24th International Symposium on Automation & Robotics in Construction (ISARC 2007)
Construction Automation Group, I.I.T. Madras

COORDINATION STRATEGY FOR PATH PLANNING OF MULTIPLE MANIPULATORS IN WORKCELL ENVIRONMENT

Shital S. Chiddarwar and N. Ramesh Babu
Manufacturing Engineering Section
Dept of Mechanical Engineering
Indian Institute of Technology Madras
Chennai – 600 036 India
{shital_c, nrbabu }@iitm.ac.in

ABSTRACT

In this paper, a coordination strategy for determining collision free path of end effectors of two manipulators involved in coordinated manipulation is proposed. This strategy consists of collision checking and path planning modules. Collision checking is done by modeling the links and environment of manipulator using sphere swept volume technique and utilizing minimum distance heuristic for interference check. Path planning is based on a two step approach for which breadth first search, A* algorithm and incremental A* algorithms are used to search the optimal collision free path. While determining the path of the end effector of manipulators, the obstacles present in the work space are considered as static obstacles and the links of the manipulator are viewed as dynamic obstacles by the other manipulator. The proposed coordination strategy is demonstrated by considering two manipulators with 6 degrees of freedom operating in 3D work cell environment with certain static obstacles.

KEYWORDS

Coordination Strategy, Multiple Manipulators, Collision Checking, Incremental Search

1. INTRODUCTION

Industrial manipulators/robots are employed for a wide range of applications wherein a single or multiple robots assist the execution of tasks in a work cell environment. These tasks can range from very simple pick and place to a precise assembly tasks. In case of certain assembly tasks such as spot welding of an automobile body, multiple manipulators are needed. Moreover, these manipulators have to be activated simultaneously to perform the tasks in a desirable manner. In order to utilize these manipulators for any specified task, it is essential to plan the sequence of movements of each manipulator in the work space with the consideration of obstacles and objects. This basically involves path planning and trajectory planning. Many attempts were made towards path planning and trajectory planning of single manipulator. Limited attempts were made to generate the path and trajectory for multiple manipulators considering their operation in cooperative and coordinated mode. Coordinated motion planning of multiple manipulators assumes considerable significance in case of precision assembly and manufacturing tasks carried out with multiple manipulators. Attempts made on multiple manipulator coordinated manipulation are essentially directed towards collision free trajectory planning. Some attempts made in this direction include (i) leader follower strategy wherein the movements of one manipulator have priority over the movements of other manipulator [1], (ii) time delay strategy wherein the motion of one manipulator is delayed by minimum time to avoid possible collision with other robots [2], and (iii) speed alteration strategy in which the speed of the links of one manipulator is made different from the speed of various links of other manipulator in order to avoid any collision with it [3]. All these approaches are based on the assumption that the environment is static, well known apriori and focus on the trajectory for collision free task execution. Hence, they are mostly suited for online planning and execution of tasks in coordinated manipulation. As this process hinders the utility of manipulators and requires enormous time for trajectory planning, there exists a need to look at methods that can generate the path for coordinated manipulation in offline mode. This approach can help in offline generation of path for coordinated motion of manipulators i.e. path planning which in turn can be utilized for trajectory generation and execution with any specific controller with minimal effort and time. In order to develop such an approach for offline generation of path for coordinated motion of manipulators, it is necessary to consider the movement of links of one manipulator with respect to the links of other manipulator as varying with time, thus making the environment dynamic due to the varying position of links of the manipulators in the task performance. Thus, the present work attempts to focus on the development of coordination strategy for path planning of multiple manipulators. This strategy essentially requires the determination of collision free space and planning the path for multiple manipulators coordinating task while considering the links of manipulators as dynamic obstacles. Collision free space determination essentially demands the modeling of work space and interference checking between interacting entities. Among the various techniques available for modeling, oriented bounding boxes [4] and minimum volume ellipsoid [5] techniques are commonly used to model the links of manipulator and environment. Since these techniques are computationally complex and time consuming, swept sphere volume technique (SSV) is found to be the most effective and accurate technique for modeling the links of manipulator and the environment [6], [7]. Collision free configuration space can be found by performing the interference check among the interacting elements of workspace using different methods like minimum distance heuristic, overlap of geometries etc. Path planning essentially deals with the determination of optimal collision free path for executing the task. For this purpose, several approaches can be found in the literature. Out of them, probabilistic roadmap approach and grid search techniques are the most popular. But, if the environment is dynamically changing like in multiple manipulator workcell environment, the most widely used probabilistic roadmap approach (lazy PRM) may fail as the movement of obstacle will lead to elimination of roadmap edges [8]. On the other hand, grid search technique poses certain difficulties like increase in complexity of search in exponential manner with an increase in dimension of configuration space. Attempts made to reduce the complexity of search include the reduction in the number of neighboring nodes to be searched with a heuristic combined with grid search technique [9], incremental building of grids in higher dimensional spaces based on random sampling [10] and incremental version of A* to reduce the number of nodes to be recomputed by reusing the information from previous searches [11]. All these approaches can be employed to obtain the path for manipulators with both static and dynamic obstacles. In the present work, a new strategy is proposed to generate collision free path that can ensure the avoidance of collision of manipulator with static as well as dynamic obstacles. The proposed strategy attempts to utilize SSV for modeling of various links of manipulators, obstacles and objects, minimum distance heuristic for interference checking and collision avoidance with an incremental A* approach for obtaining an optimal path for coordinated manipulation.

2. OUTLINE OF PROPOSED APPROACH

In figure 1, strategy proposed for coordinated manipulation is outlined. The proposed strategy is a two step approach in which the path for coordinated manipulation is generated with a coupled interaction between collision checking and path planning module. In the first step, collision checking module aids in determining collision free configuration space for each manipulator with respect to static obstacles and the path planning module determines an optimal collision free path for achieving the desired task among the static obstacles only. In the second step, the collision free path determined in previous step i.e. various configurations of first manipulator are checked for any mutual collision with moving links of second manipulator. In this process, the links of other manipulators are considered as dynamic obstacles. If collision occurs, the configurations are removed from the collision free configuration space of the manipulator. The remaining collision free space is used to replan the path considering both static and dynamic obstacles. This replanning makes use of both the steps of path planning. By this procedure, replanning is continued until the collision free configurations are found for both manipulators. The proposed strategy essentially includes the tasks outlined in following modules.

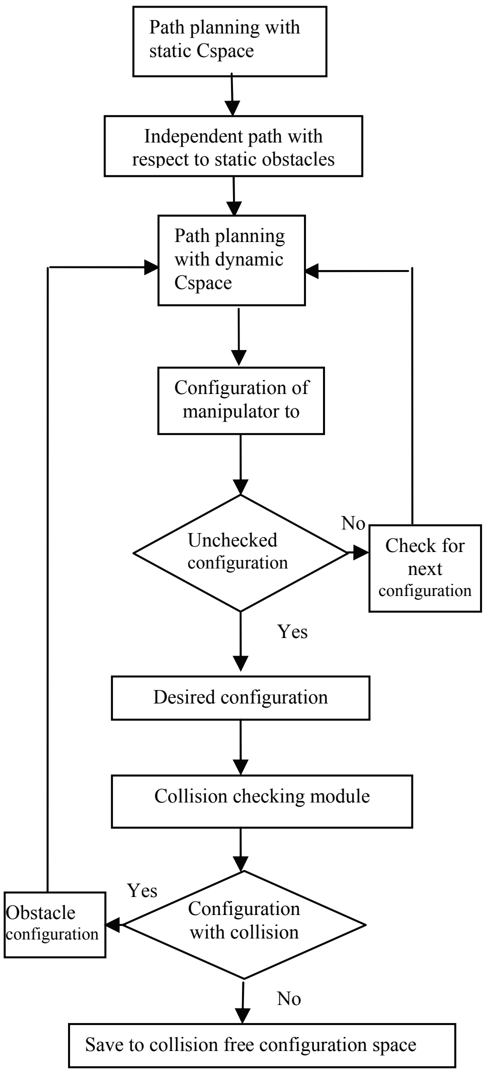


Figure 1 Strategy for Coordinated Manipulation

1. Collision checking module
   a) Modeling of manipulator links and obstacles using sphere swept volume (SSV)
   b) Checking the interference among various geometric primitives modeled using SSV
   c) Determination of collision free configuration space.
2. Path planning module
   a) Determination of collision free path for each manipulator considering only static obstacles
   b) Determination of collision free path for coordinated manipulation with static and dynamic obstacles in position

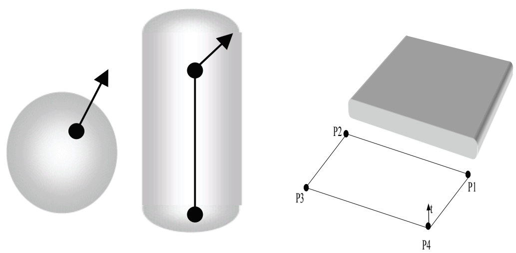
3. COLLISION CHECKING AND PATH PLANNING

3.1 Collision Checking

The collision checking involves modeling of links of manipulator and obstacles in the environment using SSV, interference checking using the minimum distance heuristic and the determination of collision free Cspace.

3.1.1 Robot link modeling using sphere swept volume (SSV) technique

SSV is a sphere that is swept out along geometric primitives such as point, line and rectangle. 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 (Figure 2). These volumes provide means for varying the shape of bounding primitive to achieve a tighter fit to underlined geometry. The link of manipulator is modeled as a cylinder with hemispherical ends. The complex shaped obstacles are enclosed in spherical volume and obstacles with rectangular cross section are modeled as platform with rounded edges and corners. The interference between links of manipulator and static or dynamic obstacles is determined by using minimum distance heuristic. The minimum distance among different geometric primitives is determined by using computational geometry concepts [12]. If this minimum distance is less than the specified value, then interference occurs among primitives under study. This particular configuration of manipulator is considered as collision configuration.



(a) (b) (c)

Figure 2 Geometry of Solid Obtained by Sphere Swept Volume for (a) point (b) line (c) rectangle

3.1.2 Determination of Cartesian Configuration Space (C space)

In order to analyze the motion of manipulator end effector for coordinated task Cartesian coordinate frames are assigned to each link to transform joint angles into task space coordinates. The task space coordinates are nothing but the position and orientation of the end effector with respect to reference base frame. The relative position and orientation of coordinate frames of two adjacent links is expressed in terms of transformation matrices defined as follows:

$$^{i}A_{i-1}=\begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

i = 1,2,3…n, where n is the number of links and a_i, α_i, θ_i, d_i are link length, twist angle, joint angle and joint offset variables respectively. Joint angles of the manipulator θ_1 to θ_6 are varied by a unit degree within the range of angles specified by kinematic constraint of manipulator, to generate manipulator configurations in terms of $[\theta_1,\theta_2,\theta_3,\theta_4,\theta_5,\theta_6]$. These configurations are given as input to collision checking module to ensure them to be collision free and then subjected to singularity check. Singularities are those manipulator configurations for which robot transitorily loses one or more degrees of freedom. The singularities occur at manipulator configuration for which the determinant of Jacobian matrix is zero. Hence, Jacobian matrix for each manipulator is determined. The manipulator configuration which gives the determinant of Jacobian matrix as zero is eliminated though it is collision free. Then these collision free, non singular joint angle configurations $[\theta_1,\theta_2,\theta_3,\theta_4,\theta_5,\theta_6]$ are substituted in transformation matrices and are converted into Cartesian variables [x, y, z]. All such collision free singularity free configurations forms a Cartesian Cspace which is then used for determining the path using path planning approach proposed in the next section.

3.2 Path Planning

As discussed earlier, path planning is carried out in two steps. In the first step, collision free Cartesian Cspace of manipulator with respect to static obstacles is obtained by means of collision checking module. This Cspace is searched for optimal collision free path using breadth first search (BFS) and A* algorithms. These are very widely used grid search algorithms especially suited for obtaining optimal path in static search space. BFS carries out exhaustive search of the Cspace and visits almost all the nodes in the Cspace to find optimal path. In contrast to this, A* algorithm uses heuristic to calculate the cost of movement in the Cspace and then to guide the search. This search is faster than BFS since less number of nodes are used for search. In the second step, the collision free path of manipulator determined in previous step is executed and checked for any mutual collision with links of second manipulator as they become dynamic obstacles for the manipulator under consideration. If these configurations are not collision free, these configurations are removed and collision free Cspace is updated. This updated collision free space is considered to replan the path considering the static and dynamic obstacles. Replanning of path is done by using an incremental A* algorithm [11]. It reuses information gathered in previous searches and hence effective for replanning purpose. The process of replanning is continued till optimal collision free is found with consideration to static and dynamic obstacles. In order to highlight the effectiveness of incremental A* algorithm, replanning is also done by using BFS and A* algorithm.

4. SIMULATION EXAMPLE

In order to illustrate the application of the proposed coordination strategy, two identical Kuka Kr industrial manipulators with 6 rotary joints are considered for coordinated manipulation. Table 1 presents Denavit Hartenberg parameters of real size Kuka Kr manipulator.

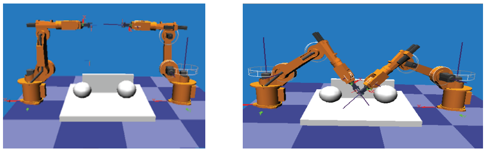
Table 1 Denavit Hartenberg Parameters [13]

Link	θi	αi	ai	di	Joint angle range
1	θ_1	+90	280	715	$-180^O$to 150^O
2	θ_2	0	615	0	-30^O to -55^O
3	θ_3	+90	0	0	-60^O to -210^O
4	θ_4	+90	0	540	-270^O to 90^O
5	θ_5	+90	0	0	-300^O to 54^O
6	θ_6	0	0	0	$-360^O$to 310^O

In the workcell, there are two manipulators with two spheres and one rectangular block as obstacle. The initial and final configuration for manipulators are selected in such a way that these manipulators will definitely collide during their movement from initial configuration (Figure 3(a)) to final configuration (Figure 3(b)).

5. RESULTS AND DISCUSSION

The proposed coordination strategy utilized SSV technique for modeling of the links of manipulator and environment since it requires minimum geometric information of manipulator links or obstacles. This technique facilitates easy computation of distance heuristic which in turn speeded up the collision checking process. The collision free configurations are transformed to Cartesian coordinates using transformation matrices. This collision free Cartesian Cspace is searched for optimal collision free path using various search algorithms. The results obtained with various search algorithms employed in the proposed strategy are shown in Table 2. Figure 4 shows the initial path (i.e. optimal collision free path with respect to static obstacles only) and final path (i.e. optimal collision free path with respect to static as well as dynamic obstacles) of both the manipulators. From the results presented in Table 2, it can be seen that there is vast difference in number of nodes searched by BFS and A* algorithm.



(a) (b)

Figure 3 Coordinated Task with 2 Manipulators (a) Initial Configuration (b) Final Configuration

Table 2 Results Obtained With Different Path Planning Algorithms

Search algorithm	Robot	No. of nodes searched to determine initial path (step1)	No. of nodes searched to determine final path (step2)	Replanning cycles
BFS	R1	163	146	20
	R2	165	144	26
A*	R1	6	3	7
	R2	5	4	15
Incremental A*	R1	6*	6	5
	R2	5*	6	7

This can be attributed to the exhaustive search carried out by BFS which involves the search of almost all nodes of the Cspace for finding an optimal path. On contrary, A* algorithm is a guided search with a heuristic that has reduced the number of nodes searched in the process. The initial path obtained by BFS and A* is given in the figure 4(a) for both robots. Incremental A* algorithm uses the path generated by A* algorithm as initial path for each robot. Thus the number of nodes searched by this algorithm is the same as that of A* algorithm (6* nodes and 5*nodes) to obtain the initial path. To obtain the final path (i.e. collision free path with respect to static as well as dynamic obstacles) for coordinated task, replanning is required. The final path obtained by BFS and A* algorithm is given in figure 4(b) and that obtained by incremental A* algorithm in figure 4(c) for both the manipulators. Since BFS and A* search techniques are one time computation algorithms which reconstruct a new path every time the state of the environment changes, they require more number of replanning cycles. 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.

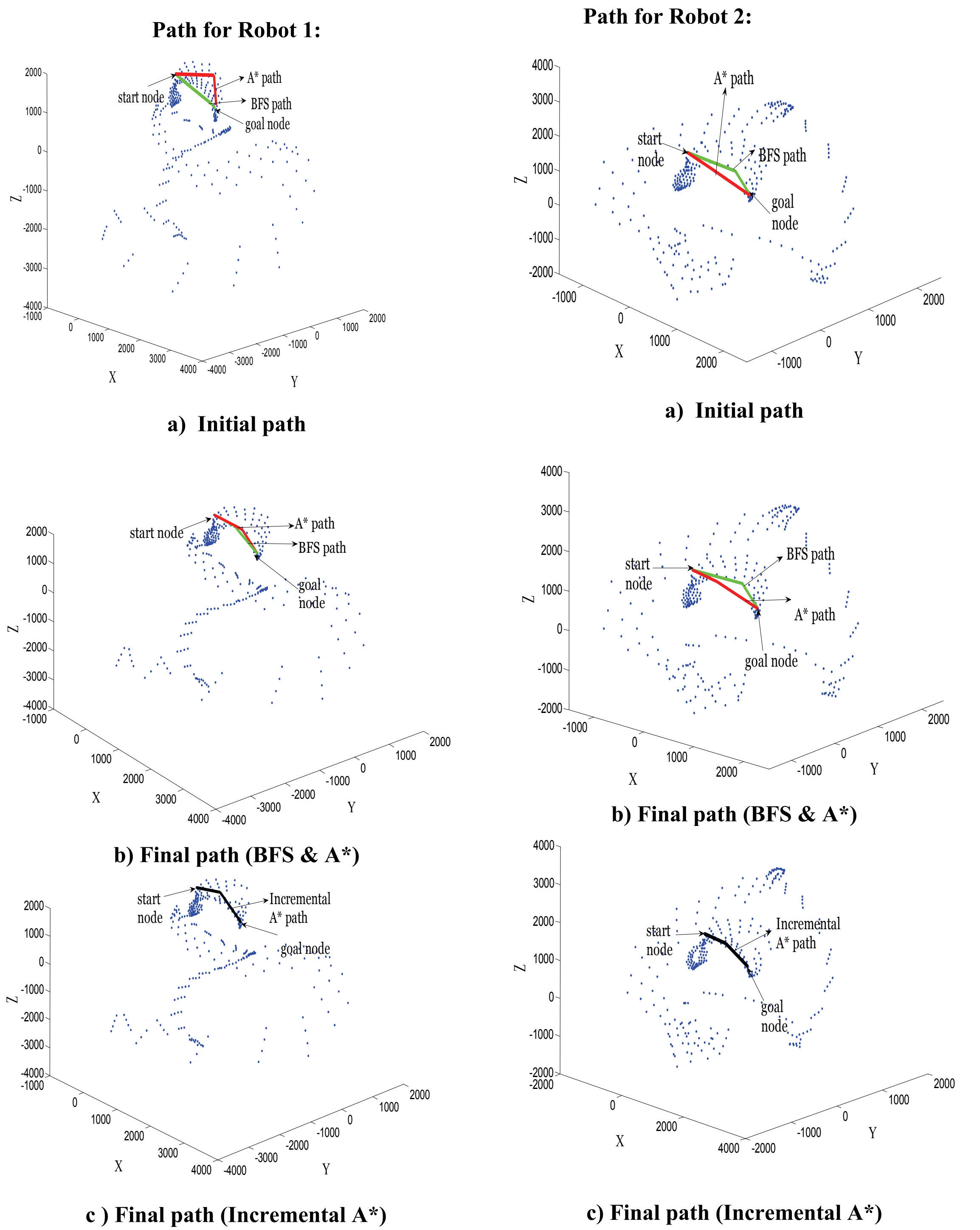


Figure 4: Collision Free Paths Generated for Robot 1 and Robot 2 using BFS, A* and Incremental A* Algorithms

It generates the initial path using A* algorithm and uses it for further replanning so as to obtain the final path.

6. CONCLUSIONS

The proposed coordination strategy for collision free path planning of multiple manipulators made use of SSV technique for modeling of links of manipulators 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. Path planning approach is unique in a way that the collision free path for each manipulator 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 manipulators as dynamic obstacles. Incremental A* algorithm chosen for coordinated path planning is found to reduce the number of replanning cycles substantially thus showing its promise for path planning of manipulators with dynamic obstacles. Though the proposed strategy considered the coordinated path planning of two manipulators with certain geometry of obstacles, it is capable of generating the collision free path for more than two i.e. multiple manipulators considering different geometry of obstacles.

7. REFERENCES

[1] Zheng ,Y. (1989) Kinematics and dynamics of two industrial robots in assembly ,Proc IEEE Int.Conf.on Robotics & Automation, Scottdale, USA,May 14-19 , vol 3, 1360 – 1365.

[2] Chang, C., Chung, M., Lee, B.(1994) Collision avoidance of two general robot manipulators by minimum delay time, IEEE Trans on Systems, Man&Cybernetics, vol24,no3,517-522.

[3] Hwang , K., Ju, M., Chen , Y. (2003) Speed alteration strategy for multijoint robots in co-working environment, IEEE Trans on Industrial Electronics, vol 50, no. 2, 385-393.

[4] Gottschalk , S., Lin , M., Manocha ,D.(1996) OBB Tree : A hierarchical structure for rapid interference detection, Proc. 23rd Int. Conf. on Computer Graphics and Interactive Techniques, Aug 4-9, New Orleans, USA.

[5] Ju, M., Liu, J., Hwang, K.(2001) Ellipsoidal modeling for articulated robot manipulators for interactive motion planning, Technical Report, TR-IIS-00-008, Academia Sinica

[6] Fares,C. and Hamam, Y.(2005) Collision detection for rigid bodies : a state of the art review , keynote paper, Proc 15th Int.Conf. on Computer Graphics and Applications, June 20-24, Novosibirsk Akademgorodok, Russia.

[7] Harden,T., Kapoor,C., Tesar, D.(2005) Obstacle avoidance influence coefficients for manipulator motion planning, Proc ASME IDETC/CIE,Long beach,USA, Sep 24-28,1-13.

[8] Song, G., Thomas ,S., Amato, N.(2003) A general frame work for PRM motion planning, Proc. IEEE Int .Conf. on Robotics & Automation, Taipei, Taiwan, Sep 14-19, 4445-4450.

[9] Kuffner ,J.(2004) Efficient optimal search of uniform cost grid and lattices, Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, Sendai,Japan, Sept28-Oct2, vol 2,1946 – 1951.

[10] LaValle, S., Branicky , M., Lindemann , S. (2004) On the relationship between classical grid search and probabilistic roadmap, Int. Jnl. of Robotics Research, vol 23, no 7-8, 673-692.

[11] Koenig, S., Likhachev, M. (2006) A New Principle for Incremental Heuristic Search: Theoretical Results [Poster Abstract].Procs Int. Conf. on Automated Planning and Scheduling (ICAPS), English Lake District, Cumbria, UK, June 6-10, 402-405.

[12] O' Rourke, J. (1994) Computational geometry in C, Cambridge University Press.

[13] www.kuka.com