Robotica (2010) volume 28, pp. 477–491. © Cambridge University Press 2009
doi:10.1017/S0263574709005773

Offline decoupled path planning approach for effective coordination of multiple robots

Shital S. Chiddarwar and N. Ramesh Babu*

Manufacturing Engineering Section, Department of Mechanical Engineering, Indian Institute of Technology Madras, Chennai 600036, Tamilnadu, India

(Received in Final Form: April 21, 2009. First published online: May 26, 2009)

SUMMARY
In this paper, a decoupled offline path planning approach for determining the collision-free path of end effectors of multiple robots involved in coordinated manipulation is proposed. The proposed approach for decoupled path planning is a two-phase approach in which the path for coordinated manipulation is generated with a coupled interaction between collision checking and path planning techniques. Collision checking is done by modelling the links and environment of robot using swept sphere volume technique and utilizing minimum distance heuristic for interference check. While determining the path of the end effector of robots involved in coordinated manipulation, the obstacles present in the workspace are considered as static obstacles and the links of the robots are viewed as dynamic obstacles by the other robot. Coordination is done in offline mode by implementing replanning strategy which adopts incremental A* algorithm for searching the collision-free path. The effectiveness of proposed decoupled approach is demonstrated by considering two examples having multiple six degrees of freedom robots operating in 3D work cell environment with certain static obstacles.

KEYWORDS: Collision checking; Decoupled path planning; Heuristic search; Coordinated robots.

1. Introduction

In the field of robotics, motion planning of multiple robots is one of the challenging issues since it deals with the generation of collision-free, coordinated paths for multiple robots operating in geometrically complex environments. Motion planning refers to both path and trajectory planning. In path planning, the sequence of movements for a robot in a given workspace is generated. However, in trajectory planning, the time history of position, velocity and acceleration are derived. As the trajectory of a robot is derived from the collision-free path of robots, the task of collision-free path planning assumes considerable significance. Path planning of multiple robots by manual methods is quite tedious and time consuming since the task becomes increasingly complex with increasing number of robots. Thus, an automated path planning of multiple robots is becoming more and more important.[1,2]

Established approaches for coordinated motion planning of multiple robots are centralised and decoupled approaches. In centralised approach, multiple robots operating in a workspace are treated as a single multi-bodied robot operating in a composite configuration space, i.e., the set of product of the configuration spaces of individual robots. Centralised planning is possible only if the robot knows the global state of the system or the goal positions of all other robots. Though this approach is complete in determining the path of robots, it is computationally demanding and applicable to robots, operating in less dense environments with less number of degrees of freedom. In practical scenario, the centralised planning approach has been beyond the capabilities of existing planning techniques, as it requires search in the configuration space (Cspace) with many dimensions.

Practical multi-robot path planning problems are addressed by a two-phase decoupled approach. In the first phase, a collision-free path for each robot is computed by considering the static obstacles and ignoring the presence of other robots in the environment. In the second phase, the coordinated path of each robot is computed with respect to other robots in the environment. Though the decoupled path planning is inherently incomplete and is not guaranteed to give a solution always, the loss of completeness is relatively small and worth computational gain. This approach may be reliable for applications where interactions among the robots are less constraining. In cases where the interaction among various robots severely constrains their movement in the environment, the decoupled planning becomes ineffective and unreliable. However, it can still be chosen if the path planner receives interactive hints from the user.[3] In view of the unique benefits of decoupled methods for path planning of multiple robots, it is still an active area of research.

The remaining part of the paper is organised as follows: the next section gives a brief overview of work in multi-robot motion planning, robot modelling and collision-detection approaches and various search algorithms used for searching the collision-free path of robots. Various issues considered in planning approach for coordinated path planning of multiple robots are detailed with their demonstration in the subsequent sections. Finally, the results of the study and the conclusions drawn from this study are covered.

* Corresponding author. Email: nrbabu@iitm.ac.in

2. Related Work

2.1. Motion planning of multiple robots

In centralised approach, the path planning of multiple robots is performed by considering all the robots as one single composite system operating in a composite configuration space.[4] Both geometric path and velocity profiles are determined by considering the coordination of multiple robots. Thus, the generation of path and the coordination of robots are inseparable processes. An attempt was made to generate the collision-free path for two or three disc-shaped robots operating in low-density workspace with polygonal obstacles.[2] No doubt, this approach can generate the path within a short time but could not be easily extended to generate the path for multiple six degrees of freedom robots. This is due to increased time complexity with an increase in the dimensionality of composite configuration space. The issue of path planning of multiple degrees of freedom robots in complex search space was addressed by certain heuristic methods.[3,5] Among the various heuristic methods, potential field and roadmap methods are the most widely employed. In case of potential field approach, the problem of local minima is addressed by Monte-Carlo technique[5] and Cspace–time approach with penalty function.[6] Roadmap methods restrict the path of robots to lie on independent roadmaps determined with the help of certain heuristics and achieve the coordination by searching Cartesian product of separate roadmaps using probabilistic roadmap approach.[3] Grouping of robots[7] was proposed to essentially reduce the dimension of search space. In this approach, hierarchical sphere tree structure was used to group the robots and the path was searched by using potential field based planner. Centralised path planning is a complete one for systematic search of composite configuration space and is feasible when the size of search space is small. However, it becomes impractical in case of multiple robots path planning where the size of composite Cspace is overwhelming due to the presence of more degrees of freedom robots. This has led to the development of decoupled approaches for multiple robot path planning, where the completeness is sacrificed in favour of complexity.[8,9]

Decoupled methods present a coordination phase separated from the path planning phase. Coordination among the robots can be achieved by adjusting the geometric paths,[10] introducing time delay[8] or modifying the velocity profiles.[11] The adjustment in the geometric path was done by identifying the regions of the space swept by the robots and then modifying the path planned *a priori* so that the robots do not occupy these regions simultaneously.[10] If it is not possible to modify the path of robot, then the sequence of tasks will have to be modified so that the regions of conflict are occupied by one robot at a time. The problems existing with this method are overcome by delaying the motion of robot by a pre-computed value at the beginning of the execution of the movements of one robot so that it does not collide with other robots.[8] Another way of coordinating the motion of robot was achieved by modifying or tuning the velocity profile without changing the geometric path. By velocity tuning, i.e., by assigning different velocities to various links of robots, the collision among the robots, along the paths computed in the path planning phase, was resolved.[11] A variant of the basic decoupled approach is prioritised path planning in which the motion of each robot is determined consecutively in the order given by prioritisation scheme. Various approaches were devised for determination of prioritisation scheme. These schemes made use of various techniques for priority determination and assignment. The most straightforward way of priority assignment was to fix the priority of robot movement based on the size of robot. Highest priority was given to a large robot while planning its collision-free path with the stationary obstacles in the workspace. Then, the path for the next lower priority robot was planned by velocity tuning so as to avoid both the stationary obstacles and the higher priority robot, treated as a moving obstacle.[1] This approach is mainly applicable for heterogeneous robots and cannot be used for applications involving the same type of robots. To avoid this problem, a simple heuristic was proposed for priority assignment.[12] In order to compute the priority of robot, the shortest distance between the start and goal configurations of each robot was used as a heuristic. In this case, the path planning was carried out in such a way that the robots that traverse long distances were allowed to move with higher priority, while the robots that had to traverse short distances were stopped to avoid interaction with robots moving with higher priority. This approach is suited for situations where intermediate stoppage of robots is permitted but cannot be employed to the applications demanding for continuous operation of robot. Predefined prioritisation scheme[13] was developed for different applications. In this case, the operation sequence was utilized to define the priority of the robot. All these approaches are found to be suitable for certain specific situations. Moreover, the success of these methods mainly depends upon the assignment of priority to the robot.

The coordination of robots can be carried out before the task execution (offline) or during the movement of robots (online).[14] Online coordinated task planning of multiple robots is not effective since it essentially demands for stopping of robots during the path planning. Moreover, the computational requirements are quite extensive. Hence, this approach is mostly suited for situations where the tasks are simple and can be taught to robot by trial and error. In typical manufacturing processes like spray painting, spot welding or precision assembly, it is essential to generate the path in advance by offline methods so as to avoid the interruption of robots performing the tasks continuously. In this context, decoupled path planning approaches are the most appropriate ones for multi-robot motion planning. Though this approach is an incomplete approach,[4] it is faster than the centralised approach in view of the search space explored by decoupled planner has lower dimensionality as compared to composite configuration space searched by centralised planner. Though the prioritised path planning seems to be a prudent variation of decoupled path planning approach, it has an inherent drawback of deadlock situation,[4] which arises when higher priority robot blocks the path of lower priority robot. Under such circumstances, the planner fails to give the feasible path and do not offer any scope for backtracking. Ultimately, prioritised path planner has to quit and alternate planner needs to be invoked for further path planning.

Thus, the above points clearly bring out the need to develop an effective decoupled path planning approach for offline coordinated path planning of multiple robots so that the need for any prioritisation scheme and online modification of velocity profiles can be completely avoided.

2.2. Robot modelling and collision detection

Basic path planning approach demands for mapping of robots and obstacles in the configuration space using suitable modelling technique and determination of collision-free configuration space using collision-detection algorithm. Collision-detection algorithm is often based on bounding volume used for modelling of the robot and its environment. Oriented Bounding Box (OBB) tree, ellipsoid, octree, sphere tree, Axis Aligned Bounding Boxes (AABB) and Swept Sphere Volumes (SSVs) are few bounding volumes used in practice. OBB – tree is the hierarchical representation of an object modelled using oriented bounding boxes. OBB is the smallest possible bounding box of arbitrary orientation that can enclose the required geometry.[15] This method is very good for fast rejection tests but has certain disadvantages like slow update rate and orientation sensitivity. Many algorithms are available for fitting minimum volume ellipsoid to robot links and obstacles. In this case, the collision detection is achieved by checking interference between two ellipsoids.[16] This technique is computationally complex and time consuming. The octree[17] representation involves recursively sub-dividing the volume containing an object into eight octants. Such data structure is simple to produce but has the disadvantage that each level of the hierarchy does not fit the underlying geometry very tightly. Spheres are rotationally invariant and hence can be updated very fast.[18,19] However, the sphere does not approximate certain types of objects very efficiently. Axis Aligned Bounding Boxes[20] (AABBs) is one more bounding volume method used in common practice. It is very simple to model the object but cannot be efficiently used for longitudinal objects. Hence, these methods have not found widespread use in robot link modelling and collision checking since they employ expensive interaction tests and are unable to give tighter fit to an underlying geometry. Swept Sphere Volume (SSV) is a sphere that is swept out along a geometric primitive, such as a point, line or a rectangle. These volumes are found to be the most effective and accurate for modelling of the links of a robot and the environment.[21,22] SSV gives inherently better fit due to the geometry of bounding volumes and hence leads to more accurate collision tests, especially if there is no penetration check required among the geometries. Thus, the above discussion clearly indicates that the SSV is most effective method for modelling of links of the robot and its environment. Among the various methods of collision checking, the overlapped geometry and minimum distance heuristics are the most promising one.

2.3. Search algorithms

Path planning essentially deals with the search for an optimal path in the collision-free configuration space of robot determined by collision-detection scheme. Among the various approaches used for searching the path, Probabilistic roadmap (PRM) approach and grid search techniques are widely used. The PRM approach samples the configuration space for collision-free placements. These are added as nodes to a roadmap graph. Pairs of promising nodes in the graph are chosen and a simple local motion planner is used in order to connect such placements to form a path. If this connection is feasible, an edge is added between the nodes in the graph. This process continues until the graph represents the connectedness of the space. On the other hand, grid search involves the search for collision-free configuration space using certain search algorithms. Grid-based search is considered to be the most straightforward form of path planning as compared to PRM approach. The roadmap edges of PRM path planner create problems when the environment is dynamically changing. In such situations, even modified form of PRM approaches in their lighter mode, i.e., lazy PRM may be unsuccessful to address the path planning problem. The reason behind this can be the abolition of roadmap edges due to the movement of obstacle that may demolish the graph structure connectivity and then leads to search failure.[23] A recent study[24] compared centralised and decoupled approaches by implementing their basic adaptation in PRM framework. The study concluded that PRM gives unreliable results when implemented for decoupled path planning.

On the other side, the path planning with grid search techniques poses the problem of exponential growth of application complexity with increasing dimensions of configuration space. The recent attempts made for the effective use of grid search methods for path planning in continuously varying search space involve the speeding up of search and improving the heuristics.[24–29] For speeding up the search, differential form of A* algorithm was developed. It was based on the assumption that the new search space generated due to variation in the environment will differ only slightly from the old. Instead of performing the full A* search on the new search space, the necessary nodes were computed to obtain the revised solution by A* algorithm.[25] An incremental version of A* algorithm (LPA: Life long planning algorithm) which reuses the path information from the previous searches was developed in this work. It saves the information obtained from the previous searches and utilizes it for further path search. This characteristic of incremental A* algorithm considerably reduces the number of grid nodes to be recomputed and hence ultimately speeds up the search.[26–28] Effectiveness of grid search for high-dimensional search spaces can be improved by optimising the search and modifying the approach used for building search space. Optimisation of search can be done by reducing the number of neighbouring nodes to be searched. A heuristic method was developed for achieving the reduction in neighbouring nodes.[29] The ease in building search space is dependent on the dimensions of search space. High-dimension search spaces are complex and time consuming. Hence, the concept of deterministic and incremental building of search spaces was developed to facilitate the generation of high-dimensional search spaces.[30]

When the nature of obstacles in the environment is dynamic, dynamic A*, i.e., D* algorithm[31] is one of the common choices for path planning. This grid search algorithm is presented in various forms like D* lite,[32]

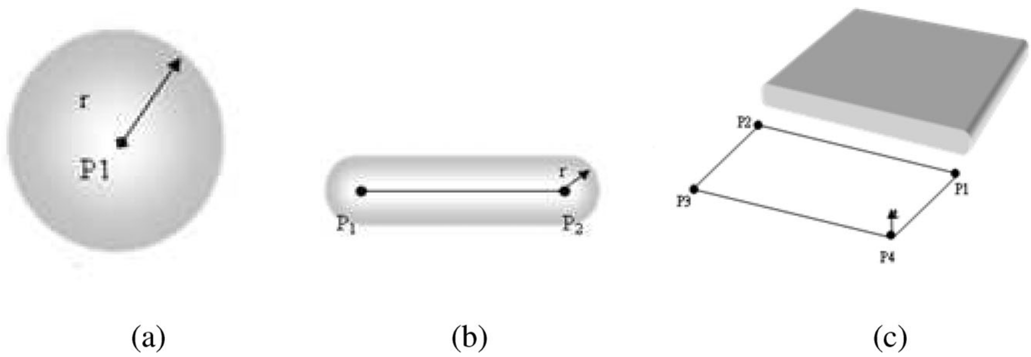


Fig. 1. Geometry of solid obtained by Swept Sphere Volume for (a) point, (b) line and (c) rectangle.

Focused D*[33], etc. The tradeoff between incremental A* and dynamic A* lies on the source of information. While planning the path for a robot, D* algorithm assumes initial conditions and carries out online modification of path based on information given by the sensors whereas the incremental A* algorithm finds out the area of search space which is changed due to movement of dynamic obstacles and plans the path accordingly. This explorative nature of incremental A* algorithm makes it suitable for offline path planning of robots. Incremental A* algorithm shows the ability to improve the solution quality after every replanning episode for any number of replanning episodes whereas the quality of search carried out by D* algorithm degrades with an increase in the number of replanning episodes. Moreover, in many situations, D* is slower than conventional A* algorithm.[28] Hence, incremental A* algorithm may become relevant for decoupled path planning of multiple robots in offline mode.

In view of the above, the present work proposes an offline decoupled approach for effective coordinated path planning of multiple robots. The approach is made effective by accurate modelling of robot links and its environment with SSV method, and realising the effective coordination of robots with replanning strategy utilising incremental search technique. Though the motion planning of multiple robots will be complete with the derivation of time history along the geometric path, the scope of the present work is limited to path planning of multiple robots.

3. Issues Considered in Problem Formulation

In planning the path for multiple robots, the following main issues have to be considered:

- Geometric modelling of robot and its environment
- Determination of collision-free Cartesian Cspace
- Coordinated path determination with proposed decoupled approach

3.1. Geometric modelling of robot and its environment with Swept Sphere Volume (SSV) technique

As discussed earlier, SSV technique is the most effective method for modelling of objects. SSV is a sphere that is swept out along the geometric primitives such as point, line and rectangle (Fig. 1). It results in a sphere around a point, cylinder with hemispherical ends around a line and a block with rounded edges and corners around a rectangle. These volumes provide means for varying the shape of bounding primitive to achieve a tighter fit to underlying geometry. In the present work, the link of robot is modelled as a cylinder with hemispherical ends. The complex-shaped obstacles are enclosed in spherical volume and obstacles with rectangular cross-section are modelled as platform with rounded edges and corners.

3.2. Determination of collision-free Cartesian Cspace

Collision-free Cartesian Cspace of robots can be obtained by removing the configurations that cause interference of robots among themselves and with the obstacles. These configurations can be determined by collision checking procedure. Collision checking involves the modelling of links of robot and obstacles in the environment using SSV, and checking the interference between links of robot and static or dynamic obstacles by using the minimum distance heuristic. In this work, the algorithms used to determine the minimum safe distance between various geometric entities are based on the concepts of computational geometry. These algorithms are based on the fact that the SSV representation of links of robot and the obstacles reduces the complex objects into simplest form of point, line and rectangle.[22] The line representation can be simplified to two points and the rectangle to four lines. The algorithm used to determine the minimum distance between the sphere and the cylinder with hemispherical ends is presented in Fig. 2. From this, it can be seen that the link of robot modelled as a cylinder with hemispherical ends can be represented by a straight line and the obstacle modelled as a sphere can be represented by a point. Thus, the task is reduced to the determination of the minimum distance between these two primitives. The equation of line in parametric form can be given as $(x, y, z) = (a, b, c) + t\,(u, v, w)$ where (a, b, c) is the point through which the line passes and (u, v, w) is the vector parallel to this line. In order to determine the shortest distance between the sphere represented as a point P_3 and the cylinder with hemispherical ends represented as a line, Line 1, the point P_4 on Line 1 joining the points P_1 and P_2 is to be found out in such a way that the Line 2 joining P_3 and P_4 will be perpendicular to Line 1. Since Line 1 and Line 2 are perpendicular, the dot product of their slopes will be zero. Using this fact, the parameter '$t1$' is determined. Solving for '$t1$' will give the co-ordinates of point P_4 on Line 1 but it also lies on Line 2. Hence, another parameter 't' is considered. The coordinates

https://doi.org/10.1017/S0263574709005773 Published online by Cambridge University Press

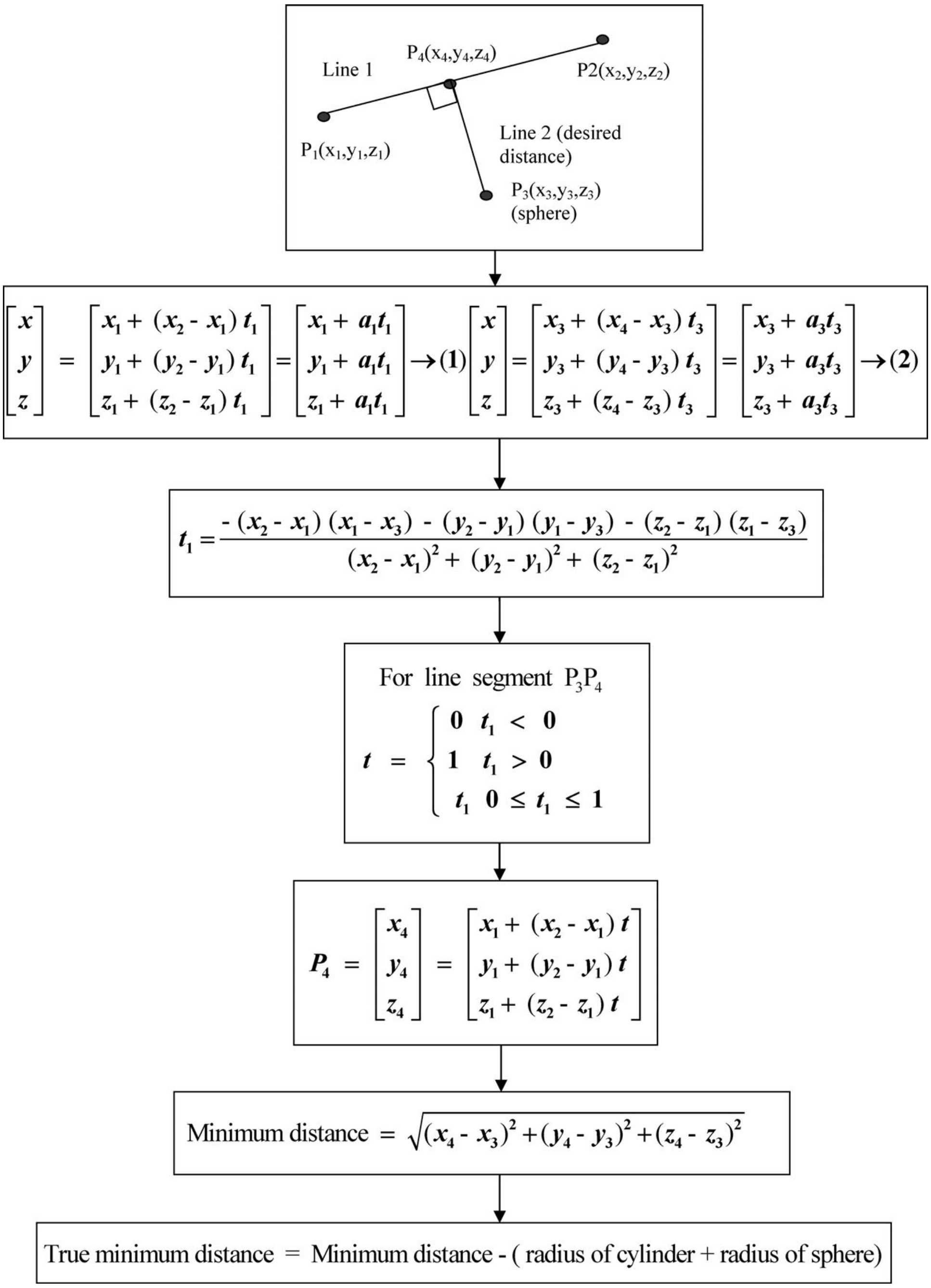


Fig. 2. Algorithm to determine the minimum distance between sphere and cylinder with hemispherical ends.

of point P_4 are determined by using this parameter 't'. Then the shortest distance between the sphere and the cylinder with hemispherical ends is just the distance between point P_3 and point P_4 and can be easily determined by using the distance formula. The true minimum distance is obtained by deducting the radius of link modelled as a cylinder with hemispherical ends and obstacle modelled as a sphere from the computed minimum distance. Figure 3 represents the algorithm used to determine the minimum distance between the two cylinders with hemispherical ends. It works on the same concept as explained in Fig. 2. If the line segments are intersecting, i.e., the condition $t_1 = t_{11}$ and $t_3 = t_{33}$ is not satisfied, then the algorithm presented in Fig. 3 will not give the correct minimum distance. In such a situation, the algorithm for determination of the minimum distance between the sphere and the cylinder with hemispherical ends is to be used (Fig. 2). This will lead to the computation of four minimum distances. The minimum of these four computed distances will be the true minimum distance between cylinders with hemispherical ends. Figure 4 provides the algorithm used to find the minimum distance between a cylinder with hemispherical ends and a rectangular block with rounded edges. In this case, the link is represented as a line and the platform as a rectangle. The two projection points P_7 and P_8 are determined as shown in Fig. 4. The algorithm presented in Fig. 4 is valid if both points P_1 and P_2 are on the same side of the platform and the projection point P_8 is on the rectangle. Otherwise, the edges of rectangular block are to be considered as cylinders with hemispherical ends and the algorithm presented in Fig. 3 is to be used for determining

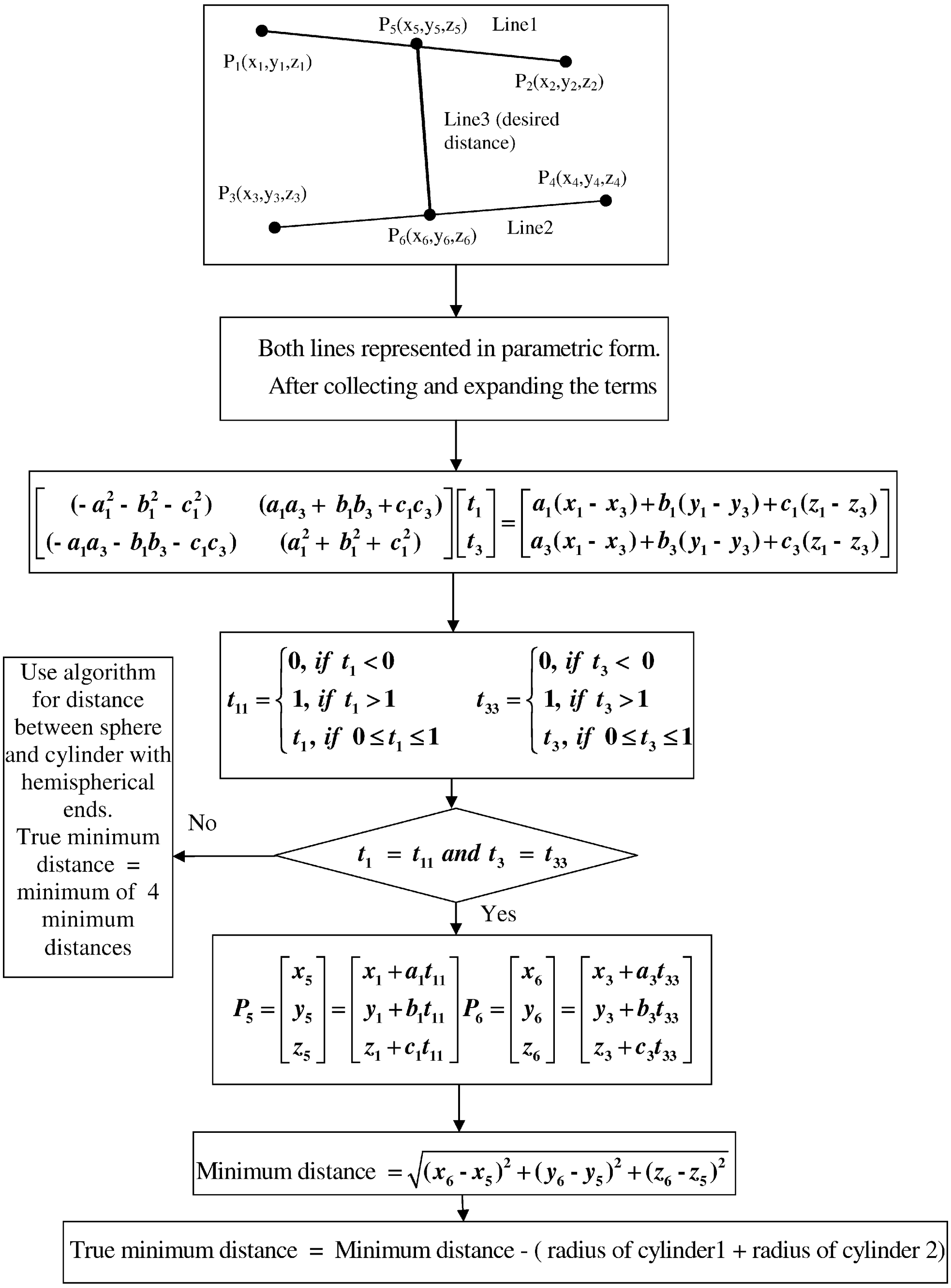


Fig. 3. Algorithm to determine the minimum distance between two cylinders with hemispherical ends.

the desired minimum distance. If this minimum distance is less than the specified value, then interference occurs among the primitives under study. This particular configuration of robot is considered as collision configuration. In this way, the collision checking for different configurations of robot can be performed.

By excluding the collision configurations from the Cspace, the collision-free Cartesian Cspace of robot can be obtained. These configurations are subjected to a singularity check by computing the determinant of Jacobian for each of these configurations. Singular configurations are those configurations at which robot transitorily lose one or more degrees of freedom. If the value of determinant of Jacobian is zero for any configuration, then that particular configuration is considered as singular and is removed from the Cspace. The remaining configurations are non-singular collision-free configurations, which form the collision-free Cartesian Cspace and are used for further path planning.

3.3. Coordinated path determination with proposed decoupled approach

3.3.1. Problem statement. Multi-robot path planning problem can be stated as follows. Let the workspace Ws, consisting of a set of stationary obstacles Ob, be shared by a set of n articulated robots $R_1, R_2, \ldots, R_n$. The geometries of the robots and obstacles are known and the robots are holonomic. Let q_i be a configuration of robot R_n in some parameterization describing the position of every point of the robot. The configuration space C_n is the set of all possible configurations q_i for the robot R_n. The number of degrees

https://doi.org/10.1017/S0263574709005773 Published online by Cambridge University Press

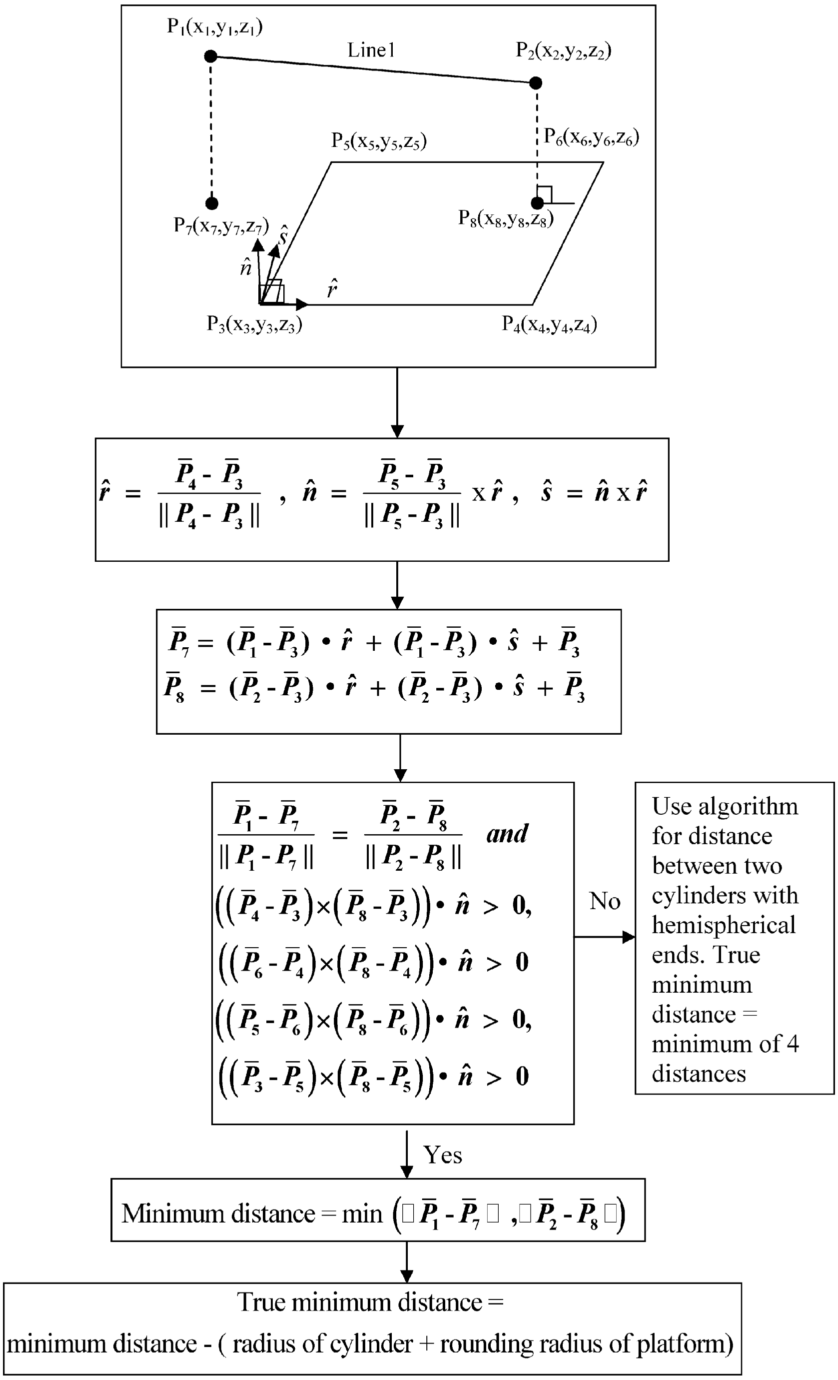


Fig. 4. Algorithm to determine the minimum distance between cylinder with hemispherical end and platform with rounded edges.

of freedom of robot R_n is the dimensionality d_n of C_n. The portion of configuration space in which the robot is collision free with respect to obstacles and other robots is the collision-free configuration space F. The multi-robot path planning problem can be represented as given a start configuration $s \subset F$ and goal configuration $g \subset F$, compute a path $p \subset F$ from s to g.

3.3.2. Proposed decoupled path planning approach. The proposed decoupled offline path planning approach with a two-phase strategy for generation of collision-free path is presented in Fig. 5. The abbreviations used in this figure are defined as j: number of replanning episodes, R_n: robot number, where n: 1, 2, 3, ..., n. R_nP_1: collision-free path of robot end effector with respect to static obstacles for robot R_n (path obtained in the first phase). $(R_nP_2)_j$: collision-free path of robot end effector with respect to static as well as dynamic obstacles for robot R_n after jth episode of replanning (path obtained in the second phase). In the first phase, the collision checking module determines the collision-free Cartesian Cspace for each robot with respect to the static obstacles. For each robot, the collision-free path with respect to the obstacles in the environment excluding the other robots is searched by using A* algorithm. In the second phase, the proposed approach considers the path planned in first phase and checks the feasibility of these paths for coordination. If

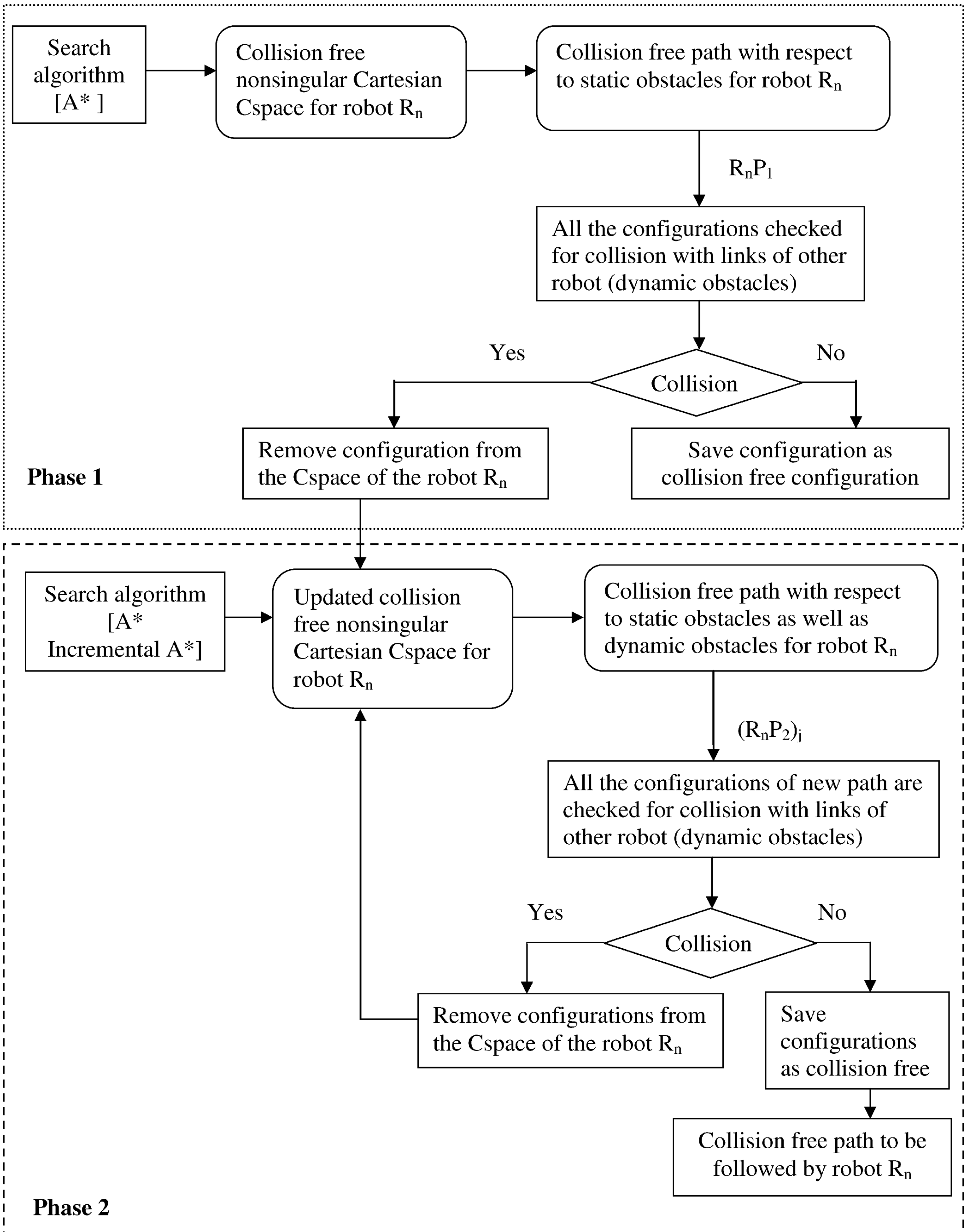


Fig. 5. Proposed decoupled path planning approach.

these paths show collision then those collision configurations are removed from the Cspace of respective robots. In this way, Cartesian Cspace of each robot is updated and an iterative search is performed. This iterative search is done by adopting the concept of replanning the path using incremental A* algorithm. This becomes the first episode of replanning. These episodes are repeated until a coordinated collision-free path of each robot is determined. Initially, incremental A* algorithm constructs a path identical to the path obtained by A* algorithm and stores it in the memory. In the next planning episode, incremental A* algorithm determines the region of the Cspace that is modified by removing the collision configurations. It reconstructs the path only in that region and keeps the path in unchanged search space intact. By this, the amount of computation needed to compute the new path can be reduced since the future paths refer to initial stored path. As incremental A* search technique reuses the information from previous searches to find solutions to a series of similar search problems, it is potentially faster than the one that solves each search problem from scratch.[27] This characteristics of incremental A* algorithm facilitate it to re-plan the path using less replanning episodes and by searching very few nodes during every replanning episode. In order to demonstrate the effectiveness of the incremental A* algorithm over one time search algorithms, replanning is also done by using A* algorithm in the present work.

3.3.3. Demonstration of proposed decoupled path planning approach. To illustrate the implementation of the proposed decoupled path planning approach a simple example of

https://doi.org/10.1017/S0263574709005773 Published online by Cambridge University Press

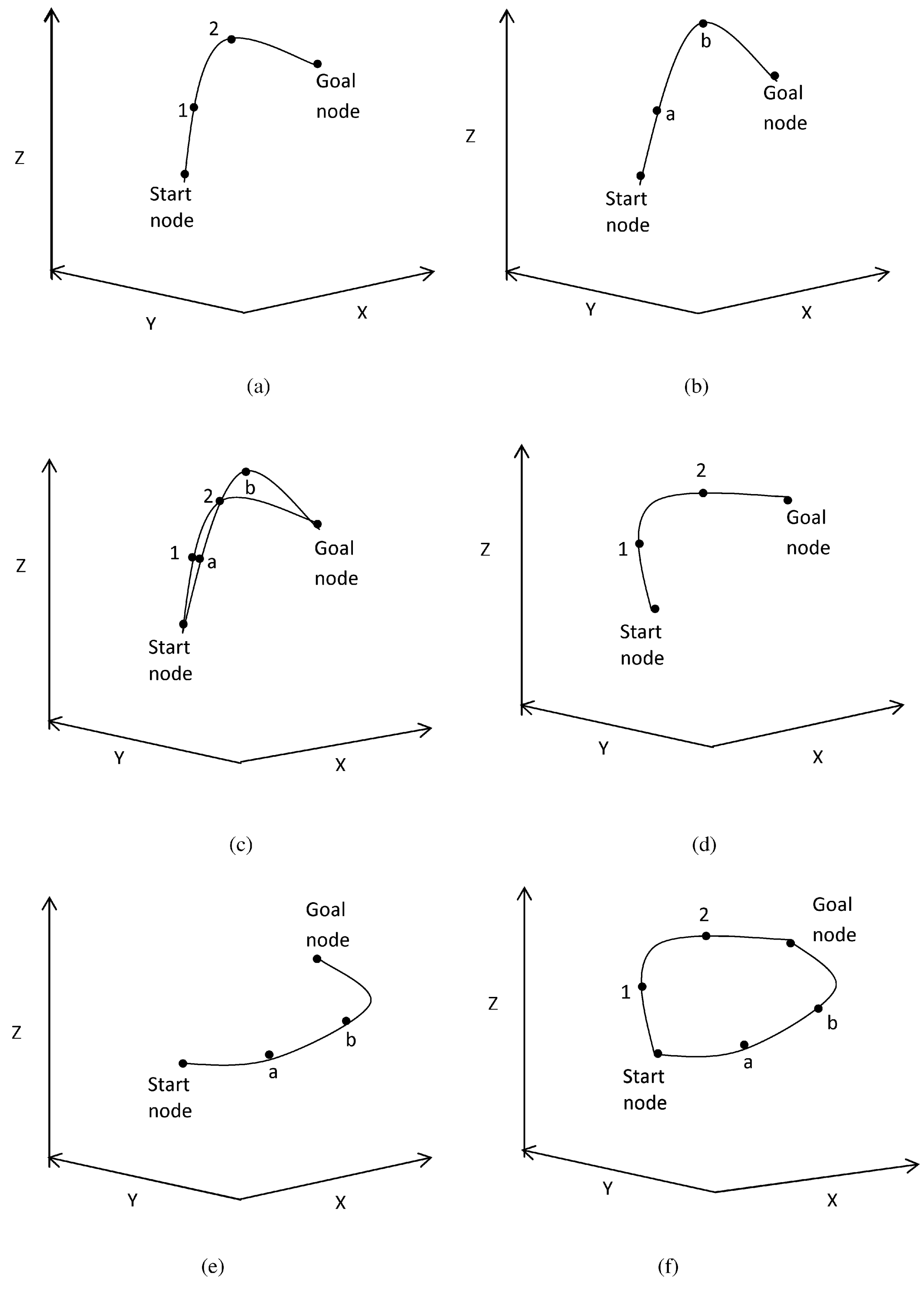


Fig. 6. Demonstration of the proposed approach (a) search Cspace for robot 1, (b) search Cspace for robot 2, (c) coordination space for both robots, (d) search Cspace for robot 1, (e) search Cspace for robot 2 and (f) coordination space for both robots.

two robots operating in workspace with static obstacles is considered (Fig. 6). In the first phase, the collision-free Cartesian Cspace of robot with respect to static obstacles is obtained by means of collision checking module. This Cspace is searched for an optimal collision-free path (R_nP_1) using A* algorithm. The path for end effector of both robots are independently planned using A* algorithm and this path is collision free with respect to static obstacles and is shown in Fig. 6(a) for robot 1 and in Fig. 6(b) for robot 2. In the second phase, the coordination among the robots is determined by replanning the paths using incremental search algorithm in such a way that the robots avoid collisions with each other while moving along their respective paths. For this purpose, the path planned in the first phase for the end effectors of both robots, i.e., Fig. 6(a) and 6(b) is placed in the coordination space as shown in Fig. 6(c). From the figure, it can be seen that there will be collision between robots if they continue to move on the same paths. The configurations of both the robots, which show collision, are determined by using collision checking module explained in the previous section.

https://doi.org/10.1017/S0263574709005773 Published online by Cambridge University Press

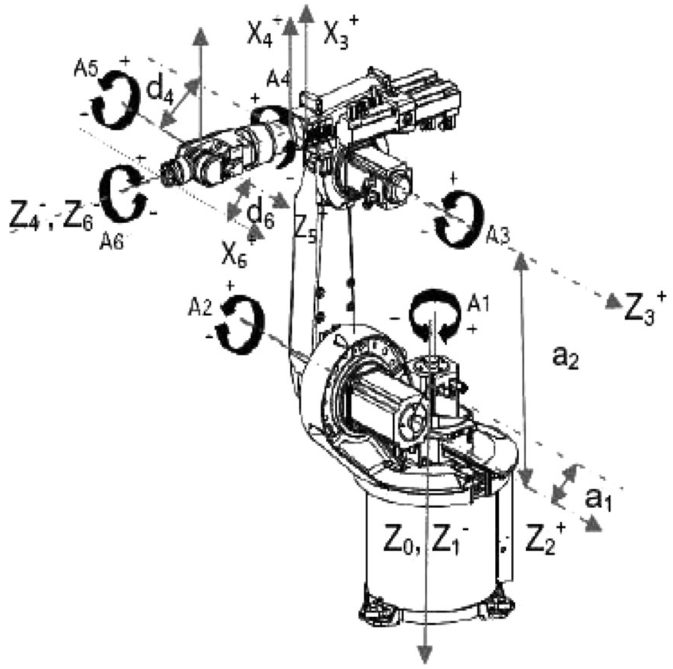


Fig. 7. Six degree of freedom robot with link frames.

These collision configurations are removed from the Cspace of respective robots. The updated Cspace is used to replan the path. As shown in Fig. 6, this process is continued until the path with all collision-free configurations is found out. The collision-free paths for end effectors of both robots are shown in Fig. 6(d) for robot 1 and Fig. 6(e) for robot 2. When both the paths are put in the coordination space, they are found to be collision free (Fig. 6(f)). Hence, the obtained paths are collision free with respect to static as well as dynamic obstacles. The proposed decoupled approach can be easily extended to a system with any number of robots.

4. Results and Discussion

The effectiveness of the proposed approach for decoupled offline path planning of multiple robots is demonstrated with two different examples. For this purpose, the robot shown in Fig. 7 is used. It is a six-axis industrial robot with jointed arm kinematics for all point-to-point and continuous control tasks. The Denavit Hartenberg (DH) parameters of the robot are given in Table I where a_i, α_i, θ_i, d_i are link length, link twist angle, joint angle and joint offset variables, respectively. In order to generate the complete configuration space for the robot, joint angles of the robot, i.e., θ_1 to θ_6 are varied by a unit degree within the range of angles given to each link by kinematic constraints of the robot. DH transformation matrix was used to convert the joint angle configurations into Cartesian coordinates.[4]

Table I. Denavit Hartenberg parameters of the robot.

Link	θi	α_{i-1}	a_{i-1} (mm)	d_i (mm)	Joint angle range
1	θ_1	π	0	0	$\pm 185°$
2	θ_2	$\pi/2$	300	0	$+115°$ to $-55°$
3	θ_3	0	650	0	$+70°$ to $-210°$
4	θ_4	$\pi/2$	155	-600	$\pm 350°$
5	θ_5	$-\pi/2$	0	0	$\pm 135°$
6	θ_6	$\pi/2$	0	0	$\pm 350°$

The first example considers two robots operating in the workcell with three static obstacles whereas the second example considers the workcell with three robots operating in an environment with two static obstacles. The addition of another robot leads to an increase in the number of dynamic obstacles and hence can increase the complexity of searching the collision-free space and the path for each robot. In effect, these examples can help to ascertain the flexibility of the proposed approach independent of the nature of obstacles and the number of robots sharing the workspace. In both these examples, the robots are positioned in the workspace at a suitable distance from each other. The size of robot used for simulation purpose is identical to real size of the robot. The target configurations for each of these robots are selected in such a way that these robots will definitely collide during the execution of desired task.

Example 1: *In this example, the workcell consists of two identical robots, each with six degrees of freedom and three obstacles of different shapes. The two obstacles are modelled as spheres with radius of* 25 *mm and third one as a block of* 50 *mm* × 25 *mm* × 5 *mm with rounded edges using SSV technique. Figure* 8*(a,b) shows the initial and final configurations of each robot. Both the robots can start their movement at the same time to move its end effector from its initial configuration* (−403.946, 76.0782, 1275.03) *to final configuration* (346.756, 254.275, 650.925) *among the obstacles.*

Example 2: *In this example, the workcell consists of three identical robots, each with six degrees of freedom and two blocks of* 50 *mm* × 25 *mm* × 5 *mm as static obstacles. These obstacles are modelled as blocks with rounded edges using SSV technique. Figure* 9*(a,b) shows the initial and final configurations of robots in the workcell. The robot has to move its end effector from its initial configuration* (−726.348, −888.956, −580.854) *to final configuration* (2006.88, 266.758, −1130.2) *among the obstacles.*

The proposed decoupled offline path planning approach is employed to plan the path for coordinated motion of robots considered in the example 1. The collision-free paths determined for robot R_1 (R_1P_1) and robot R_2 (R_2P_1) by considering only static obstacles but without considering other robot in the workspace are shown in Figs. 10(a) and 10(b), respectively. These paths are relevant when these robots are independently operated without their simultaneous movement. But for coordinated movement of these robots, the coordinated path was determined by adopting replanning strategy with A* and incremental A* algorithms. The path determined by A* algorithm for robot R_1 $[(R_1P_2)_7]$ and for robot R_2 $[(R_2P_2)_{15}]$ is given in Fig. 10(c, d) whereas the paths obtained with incremental A* algorithm for both robots are shown as $(R_1P_2)_5$ and $(R_2P_2)_7$ in Figs. 10(e) and 10(f). From the results presented in Table II, it can be noticed

https://doi.org/10.1017/S0263574709005773 Published online by Cambridge University Press

Table II. Results obtained with the proposed approach.

	Example 1				Example 2					
	A*		Incremental A*		A*			Incremental A*		
Parameter	R_1	R_2	R_1	R_2	R_1	R_2	R_3	R_1	R_2	R_3
No. of replanning episodes	7	15	5	7	1	6	6	1	5	3
Total nodes in Cspace	332	318	325	321	172	143	156	172	156	163
No. of nodes searched	6	6	3	4	116	117	94	116	81	9
Length of coordinated Path $(R_nP_2)_j$ (mm)	1133.68	1146.35	1077.89	1092.91	3096.11	2888.84	5896.68	3096.11	2729	4359.27
Length of path without coordination (R_nP_1) (mm)	1069.56	1077.89	1069.56	1077.89	3068.45	2644.08	4077.46	3068.45	2644.08	4077.46
Deviation in path length due to coordination (mm)	64.12	68.45	8.33	15.01	27.66	244.75	1819.21	27.66	84.91	281.80

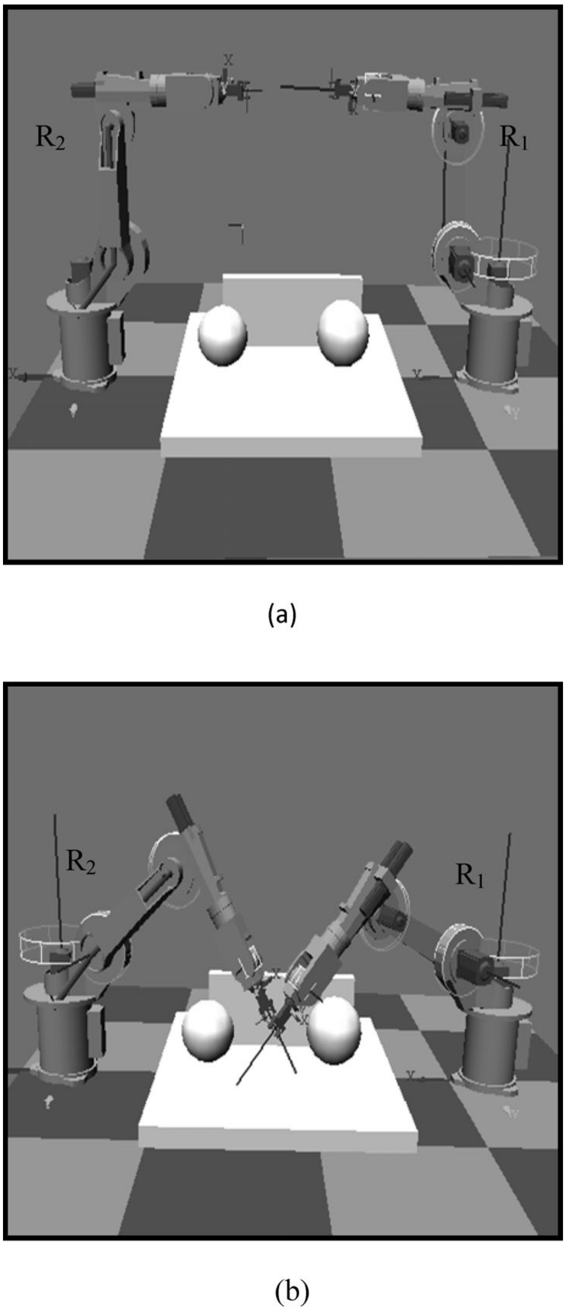


Fig. 8. Coordinated task with two robots (Example 1) (a) initial configuration of robots and (b) final configuration of robots.

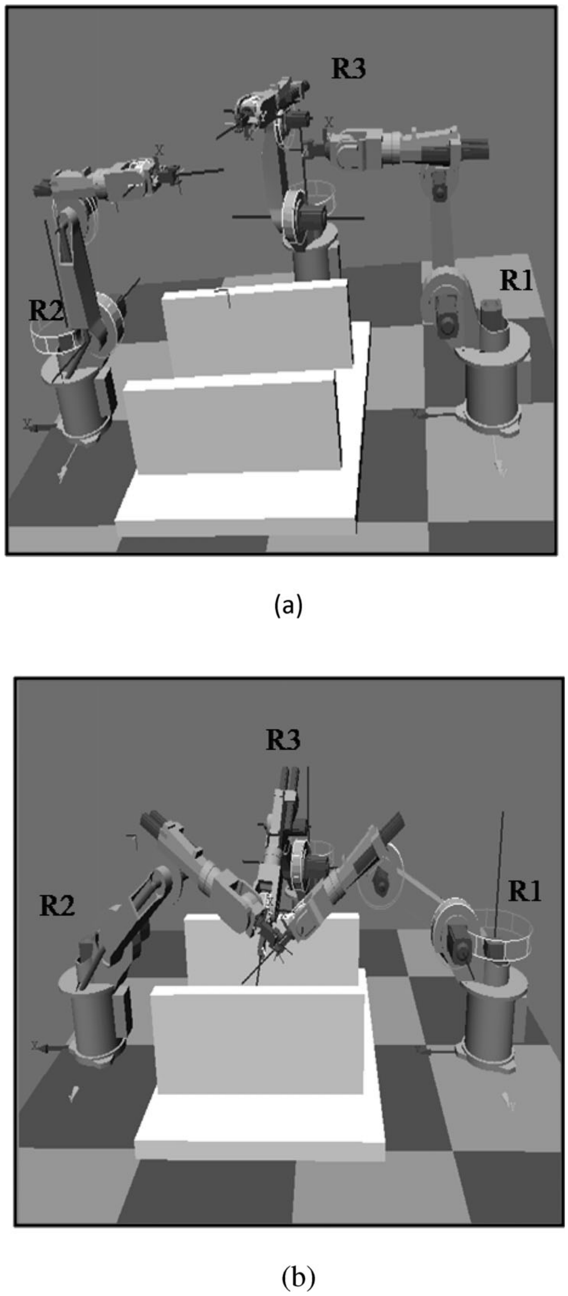


Fig. 9. Coordinated manipulation with three robots (Example 2) (a) initial configuration of robots and (b) final configuration of robots.

https://doi.org/10.1017/S0263574709005773 Published online by Cambridge University Press

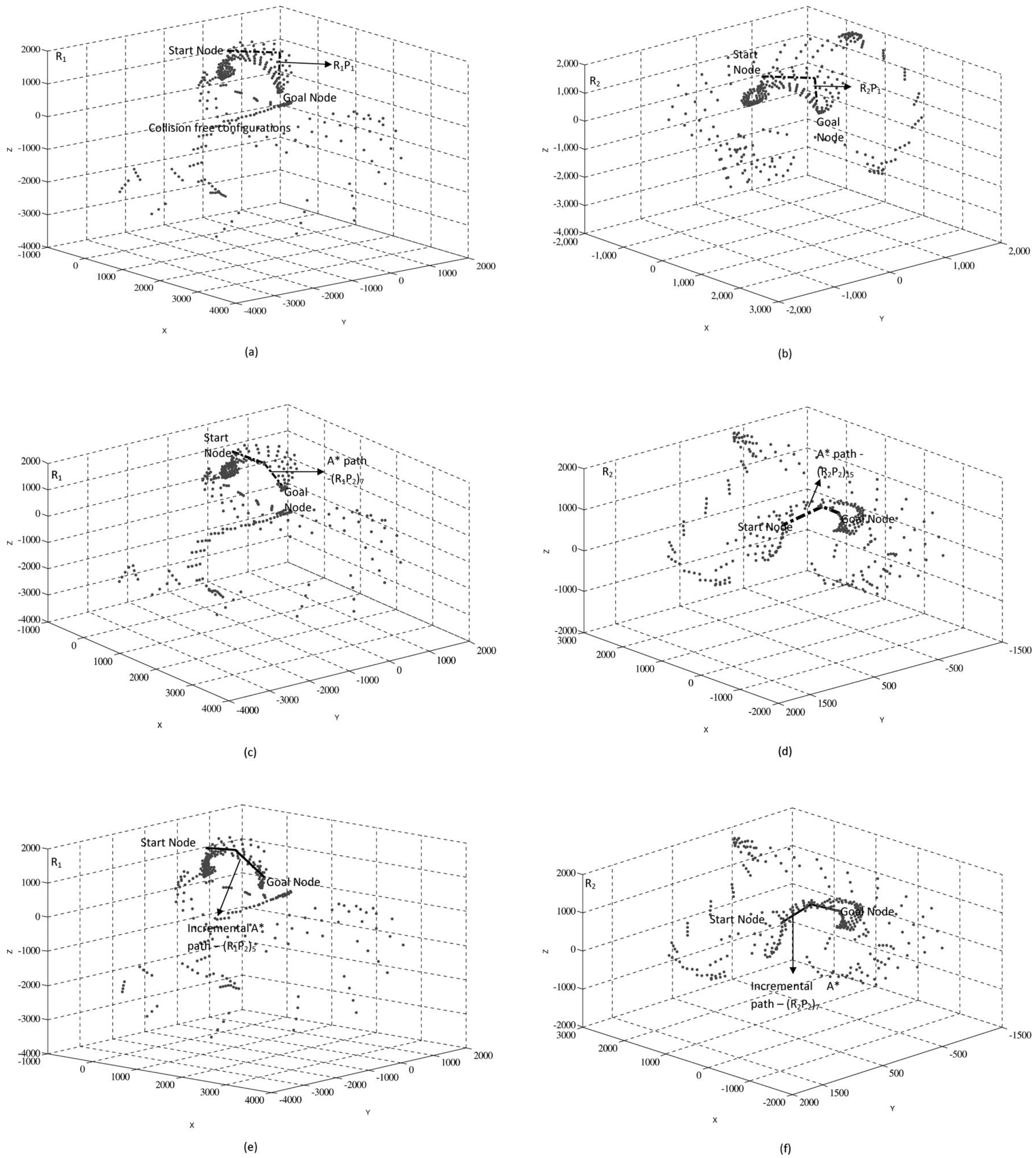


Fig. 10. Collision-free path for robots R_1 and R_2, using A* and incremental A* algorithm (Example 1). Collision-free path with respect to static obstacles (a) for robot 1 – R_1P_1 and (b) for robot 2 – R_2P_1. Coordinated collision-free path (c) A* path – $(R_1P_2)_7$, (d) A* path – $(R_2P_2)_{15}$, (e) incremental A* path – $(R_1P_2)_5$ and (f) incremental A* path – $(R_2P_2)_7$.

that the coordinated path for robot R_1 using A* algorithm is obtained after seven replanning episodes and the length of path is 1133.68 mm. This path is longer by 64.12 mm over the length of shortest uncoordinated path R_1P_1. In contrast to this, the coordinated path for robot R_1 was realized after five replanning episodes with incremental A* algorithm. The path length is 1077.89 mm which is slightly, i.e., 8.33 mm longer than the shortest uncoordinated path length R_1P_1. For obtaining this path, A* algorithm visited 6 nodes out of 332 nodes whereas the incremental A* algorithm visited only 3 nodes out of 325 nodes of Cspace for R_1. Similar trends are noticed while obtaining the coordinated path for robot R_2.

https://doi.org/10.1017/S0263574709005773 Published online by Cambridge University Press

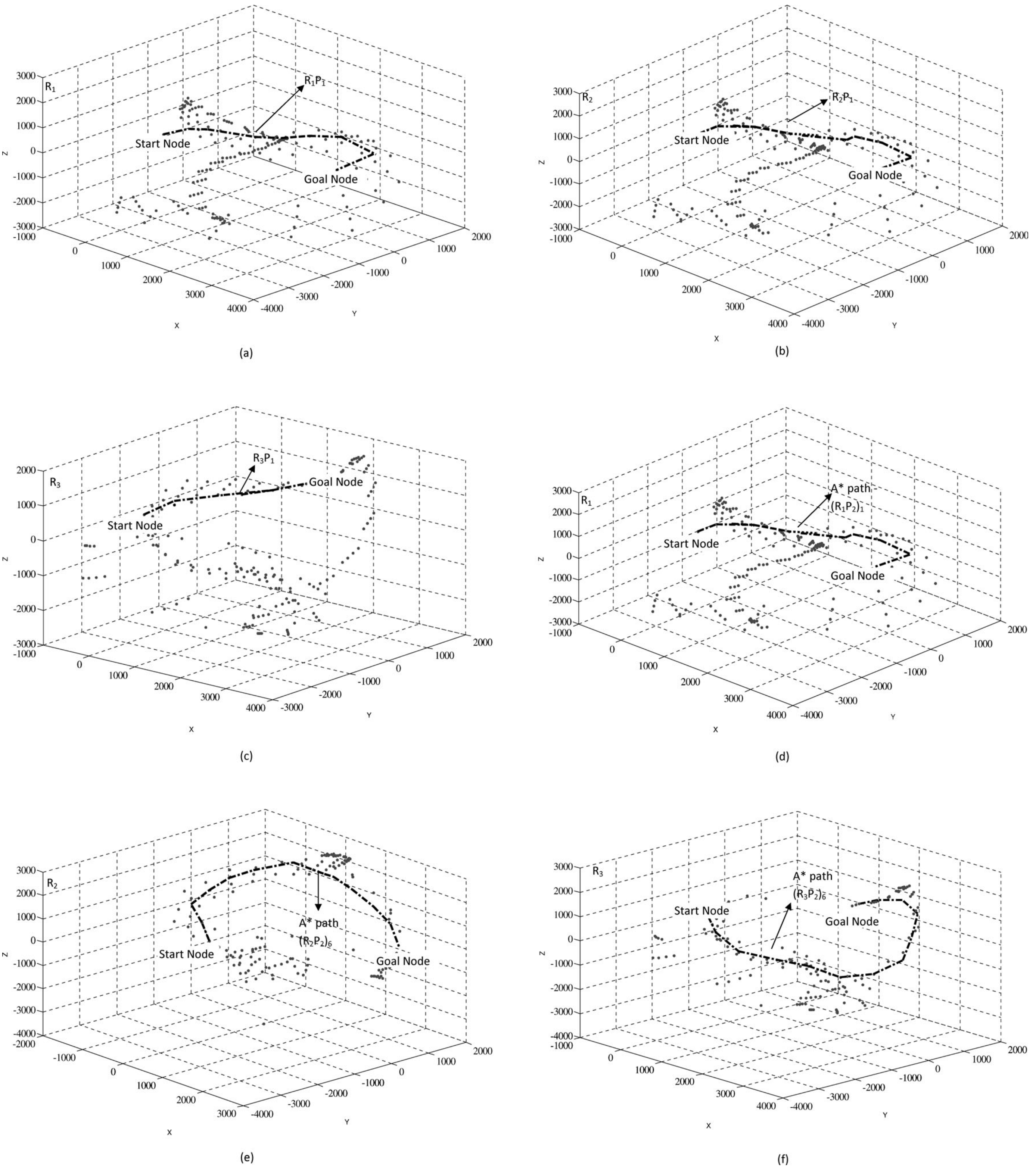
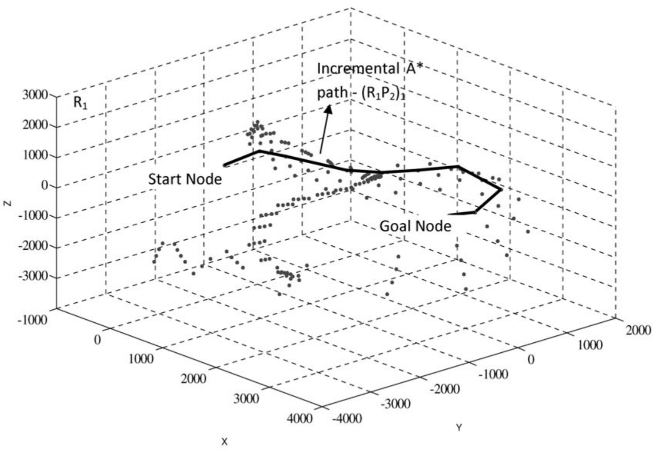


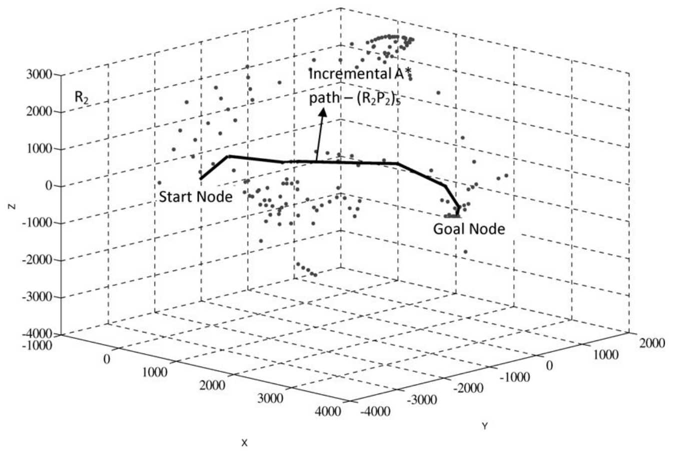
Fig. 11. For legend see next page.

The proposed approach was also employed to generate coordinated path for three robots considered in the second example. Figure 11(a–c) shows the collision-free paths of all the three robots after the first phase of path planning. Figure 11(d–f) shows the coordinated paths of all the three robots R_1, R_2 and R_3 using A* algorithm and Fig. 11(g–i) shows coordinated path obtained using incremental A* algorithm for realising the task considered in the Example 2.

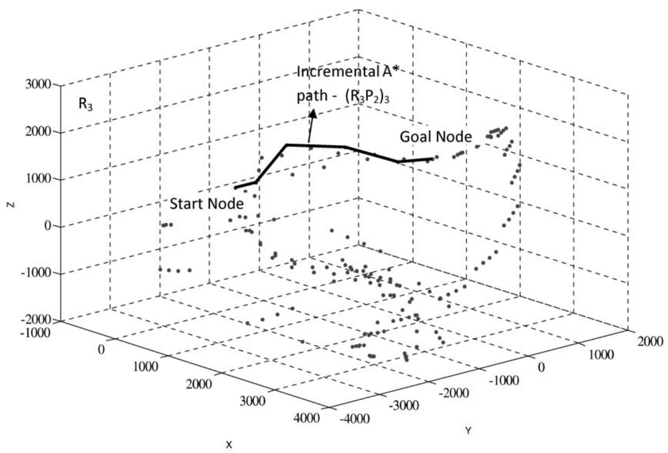
From the results presented in Table II, it can be seen that an incremental A* algorithm outperforms A* algorithm in terms of the number of replanning episodes, the number of nodes searched to find collision-free path and the path length covered by each of the robots for task execution. For robot R_2, A* algorithm found the coordinated path after six replanning episodes and the length of path is 2888.84 mm which is 244.75 mm more than the shortest path length



(g)



(h)



(i)

Fig. 11. Collision-free path for robots R_1, R_2, R_3 using A* and incremental A* algorithm (Example 2). Collision-free path with respect to static obstacles (a) for robot 1 – R_1P_1, (b) for robot 2 – R_2P_1 and (c) for robot 3 – R_3P_1. Coordinated collision-free path (d) A* path – $(R_1P_2)_1$, (e) A* path – $(R_2P_2)_6$, (f) A* path – $(R_3P_2)_6$, (g) incremental A* path – $(R_1P_2)_1$, (h) incremental A* path – $(R_2P_2)_5$ and (i) incremental A* path – $(R_3P_2)_3$.

obtained without considering the coordination among robots. The same task is achieved by incremental A* algorithm in five replanning episodes with a path length of 2729 mm which is 84.91 mm longer than the shortest path. To obtain this path, A* algorithm visited 117 nodes out of 143 nodes whereas the incremental A* algorithm performed the same task by visiting only 81 nodes out of 156 nodes. Similar trends are also noticed for robot R_1 and R_3 in the Example 2.

Since A* search technique is one time computation algorithm which reconstructs the new path every time the state of the environment changes, it requires more number of replanning episodes. In contrast to this, incremental A* algorithm works on reuse strategy than a replan strategy and reconstructs only the areas affected by the changes to the state of environment. It considers the initial path generated by A* algorithm and uses it for further replanning to obtain the final path. As incremental A* algorithm carries out search only in the affected zones, the number of nodes visited during the search are less when compared to the number of nodes visited during the search with A* algorithm.

From the above, it is clear that the proposed decoupled approach generates feasible and effective path for coordinated motion of multiple robots with multiple obstacles in the workspace.

5. Conclusions

The proposed decoupled offline path planning approach for collision-free path planning of multiple robots made use of SSV technique for modelling of links of robots and obstacles, minimum distance heuristic as interference checking criterion and a two-step approach for path planning with A* and incremental A* algorithms as search algorithms. The effectiveness of proposed approach is demonstrated with the help of two case studies each with different degree of complexity of path planning. This approach is unique in a way that the collision-free path for each robot is determined by considering the obstacles in the workspace as static obstacles and the collision-free path for coordinated manipulation is obtained by considering the links of interacting robots as dynamic obstacles. With the case studies considered, it is clear that the proposed decoupled offline path planning approach is independent of the number of robots sharing the workspace and the nature of obstacles.

https://doi.org/10.1017/S0263574709005773 Published online by Cambridge University Press

Moreover, this approach does not involve any prioritisation or complex optimisation function. The advantage of using incremental A* algorithm for coordinated path planning is evident from the substantial reduction in the number of replanning episodes. Moreover the length of path determined by using this algorithm is shorter as compared to the path obtained by using A* algorithm, thus showing its promise for path planning of robots among dynamic obstacles. The approach can be extended to generate the coordinated path for robots operating in cluttered environment with the nature of obstacles and their movement is partially known or completely unknown. Moreover, nature of the proposed approach is generic and hence can be implemented to coordinated path planning of team of articulated robots.

References

1. J. P. van der Berg and M. H. Overmars, "Prioritized Motion Planning for Multiple Robots," *Proceedings of IEEE/RSJ International Conference on Robotics and Systems*, Edmonton, Canada (2005) pp. 2217–2222.
2. B. Aronov, M. de Berg, A. F. Van der Stappen, P. Svestka and J. Vleugels, "Motion planning for multiple robots," *Discr. Comput. Geomet.* **22**, 505–525 (1999).
3. G. Sanchez and J. C. Latombe, "Using a PRM Planner to Compare Centralized and Decoupled Planning for Multi-Robot Systems," *Proceedings of IEEE International Conference on Robotics and Automation*, Washington, DC, Vol. 2 (2002) pp. 2112–2119.
4. J. C. Latombe, *Robot Motion Planning*. (Kluwer academic press, Boston, 1991).
5. Barraquand and J. C. Latombe, "A Monte-Carlo Algorithm for Path Planning with Many Degrees Of Freedom," *Proceedings of IEEE International Conference on Robotics and Automation*, Cincinnati, OH, Vol. 3, (1990) pp. 1712–1717.
6. C. Warren, "Multiple Robot Path Coordination Using Artificial Potential Fields," *Proceedings of IEEE International Conference on Robotics and Automation*, Cincinnati, OH (1990) pp. 500–505.
7. T. Y. Li and H. C. Chou, "Motion Planning for a Crowd of Robots," *Proceedings of IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, ROC (2003) pp. 4215–4221.
8. P. A. O'Donnell and T. Lozano-Perez, "Deadlock-Free and Collision-Free Coordination of two Robot Manipulators," *Proceedings of IEEE International Conference on Robotics and Automation*, Scottdale, AZ (1989) pp. 484–489.
9. K. Hwang, M. Ju and Y. Chen, "Speed alteration strategy for multijoint robots in co-working environment," *IEEE Trans. Indus. Electr.* **50**(2), 385–393 (2003).
10. X. Cheng, "Online Collision Free Path Planning for Service and Assembly Tasks by a Two Arm Robot," *Proceedings of IEEE International Conference on Robotics and Automation*, Nagoya, Japan (1995) pp. 1523–1528.
11. K. G. Kant and S. W. Zucker, "Towards efficient trajectory planning: Path velocity decomposition," *Int. J. Rob. Res.* **5**, 72–89 (1986).
12. M. Bennewitz, W. Burgard and S. Thru, "Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobile robots," *Rob. Autonom. Syst.* **41**(2), 89–99 (2002).
13. Th. Fraichard, "Trajectory: a planning in a dynamic workspace state time space approach," *Adv. Rob.* **13**(1), 75–94 (1999).
14. E. Todt, G. Raush and R. Suarez, "Analysis and Classification of Multiple Robot Coordination Methods," *Proceedings of IEEE International Conference on Robotics and Automation*, San Francisco, CA, Vol. 4, (2000) pp. 3152–3157.
15. S. Gottschalk, M. Lin M and D. Manocha, "OBB Tree : A Hierarchical Structure for Rapid Interference Detection," *Proceedings of 23rd International Conference on Computer Graphics and Interactive Techniques*, New Orleans, LA (Aug. 4–9, 1996).
16. M. Ju, J. Liu and K. Hwang, "Ellipsoidal modeling for articulated robot manipulators for interactive motion planning," *Technical Report*, TR-IIS-00–008, Academia Sinica (2001).
17. H. Sammet and R. Webber, "Hierarchical data structures and algorithms for computer graphics," *IEEE Trans. Comp. Graph. Appl.* **4**(3), 46–68 (1998).
18. I. Palmer and R. Grimsdale, "Collision detection for animation using sphere trees," *Comp. Graph. Forum* **14**(2), 105–116 (1995).
19. O. Brock and O. Khatib, "Real Time Obstacles Avoidance and Motion Coordination in a Multi-Robot Workcell," *IEEE International Symposium on Assembly and Task planning*, Porto, Portugal (1999) pp. 274–279.
20. C. Fares and Y. Hamam, "Collision Detection Between Virtual Objects Using Optimization Techniques," *Information processing: Recent Mathematical Advances in Optimization and Control*, Paris (2005).
21. C. Fares and Y. Hamam, "Collision Detection for Rigid Bodies: A State of the Art Review, Keynote Paper," *Proceedings of 15th International Conference on Computer Graphics and Applications*, Novosibirsk Akademgorodok, Russia (Jun. 20–24, 2005).
22. T. Harden, C. Kapoor and D. Tesar, "Obstacle Avoidance Influence Coefficients for Manipulator Motion Planning," *Proceedings of ASME IDETC/CIE*, Long beach, CA (Sep. 24–28 2005) pp. 1–13.
23. G. Song, S. Thomas and N. Amato, "A general frame work for PRM motion planning," *Proc. IEEE Int. Conf. on Robotics & Automation*, Taipei, Taiwan, (Sep. 14–19, 2003) pp. 4445–4450.
24. G. Sanchez and J. C. Latombe, "A Single-Query Bi-Directional Probabilistic Roadmap Planner with Lazy Collision Checking," *Proceedings of International Symposium on Robotics Research*, Siena, Italy (2003) pp. 404–417.
25. K. I. Trovato and L. Dorst, "Differential A*," *IEEE Trans. Knowledge Data Eng.* **14**(6), 1218–1229 (2002).
26. S. Koenig and M. Likhachev, "Incremental A*," *J. Adv. Neural Info. Process. Syst.* **2**(14), 1539–1546 (2002).
27. S. Koenig, M. Likhachev and D. Furcy, "Lifelong planning A*," *J. Artific. Intell.* **155**(1–2), 93–146 (2004).
28. S. Koenig and M. Likhachev, "A New Principle for Incremental Heuristic Search: Theoretical Results," [Poster Abstract] *International Conference on Automated Planning and Scheduling (ICAPS)*, The English Lake District, Cumbria, UK (2006) pp. 402–405.
29. J. Kuffner, "Efficient Optimal Search of Uniform Cost Grid and Lattices," *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, Vol. 2 (Sep. 28–Oct. 2, 2004) pp. 1946 – 1951.
30. S. LaValle, M. Branicky and S. Lindemann, "On the relationship between classical grid search and probabilistic roadmap," *Int. J. Rob. Res.* **23**(7–8), 673–692 (2004).
31. A. Stentz, "The Focussed D* Algorithm for Real-Time Replanning," *Proceedings of the International Joint Conference on Artificial Intelligence*, Montreal, Quebec, Canada (1995).
32. S. Koenig and M. Likhachev, "D* Lite," *AAAI Conference of Artificial Intelligence (AAAI)*, Edmonton, Alberta, Canada (2002) pp. 476–483.
33. D. Ferguson and A. Stentz, "The Delayed D* Algorithm for Efficient Path Replanning," *IEEE International Conference on Robotics and Automation*, Barcelona, Spain (2005) pp. 2045–2050.