 5-DOF mounted, planar and revolute-joint manipulator. Two rotational joints have been considered at the bottoms and the tips of both manipulators with the aim of making the system applicable to more practical situations. It is assumed that the motion of vehicles is restricted to the horizontal plane and both of the end effectors catch the object tightly. The assumption that the joints are rotational can be easily relieved, however, since the mixed use of both prismatic and rotational joints will complicate some analytical derivations. However, the authors feel they should preserve clarity in the presentation by using one type only.

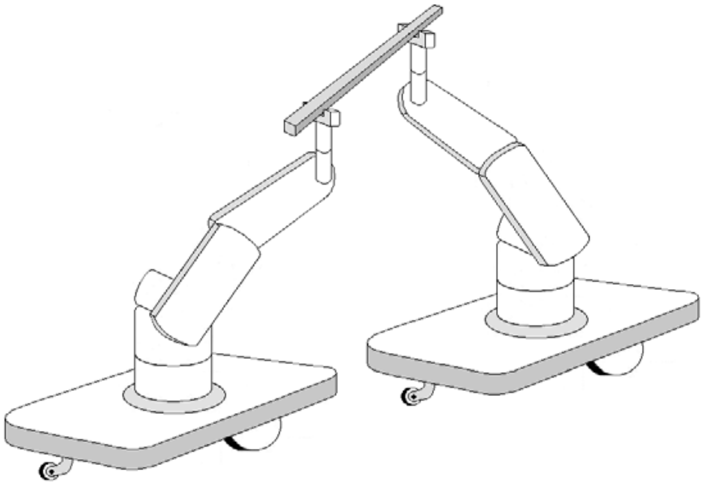


FIGURE 1. Cooperative transportation by a dual mobile manipulator system

The problem is now 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 fixed and moving 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, non-holonomic and dynamic constraints) in such a system makes the motion planning problem complicated and requires careful evaluation to realize the payload manipulation task efficiently. Therefore, in the remainder of this section, we use the above model to generate the constraint equations of the system.

2.1. **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 one 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:

$l_i\,(i = 1, \ldots, 6)$: The length of i^{th} link in the closed kinematic chain;

$\theta_i\,(i = 1, \ldots, 6)$: The i^{th} joint angle rotating in the vertical plane in the closed kinematic chain;

$\theta_{pi}\,(i = 1, \ldots, 4)$: The i^{th} joint angle rotating in the horizontal plane in the closed kinematic chain;

l_{obj}: The length of object between two end effectors;

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

$\theta_{b_1}, \theta_{b_2}$: Platform orientations of the mobile manipulators.

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

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

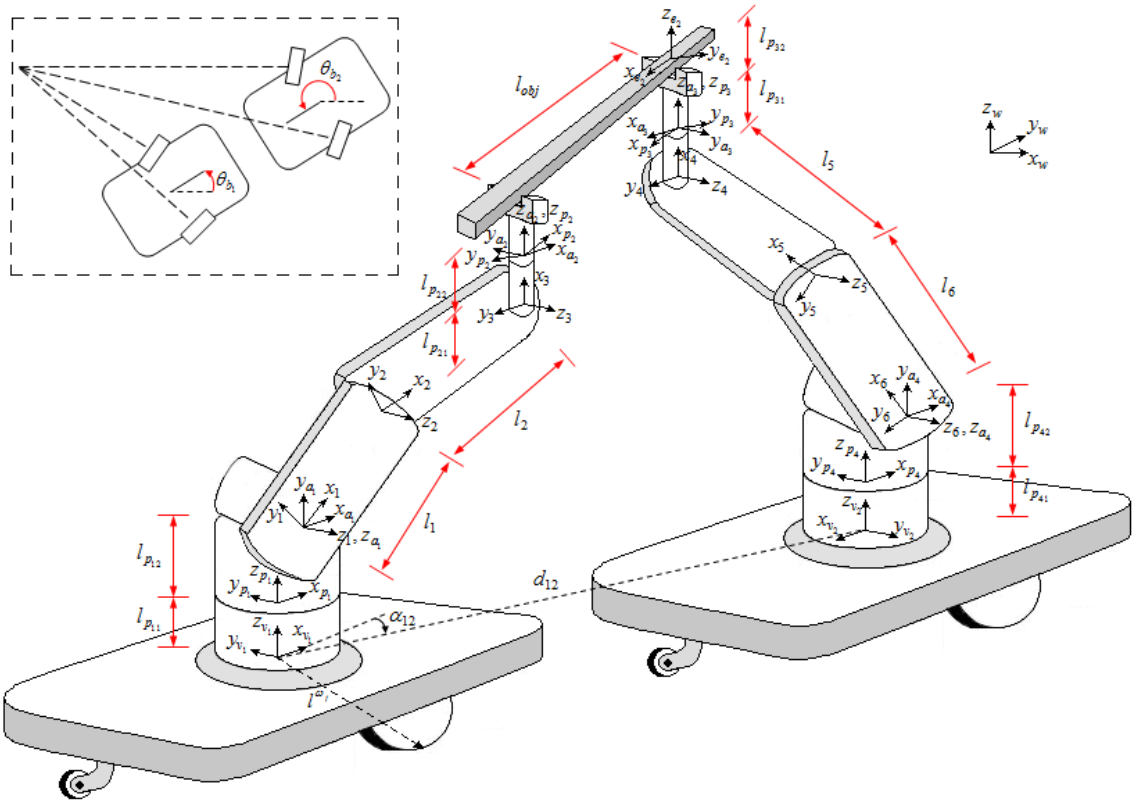


FIGURE 2. Cooperative system with its attached coordinate systems

Proof of this equation can be found in Appendix A. However, in order to lower the computational cost of our method, we design the desired trajectory in a way that both mobile bases have fixed positions and orientations relative to each other. To this end, we define the following conditions:

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

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

$$\theta_{p_1} = \theta_{p_2} = \theta_{p_3} = \theta_{p_4} = 0 \tag{3}$$

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

$$\begin{aligned} &\theta_1 + \theta_2 + \theta_3 = \theta_4 + \theta_5 + \theta_6 \\ &l_3 c_{123} + l_2 c_{12} + l_1 c_1 + l_{obj} s_{456} = l_4 c_{456} + l_5 c_{56} + l_6 c_6 + g_b \\ &l_3 s_{123} + l_2 s_{12} + l_1 s_1 - l_{obj} c_{456} = l_4 s_{456} + l_5 s_{56} + l_6 s_6 \end{aligned} \tag{4}$$

in which, we define the following parameters:

$$\begin{aligned} &c_{ijk} = \cos(\theta_i + \theta_j + \theta_k), \ c_{ij} = \cos(\theta_i + \theta_j), \ c_i = \cos(\theta_i) \ \ i, j, k \in \{1, 2, 3, 4, 5, 6\} \\ &s_{ijk} = \sin(\theta_i + \theta_j + \theta_k,), \ s_{ij} = \sin(\theta_i + \theta_j), s_i = \sin(\theta_i) \ \ i, j, k \in \{1, 2, 3, 4, 5, 6\} \\ &\quad l_3 = l_{p_{21}} + l_{p_{22}}, \ l_4 = l_{p_{31}} + l_{p_{32}} \end{aligned} \tag{5}$$

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

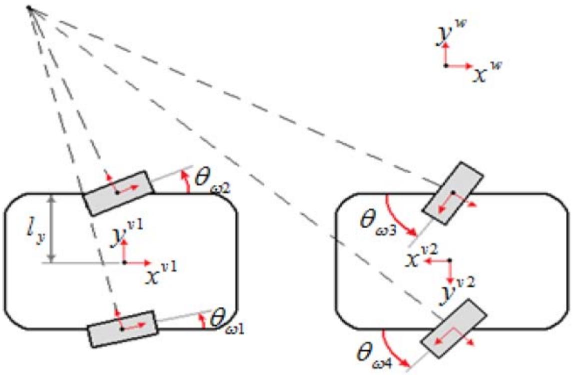


FIGURE 3. Base structure subsystem of the cooperative system

To derive the equations due to these constraints, we need to compute the velocity of each wheel in its attached coordinate system (see Figure 3). In this regard, we can write [51,52]:

$$v^{\omega_j} = \dot{x}_{b_i}\mathbf{x^w} + \dot{y}_{b_i}\mathbf{y^w} + \dot{\theta}_{b_i}\left(\mathbf{z}^{v_i} \times l^{\omega_j}\right) \tag{6}$$

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 (see Figure 2). Hence, the velocity of each wheel can be written as:

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

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

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

The non-skidding condition implies that the second term of the velocity vector in Equation (7) vanishes. Therefore, for each of the mobile manipulators these constraints can be taken into account as:

$$-\dot{x}_{b_i}\sin\left(\theta_{b_i} + \theta_{\omega_j}\right) + \dot{y}_{b_i}\cos\left(\theta_{b_i} + \theta_{\omega_j}\right) + \dot{\theta}_{b_i}\left(l_y^{\omega_j}\sin\theta_{\omega_j}\right) = 0 \tag{9}$$

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

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

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

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

Furthermore, defining r^{ω_j} as radius of the j^{th} wheel, non-slipping constraints can be written as [51,52]:

$$v_i^{\omega_j}=\omega_i^{\omega_j}\times r^{\omega_j}z^{\omega_j}\tag{12}$$

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

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

3. **Preliminaries.** The success of PRM approaches is mainly due to their great efficiency, reliable performance, conceptual simplicity and applicability to many different types of problems. The general methodology of PRMs is to construct a graph (roadmap) during a preprocessing stage that represents the connectivity of the robot's free configuration space and then to query the roadmap using an optimal graph-search algorithm (e.g., A*) in order to find the shortest possible path between start and final configurations. In this section, we apply this method to our cooperative mobile manipulators system. Focusing on the system's model, we define a generalized coordinate vector as:

$$q=\left[\begin{array}{cccccc}x_b & y_b & \theta_b & \theta_1 & \cdots & \theta_6\end{array}\right]^T\tag{14}$$

where x_b and y_b denote the position of the first vehicle in the world coordinate system $\{w\}$ and θ_b is its angular position about axis z^w.

3.1. **Roadmap construction.** Some efforts have been recently made to apply the PRM method to closed chain systems [28-31,53,54]. However, in the large workspace of a mobile manipulator system, PRM methods require several hours of computation time to generate a well-connected roadmap. To solve this problem, we fix the position and orientation of both manipulator bases first, and construct a roadmap (fixed-base roadmap) that contains n different self-collision-free closure configurations. Then, we populate the environment with copies of the kinematic roadmap and connect configurations of the same closure type.

However, a main problem in the first stage is that too many samples may 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 PRM approach with a geometric guided sampling method called Random Loop Generator [30], which notably increases the probability of obtaining real solutions when solving the loop closure equations. The basic principle as stated in Algorithm 1 is grounded on separating configuration variables into two sets: active variables (q_a) and passive variables (q_p).

Algorithm 1. RLG method for guided random sampling in the roadmap construction phase

```
specify active and passive subchains as $q_a$ and $q_p$
for each active variable
    compute an interval that closed-chain constraint equations have a solution in it.
    choose the active variable randomly from the computed interval
end for
solve the closed-chain constraint equations for passive variables after substituting
  active variables.
if there is a solution
    use $q_a$ and $q_p$ to construct the closed-chain random configuration
(repeat until attaining a real solution)
```

The only limitation is to require that the joint space variables in q_a and q_p correspond to consecutive joints in the mechanism. In the remainder of this paper, we refer to active and passive variables as follows:

$$\begin{aligned} q_a &= [\theta_1, \theta_2, \theta_6]^T \\ q_p &= [\theta_3, \theta_4, \theta_5]^T \end{aligned} \tag{15}$$

The planner directly acts on the active variables while the passive ones are obtained by solving loop-closure equations. One of the important parts of Algorithm 1 is to compute an interval for each of the active variables, thereby increasing the probability of having real solutions for the 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. Because illustration of the workspace of the closed-chain, is a very complicated task, we have proposed an approximate approach. A simplified model of our system is shown in Figure 4.

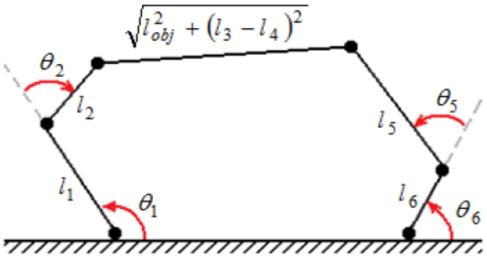


Figure 4. A simplified model of the fixed base system to be used in RLG algorithm

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:

$$\begin{aligned} \hat{r}_{ext} &= \sum_{i=1}^{k} L_i \\ \hat{r}_{int} &= \begin{cases} 2L_{\max} - \hat{r}_{ext} & 2L_{\max} > \hat{r}_{ext} \\ 0 & o.w. \end{cases} \end{aligned} \tag{16}$$

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.

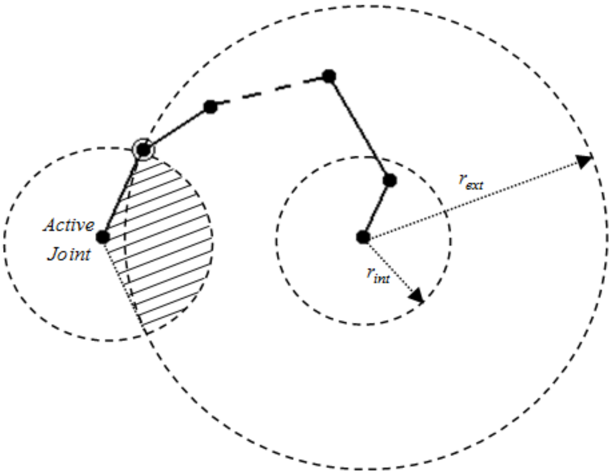


Figure 5. Subset of values for an active joint to be reachable by the remaining chain of the system

After choosing each active variable in its computed interval, we solve the closed-chain constraint equations for the passive variables. Now, the configuration vector of the fixed-base system, which is a minimal set of parameters defining the location of the system in the world frame, is written as:

$$\theta = \left[q_a^T, q_p^T\right]^T \tag{17}$$

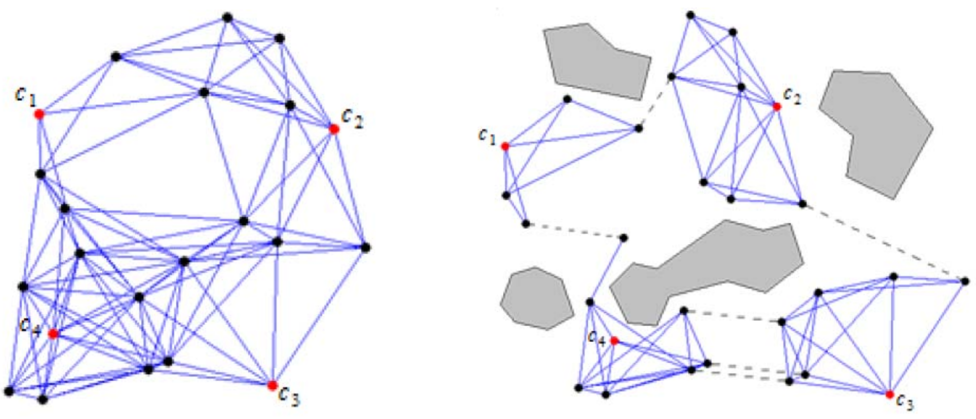
In the second stage, we consider the base structure mobility, utilizing Algorithm 2, and populate the entire workspace with randomly selected parts of the initial roadmap. Towards this end, we choose the base configuration, $g_{\omega b}$, as a random vector including the position and orientation of the first truck relative to the world frame:

$$g_{\omega b} = (x_b, y_b, \theta_b) \tag{18}$$

Algorithm 2. Populating the environment with copies of the fixed-base roadmap

```
generate random base configuration $g_{\omega b}$
choose random vertex $\theta$ repeatedly from the fixed-base roadmap until attaining
 a collision-free configuration ($g_{\omega b}$, $\theta$)
if there exist a collision-free configuration
    retain ($g_{\omega b}$, $\theta$) as a roadmap vertex
    for each neighbor of $\theta$ , say $\tilde{\theta}$, in the fixed-base roadmap
        if ($g_{\omega b}$, $\tilde{\theta}$) is collision-free
            retain ($g_{\omega b}$, $\tilde{\theta}$) as a roadmap vertex
            retrieve the path $\theta(t)$ connecting $\theta$ and $\tilde{\theta}$ from the fixed-base roadmap
            if ($g_{\omega b}$, $\theta(t)$) is collision-free for all intermediate configurations along the path
                add an edge between ($g_{\omega b}$, $\theta$) and ($g_{\omega b}$, $\tilde{\theta}$)
(repeat as desired)
```

Then, we sample a node, θ, randomly from the initial roadmap and check the combined vector of ($g_{\omega b}$, 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 collect all roadmap nodes with the same closed configuration in a set and use PRM connection method to connect the nodes in the set as illustrated in Figure 6. Finally, we search the graph for the shortest possible path between the start and final configurations with an optimal graph search algorithm.



(a) (b)

FIGURE 6. (a) Initial roadmap without base structure mobility and (b) distributing the initial roadmap to consider the base structure mobility

3.2. **Collision detection.** 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 collisions or incorrectly detecting collisions, even when the workspace has complex geometry.

We use two types of collision checks in our motion planner: static checks, which are used to test whether a sampled configuration in the roadmap is in the free space, and dynamic checks to test its local paths, which are continuous sets of configurations. In static methods, a common approach is to break complex objects (robot link, obstacle, etc.) down using a bounding volume hierarchy (BVH), which is a hierarchy of BVs (e.g., spheres) that approximates the geometry of the object at successive levels of detail [55].

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 quickly test for collision with each other in conjunction with a more precise but expensive type of bounding volume according to Figure 7.

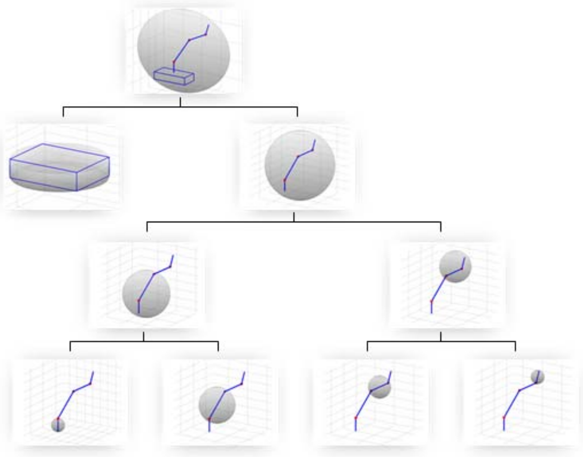


FIGURE 7. A BVH model for the mobile manipulator system

Static Collision Checking. This method checks two objects for collisions by searching their BVHs from the top down, making it possible to quickly discard large subsets of objects contained in disjoint BVs. In other word, if the BVs at the top level are in collision, then their children are also checked to determine if they are in 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 [56] 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 using the following methodology:

$A_i(q)$ denotes 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 nontrivial lower bound on the Euclidian distance between $A_i(q)$ and $A_j(q)$. $\lambda_i(q_a, q_b)$ is 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 configuration space is [56]:

$$\lambda_i\left(q_a, q_b\right) + \lambda_j\left(q_a, q_b\right) < n_{ij}\left(q_a\right) + n_{ij}\left(q_b\right) \tag{19}$$

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, a classical BVH collision checker is used. However, instead of testing if two BVs intersect, we compute the distance between them to find the closest distance.

3.3. **Distance metric.** In addition to collision detection, which is an important primitive operation in any motion planner, all PRMs make heavy use of distance computations. As it would be infeasible to attempt to connect all possible pairs of nodes, distance metrics are used to determine which pairs we should try to connect when building the roadmap. Similarly, to obtain fast query items, it is necessary to limit the number of connection attempts from the start and final configurations to the roadmap nodes. Thus, it should be noted that the connectivity of the roadmap and the success of the queries depends largely on the correct selection of an efficient and fast distance metric. Assuming that c_1 and c_2 are two configurations in the configuration space, we use a metric in the form of Manhattan formulation as follows [57]:

$$d\left(c_1, c_2\right) = \sum_{k=x_b, y_b} P_{12}\left(k\right) + \sum_{k=\theta,\theta_1,\theta_2,\ldots,\theta_6} Q_{12}\left(k\right) \tag{20}$$

where:

$$\begin{aligned} Q_{12}\left(k\right) &= \left|c_2\left(k\right) - c_1\left(k\right)\right|, \quad k \in \{\theta_b, \theta_1, \ldots, \theta_6\} \\ P_{12}\left(k\right) &= n_{12}\left|c_2\left(k\right) - c_1\left(k\right)\right|, \quad k \in \{x_b, y_b\} \end{aligned} \tag{21}$$

and n_{12} is a weighing factor. It should be noted that choosing the weighing factor involves a tradeoff between the path smoothness of the joint space variables and that of the x-y-z position of the object towards the final configuration.

4. Design of the Trajectory under Closed-chain and Differential Constraints.

Most of the motion planning techniques for closed chain mechanisms could not directly account for differential constraints, which could render the planned trajectory infeasible. In this section, we find a trajectory between intermediate points generated during the path planning process, considering constraints in the form of differential equations. A heuristic method is also proposed to keep the system from colliding with moving obstacles.

4.1. **Trajectory generation without moving obstacles.** A challenging problem during the planning process is that the differential constraints must be satisfied with regard to the closed-chain constraints. Here, we present an extension of the RLG algorithm to solve such a trajectory planning problem. First, we describe the trajectory of each active variable between the sequential pairs of intermediate points by 3rd order polynomial equations, which are expressed in a normalized interval of time as follows:

$$\begin{aligned} q_{ik}(s) &= a_{0ik} + a_{1ik}s + a_{2ik}s^2 + a_{3ik}s^3 \\ & \qquad i \in \{x_b, y_b, \theta_b, \theta_1, \theta_2, \theta_6\},\ k = 1, 2, \ldots, n_{seg},\ 0 \le s \le 1 \end{aligned} \tag{22}$$

where n_{seg} is the number of path segments and s is defined as:

$$s \triangleq \frac{t}{t_{fik}} \tag{23}$$

where t_{fik} is the time required to traverse the k^{th} segment of the path. We use the position and velocity values at the endpoints of the path segment to calculate the coefficients of q_{ik} in Equation (22) as follows:

$$\begin{aligned} a_{0ik} &= q_{ik}(0) \\ a_{1ik} &= \left.\frac{dq_{ik}}{ds}\right|_{s=0} \\ a_{2ik} &= 3\left(q_{ik}(1) - q_{ik}(0)\right) - \left(2\left.\frac{dq_{ik}}{ds}\right|_{s=0} + \left.\frac{dq_{ik}}{ds}\right|_{s=1}\right) \\ a_{3ik} &= \left.\frac{dq_{ik}}{ds}\right|_{s=0} + \left.\frac{dq_{ik}}{ds}\right|_{s=1} - 2\left(q_{ik}(1) - q_{ik}(0)\right) \end{aligned} \tag{24}$$

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

$$\begin{cases} \sin(\xi + \theta_4) = \dfrac{(l_3 - l_4)^2 + {l_{obj}}^2 + l_5^2 - \beta^2 - \gamma^2}{2l_5\sqrt{{l_{obj}}^2 + (l_3 - l_4)^2}} \\ \theta_3 = \operatorname{atan2}\left(\dfrac{\beta}{\alpha}, -\dfrac{\gamma}{\alpha}\right) - \operatorname{atan2}\left(\dfrac{d}{\alpha}, \dfrac{d'}{\alpha}\right) \\ \theta_5 = \theta_1 + \theta_2 + \theta_3 - \theta_4 - \theta_6 \end{cases} \tag{25}$$

where:

$$\begin{cases} \xi = \arctan\left(\dfrac{l_3 - l_4}{l_{obj}}\right) \\ \beta = g + l_6\cos(\theta_6) - l_1\cos(\theta_1) - l_2\cos(\theta_1 + \theta_2) \\ \gamma = l_6\sin(\theta_6) - l_1\sin(\theta_1) - l_2\sin(\theta_1 + \theta_2) \\ \alpha = \sqrt{\beta^2 + \gamma^2} \\ d = (l_3 - l_4)\cos(\theta_1 + \theta_2) + l_{obj}\sin(\theta_1 + \theta_2) - l_5\cos(\theta_1 + \theta_2 - \theta_4) \\ d' = -(l_3 - l_4)\sin(\theta_1 + \theta_2) + l_{obj}\cos(\theta_1 + \theta_2) + l_5\sin(\theta_1 + \theta_2 - \theta_4) \end{cases} \tag{26}$$

Now, we calculate the minimum time to go from one intermediate point to the next with regard to the maximum allowed acceleration. Towards this end, the following theorem establishes a lower limit for the trajectories in the form of cubic polynomials such as those in Equation (22).

Theorem 4.1. *Consider the cubic polynomial given by the following equation:*

$$q_{ik}(s) = a_{0ik} + a_{1ik}s + a_{2ik}s^2 + a_{3ik}s^3, \quad 0 \le s \le 1 \tag{27}$$

with $s \triangleq \dfrac{t}{t_{fik}}$ *and* $\left|\dfrac{d^2 q_{ik}}{dt^2}\right| \le \alpha_{ik}$*. The lower bound on* t_{fik} *that satisfies the acceleration limits of* q_{ik} *is:*

$$t_{fik} \ge \sqrt{\frac{2\max\left(\left|a_{2ik}\right|, \left|a_{2ik} + 3a_{3ik}\right|\right)}{\alpha_{ik}}} \tag{28}$$

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

$$\left|\frac{d^2 q_{ik}}{ds^2}\right| \le \max\left(\left|\frac{d^2 q_{ik}}{ds^2}\right|_{s=0}, \left|\frac{d^2 q_{ik}}{ds^2}\right|_{s=1}\right) \tag{29}$$

In other words:

$$\left|\frac{d^2 q_{ik}}{ds^2}\right| \le \max(\left|2a_{2ik}\right|, \left|2a_{2ik} + 6a_{3ik}\right|) \tag{30}$$

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

$$\left|\frac{d^2 q_{ik}}{dt^2}\right| \le \frac{2}{t_{fik}^2}\max(\left|a_{2ik}\right|, \left|a_{2ik} + 3a_{3ik}\right|) \tag{31}$$

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

$$\frac{2}{t_{fik}^2}\max(\left|a_{2ik}\right|, \left|a_{2ik} + 3a_{3ik}\right|) \le \alpha_{ik} \tag{32}$$

and Equation (28) 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 (25), we compute a linear approximation for the acceleration and write the following condition:

$$t_{fjk}^2 \ge \beta_{ins}\frac{\left.\dfrac{dq_{jk}}{ds}\right|_{s=1} - \left.\dfrac{dq_{jk}}{ds}\right|_{s=0}}{\alpha_{jk}}, \tag{33}$$
$$j \in \left\{\theta_3, \theta_4, \theta_5, \theta_{\omega_{1,2}}, \psi_{1,2}, \theta_{\omega_{3,4}}, \psi_{3,4}\right\}, \quad k = 1, 2, \ldots, n_{seg}$$

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 a safe bound. Obviously, 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 Equations (28) and (33):

$$t_{fk} = \max_{i,j} \left(t_{fik}, t_{fjk} \right) \tag{34}$$

To find these lower bounds, due to the lack of time response information, we utilized an approximation of dq_{ik}/ds in place of the actual velocities. In other words, we assumed that:

$$\left. \frac{dq_{ik}}{ds} \right|_{s=1} = \left. \frac{dq_{i(k+1)}}{ds} \right|_{s=0} \tag{35}$$

Hence, from Equation (23) we can write:

$$t_{fk} \left. \frac{dq_{ik}}{dt} \right|_{t=t_{fk}} = t_{f(k+1)} \left. \frac{dq_{i(k+1)}}{dt} \right|_{t=0} \tag{36}$$

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 Equation (34) to find an approximated velocity in each intermediate point through the previously mentioned numerical approach. Then, we can write:

$$\begin{aligned} &a_{0ik} = q_{ik}(0), \quad a_{1ik} = t_{fk} \left. \frac{dq_{ik}}{dt} \right|_{t=0} \\ &a_{2ik} = 3\left(q_{ik}(t_{fk}) - q_{ik}(0)\right) - t_{fk} \left(2 \left. \frac{dq_{ik}}{dt} \right|_{t=0} + \left. \frac{dq_{ik}}{dt} \right|_{t=t_{fk}} \right) \\ &a_{3ik} = t_{fk} \left(\left. \frac{dq_{ik}}{dt} \right|_{t=0} + \left. \frac{dq_{ik}}{dt} \right|_{t=t_{fk}} \right) - 2\left(q_{ik}(t_{fk}) - q_{ik}(0)\right) \end{aligned} \tag{37}$$

Finally, by substituting Equation (37) into Equation (28) and using Equations (33) and (34) we can compute new values for t_{fk}.

4.2. **Avoiding moving obstacles based on velocity adjustment.** In this section, to solve the problem of collision avoidance of dynamic obstacles, a heuristic method is proposed in which the pattern of human behavior in driving is modeled. When a typical driver considers the probability of a crash on the road, he/she usually prefers to reduce his/her speed instead of changing his/her path, and after the obstacle, he/she increases his/her speed to the previous rate. Moreover, if the delay due to crossing the obstacle is greater than a certain amount of time, changing the path is preferred. Using this model eliminates the need for replanning in many situations. However, it shall be noted that any change in velocity in our cooperative system may lead to violation of kinematic and differential constraints. In this regard, the following theorem converts the challenge of speed adjustment to the problem of designing a time-dependent factor without violating the system constraints.

Theorem 4.2. *Assume that $q(t)$ and $\dot{q}(t)$ are the position and velocity vectors of the closed chain system in the presence of nonholonomic constraints. Consider $\beta(t)$ as an arbitrary time scaling function. Then, substituting $q(\beta(t))$ and $\dot{\beta}(t)\,\dot{q}(\beta(t))$ for the previous position and velocity vectors still satisfies the closed chain and nonholonomic constraints.*

Proof: This theorem can be easily verified by using Equations (4), (9) and (13).

Now, let us define $q_d(t)$ and $\dot{q}_d(t)$ as position and velocity vectors without considering the moving obstacles. Hence, from Theorem 4.2, these vectors in the presence of moving obstacles can be written as:

$$\begin{aligned} \dot{q}_{dm}(t) &= \alpha(t)\, \dot{q}_d(\beta(t)) \\ q_{dm}(t) &= q_d(\beta(t)) \end{aligned} \tag{38}$$

where:

$$\beta(t) = \int_0^t \alpha(t)\, dt \tag{39}$$

and $\alpha(t)$ is a continuous function which can be utilized to adjust the velocity of the system to keep it from colliding with moving obstacles. To design a suitable velocity adjustment factor, we need to predict the distance from the moving obstacles. Therefore, we use a linear estimator that periodically samples the position of each obstacle with a suitable sampling period, Δ, and evaluates its position at the next sampling time. Now, let d_{th} be a threshold value for the distance from moving obstacles that determines the necessity of velocity adjustment. Then, considering $d_i(k\Delta)$ as the distance from i^{th} moving obstacle at the k^{th} sampling time, we can define the following factor in the same sampling time as shown in Figure 8:

$$\alpha(k\Delta) = \min\left(\min_i\left(\frac{d_i(k\Delta)}{d_{th}}\right), 1\right) \quad k = 0, 1, 2, \ldots \tag{40}$$

which decreases the velocity of the system as a function of the minimum distance from moving obstacles if they are closer than d_{th}. Finally, from Figure 8 together with Equation (40), the continuous-time velocity adjustment factor can be expressed as:

$$\alpha(t) = \alpha(k\Delta) + \frac{[\alpha((k+1)\Delta) - \alpha(k\Delta)]}{\Delta}(t - k\Delta) \tag{41}$$

where:

$$k = Integer\ Part\left(\frac{t}{\Delta}\right) \tag{42}$$

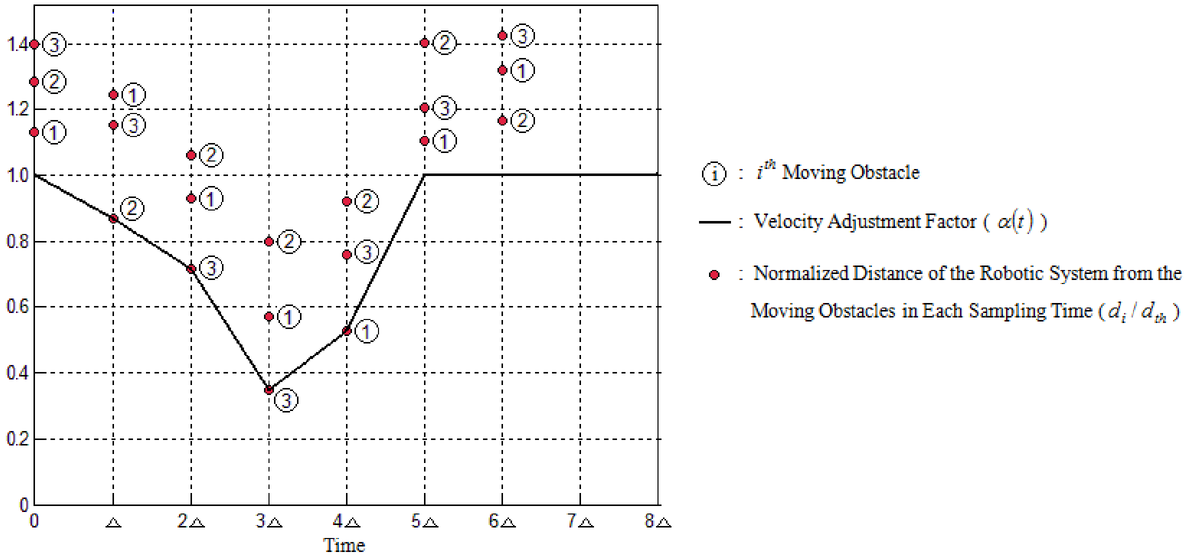


Figure 8. Continuous-time velocity adjustment factor (see Equations (40) and (41))

Theorem 4.3. *The velocity adjustment method defined in Equations (38)-(42), preserves the previous upper bounds on the accelerations.*

Proof: From Equation (40) we can write:

$$\alpha(t) \leq 1 \tag{43}$$

which by using Equation (38), implies that:

$$\dot{q}_{dm}(t) \leq \dot{q}_d(\beta(t)) \tag{44}$$

Now, differentiating both sides of the above inequality gives:

$$\ddot{q}_{dm}(t) \leq \alpha(t)\, \ddot{q}_d(\beta(t)) \tag{45}$$

Therefore, from Equation (43) it is easy to conclude that the method does not violate the previous bounds on the accelerations.

5. **Simulation Results.**

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

- Each mobile manipulator is subject to the differential constraints mentioned in Equations (9) and (13) and its goal is to cooperate with the other in a manner that satisfies the closed-chain constraints in Equation (1).
- The transporting object is a rigid body, which cannot be deformed.
- Some important parameters used in the planner are chosen as per the following table:

Table 1. Simulation parameters

Parameter	n	m	g	l_{obj}	l_1	l_2	l_3	l_4	l_5	l_6	d_{th}
Value	300	200	1.2	1.2	0.8	0.5	0.3	0.3	0.5	0.8	3

- The start and final configurations are set to:

$$q_{start} = \left[0, 0, 0, \frac{\pi}{2}, 0, 0, 0, 0, \frac{\pi}{2}\right]^T, \quad q_{final} = \left[10, 10, \frac{\pi}{4}, 0, \frac{\pi}{2}, 0, 0, \frac{\pi}{2}, 0\right]^T \tag{46}$$

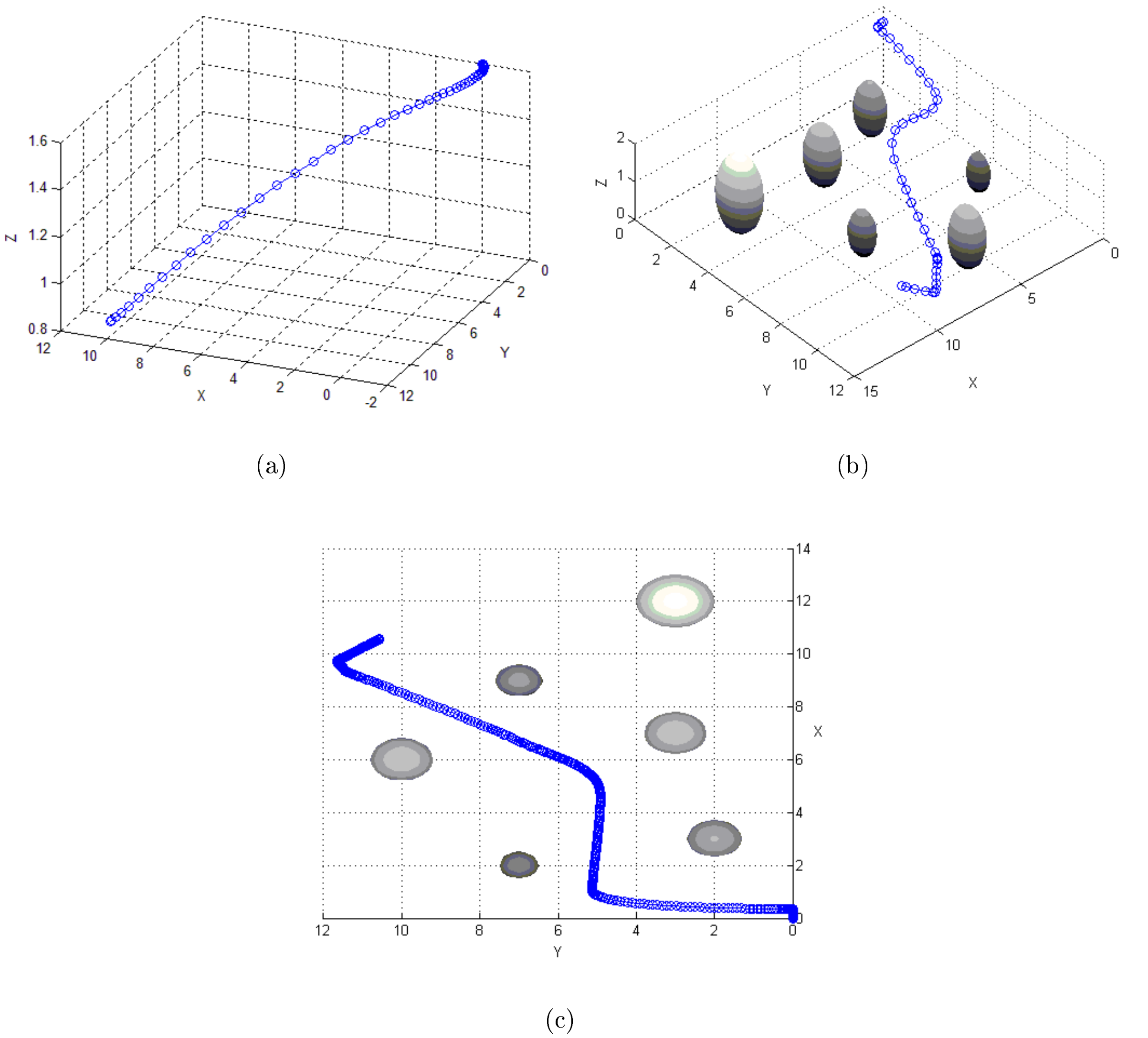
- The environment is populated with six static obstacles in the form of spheres with specified positions and radii as shown in Figures 9(b) and 9(c).
- A moving obstacle passes the workspace with the following trajectory:

$$\begin{cases} x_{obs}(t) = \dfrac{t}{2} \\ y_{obs}(t) = 10 - \dfrac{t}{2} \end{cases} \tag{47}$$

Figure 9 shows a 3-dimensional visualization for the trajectory of the object with and without static obstacles. As illustrated, there is a smooth trajectory between the start and final configurations in both cases. Furthermore, accelerations of the computed trajectories for joint space variables are shown in Figure 10 and it can be seen that they are bounded within the predefined maximum allowed accelerations (1 rad/s^2).

The quality of the designed trajectory is shown in Figure 11. This figure illustrates the deviation of the joint space variables from their final values neglecting moving obstacles. The errors converge to zero as time goes on.

However, as shown in Figure 12(a), in presence of a moving obstacle, this leads to a collision with the obstacle (collision zone). Therefore, we use the velocity adjustment approach to avoid collision. In this regard, based on the theory presented in the previous section we evaluate $\alpha(t)$ as shown in Figure 13. This has the benefit of collision avoidance with the moving obstacle as illustrated in Figure 12(b).



(a)

(b)

(c)

FIGURE 9. Object's trajectory between predefined start and goal configurations: (a) without obstacle, (b) with obstacles and (c) with obstacles (top view)

If the moving obstacle moves into the velocity adjustment zone, the velocity adjustment strategy is adopted to avoid it as shown in Figure 14. Every gray circle in the figure denotes the simplified model of the cooperative mobile manipulator system in its current position and those in black represent the moving obstacle's current position. The number within the circles marks the sampling order. Figure 14(a) shows that a collision occurs in the 3^{rd} sampling time by utilizing the initial trajectory. However, by using the velocity adjustment method, the obstacle is avoided as illustrated in Figure 14(b), though it takes longer to reach the final configuration (see Figure (15)).

6. **Conclusions.** A heuristic trajectory planning method has been proposed for cooperative transportation of an object utilizing a dual mobile manipulator system in presence of fixed and moving obstacles. The method utilizes the advantages of direct and decoupled approaches along with the ability of probabilistic methods to present a reliable and fast

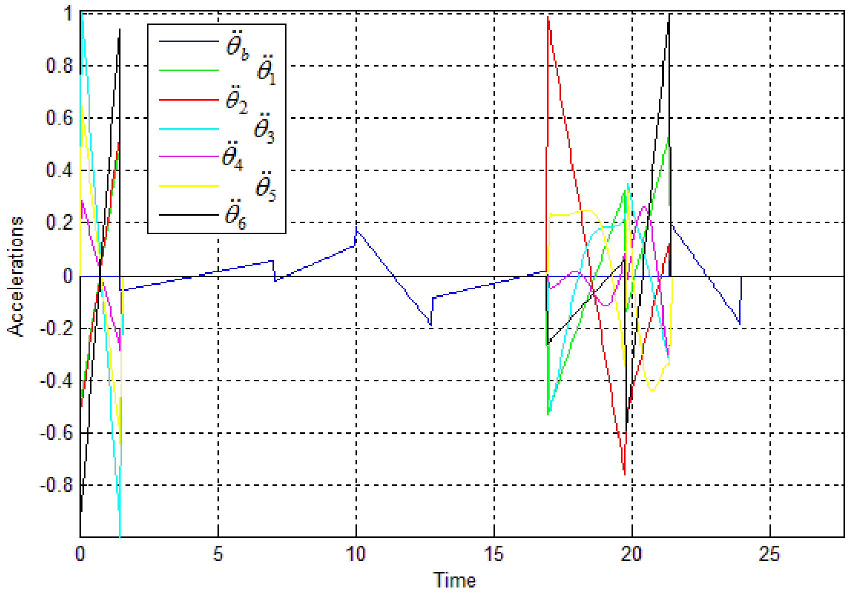


FIGURE 10. Joint accelerations of the manipulators

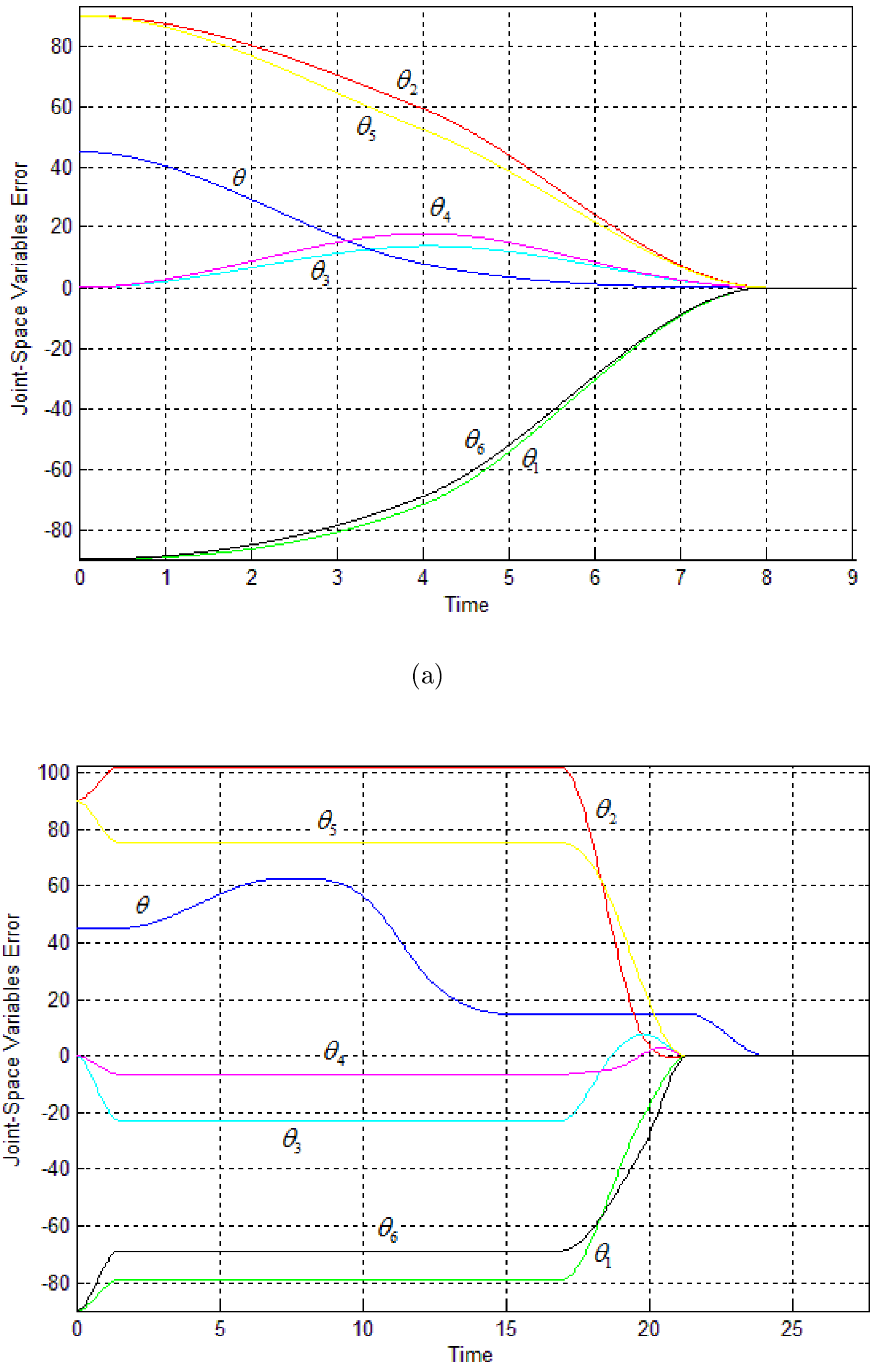
solution. Furthermore, the results of computer simulations confirmed the effectiveness of the method.

In comparison with other approaches discussed in the literature, advantages of the proposed method are summarized as follows and have been shown in greater detail in Table 2.

TABLE 2. 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 Dynamic Environments**	**Applicable to High Dimensional Systems**
Complete Methods	Optimal Control Based [21,22]	Very High	Yes	Yes	No	No
	Numerical Optimization [23,24]	High	Yes	No	No	No
	Artificial Potential Fields [27]	Very High	Yes	Yes	No	No
Randomized or Probabilistic Complete Methods	Single Query Methods [28]	Moderate	No	Yes	No	Yes
	Multiple Query Methods [28-31]	Low	No	Yes	No	Yes
Our method		Low	Yes	Yes	Yes	Yes

1) Using advantage of decoupled methods, the complexity of trajectory planning can be decomposed in two steps, which increases the simplicity of the planning process.
2) Because of its structure, the method can handle high dimensional configuration spaces efficiently and it is not necessary to construct an exact representation of the free space. Therefore, a simple and fast method is achieved instead of a more accurate but slower one.



(a)

(b)

FIGURE 11. Errors of the joint space variables: (a) without obstacle and (b) with obstacles

3) Various types of constraints such as nonholonomic and closed-chain constraints, along with joint angle and acceleration limits, can be easily dealt with in presence of collision constraints with fixed and moving obstacles.

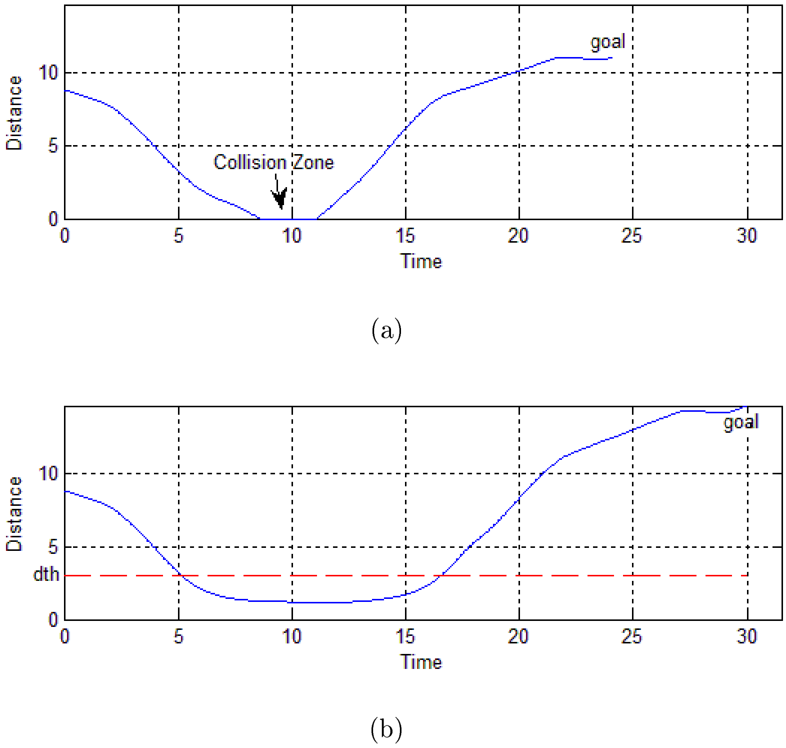


FIGURE 12. Minimum distance from moving obstacle: (a) without velocity adjustment and (b) with velocity adjustment

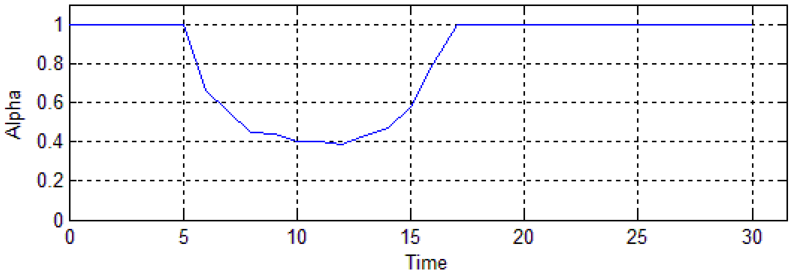
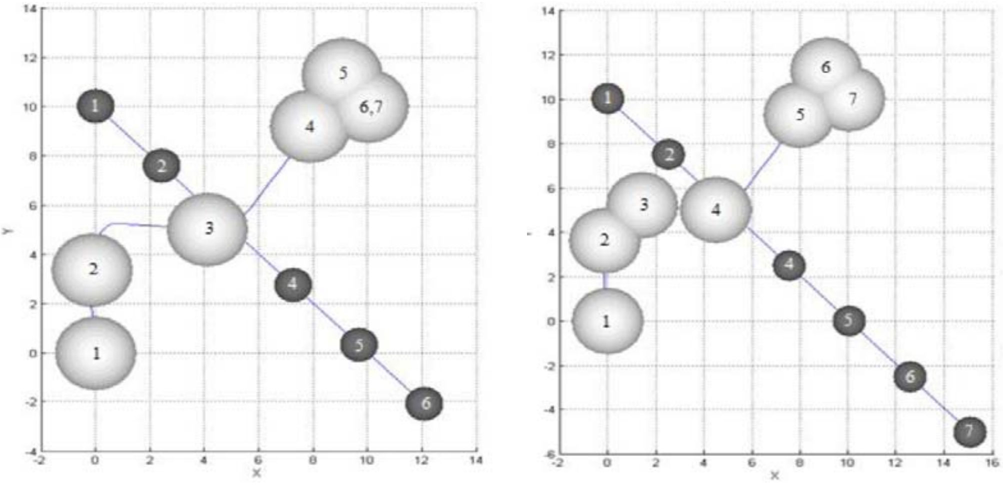


FIGURE 13. Velocity adjustment factor

4) Moving obstacles with unknown dynamics are avoided through a heuristic velocity adjustment method, which does not violate closed-chain and differential constraints of the system. The method has the desirable benefit of avoiding replanning.
5) The method can avoid any number of moving obstacles without a significant increase in the computational complexity and memory requirements.

Although it is usually possible to keep the system from colliding with moving obstacles by adjusting its velocity, in some emergency cases, the system must escape out the moving field of the obstacles by changing the designated trajectory. Therefore, a good idea to improve the algorithm proposed in this work is to design an optimal detour with minimum deviation from the initial trajectory when moving obstacles enter a predetermined security area. Also, investigating the trajectory planning of more than two cooperating mobile manipulators to perform the carrying task seems to be an interesting way to follow.



(a) (b)

Figure 14. Escaping out the moving field of the obstacle by velocity adjustment: (a) initial trajectory and (b) modified trajectory

REFERENCES

[1] C. Chen and D. Dong, Grey system based reactive navigation of mobile robots using reinforcement learning, *International Journal of Innovative Computing, Information and Control*, vol.6, no.2, pp.789-800, 2010.

[2] V. Kroumov, J. Yu and K. Shibayama, 3D path planning for mobile robots using simulated anealing neural network, *International Journal of Innovative Computing, Information and Control*, vol.6, no.7, pp.2885-2899, 2010.

[3] X. Dai, X. Ning and Y. Shi, A novel path planning algorithm for mobile robots based on cloud model, *ICIC Express Letters*, vol.3, no.4(A), pp.877-881, 2009.

[4] Y. Cheng, X. Wang and R. Lei, A fuzzy control system for path following of mobile robots, *ICIC Express Letters*, vol.3, no.3(A), pp.403-408, 2009.

[5] Y. Cao, A. S. Fukunaga and A. B. Kahng, Cooperative mobile robotics: Antecedents and directions, *Autonomous Robots*, vol.4, no.1, pp.7-27, 1997.

[6] A. Farinelli, L. Iocchi and D. Nardi, Multi-robot systems: A classification focused on coordination, *IEEE Transactions on Systems, Manufacturing and Cybernetics*, vol.34, no.5, pp.2015-2028, 2004.

[7] J. Zheng, H. Yu, M. Zheng, W. Liang and P. Zeng, Coordination of multiple mobile robots with limited communication range in pursuit of single mobile target in cluttered environment, *Journal of Control Theory and Applications*, vol.8, no.4, pp.441-446, 2010.

[8] S. Liu, D. Sun and C. Zhu, Coordinated motion planning for multiple mobile robots along designed paths with formation requirement, *IEEE/ASME Transactions on Mechatronics*, 2004.

[9] M. D. Zivanovic and M. K. Vukobratovic, *Multi-Arm Cooperating Robots: Dynamic and Control*, Springer, 2006.

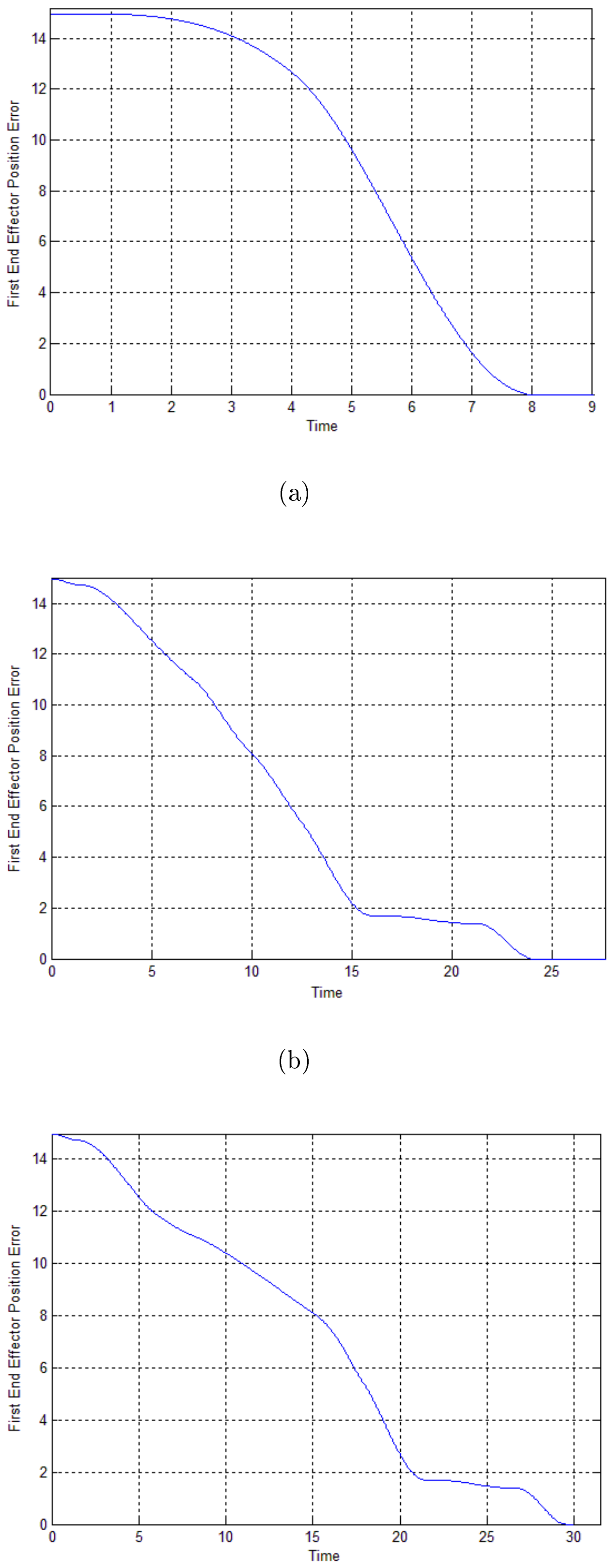
[10] H. Homaei and M. Keshmiri, Optimal trajectory planning for minimum vibration of flexible redundant cooperative manipulators, *Advanced Robotics*, vol.23, no.12-13, pp.1799-1816, 2009.

[11] B. Balaguer and S. Carpin, Motion planning for cooperative manipulators folding flexible planner objects, *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.3842-3847, 2010.

[12] J. K. Salisbury and J. J. Craig, Articulated hands: Force control and kinematic issues, *The International Journal of Robotics Research*, vol.1, no.1, pp.4-17, 1982.

[13] J. Kerr and B. Roth, Analysis of multifingered hands, *The International Journal of Robotics Research*, vol.4, no.4, pp.3-17, 1986.

[14] T. Watanabe and M. Beetz, Grasp motion planning for box opening task by multi-fingered hands, *Proc. of IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp.1-7, 2009.



(a)

(b)

(c)

FIGURE 15. The distance between current and goal positions of the first end effector: (a) without any obstacle, (b) with static obstacles and (c) with static and moving obstacles

[15] J. P. Saut, A. Sahbani and R. Perdereau, Generic motion planner for robot multi-fingered manipulation, *Advanced Robotics*, vol.25, no.1-2, pp.23-46, 2011.

[16] S. M. Song and K. J. Waldron, *Machines That Walk*, MIT Press, Cambridge, MA, 1989.

[17] J. A. Adams, R. Bajcsy, J. Kosecka, V. Kumar, R. Mandelbaum, M. Mintz, R. Paul, C. C. Wang, Y. Yamamoto and X. Yun, Cooperative material handling by human and robotic agents, *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.200-205, 1995.

[18] M. Abou-Samah, C. P. Tang, R. M. Bhatt and V. Krovi, A kinematically compatible framework for cooperative payload transport by nonholonomic mobile manipulators, *Autonomous Robot*, vol.21, pp.227-242, 2006.

[19] R. M. Bhatt, *Towards Modular Cooperation between Multiple Nonholonomic Mobile Manipulators*, Ph.D. Thesis, State University of New York at Buffalo, 2007.

[20] C. P. Tang, R. Bhatt and V. Krovi, Decentralized kinematic control of payload transport by a system of mobile manipulators, *Proc. of IEEE International Conference on Robotics and Automation*, pp.2462-2467, 2004.

[21] J. Desai, C. C. Wang, M. Zefran and V. Kumar, Motion planning for multiple mobile manipulators, *Proc. of IEEE International Conference on Robotics and Automation*, pp.2073-2078, 1996.

[22] J. P. Desai and V. Kumar, Nonholonomic motion planning for multiple mobile manipulators, *Proc. of IEEE International Conference on Robotics and Automation*, pp.3409-3414, 1997.

[23] J. P. Desai and V. Kumar, Motion planning for cooperating mobile manipulators, *Journal of Robotic Systems*, vol.16, no.10, pp.557-579, 1999.

[24] S. Furuno, M. Yamamoto and A. Mohri, Trajectory planning of cooperative multiple mobile manipulators, *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.136-141, 2003.

[25] J. Albaric and R. Zapata, Motion planning of cooperative nonholonomic mobile manipulators, *Proc. of IEEE International Conference on Systems, Man and Cybernetics*, 2002.

[26] Y. Yamamoto and S. Fukuda, Trajectory planning of multiple mobile manipulators with collision avoidance capability, *Proc. of IEEE International Conference on Robotics and Automation*, pp.3565-3570, 2002.

[27] H. G. Tanner, S. G. Loizou and K. J. Kyriakopoulos, Nonholonomic navigation and control of cooperating mobile manipulators, *IEEE Transactions on Robotics and Automation*, vol.19, no.1, pp.53-64, 2003.

[28] J. Cortes, *Motion Planning Algorithms for General Closed-Chain Mechanisms*, Ph.D. Thesis, Institut National Polytechnique de Toulouse, 2003.

[29] L. Han and N. Amato, A kinematics-based probabilistic roadmap method for closed chain systems, *Proc. of the 4th International Workshop on Algorithmic Foundations of Robotics*, pp.233-245, 2000.

[30] J. Cortes, T. Simeon and J. P. Laumond, A random loop generator for planning the motions of closed kinematics chains using PRM methods, *Proc. of IEEE International Conference on Robotics and Automation*, pp.2141-2146, 2002.

[31] J. Cortes and T. Simeon, Sampling-based motion planning under kinematic loop-closure constraints, *Proc. of the 6th International Workshop on Algorithmic Foundations of Robotics*, 2004.

[32] J. P. Laumond, *Robot Motion Planning and Control*, Springer, 1998.

[33] S. M. Lavalle, *Planning Algorithms*, Cambridge University Press, 2006.

[34] G. Hirano, M. Yamamoto and A. Mohri, Trajectory planning for cooperative multiple manipulators with passive joints, *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.2339-2344, 2000.

[35] Y. Liu and S. Arimoto, Minimum-time trajectory planning for multiple manipulators handling an object with geometric path constraints, *IEEE/RSJ International Workshop on Intelligent Robots and Systems*, pp.322-327, 1991.

[36] A. Mohri, M. Yamamoto and S. Marushima, Collision-free trajectory planning for two manipulators using virtual coordination space, *Proc. of IEEE International Conference on Robotics and Automation*, pp.674-679, 1993.

[37] A. Mohri, M. Yamamoto and G. Hirano, A trajectory planning algorithm with path search for cooperative multiple manipulators, *Proc. of IEEE International Conference on Systems, Man, and Cybernetics*, pp.898-903, 1995.

[38] A. Mohri, G. Hirano and M. Yamamoto, Collision-free trajectory planning for cooperative multiple manipulators, *Proc. of IEEE International Conference on Systems, Man, and Cybernetics*, pp.128-133, 1996.

[39] G. J. Garvin, M. Zefran, E. A. Henis and V. Kumar, Two-arm trajectory planning in a manipulation task, *Biological Cybernetics*, pp.53-62, 1997.

[40] T. Tsuji, A. Jazidie and M. Kaneko, Distributed trajectory generation for cooperative multi-arm robots via virtual force interactions, *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, vol.27, no.5, 1997.

[41] E. Todt, G. Rush and R. Suarez, Analysis and classification of multiple robot coordination methods, *Proc. of IEEE International Conference on Robotics and Automation*, pp.3158-3163, 2000.

[42] H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementation*, MIT Press, 2005.

[43] D. Zameroski, G. Starr, J. Wood and R. Lumia, Swing-free trajectory generation for dual cooperative manipulators using dynamic programming, *Proc. of IEEE International Conference on Robotics and Automation*, pp.1997-2003, 2006.

[44] G. Pajak, I. Pajak and M. Galicki, Trajectory planning of multiple manipulators, *Proc. of the 4th International Workshop on Robot Motion and Control*, pp.121-126, 2004.

[45] J. P. Vandenberg, *Path Planning in Dynamic Environments*, Ph.D. Thesis, Utrecht University, 2007.

[46] D. Ferguson, N. Kalra and A. Stenz, Replanning with RRT, *Proc. of IEEE International Conference on Robotics and Auromation*, 2006.

[47] J. P. VandenBerg, D. Ferguson and J. Kuffner, Anytime path planning and replanning in dynamic environments, *Proc. of the IEEE International Conference on Robotics and Automation*, pp.2366-2371, 2006.

[48] D. Ferguson and A. Stentz, Anytime, dynamic planning in high-dimensional search spaces, *Proc. of IEEE International Conference on Robotics and Automation*, pp.1310-1315, 2007.

[49] L. Jaillet and T. Simeon, A PRM-based motion planner for dynamically changing environments, *Proc. of IEEE International Conference on Intelligent Robots and Systems*, 2004.

[50] J. P. Vandenberg and M. H. Overmars, Roadmap-based motion planning in dynamic environments, *IEEE Transactions on Robotics and Automation*, pp.885-897, 2005.

[51] H. G. Tanner and K. J. Kyriakopoulos, Mobile manipulator modeling with Kane's approach, *Robotica*, vol.19, pp.675-690, 2001.

[52] H. G. Tanner, K. J. Kyriakopoulos and N. I. Krikelis, Modeling of multiple mobile manipulators handling a common deformable object, *Journal of Robotic Systems*, vol.15, no.11, pp.599-623, 1998.

[53] S. M. LaValle, J. H. Yakey and L. E. Kavraki, A probabilistic roadmap approach for systems with closed kinematic chains, *Proc. of IEEE International Conference on Robotics and Automation*, 1999.

[54] J. H. Yakey, S. M. LaValle and L. E. Kavraki, Randomized path planning for linkages with closed kinematic chains, *IEEE Transactions on Robotics and Automation*, vol.17, no.6, pp.951-958, 2001.

[55] G. Bergen, *Collision Detection in Interactive 3D Environments*, Elsevier, 2004.

[56] F. Schwarzer, M. Saha and J. C. Latombe, Adaptive dynamic collision checking for single and multiple articulated robots in complex environments, *IEEE Transactions on Robotics*, vol.21, no.3, pp.338-353, 2005.

[57] N. M. Amato, O. B. Bayazit, L. K. Dale, C. Jones and D. Vallejo, Choosing good distance metrics and local planners for probabilistic roadmap methods, *Proc. of IEEE International Conference on Robotics and Automation*, pp.630-637, 1998.

Appendix A. Proof of Equation (1).

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

$$ {}^{v_1}_{p_1}T a_1 {}^{p_1}T^{a_1}_{1}T\,{}^{1}_{2}T\,{}^{2}_{3}T\,{}^{3}_{a_2}T\,{}^{a_2}_{p_2}T\,{}^{p_2}_{e_2}T = {}^{v_1}_{v_2}T\,{}^{v_2}_{p_4}T\,{}^{p_4}_{a_4}T\,{}^{a_4}_{6}T\,{}^{6}_{5}T\,{}^{5}_{4}T\,{}^{4}_{a_3}T\,{}^{a_3}_{p_3}T\,{}^{p_3}_{e_2}T \tag{A.1} $$

in which ${}^{A}_{B}T$ is a matrix that transforms coordinate system $\{B\}$ into $\{A\}$ and for the different coordinate systems in the above equation is defined as follows:

$$ {}^{v_1}_{p_1}T = \begin{bmatrix} \cos(\theta_{p_1}) & -\sin(\theta_{p_1}) & 0 & 0 \\ \sin(\theta_{p_1}) & \cos(\theta_{p_1}) & 0 & 0 \\ 0 & 0 & 1 & l_{p_{11}} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{p_1}_{a_1}T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & l_{p_{12}} \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{a_1}_{1}T = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{1}_{2}T = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & l_1 \\ \sin(\theta_2) & \cos(\theta_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{2}_{3}T = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & l_2 \\ \sin(\theta_3) & \cos(\theta_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{3}_{a_2}T = \begin{bmatrix} 0 & 0 & 1 & l_{p_{21}} \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{a_2}_{p_2}T = \begin{bmatrix} \cos(\theta_{p_2}) & -\sin(\theta_{p_2}) & 0 & 0 \\ \sin(\theta_{p_2}) & \cos(\theta_{p_2}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{p_2}_{e_2}T = \begin{bmatrix} -1 & 0 & 0 & l_{obj} \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & l_{p_{22}} \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

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

$$ {}^{v_2}_{p_4}T = \begin{bmatrix} \cos(\theta_{p_4}) & -\sin(\theta_{p_4}) & 0 & 0 \\ \sin(\theta_{p_4}) & \cos(\theta_{p_4}) & 0 & 0 \\ 0 & 0 & 1 & l_{p_{41}} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{p_4}_{a_4}T = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & l_{p_{42}} \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{a_4}_{6}T = \begin{bmatrix} \cos(\theta_6) & -\sin(\theta_6) & 0 & 0 \\ \sin(\theta_6) & \cos(\theta_6) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{6}_{5}T = \begin{bmatrix} \cos(\theta_5) & -\sin(\theta_5) & 0 & l_6 \\ \sin(\theta_5) & \cos(\theta_5) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{5}_{4}T = \begin{bmatrix} \cos(\theta_4) & -\sin(\theta_4) & 0 & l_5 \\ \sin(\theta_4) & \cos(\theta_4) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{4}_{a_3}T = \begin{bmatrix} 0 & 0 & 1 & l_{p_{31}} \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{a_3}_{p_3}T = \begin{bmatrix} \cos(\theta_{p_3}) & -\sin(\theta_{p_3}) & 0 & 0 \\ \sin(\theta_{p_3}) & \cos(\theta_{p_3}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{p_3}_{e_2}T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & l_{p_{32}} \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

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

Appendix B. Proof of Equation (16).

The reachable workspace of any articulated mechanism can be bound by two concentric spheres centered at its base as shown in Figure 5. The radii of these spheres correspond to the length of the mechanism in case of minimum and maximum extensions. This length is defined by the distance between the origins of the base-frame and the end-frame and can be expressed as:

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

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:

$$r_{\max} = \sum_{i=1}^{k} L_i \tag{B.2}$$

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

$$r = |L_1 a_1 + L_2 a_2 + L_3 a_3 + \cdots + L_{\max} a_j + \cdots + L_k a_k| \tag{B.3}$$

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

$$r_{\min} = \begin{cases} -L_1 - L_2 - \cdots - L_{j-1} + L_{\max} - L_{j+1} \cdots - L_k \\ \qquad\qquad \text{if} \quad L_{\max} > L_1 + L_2 + \cdots + L_{j-1} + L_{j+1} \cdots + L_k \\ 0 \\ \qquad\qquad \text{if} \quad L_{\max} \le L_1 + L_2 + \cdots + L_{j-1} + L_{j+1} \cdots + L_k \end{cases} \tag{B.4}$$

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

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

which shows the validity of Equation (16).