

01 Jan 1990

## Planning Optimal Robot Trajectories by Cell Mapping

W. H. Zhu

Ming-Chuan Leu

Missouri University of Science and Technology, mleu@mst.edu

Follow this and additional works at: [https://scholarsmine.mst.edu/mec\\_aereng\\_facwork](https://scholarsmine.mst.edu/mec_aereng_facwork)



Part of the [Aerospace Engineering Commons](#), and the [Mechanical Engineering Commons](#)

---

### Recommended Citation

W. H. Zhu and M. Leu, "Planning Optimal Robot Trajectories by Cell Mapping," *Proceedings of the 1990 IEEE International Conference on Robotics and Automation, 1990*, Institute of Electrical and Electronics Engineers (IEEE), Jan 1990.

The definitive version is available at <https://doi.org/10.1109/ROBOT.1990.126259>

This Article - Conference proceedings is brought to you for free and open access by Scholars' Mine. It has been accepted for inclusion in Mechanical and Aerospace Engineering Faculty Research & Creative Works by an authorized administrator of Scholars' Mine. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact [scholarsmine@mst.edu](mailto:scholarsmine@mst.edu).

# Planning Optimal Robot Trajectories by Cell Mapping

Wen H. Zhu    Ming C. Leu

Department of Mechanical and Industrial Engineering  
New Jersey Institute of Technology  
Newark, NJ 07102

## ABSTRACT

A cell mapping method is introduced for planning global trajectories of robotic manipulators, where the cell space is composed of combination pairs of plane cells. With the proposed method, optimal trajectory problems both in the free field and in the obstacle constrained field are studied. Two numerical examples are given to show the obtained optimal trajectories and controls.

## 1. Introduction

The optimal robot trajectory planning problem is described as follows: Given the dynamics of a manipulator and a geometric description of the manipulator's work space, plan a trajectory (i.e. path as a function of time) between two specified end states such that the manipulator avoids collision with obstacles in the work space and is optimal with respect to a performance index. This problem was initially dealt with by researchers in the form of two sub-problems: path planning and trajectory finding problems. In the path planning problem, the time variable is not considered. It has been widely investigated and discussions can be found in [1-4] and others. In the trajectory finding problem, the path to be followed by the manipulator is given (which may be a result of path planning). This problem has been studied with the dynamic programming method [5,6] and other approaches [7]. The general trajectory planning problem, i.e. the combined path planning and trajectory finding problem, has also been discussed recently. In [8,9] a hierarchical search and planning algorithm was used with an optimization procedure to yield a time minimum solution. In [10] a recursive quadratic programming algorithm based on Pschenichny's linearization method was used to obtain the optimal trajectory for an arbitrary performance index.

Cell mapping was proposed by Hsu [11] as a general method for global analysis of nonlinear dynamic systems. It involves dividing the continuous state space into finite discrete cells and numbering them in order. This method has also been developed to solve optimal control problems [12,13]. There are three main features that distinguish this method from others in optimal control studies. First, it performs a global analysis for any given state space, and optimal trajectories for all possible initial states can be determined simultaneously. Second, obstacles in the work space of the manipulator do not increase difficulties in analysis but, instead, lessen the computation burden. Also, nonlinear constraints on the control force/torque (for example, saturation) and a variety of performance indices can be easily taken into consideration. Third, the method has efficient calculation and storage saving features when a well designed data structure is used to hold information.

Here we introduce a cell mapping method which can be applied to the optimal trajectory planning of manipulators. We will consider first the free space problem and then the obstacle constrained problem.

The same mapping method can be used in both cases after the corresponding cell space for each of them is created. A general expression of performance index, such as traveling time, control energy, or their combinations can be used. Generally, performance indices are additive, e.g. a performance index can be selected as :

$$L = \sum_{i=1}^s (t_i + \alpha(u^T u)_i) \quad (1.1)$$

where  $i$  is a number used to indicate the mapping step,  $s$  is the total step number from the start cell to the target set, and  $\alpha$  is an appropriate coefficient used to weigh the two terms in  $L$ . For time optimal control  $\alpha$  is zero. If it is important to save control energy then  $\alpha$  should be large. Of course, other performance indices may also be used.

## 2. Point mapping in state space

The state of joint  $i$  can be described by a point  $p_i$  in the state plane formed by the joint displacement  $q_i$  and its time derivative  $\dot{q}_i$ . We construct  $n$  independent state planes which can be used to form the admissible state of an  $n$ -link manipulator in a  $2n$ -dimensional state space. For solving a manipulator trajectory planning problem we determine the joint state of the manipulator at time  $t$ . This requires integration of a series of coupled, nonlinear autonomous differential equations in the form of

$$\sum_{i=1}^n M_{ij}(q_1, \dots, q_n) \ddot{q}_i + h_j(q_1, \dot{q}_1, \dots, q_n, \dot{q}_n) = u_j \quad j = 1, 2, \dots, n \quad (2.1)$$

where  $M_{ij}$  represents elements of the inertial matrix,  $h_j$  represents the Coriolis, centrifugal and gravitational effects, and  $u_j$  represents input torques. If we study the manipulator trajectory for a series of time intervals  $(t_1, t_2 \dots t_k, t_{k+1} \dots)$  we can establish a state point mapping in the form of

$$z(k+1) = G(z(k)) \quad (2.2)$$

where  $z = z(q_1, \dot{q}_1, \dots, q_n, \dot{q}_n)$  is a state vector of dimension  $2n$ , and  $k$  is an integer representing the mapping step. For any non-singular points in the state space, mapping is always determinate.

For simplicity we denote  $q(k)$  and  $q(k+1)$  as two different position  $n$ -vectors at  $t = t_k$  and  $t = t_{k+1}$ , respectively. Their contents are in the form of  $(q_1, \dots, q_n)^T$ . Let  $\dot{q}(k)$ ,  $\ddot{q}(k)$ ,  $\dot{q}(k+1)$ , and  $\ddot{q}(k+1)$  be their corresponding velocity and acceleration  $n$ -vectors. It is obvious that

$$\dot{q}(k+1) = \dot{q}(k) + \int_{t_k}^{t_{k+1}} \ddot{q}(t) dt \quad (2.3)$$

With the method of integration by parts we obtain

$$\begin{aligned} q(k+1) &= q(k) + \dot{q}(k)\Delta t + \int_{t_k}^{t_{k+1}} (t_{k+1} - t)\ddot{q}(t)dt \\ &= q(k) + \dot{q}(k)\Delta t + \int_0^{\Delta t} t\ddot{q}(t_{k+1} - t)dt \end{aligned} \quad (2.4)$$

where  $\Delta t = t_{k+1} - t_k$  is a small time step. With the help of the mean value theorem of a continuous function we can get

$$\dot{q}(k+1) = \dot{q}(k) + ((1-\gamma)\ddot{q}(k) + \gamma\ddot{q}(k+1))\Delta t \quad 0 \leq \gamma \leq 1 \quad (2.5)$$

$$q(k+1) = q(k) + \dot{q}(k)\Delta t + \left(\frac{1}{2} - \beta\right)\ddot{q}(k) + \beta\ddot{q}(k+1)(\Delta t)^2 \quad 0 \leq \beta \leq 0.5 \quad (2.6)$$

If  $\gamma$  is close to 0, the velocity  $\dot{q}(k+1)$  at time  $t_{k+1}$  depends mostly on acceleration at time  $t_k$ , and it receives very little contribution from acceleration at time  $t_{k+1}$ . If  $\gamma$  is close to 1, the situation is reversed. The relative influences of the accelerations of the two end instances of a time period to the velocity at time  $t_{k+1}$  can be adjusted by changing the value of coefficient  $\gamma$ . The same principle can be applied to  $q(k+1)$  in (2.6) with respect to coefficient  $\beta$ . If  $\gamma$  and  $\beta$  are set to be constant for the entire integration procedure we have a simple linear acceleration integration technique. Because the dynamic equations (2.1) are highly nonlinear, integration of these equations using a linear acceleration method may cause considerable errors both in velocity and displacement, especially near singular configurations. To improve the accuracy of integration, we use varied  $\gamma$  and  $\beta$  for different time intervals. The values of  $\gamma$  and  $\beta$  can be chosen by a one-dimensional search method.

### 3. Cell mapping procedure

After we establish the point mapping described by (2.2) in the state space, we are ready to form cell mapping. For simplicity, plane cell  $c_i$  is formed in rectangular shape with  $p_i$  as its representative point. Cell  $j$  of the  $i$ -th state plane can combine with cell  $k$  of the  $m$ -th state plane for all  $m = 1, 2, \dots, i-1, i+1, \dots, n$ . One combination pair formed by  $n$  plane cells, each of them located in a different state plane, represents a set of joint states of the manipulator. All the possible combination pairs form a cell space. All states outside of this space form a sink cell. Simple cell mapping uses the center point of a cell as its state representation. The evolution of simple cells is always determinate. The contents contained by simple cells in our optimal manipulator trajectory planning problem include: name of combination pair, control, time duration in one evolution step, performance index from a combination pair to the target, status of a pair and number of steps to the target, location deviation, and image name. These parts (called fields) are closely linked together to compose a data structure. The field of combination pair name is used to determine the cell location in the state space, which includes the composition of combination pair of every plane cell. The control field describes control input used for every joint, with which this combination pair will develop to its image in one step. In the location deviation field, deviation from the ideal route caused by specific cell mapping to the target is stored. † The last field contains the name of image combination pair which will be arrived at under the control input and time duration specified. Every field is accessible in the mapping procedure only if its name is referred to.

The cell mapping method for optimal manipulator trajectory planning is essentially to complete the data structure for every combination pair. Since a  $2n$ -dimensional cell used to represent the state of a robot

† In simple cell mapping, we define the ideal mapping route as follows: starting from a representative point of a cell, after one mapping step its image point is exactly the representative point of the image cell, and this property holds until the target is reached. In other words, along the ideal cell mapping route representative points in every step of mapping coincide with those of point mapping[13].

manipulator is described by a combination pair of  $n$  plane cells, one field in the data structure should be divided into  $n$  components, each of them related to one of  $n$  different state planes.

In most cases there are only a finite number of torques/forces that can be used for a joint. All possible control torques/forces form a countable finite set  $U$ . In reality, the action time of  $u$  cannot be infinitesimally small. We denote  $t_0$  the smallest time duration of evolution. Any other time duration  $t$  used to describe the evolution of system is a multiple of  $t_0$ , and all possible  $t$ 's compose an unlimited countable set  $T$ . For every  $u \in U$  and  $t \in T$  a simple cell-to-cell mapping of manipulator states can be carried out when needed.

To express the general mapping procedure clearly, let us consider an unconstrained work space first. Mapping in the free space has the following features in cell meaning: a) all position variables  $q_i$  are monotonous; b) all velocity variables  $\dot{q}_i$  are positive; c) only one image cell for every  $(u, t)$  is reachable from any starting cell.

To follow the convention of coordinates used in describing manipulator dynamics, the origin of the cell combination pair is set at the starting position. The cells in the target set are the last part of the cell space in the numbering of combination pairs. If the mapping order began also from the lowest numbered cell, it would take a large number of mapping steps to arrive at the target. Since the sorting procedure has to start from the target set, many intermediate data would have to be reserved until the optimal route from a starting cell to the target is determined. Since branch mapping happens frequently in the procedure, the storage needed for all temporary data might be so large that the memory of a computer is quickly exhausted. We avoid this problem by re-arrangement of cell space and mapping with recursive orders as described below.

The combination pair number can be calculated from its components by[13]

$$c = \sum_{i=1}^{n-1} \left( \prod_{j=i+1}^n Nc_j \right) (c_i - 1) + c_n \quad (3.1)$$

where  $Nc_i$  denotes the total number of plane cells in the  $i$ -th state plane and  $c_i$  denotes a cell number in the  $i$ -th plane. From a combination pair number  $c$  every cell component  $c_i$  can be calculated by the following method:

$$\begin{aligned} &\text{if } R_1 = \text{mod}(c, \prod_{j=1}^{n-1} Nc_j) \neq 0, \text{ then} \\ &\quad c_1 = \text{Int}(c / \prod_{j=i+1}^{n-1} Nc_j) + 1 \text{ and } R_2 = R_1; \\ &\text{if } R_1 = \text{mod}(c, \prod_{j=1}^{n-1} Nc_j) = 0, \text{ then} \\ &\quad c_1 = \text{Int}(c / \prod_{j=i+1}^{n-1} Nc_j) \text{ and } R_2 = \prod_{j=1}^{n-1} Nc_j; \\ &\dots \\ &\text{if } R_n = \text{mod}(R_{n-1}, Nc_{n-1}) \neq 0, \text{ then} \\ &\quad c_{n-1} = \text{Int}(R_{n-1} / Nc_n) + 1; \\ &\text{if } R_n = \text{mod}(R_{n-1}, Nc_{n-1}) = 0, \text{ then} \\ &\quad c_{n-1} = \text{Int}(R_{n-1} / Nc_n); \\ &\quad c_n = R_n. \end{aligned} \quad (3.2)$$

With (3.1) and (3.2) we convert the combination pair number  $c$  to its components and vice versa. For the reason mentioned above,  $c$  is

re-arranged to start outward from the closest wrap layer of the target. This new order can be changed back to the original one when needed.

We set up the relation of  $c_i$  with  $q_i$  and  $\dot{q}_i$  as follows. If  $(d_i, \dot{d}_i)$  represents the cell coordinates of the  $i$ -th state plane, then they can be calculated from the state variables  $(q_i, \dot{q}_i)$  by

$$\begin{aligned} d_i &= \text{Int}\left(\frac{q_i}{h\dot{q}_i} + \frac{1}{2}\right) \\ \dot{d}_i &= \text{Int}\left(\frac{\dot{q}_i}{h\dot{q}_i} + \frac{1}{2}\right). \end{aligned} \quad (3.3)$$

in which  $(h\dot{q}_i, h\dot{q}_i)$  is the size of plane cell  $c_i$ . It is not difficult to get

$$c_i = Nc_i - \sum_{j=1}^{d_i-1} (\dot{d}_j)_m + \dot{d}_i \quad (3.4)$$

where

$$(\dot{d}_i)_m = \text{Int}\left(\frac{(\dot{q}_i)_m}{h\dot{q}_i} + \frac{1}{2}\right) \quad (3.5)$$

and  $(\dot{q}_i)_m$  is the upper bound on velocity, under the control set at position  $q_i$  [14].

Let  $\omega$  be the maximum integer that satisfies the following expression:

$$\text{Max}_\omega(Nc_i - \sum_{j=1}^{\omega} (\dot{d}_j)_m \geq c_i)$$

then

$$\begin{aligned} d_i &= \omega + 1 \\ \dot{d}_i &= (\dot{d}_i)_m - (Nc_i - \sum_{j=1}^{\omega} (\dot{d}_j)_m) \end{aligned} \quad (3.6)$$

The cell map is written in the form of

$$c(k+1) = G(c(k), u, t) \quad (3.7)$$

in which argument  $k$  is the step variable and  $u \in U$ . The map is computed for every  $c$  that does not belong to the target set  $Q$  and whose status is unprocessed. To determine the time duration  $t$  between steps we let  $t = t_0$  at the beginning. If  $c(k+1) = c(k)$ , the time is too short to develop a mapping along a cell route, thus  $t$  should be increased to  $2t_0, 3t_0, \dots$ , etc. A better way is to estimate the time duration  $t$  before mapping. Of course this estimation is system dependent; i.e. different system inertia, different control set and different starting cell will result in different suitable time durations. After mapping, the status of  $c(k+1)$  is checked. If the status is processed, a new mapping from  $c(k)$  begins with another  $u \in U$ . When all  $u \in U$  have been used we can begin a sorting procedure. If the status of  $c(k+1)$  is unprocessed, we carry out a same recursive procedure starting from  $c(k+1)$ . This procedure is repeatedly pursued if the same situation is encountered at any stage, until all the images of a considered cell have become processed cells. A sorting procedure follows immediately to determine optimal routes from the processing cells to the target. If we find a cell whose images under  $u \in U$  and  $t \in T$  all belong to the sink cell, this cell is regarded uncontrollable. If all the images of a cell are uncontrollable or belong to the sink cell, the cell is considered uncontrollable, too. It is obvious that "controllability" here refers to the conditions of  $T, U$  and the cell space selected. An uncontrollable cell may become controllable for a different cell space and control input. After a sorting procedure is completed for a processing combination pair, the optimal route from this combination pair to the target can be determined. Thereafter, a processed status is assigned to it, and all fields of data for this combination pair are filled with the corresponding information.

It is possible during the sorting process that several candidates are competing for an optimal trajectory by offering different routes to the target with the same performance index. A discrimination procedure has to be employed in this situation. In order to design a common

principle for various cases, a deviation index is used. The deviation index is computed in every step based on its deviation from the ideal cell mapping route. The summation of the value of this index along a route is stored in its location deviation field if the combination pair is processed. We have two different deviation measures: a local one just for a considered step and a global one including all deviations in the mapping history from a combination pair to the target. With the aid of local and global deviation indices we can distinguish competing candidates. This principle essentially has the meaning of accuracy in simple cell mapping. We can judge the global accuracy of a determined optimal control sequence by examining its deviation index, too.

#### 4. Cell space for an obstacle constrained field

Obstacles in the work space can be modeled by enveloping them with larger but simpler objects, called obstacle shadows. If the shadows exist all the time and do not change its geometric characteristics in space then the constraints are static. Dynamic constraints exist if the shadows are time dependent. Static constraints are commonly encountered when some fixed obstacles exist in the work space. In this paper we will only consider static constraints.

We use the idea of extended cell space for the constrained problem. To explain it we begin with an example plotted in Figure 1. A 2-link manipulator is to move from an initial position to a target in a work space which contains obstacles modeled by two obstacle shadows. Suppose that the displacement  $q_2$  increases toward obstacle shadow 1 and then decreases toward obstacle shadow 2 and then increases until it reaches its ultimate displacement value at the target. The value of  $q_2$  is not monotonous in the whole motion procedure. Velocity  $\dot{q}_2$  can be sometimes positive and sometimes negative. For a given  $q_2$  there may be several different  $\dot{q}_2$  as its correspondences. The conditions put forward in Section 3 for the free field case cannot be maintained for the constrained case. The straightforward transform from the state space to the cell space will cause the problem of non-uniqueness, and thus we have to modify our cell space description method. We form the cell space by  $2k+1$  partitions if there are  $k$  obstacle shadows in the work space. Every shadow occupies a partition and forms two free partitions in between. Usually, the first and last partitions which have included the start and target states cannot be any shadow partitions. Since the cell space is formed by combination pairs of  $n$  plane cells for an  $n$ -link manipulator, every state plane also consists of  $2k+1$  plane partitions. The admissible combination pairs are composed of  $n$  plane cells located correspondingly in the  $n$  plane partitions of same names. The combination pairs formed by cells of cross-plane partitions are expelled from the cell space. The total number of combination pairs in the cell space is reduced from that of the unconstrained case.

The composition of cell space in a constrained field thus has the following major features. The displacement component in the  $i$ -th plane is partitioned if an obstacle shadow exists, and it is extended if a non-monotonous  $q_i$  occurs. Several occurrences of a same value of  $q_i$  in different time instants are reflected by different plane cells as well as combination pairs. Corresponding to the partitioned cell plane, other cell planes are also partitioned under the same name. These partitioned cell planes combine themselves tightly in organization, instead of being independent like the state planes in the state space. A cell mapping relation is created to ensure that the velocity component in a plane cell is always positive, though the real velocity of a manipulator link may be positive or negative. The state space is now topologically mapped to an extended, partitioned cell space which is piecewise monotonous in displacement direction of each cell plane. Any results of calculation completed in the cell space can thus be transformed to the state space uniquely.

#### 5. Example 1: Optimal trajectory planning in the free field

As an example, the method described above is used to obtain optimal trajectories in the free field for a planar manipulator composed of two uniform bars as the links, each with a revolute joint. Let

$l_1, l_2, m_1, m_2$  be the lengths and masses of these two bars, respectively,  $q_1$  and  $q_2$  be the joint rotation angles. The dynamic equations are [15]:

$$\begin{aligned} \ddot{q}_1 \left( \frac{l_1^2 m_1 + l_2^2 m_2}{3} + m_2 l_1 (l_1 + l_2 \cos q_2) \right) + \ddot{q}_2 m_2 l_2 (l_2/3 + l_1 \cos q_2/2) \\ = f_1 + m_2 l_1 l_2 \dot{q}_2 \sin q_2 (\dot{q}_1 + \dot{q}_2/2) \\ \ddot{q}_1 (l_2^2 m_2/3 + m_2 l_1 l_2 \cos q_2/2) + \ddot{q}_2 m_2 l_2^2/3 = f_2 - \dot{q}_1^2 l_1 l_2 m_2 \sin q_2/2 \end{aligned} \quad (5.1)$$

where  $f_1$  and  $f_2$  are the two torques applied to joints 1 and 2, respectively. For the sake of simplicity, let  $l_1 = 2l_2 = 2$  units of length,  $m_1 = 2m_2 = 2$  units of mass, then we have

$$\begin{aligned} \ddot{q}_1 (7 + 2 \cos q_2) + \ddot{q}_2 (1/3 + \cos q_2) = f_1 + \dot{q}_2 (2\dot{q}_1 + \dot{q}_2) \sin q_2 \\ \ddot{q}_1 (1/3 + \cos q_2) + \ddot{q}_2/3 = f_2 - \dot{q}_1^2 \sin q_2 \end{aligned} \quad (5.2)$$

Suppose the target location is at  $r_0 = 2\sqrt{2}$ ,  $\theta = \frac{2\pi}{3}$ , and  $\theta$  is measured from the initial position with  $q_1 = 0$ ,  $q_2 = 0$ , then the target position ( $q_1^*$ ,  $q_2^*$ ) can be determined by

$$\begin{aligned} 2 \sin q_1^* + \sin(q_1^* + q_2^*) = \sqrt{3/2} \\ 2 \cos q_1^* + \cos(q_1^* + q_2^*) = -\sqrt{1/8} \end{aligned} \quad (5.3)$$

It is not difficult to show that

$$\tan(q_1^* + q_2^*) = \frac{2 \tan q_1^* - \sqrt{3(1 + \tan^2 q_1^*)}}{2 + \sqrt{1 + \tan^2 q_1^*}} \quad (5.4)$$

and

$$q_1^* = \frac{2\pi}{3} - \arccos\left(\frac{11}{8\sqrt{2}}\right) = 1.858$$

Solving (5.4) numerically we get  $q_2^* = 0.719$ . We also set  $\dot{q}_1^* = \dot{q}_2^* = 0$ , and we require  $q_1 \geq 0$  and  $q_2 \geq 0$ . The target set contains one combination pair only.

To construct the cell space, first we set up two cell planes. In the first plane 49 intervals are equally divided along  $q_1$ , from  $q_1 = 0$  to  $q_1 = q_1^*$ . The two end points are located at the centers of the cells. Thus there are 50 cell coordinates along  $q_1$ . The length of the cell along this direction is  $h_{q_1} = 0.03795$ . In the same manner 24 cell coordinates along  $q_2$  are created with the length of  $h_{q_2} = 0.03125$  in the second cell plane. In order to study motions versus time in detail, small torques are purposely assigned to the actuators. The maximum torques  $f_1$  and  $f_2$  put on joint 1 and joint 2 are 0.5 and 0.05, respectively. There are 9 elements in  $U$ , each element composed by selecting one from  $[f_1, 0, -f_1]$  and one from  $[f_2, 0, -f_2]$ . The minimum time duration is  $t_0 = 0.15$  time unit.

With the help of the method described in [14], the maximum velocities  $\dot{q}_1$  and  $\dot{q}_2$  versus displacement  $q_1$  and  $q_2$  are estimated and the areas of analysis in the state planes of both link 1 and link 2 are located. We have 379 plane cells in cell plane 1, with the cell length of 0.03735 along  $q_1$ , and 179 plane cells in cell plane 2, with the cell length of 0.03284 along  $q_2$ . A total of 67,841 combination pairs form the whole cell space. The value  $\alpha = 0.05$  ( $\text{sec}/(m - N)^2$ ) is used for the performance index to reflect the combined requirements of optimizing both time and control energy for the manipulator.

The optimal route describing the angular displacement  $q_1$  versus time is given in Figure 2. The horizontal time shift of character spacing is 0.15, which is also used for all the figures that follow. Every vertical position shift represents one cell length in  $q_1$ . Numbers are printed on the top line to indicate the time scale. In figures 3, 4, and 5, optimal  $q_2$ ,  $\dot{q}_1$ ,  $\dot{q}_2$  versus time are presented. The vertical shifts of the characters are in their own unit lengths. The rightmost locations of the character A in these 4 figures correspond to target values of  $q_1^*$ ,  $q_2^*$ ,  $\dot{q}_1^*$ , and  $\dot{q}_2^*$ , respectively. For the sake of compactness of the figures, several routes marked by different characters are put together to show the global characteristics of the optimal trajectories. Figure 6 gives the

time history of the optimal control torques applied to joint 1 and joint 2. Every vertical shift of character position corresponds to 0.5 for  $f_1$  and 0.05 for  $f_2$ .

Let us first study the parts of these figures that are symbolized by character A, which correspond to the trajectory starting from the initial stationary position. At the beginning, link 2 gains velocity due to the torques  $f_1 = -0.5$ , and  $f_2 = -0.05$  applied to joint 1 and joint 2, respectively, for a time duration of  $t_0$ . It takes  $4t_0$  period of time for link 2 to move to a new cell position, while in the same time period  $q_1$  and  $\dot{q}_1$  are so small that link 1 stays in the same cell of state plane 1. At the 6th time interval  $f_2 = 0.05$  is applied to joint 2. As a result velocity  $\dot{q}_2$  jumps to a higher level, and displacement  $q_2$  increases in the following time intervals. A sequence of non-negative torques are then applied to joint 2 in such a way that velocity  $\dot{q}_1$  remains in a very low level and the displacement  $q_1$  does not exceed the limitation imposed. It is obvious that positive  $f_2$  will cause negative  $\dot{q}_1$ , so positive  $f_1$  and  $f_2$  are applied to joint 1 and joint 2 simultaneously at time interval 14 to ensure that the cell location in plane 1 does not exceed the boundary. But positive  $f_1$  induces a drop in velocity  $\dot{q}_2$ . For adjustment a negative  $f_1$  and a positive  $f_2$  are applied at the next time interval. As a result,  $\dot{q}_2$  gets a big jump as shown in figure 5 at time interval 16. In order to speed up  $q_1$ , a sequence of  $f_1$  are then applied to joint 1, with which  $\dot{q}_2$  drops quickly while  $q_2$  increases steadily. At time interval 26,  $q_1$  is big enough to reach a new cell in plane 1, so does displacement  $q_1$  at time interval 29. In the following time intervals  $f_1 = 0.5$  is steadily applied to joint 1, consequently  $q_1$  increases constantly. Also,  $f_2 = 0.05$  is applied to joint 2 to balance the influence of  $f_1$  on link 2. Velocity  $\dot{q}_2$  stays in a low level because  $q_2$  is already very close to its target position. A series of complex  $f_1$  and  $f_2$  indicated in Figure 6 are then applied to the two joints to ensure the evolution of cell state to the target. Roughly speaking, in the first time section, displacement  $q_1$  and velocity  $\dot{q}_1$  are so small that link 1 almost does not move, but the displacement of link 2 increases continuously. In the second time section, link 1 has a positive, nearly constant velocity and its displacement increases constantly. The velocity of link 2 remains in a low level and its displacement does not increase much. In the third time section, a complex torque sequence is applied. Consequently, the displacements and velocities of both link 1 and link 2 quickly reach their corresponding target values. The total time needed to move by the manipulator from the starting position to the target is  $81t_0$ , about 12.15 time units.

The nature of the algorithm allows us to obtain a large number of globally optimal trajectories simultaneously. From Figure 2 to Figure 6, 25 different optimal trajectories and controls for 25 different initial states are given in a condensed form. For example, the optimal trajectory represented by character B has an initial state which corresponds to the combination pair [(2,1) (2,1)]. The state variables of this optimal trajectory move along the route described by character B according to Figure 2 to Figure 5 with the control sequence defined in Figure 6. In these figures character B does not appear whenever it coincides with the position of character A in the figures. Starting from time interval 32, the control torques and trajectory routes are the same as those symbolized by A. It takes  $76t_0$  for B to arrive at the target. The control sequence of B is different from that of A only in the beginning part, because the starting position of B is not on the boundary of the workspace. These figures imply that if the state of the system is driven to the combination pair [(2,1) (2,1)] from the optimal route symbolized by A due to disturbances, the route will no longer be optimal if it is forced immediately to join the trajectory symbolized by A. Instead, this route should join the optimal route of A at time  $31t_0$ . Likewise, suppose the system is driven to the combination pair [(5,5) (5,5)] from the optimal route of A because of some disturbances, the optimal route and control from this state will be those indicated by character O which joins A in the last few time steps. The computed results have given an optimal route and the corresponding control for every controllable cell state. The quantity of information obtained is so large that it is impossible to express them in a few figures. The optimal routes represented by the alphabets are printed in the order such that once a cell is printed with a character, it will not be printed with other

characters following it. The presentation here is aimed at showing the global nature of the obtained optimal trajectories. Much information has been suppressed in these figures because of the overlay effect, for example, in Figure 2 character O first appears at location 13 instead of 5. Only some control sequences are printed in Figure 6 due to space limitation. We have only presented less than 1 percent of total optimal trajectories obtained, since there are 59,750 controllable combination pairs.

It is true that the optimal trajectories obtained from the cell mapping method may be different, if different cell spaces are selected. But the influence of cell space selection can be reduced if we study the dynamics of the robotic manipulator carefully beforehand. Proper selection of sets U and T will be of great help for computation efficiency. In order to get good accuracy, small cell size is preferred, but increasing the total number of cells will increase the computation time rapidly. Cell refining techniques can be used if we want to obtain a specific optimal trajectory and study the characteristics of its vicinity in detail. Variable cell size techniques are attractive for those problems which require accurate position or velocity description only in some part of the trajectory (e.g. near the start and target locations). These methods will be developed in the future. As far as memory and computation efficiency are concerned, the free field problem takes most storage and is most time consuming. Data storage and computation time is reduced if there are some constraints in the workspace.

## 6. Example 2: Optimal trajectory planning in an obstacle constrained field

The same manipulator of Section 5 is used to demonstrate the use of the described cell mapping method for solving the constrained field problem. Suppose on its way to the target there is an obstacle shadow in the form of a sector which begins at  $\theta_1 = \pi/4$  and ends at  $\theta_2 = \pi/2$ , with  $r \geq r_0 = 2.4$  units of length. The constraint equations are:

$$\begin{aligned} l_1 \cos q_1 + l_2 \cos(q_1 + q_2) &= r_0 \cos \theta \\ l_1 \sin q_1 + l_2 \sin(q_1 + q_2) &= r_0 \sin \theta \end{aligned} \quad (6.1)$$

$$l_1^2 + l_2^2 + 2l_1 l_2 \cos q_2 \leq r_0^2 \quad (6.2)$$

From (6.1) we can solve  $q_{1a}$  and  $q_{2a}$ . They correspond to the starting point of the obstacle shadow area in the  $q_1$  and  $q_2$  directions.

Since  $q_2$  will not be monotonous on its whole route and  $\dot{q}_2$  can be negative in some regions, we have to extend the cell space and divide it into 3 partitions. Generally speaking, they correspond to 3 stages of manipulator motion: before entering the shadow zone, in the shadow zone, and after the shadow zone. In the cell plane 1 we still use 50 cells in the  $q_1$  direction and the total plane cell number is 379, as before. In the cell plane 2 the situations are drastically changed. The cell number in the  $q_2$  direction is increased to 66 instead of 24 because of the extension requirements. The total cell number in this plane is increased to 461, in which 340 are located in partition 1, 6 in partition 2, and 115 in partition 3. As the counterpart, plane 1 is also divided into 3 partitions. There are 51 plane cells in partition 1, 292 cells in partition 2, and 36 in partition 3. The total admissible combination pairs are 23,232, which is only one-third of that of the unconstrained case. Furthermore, there are about one-third of them uncontrollable because of violation of the constraints. Some results are listed from Figures 7 to 10. It takes 10840 to arrive at the target starting from the origin. Compared with the free field case, about one-third of time duration more is needed for the manipulator to execute the optimal trajectory for the same optimal performance index. By comparing Figures 2 and 3 with Figures 7 and 8, we see the "narrow corridor" phenomenon caused by the obstacle in partition 2. Many optimal routes starting from different states will join together, then pass through this corridor to satisfy the constraint. In partition 1 the behaviors of displacements  $q_1$  and  $q_2$  versus time are similar to those of unconstrained case. In partition 3 displacement  $q_2$  is decreasing and velocity  $\dot{q}_2$  is negative, as expected. As in the unconstrained case

we present in these figures only a very small percentage of optimal trajectories obtained.

## 7. Conclusions

A sound base has been founded for the application of cell mapping method for optimal trajectory planning of robotic manipulators. We have established main principles and techniques needed for efficiently determining optimal trajectories in the cell sense. The free field and obstacle constrained optimal trajectory planning problems are discussed, with some computational techniques developed. The method of extended cell space is proposed and successfully applied to static constraint problems. This method can be further developed to solve collision-free dynamic constraint problems.

## 8. Acknowledgement

The research was partially funded by the National Science Foundation under grant MSM 8796277 for support of a Presidential Young Investigator Award to M. C. Leu.

## References

- [1] Brooks, R. A., "Planning collision free motion for pick and place operation," *Inter. J. of Robotics Research*, Vol. 2, No. 4, 1983.
- [2] Lozano-Perez, T., "Spatial planning: a configuration space approach," *IEEE Transaction on Computers*, Vol. C-32, No. 2, 1983.
- [3] Hopcroft, J. E., Joseph, D., and Whitesides, S., "On the movement of robot arms in two dimensional bounded regions," TR 82-486, Dept of computer science, Cornell University, NY., 1983.
- [4] Schwartz, J. T. and Yap, C. K. "Advances in robotics Vol. 1 : Algorithmic and geometric aspects of robotics," Lawrence Erlbaum Associates, Hillsdale, NJ, 1987
- [5] Shin, K. G. and McKay, N.D., "A dynamic programming approach to trajectory planning of robotic manipulators," *IEEE Transactions on Automatic control*, Vol. AC-31, No. 6, 1986.
- [6] Singh, S. A. and Leu, M.C., "Optimal trajectory generation for robotic manipulators using dynamic programming," *Journal of Dynamic Systems, Measurement and Control*, Vol. 109, 1987.
- [7] Bobrow, J. E. and Dubowsky, S., "On the optimal control of robotic manipulators with actuator constraints," *Proc. American Control Conference*, San Francisco, 1983.
- [8] Dubowsky, S. and Norris, M. A., and Shiller, Z. "Time optimal trajectory planning for robotic manipulators with obstacle avoidance : a CAD approach," *Proc. IEEE Conference on Robotics and Automation*, Vol. 3, 1986.
- [9] Shiller, Z. and Dubowsky, S., "Global time optimal motion of robotic manipulators in the presence of obstacles," *Proceedings 1988 IEEE Int. Conf. on Robotics and Automation*, 1988.
- [10] Singh, S. K. and Leu, M. C. "Manipulator motion planning in the presence of obstacles and dynamic constraints," *Inter. J. of Robotics Research* (to appear)
- [11] Hsu, C. S., "A theory of cell-to-cell mapping dynamic systems," *J. of Appl. Mech.*, Vol. 47, 1980.
- [12] Hsu, C. S. "A discretized method of optimal control based on the cell state space concept," *J. Opt. Theo. Appl.*, Vol. 57, 1986.
- [13] Zhu, W. H., "An applied cell mapping method for optimal control systems," *J. of Opt. Theo. Appl.*, Vol. 60, No. 3, 1989.
- [14] Zhu, W.H. and Leu, M.C., "Optimal trajectory planning of robotic manipulators by cell state space approach," *Robotics Research Vol. 14*, 1989.
- [15] Fu, K. S., Gonzalez, R. C. and Lee, C. S. G., "Robotics: control, sensing, vision, and intelligence," McGraw-Hill Co., 1987.

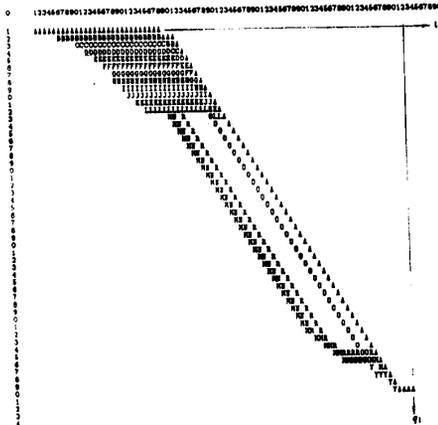


Figure 2 Displacement  $q_1$  vs. time  $t$  in the free field problem.

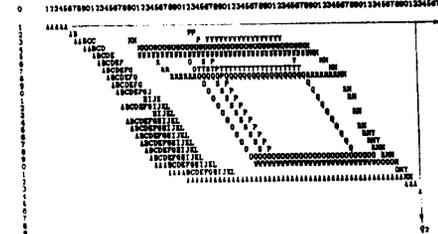


Figure 3 Displacement  $q_2$  vs. time  $t$  in the free field problem.



Figure 4 Velocity  $q_1$  vs. time  $t$  in the free field problem.



Figure 5 Velocity  $q_2$  vs. time  $t$  in the free field problem.

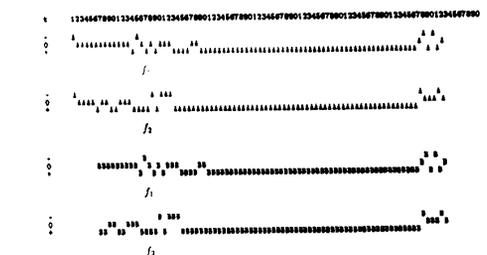


Figure 6 Control torques  $f_1$  and  $f_2$  vs. time  $t$  in the free field problem.



Figure 9 Velocity  $q_1$  vs. time  $t$  in the constraint case.

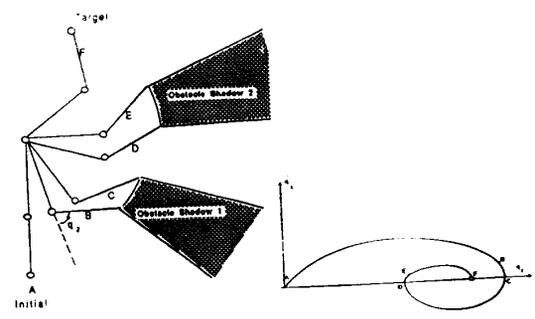


Figure 1 Effect of obstacle effects on state space trajectories.

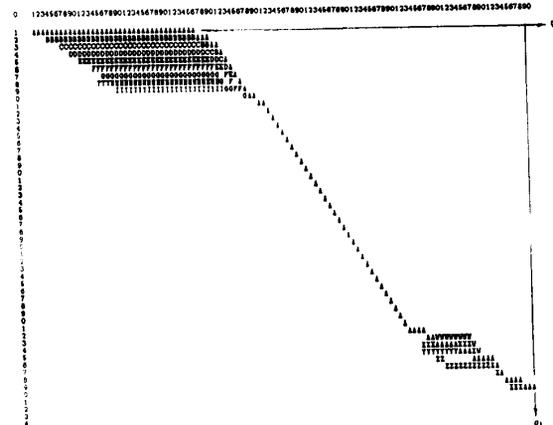


Figure 7 Displacement  $q_1$  vs. time  $t$  in the constraint case.

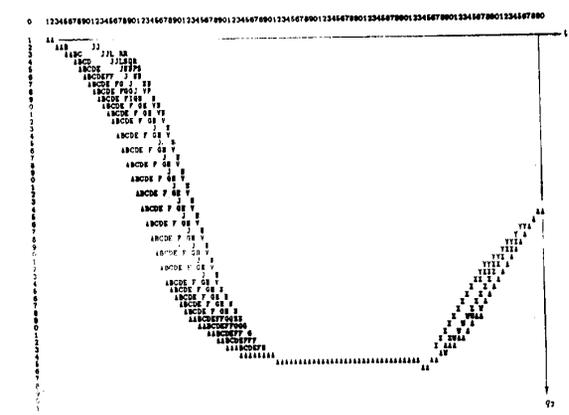


Figure 8 Displacement  $q_2$  vs. time  $t$  in the constraint case.



Figure 10 Velocity  $q_2$  vs. time  $t$  in the constraint case.