LOCATIONING OF TELEMANIPULATOR BASED ON TASK CAPABILITY

°Young Soo Park*, Jisup Yoon*, Hyung Suck Cho**

*R&D Group, NEMAC, Korea Atomic Energy Research Institute, Yusung P.O.Box 15, Taejon, R.O.K.
Tel: +82-42-868-2855; Fax: +82-42-861-4824; E-mail: yspark@nemacnis.kaeri.re.kr

**Dep. of Mechanical Engineering, KAIST, 373-1 Kusung-Dong, Yusung, Taejen, R.O.K.
Tel: +82-42-869-3213; Fax: +82-42-869-3210; E-mail: hscho@sorak.kaist.ac.kr

Abstracts This paper presents a time efficient method for determining a sequence of locations of a mobile manipulator that facilitates tracking of continuous path in cluttered environment. Given the task trajectory in the form of octree data structure, the algorithm performs characterization of task space and subsequent multistage optimization process to determine task feasible locations of the robot. Firstly, the collision free portion of the trajectory is determined and classified according to uniqueness domains of the inverse kinematics solutions. Then by implementing the extent of task feasible subspace into an optimization criteria, a multistage optimization problem is formulated to determines the task feasible locations of the mobile manipulator. The effectiveness of the proposed method is shown through a simulation study performed for a 3-d.o.f. manipulator with generic kinematic structure.

Keywords t-connectivity, octree, optimization by direct search, multistage optimization, mobile manipulator

1. INTRODUCTION

To successfully perform telerobotic tasks covering a large task area, it is required to frequently reposition the manipulator in the work area so that it has continuous access to the vicinity of the task trajectory. However, in clutter environments, such a continuous trajectory motion can generally be achieved only within a fractional domain of the free workspace, and the decision making on where and how to locate the manipulator requires complex geometric reasoning with complications beyond operator's intuition. Therefore, a powerful off-line planner must be at the aid of the operator.

Several researchers have developed an interest in the automatic positioning of manipulators mainly pertaining to the fixed automation of industrial processes. However, their approaches lacks the understanding of global task capability. Also, several robotics CAD systems are implemented with auto placement function which locates robotic manipulator in a position capable of tracking specified trajectories. They are characterized by a trial and error approaches based solely on collision checking. Currently, very little research results have been reported that takes into account the global task capability of the workspace.

Several authors have studied on the ability of a manipulator to travel through its workspace. Borrel and Legeois[1] have introduced the notion of aspects and revealed that they characterize trajectory following capability of most industrial robots in the absence of obstacles. Wenger and Chedmail[2] extended the scope of such characterization to robots with generic kinematic structures and taking into account the presence of obstacles. A few attempts have been made on application of such a topological analysis to the base locationing problem by adopting a direct search type of optimization method[3][4].

Carrying on with this stream of research efforts, this paper presents a method for determining a multiplicity of base locations that facilitates tracking of a long trajectory. The method entails a multistage optimization routine by concatenating individual derect optimization routines as each stage component. A global optimal solution is obtained through time efficient dynamic programming technique. In this method, the task trajectory is represented in octree data structures which is coherent with workspace characterization procedure, and computational efficiency of single stage optimization is improved by carrying out the workspace analysis in Cartesian space, rather than in configuration space. The effectiveness of the proposed method is illustrated for a 3-d.o.f. regional manipulator with generic kinematic structure.

2. SINGLE STAGE OPTIMIZATION

This section elucidates the application of a direct search type optimization method to determine an optimal location of robot base in terms of continuous trajectory capability.

2.1. Representation of theTask Environment

The task environment of a manipulator is composed of various Cartesian spaces occupied by task trajectories, obstacles, and robot body. The task trajectories are represented in octree code. Octree representation has certain merits over other spatial coding methods in that it is space efficient, capable of describing complex shpes and suitable for various set operations[5]. The obstacles and robot bodies are represented in generic geometric equations.

2.2. Characterization of Task Capability

Manipulator's task capabilities, in this paper, refers only to the kinematic capability. Based on workspace connectivity analysis, various task capabilities can be defined, such as the capability to perform continuous path tracking motion, called t-connectivity, and that to perform various point-to-point motions as described in Chedmail and Wenger[2]. The maximal t-connected regions of a manipulator workspace can be completely determined by an obstacle-workspace analysis described in Park and Cho[3]. However, such a full characterization is neither computationally feasible, nor essential for solving the current problem.

Thus, instead, in this paper, the task capability is tested and characterized only over the spatial occupancy of the task trajectories. Given an octree entity of the task trajectory, it performs collision tests on each octree cells encountered in ascnning through them in morton order. Only the collision free elements are stored and, moreover, the multiple inverse kinematics solution sets of a single cell are stored separately. Then the t-connected portions of the task trajectory, $CFtraj_{j,l}(\underline{x})$, is obtained by tracing the connected components of each database, thus

$$CFtraj_{j,l}(\underline{x}) = \underset{l}{CC}(Oct(Ftraj_j(\underline{x}))) \qquad (1)$$

where $Oct(.)$ denotes spatial volume represented in octree code, $Ftraj_j(\underline{x})$ is the collision free trajectory in jth uniqueness domain defined at base location $\underline{x}$, and $\underset{l}{CC}(.)$ denotes lth connected component of the octree inside the parenthesis.

2.3. Optimization by Direct Search

The problem addressed in this section is to find the optimum location of the robot's base in the work site in such a way that t-connectivity is guaranteed over the specified task trajectory. Due to the non-algebraic nature of the relevant process, a direct search type optimization method, called the simplex method, is adopted.

A cost function is designed in such a way that its maximization may naturally lead to the optimal base location, as

$$F(\underline{x}) = \max_{j,l} Area[CFtraj_{j,l}(\underline{x})] \qquad (2)$$

where the operator $Area(Oct(.))$ returns the volume occupied by the octree entity inside the parenthesis..

With the cost function defined as this, the optimization process is achieved through searching over the range of $\underline{x}$ to find the optimal set $\underline{x}^*$ that maximizes the cost function. Fig. 1 shows the overall structure of the base positioning algorithm, which properly takes into account the octree description of t-connected subspace as cost function.

3. MULTIPLE LOCATIONING

When a long trajectory has to be tracked, or the environment is severely cluttered, the manipulator may not be able to achieve t-connectivity over the entire trajectory at one location of the robot base. In this case, it is necessary to move the manipulator through a series of discrete locations to complete the task. This section presents an efficient multiple positioning scheme.

3.1. Problem Formulation

The multiple base locationing problem is defined as follows.

Problem Definition: Given a task in the form of a continuous trajectory with length considerably longer than the cross dimension of the manipulator's workspace, find a sequence of robot base positions that achieves complete coverage of t-connected subworkspace over the entire trajectory in a least number of allocations.

The general structure of a serial multistage system is shown in Fig. 2(a), in which single decision making stages are represented schematically by appropriately numbered rectangles, with arrows used to indicate inputs and outputs to the various stages[6]. Associated with the i_{th} stage is the stage return r_i, output state variables $\tilde{s}_i$, input stare variables s_i, and decision variables d_i. These variables are related as

$$r_i = R_i(d_i, s_i) \qquad (3)$$

$$\tilde{s}_i = T_i(d_i, s_i) \qquad (4)$$

By appropriately defining the relevant variables, the current problem can be formulated into a serial multistage system defined in the above. In the current method, the task trajectory is represented parametrically with the traveled length along the trajectory from a starting point t_0 to an end point t_f, which are appended to the octree codes. Then, for each optimization stage, i, they are defined as follows:

- Input state variable, s_i : This variable denotes the end points of the trajectory segments, $\overline{t_0, s_i}$, for which ith stage optimization is performed. It is determined from $\tilde{s}_{i-1}$ by incidence identity

$$s_i = \tilde{s}_{i-1} + \alpha \ ; \qquad i = 2, \ldots, N \qquad (5)$$

  where α is an arbitrarily chosen range of values denoting the amount of increment along the trajectory for the next stage optimization. In stage 1, the values of s_i are assigned to the grid points on an arbitrarily discretized range along the trajectory.
- Decision variable, d_i : This variable denotes the base locations, $\underline{x}$, of the manipulator. For a given s_i, the optimal decision, $d_i^*(s_i)$, is found by direct search technique of the previous section.
- Stage return, r_i : The area of the portion of the trajectory on which t-connectivity is attainable by the manipulator at a base location d_i,
- Output state variable, $\tilde{s}_i$: The most distant point from t_0 of the t-connected trajectory allocated by ith stage optimization. This sets the milestone for incrementing along the trajectory for the next stage.

Then, the output and transition equations (3) and (4) are redefined problem specifically as follows,

$$r_i = \max_{j,l} Area[CFtraj_{j,l}(d_i, s_i)] \qquad (6)$$

$$\tilde{s}_i = t\max[r_i]. \qquad (7)$$

where the function $t\max[]$ returns the largest value of the trajectory's length parameter, t, from the octree representation of a trajectory. With the variables and transition equations defined for each stage as above, the entire base positioning problem can be formulated by concatenation of each stage blocks as shown in Fig. 2.

3.2. Dynamic Programming

The system defined as in the above, the problem becomes a decision problem whose goal is to find the optimal sequence $d_1^*(s_N), \ldots, d_N^*(s_N)$ maximizing the total return R, defined by

$$R = \sum_{i=1}^{N} R_i(d_i, s_i) \qquad (8)$$

To simply the computational process let s_n be treated as a parameter, and define the n-stage maximum return $f_n(s_n)$ as

$$f_n(s_n) \geq \sum_{i=1}^{n} R_i(d_i, s_i) \quad \text{for all } d_1, \ldots, d_n; \quad n = 1, \ldots, N \qquad (9)$$

with equality achieved at each stage for at least one set of decisions. The function $f_n(s_n)$ can be made to depend on the input and the decision for the previous stage $n+1$ by

$$f_n(s_n) = f_n(T_{n+1}(d_{n+1}, s_{n+1})) \equiv f_n(d_{n+1}, s_{n+1}); \quad n = 1, \ldots, N-1 \qquad (10)$$

where the state s_{n+1} is regarded as a parameter. Therefore, the dynamic programming is described as the following recursive equation,

$$f_{n+1}(s_{n+1}) = \max_{d_{n+1}}[R_{n+1}(d_{n+1}, s_{n+1}) + f_n(d_{n+1}, s_{n+1})];$$
$$n = 1, \ldots, N-1. \quad (11)$$

Since the return $R_1(d_1, s_1)$ depends only on one decision variable, determination of $f_1(s_1)$ is a one-decision, one-state optimization problem. With $f_1(s_1)$ known, one can find $f_2(s_2)$ and $d_2^*(s_2)$ by another one-decision, one-state optimization. After N such optimization, one obtains the N decision functions $d_1^*(s_1), \ldots, d_n^*(s_N)$ and the N-stage maximum return $f_N(s_N)$.

The above dynamic programming procedure can be applied directly to the multiple base locationing problem. The goal is to find a series of base locations $d_i^* = \underline{x}_i$, $i = 1, .., N$ that maximize the each stage return that is the coverage of manipulator workspace over the task area. Since, in practice, all variables are available in the form of octree quantities, the recursive expression of *ith* stage maximum return given in (18) is rewritten as

$$f_i(s_i) = \begin{cases} \max_{d_i}[R_i(d_i, s_i) \cup f_{i-1}(s_{i-1})] & \text{if } R_i(d_i, s_i) \cup f_{i-1}(s_{i-1}) \neq \varnothing \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

The number of stages, N, is not fixed, and, instead it will be determined when the output state, $\tilde{s}_i$, reaches the end of the task trajectory, t_f.

To sequentially update the trajectory segment according to (23), it is necessary to modify the single stage direct search procedure of the previous section to a constrained optimization. A particularly relevant type of constraint condition for our purpose is the mandatory inclusion of a specified part of the trajectory element. This constraint is imposed by adding a penalty function to previous optimization function as follows

$$F_{opt} = \max_{j,l}[JArea(Oct(Qt_{jl}(\underline{x})) \cap Oct(Qtraj(\underline{x}))) + \lambda * JArea(Oct(Qt_{jl}(\underline{x})) \cap Oct(Qtraj(\underline{x})))] \quad (13)$$

where λ is a penalty constant.

4. SIMULATION STUDY

The proposed multiple positioning algorithm is tested for a task trajectory. The performance of the method is evaluated in terms of completeness of coverage and computational load.

4.1. Procedure

A 3 DOF regional manipulator with generic kinematic structure isused in the simulation study as shown in Fig. 3. It has 4 sets of unique inverse kinematics solutions. The manipulator, presumably mounted on a mobile base. The obstacle environment is as shown in Fig. 4. The trajectory lies among cluttered obstacle objects $O_1, \ldots, O_n$. In the world coordinate system, the task trajectory is defined as a connected series of straight line segments whose absolute coordinates of the end points are ((45.0, 0.0, 31.25), (25.0, 0.0, 31.25)), and ((25.0, 0.0, 31.25), (25.0, 0.0, 23.25)).

The dynamic programming is carried out as follows;

1) Assign the values for stage 1 input state variables: Since the maximum span of the manipulator is 4.5m, they are assigned to points along the trajectory within this distance away from t_0. Four grid points are selected at s_{11}=4.5m, s_{12}=4.0m, s_{13}=3.5m, and s_{14}=3.0m.
2) Starting from i=1, find stage i optimal base locations, $d_i^*(s_{i1})$, $d_i^*(s_{i2})$, $d_i^*(s_{i3})$, $d_i^*(s_{i4})$ through direct search with cost function described in (13).
3) For each single stage decision, $d_i = d_i^*(s_i)$, compute and store *ith* stage maximum return according to (12). Also, compute $\tilde{s}_i = (\tilde{s}_{i1}, \tilde{s}_{i2}, \tilde{s}_{i3}, \tilde{s}_{i4})$ according to (7).
4) If $\tilde{s}_i = t_f$ at any *i=N*, then retrace the sets of single stage optimal decisions for global optimal sequence of decisions, $(d_N^*, d_{N-1}^*, \ldots, d_1^*)$, and terminate the optimization process.
5) Otherwise, increment the trajectory segment according to (20), which is in practice implemented as
$$s_{i+1,1} = \max(s_i) + \alpha_{max} \quad (14a)$$
$$s_{i+1,4} = \max(s_i) + \alpha_{min} \quad (14b)$$
and $s_{i+1,2}$, $s_{i+1,3}$ are placed at even distance betwween them. For the current study, the offset parameters are set to be α_{max}=4.5m, and α_{min} = 3.5m.
6) increment i=i+1 and repeat from 2) for the next stage.

4.2. Result and Discussion

The resulting discrete sequence of base locations of the mobile manipulator is shown in Fig. 4. Simulation process can be summarized by writing the transient results in tabular form, although not shown here due to the space limitations. In the table, for each discrete value of s_i, the corresponding values of $d_i^*(s_i)$, $f_i(s_i)$, $\tilde{s}_i$ are recorded. Also, the last column indicates the transition to the previous stage. As a result, the final stage decision of $d_4^*(s_{42})$ is selected, and backtracking the transition tables the optimal policy is found to be d_4^* =(23.45, 25.40), d_3^* =(24.71, 30.74), d_2^* =(30.45, 30.75), d_1^* =(34.64, 30.76). The results show that complete coverage is achieved with 6 locations of base positions.

5. CONCLUSIONS

As an effort to relieve human operator from the complex kinematic reasoning in the planning stage of teleoperation tasks, an off-line method for selection a series of optimal locations for mobile manipulators is developed based on obstacle-workspace analysis. The base positioning method entails characterizing the manipulator's workspace based on task capability, and a subsequent optimization process to find task feasible base location that is capable of performing the given task specifications in the presence of obstacle. Chatarcerization of task capability is carried out directly in Cartesian space, by coding the task trajectory in octree and classifying the inverse kinematic solutions. To ensure the task compatibility over a broad and cluttered task environment, individual base positioning process is concatenated into a multiple decision making system, which can be solved with computational size increase linearly with the length of the task trajectory.

REFERENCES

1. P. Borel, and A. Legeois, 1986, A study of multiple manipulator inverse kinematic solution with applications to trajectory planning and workspace determination, Pr. of IEEE International Conference on Robotics and Automation, pp. 1180-1185
2. Ph. Wenger, and P. Chedmail, 1991, On the connectivity of manipulator free workspace, Journal of Robotic Systems, Vol. 8(6), pp. 767-799

3. F. Reynier, P. Chedmail, and Ph. Wenger, 1992, Automatic positioning of robots, continuous trajectories feasibility among obstacles, IEEE Int. Conf. on Syst. Man and Cybern. '92, Chicago, USA, pp. 189-194
4. Y. S. Park, J. S. Yoon, H. S. Cho, 1995, Robot Positioning Based on Workspace Connectivity, to appear in Proc. of ASME Annual Meetin
5. I. Gargantini, 1982, An effective way to represent quadtrees, Computer Graphics and Image Processing, Vol. 25, No. 12, pp. 905-910
6. R. E. Bellman, 1957, Dynamic Programming, Princeton Univ. Press, Princeton, N.J.
7. J. A. Nelder, and R. Mead, 1965, A simplex method for function minimization, Computer Journal, Vol. 7, pp. 308-313

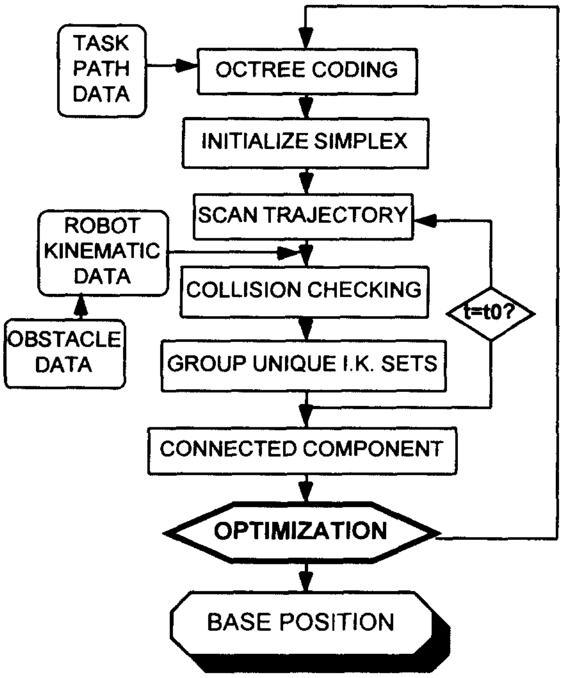
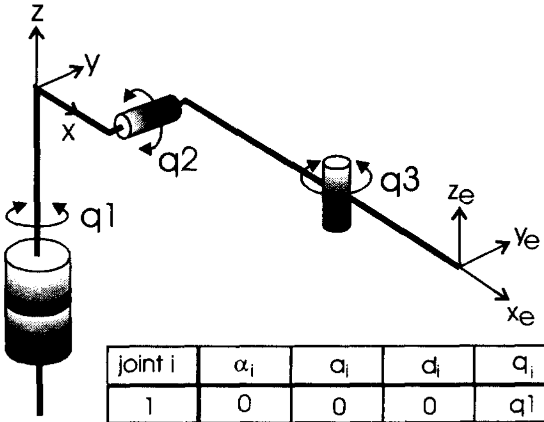


Figure 1. Structure of Base Positioning Algorithm



joint i	α_i	a_i	d_i	q_i
1	0	0	0	q1
2	-90	1	1	q2
3	90	2	0	q3
4	0	1.5	0	0

Figure 3. A 3 DOF Manipulatoror

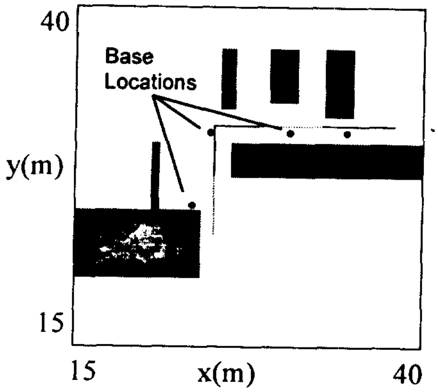


Figure 4. Simulation Result

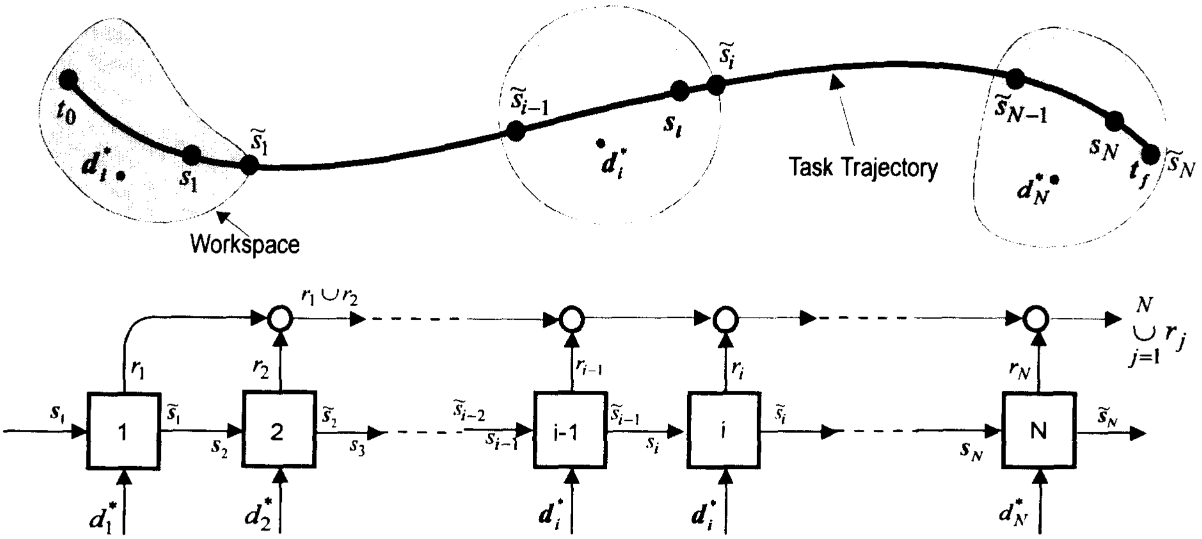


Figure 2. Definition of Variables for Multiple Base Locationing