Sequential Quadratic Programming based Global Path Re-Planner for a Mobile Manipulator

Sooyong Lee

Abstract: The mobile manipulator is expected to work in partially defined or unstructured environments. In our global/local approach to path planning, joint trajectories are generated for a desired Cartesian space path, designed by the global path planner. For a local path planner, inverse kinematics for a redundant system is used. Joint displacement limit for the manipulator links is considered in the motion planner. In an event of failure to obtain feasible trajectories, the task cannot be accomplished. At the point of failure, a deviation in the Cartesian space path is obtained and a *re-planner* gives a new path that would achieve the goal position. To calculate the deviation, a nonlinear optimization problem is formulated and solved by standard Sequential Quadratic Programming (SQP) method.

Keywords: Global path planner, local path planner, mobile manipulator, sequential quadratic programming.

1. INTRODUCTION

A mobile manipulator has a robot manipulator arm mounted atop a mobile platform. Advantages of mobile manipulator are infinite workspace and redundancy, thus gives more freedom and flexibility in control. The introduction of redundancy in the combined system increases the functionality manifold. The mobile manipulator has application in personal robots as well as service robots in various fields like factory automation, underwater exploration, robotic surgery, space robots and nuclear power plant maintenance. Furthermore, application of haptics interface to the mobile manipulator can achieve tasks that are potentially dangerous. This avoids human presence in the undesirable surroundings.

The basic expectations from a mobile manipulator, operating in a workspace, can be stated as follows

- Generate a trajectory from the initial to the goal position in Cartesian space.
- Generate physically realizable commands for the various joints.
- Avoid known and unknown obstacles.

Along with the progress in the field of robotics, the operating conditions for the robots have grown more and more complex. While performing the assigned tasks in dynamic environment, a mobile manipulator is bound to encounter static and/or moving obstacles. We plan the path for shunning an obstacle to avoid collision. The basic strategy for obstacle avoidance is to be as far from it as possible [1]. In this paper, we have sought to account for the base obstacles at the local planner. These are detected in real-time, by the sensors, as the mobile manipulator moves in the environment. It is assumed that the other obstacles are known while planning the global Cartesian space trajectory.

All practical manipulators have upper and lower limits for its joint displacements. These limits are due to the joint mechanism or due to the interference with the other links/parts of system. It is important to generate a set of joint trajectories for manipulator joints such that the joint limits are not violated, otherwise the command following is not possible.

The problem of motion planning for a manipulator has been widely studied, mostly motion planners are custom-made for a specific application and developing a path planner for general purpose is still a challenge. Generally, such approaches do not use the redundancy of the mobile manipulator, thus are less efficient ways of path planning for a mobile manipulator. Yamamoto and Yun [2] demonstrated the coordinated motion of a mobile manipulator using a feedback control algorithm. Inverse kinematics for a holonomic mobile manipulator along a geometric spline using pseudo-inverse of non-square Jacobian matrix is proposed in [3]. A non-holonomic path planner is proposed in [4] which generates a trajectory that achieves the goal position while avoiding the obstacles and the singular configurations for a two-

Manuscript received June 24, 2004; revised January 25, 2006; accepted March 29, 2006. Recommended by Editor Keum-Shik Hong. This work was supported by Korea Research Foundation Grant (KRF-2003-003-D00024).

Sooyong Lee is with the Department of Mechanical and System Design Engineering, Hongik University, 72-1 Sangsu-dong, Mapo-gu, Seoul 121-791, Korea (e-mail: sooyong@hongik.ac.kr).

link mobile manipulator. Mohri [5] used optimal controls to attain goal position, follow Cartesian space trajectory and avoid obstacles, in that order of priority, for a mobile manipulator path planning. They have implemented the algorithm for a two-link planar nonholonomic mobile manipulator.

In the following section, the structure of global/local path planner is described. Section 3 covers the Local path planner and the Global path planner is described in section 4. It also introduces the Global path Re-planner. Section 5 gives simulation results for the proposed algorithm with a three link mobile manipulator.

2. PATH PLANNER

Numerous path planners are proposed for mobile manipulator with holonomic or nonholonomic base [6,7]. In the global/local approach (Fig. 1), global path planning requires pre-conditioning. Such schemes may consider different optimizing criterions for the different parts of the path. It uses the available information about the workspace to decide on the motion plan. The global level planning is well suited for controls as a controller is used to stabilize the motion along a trajectory.

The local path planner generates joint commands to follow the path designed by the global path planner and it consists of the inverse kinematics routine for the mobile manipulator. A cost function can be used in the inverse kinematics for the unknown or dynamic obstacle avoidance in the local path planner. Such sensor-based schemes, which adapt to the dynamic environment by modifying its path are also called as dynamic path planners [8].

2.1. System definition

A three link mobile manipulator with two wheeled differentially driven mobile base is used for this paper. The local coordinate system for the mobile base is located at the center of the two driving wheels and the local coordinate system for the manipulator is the same. The joint angle for the manipulator θ_1, θ_2 and θ_3 are measured relative to the previous link orientation.

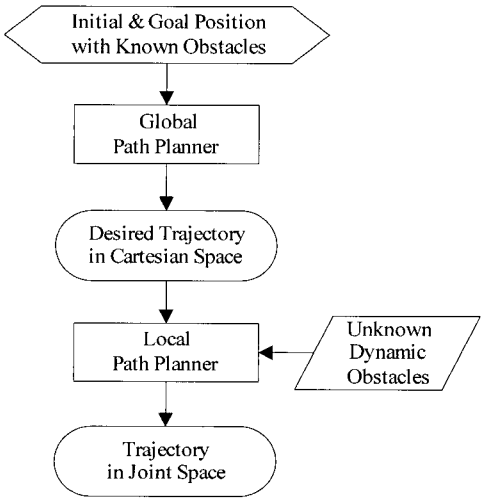


Fig. 1. The flow chart of the general global/local motion planner.

The following assumptions are made while considering the model for this system

- The slip between wheels and floor is negligible.
- Distance information is available from the sensors. Or the obstacle locations are known.
- The vehicle cannot move side-ways to maintain non-holonomic constraint.
- The mounting of the manipulator on the platform is such that the motion stability of the mobile manipulator is not an issue.
- Motion of mobile base is confined to plane of the floor.

A mobile manipulator system is kinematically described in the form of non-linear mapping of joint variables in the Cartesian space (1).

$$\mathbf{X} = f(\Theta), \tag{1}$$

where $\mathbf{X} \in R^m$ is the end effector position vector and $\Theta \in R^n$ is the joint angle vector. For our system, the joint angle vector, Θ, is composed of two wheels' angular displacements (θ_l, θ_r) and three joints' angular displacements $(\theta_1, \theta_2, \theta_3)$, thus $\mathbf{X} = [\theta_l, \theta_r, \theta_1, \theta_2, \theta_3]^T$ while the position vector is $[x, y, z]^T$. The relationship between Cartesian space velocity and the joint velocity is given by the Jacobian matrix as follows

$$\dot{\mathbf{X}} = J(\Theta)\dot{\Theta}, \tag{2}$$

where $\dot{\mathbf{X}} \in R^m$ is the Cartesian space velocity of the end-effector, $\dot{\Theta} \in R^n$ is the joint velocity vector. Jacobian matrix $J(\Theta) \in R^{m \times n}$ is a non-square matrix for a redundant system. The mobile manipulator being a redundant system gives multiple solutions for a desired Cartesian space position.

2.2. Structure of the path planner

The proposed path planner consists of three stages (Fig. 2): the first one is Global path planner which uses the consideration of universal topological features of the environment, and the second stage is the Local path planner which generates the joint trajectories that satisfy the kinematic and physical constraints. The third stage is Global Path *Re-Planner*, which is involved when a joint command is not physically implementable.

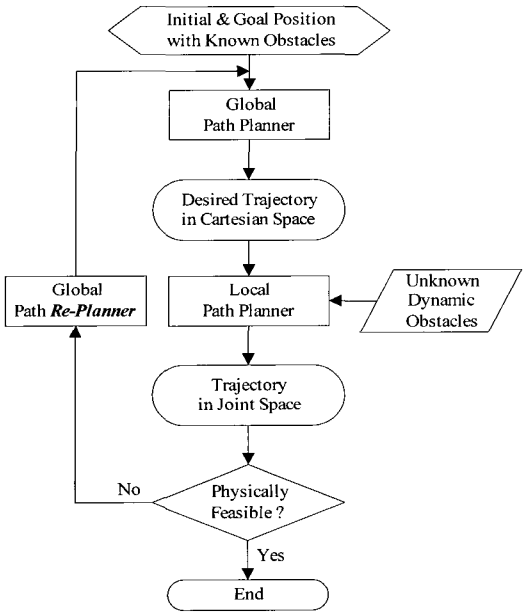


Fig. 2. The feedback structure of motion planner due to global re-planner.

3. LOCAL PATH PLANNER

Navigation of a mobile manipulator in a known or partially known environment or in a dynamic environment is done by a Local Path Planner. To accomplish a task in the 3-D world, a robotic system needs to be driven by joint velocity commands generated at regular time intervals.

A mobile manipulator being a redundant system has multiple solutions for the desired Cartesian position [9]. So it becomes hard to get a unique solution for the inverse kinematics and formulate a joint velocity trajectory. On the other hand, this has an advantage that we can select an optimal solution from the pool of multiple solutions based on some optimizing criterion. To make the planned trajectory practically realizable for the robotic system, many auxiliary criterions are to be satisfied, depending on the system definition and task requirements.

The inverse kinematics for a redundant system with only the scalar sub-task(s) can be stated as follows

$$d\Theta = J^{\#} J d\mathbf{X} + \left(I - J^{\#} J\right)\left(-k\frac{dP}{d\Theta}\right)^{T}, \tag{3}$$

where $J^{\#}$ is the pseudo-inverse of Jacobian matrix, I is the identity matrix of appropriate dimensions, P is the linear combination of all the *cost functions* used and k is the proportional constant. A cost function is the mathematical expression for the optimization criterion for the trajectory.

$$P = \sum_{j=1}^{m} \alpha_j p_j , \tag{4}$$

where α_j is the weighting factor for the respective cost function. The redundancy of the robotic system is used in the same priority order to satisfy each cost function. For joint limit constraints, the following cost function is used for each joint.

$$\theta_{j,\min} \le \theta_j \le \theta_{j,\max} , \tag{5}$$

$$p_j\left(\theta_j, \theta_{j,\min}, \theta_{j,\max}\right) = \tan\left(\frac{\pi\left(\theta_j - \theta_{j,\min}\right)}{2\left(\theta_{j,\max} - \theta_{j,\min}\right)}\right) + \tan\left(\frac{\pi\left(\theta_{j,\max} - \theta_j\right)}{2\left(\theta_{j,\max} - \theta_{j,\min}\right)}\right). \tag{6}$$

In some practical applications, assigning priorities to the subtask helps in the path implementation. A modified version of inverse kinematics (3) is used to achieve this. It sets the order in which the redundancy of the system is used to satisfy the auxiliary criteria. The proof and algorithm of the inverse kinematics considering the order of priority can be found in [10]. For the trajectory generation with two subtasks and different priorities, the inverse kinematics is given by (7).

$$d\Theta = J^{\#} d\mathbf{X}_a + \tilde{J}_b^{\#}\left(d\mathbf{X}_b - J_b J^{\#} d\mathbf{X}_a\right) + \left(I - J^{\#} J\right)\left(I - \tilde{J}_b^{\#}\tilde{J}_b\right)\mathbf{z} , \tag{7}$$

where

$$\tilde{J}_b = J_b\left(I - J^{\#} J\right) \tag{8}$$

and $\mathbf{X}_a \in R^{m_a}$, $\mathbf{X}_b \in R^{m_b}$, $m_a + m_b = m$. $\mathbf{z} \in R^n$ are in decreasing order of priority. $J_b \in R^{m_b \times n}$ is defined as Jacobian for the second order priority sub-task (9).

$$\dot{\mathbf{X}}_b = J_b(\Theta)\dot{\Theta} \tag{9}$$

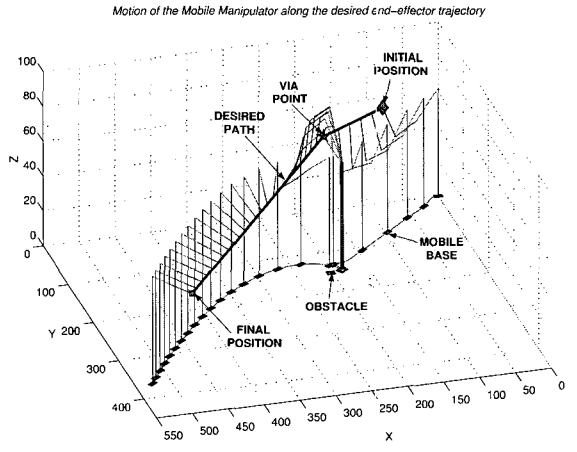


Fig. 3. Trajectory planning for 4-link mobile manipulator.

When a redundant mobile manipulator system is moving in a Cartesian space having obstacle(s), it becomes important to generate joint velocity trajectories while avoiding collision and maintaining joint displacements within physical limits of the manipulator arm. The desired path following is used as the first priority task, the obstacle avoidance as the subtask with the second priority while the joint limit constraint is assigned as the last priority.

Fig. 3 shows an example of a 4-link mobile manipulator path planning. The end point of the manipulator follows the desired trajectory while the base is avoiding collision with obstacles. The joint limit constraints are also satisfied.

4. GLOBAL PATH PLANNER

The global path planner serves a dual purpose of being an initial Cartesian Space trajectory planner and a *re-planner* in case of a convergence failure at the local path planner level. The failure to satisfy the obstacle avoidance or joint limits will render the joint trajectories to be practically not implementable.

2.1. Global path re-planner

If the local path planner fails to give a joint trajectory that satisfies all the criterions, then the Global path planner is asked to *re-plan* the Cartesian space trajectory, considering the task requirements, from the point of failure to the goal position. The local path planner, along with failure notice and the type of criterion failed, can provide feedback to the global path planner to guide it in *re-planning*. This helps in making a intelligent *guess* of the deviation in the trajectory that will satisfy the criterions.

The use of priority-based inverse kinematics (7) ensures that joint limit constraints are violated before the obstacle avoidance has failed. Thus, the violation of any joint limits acts as a trigger for the global path *re-planner*. The Jacobian matrix that gives the relation between Cartesian space velocity and joint velocities (2) can aid in decision making. The elements of this Jacobian Matrix can be considered as *sensitivity* functions of the global variables with respect to the joint variables.

$$J_{i,j} = \frac{\partial f_i}{\partial \theta_j} \simeq \frac{\Delta f_i}{\Delta \theta_j} \tag{10}$$

(10) can be used in case of joint limit violations. If j^{th} joint displacement constraints is violated at $(k+1)^{th}$ iteration, then it should be modified such that it is within joint limits

$$\Delta\theta_j = \theta_j^*(k+1) - \theta_j(k). \tag{11}$$

(10) gives Δf_i which indicates the change in the i^{th} global variable required to get a trajectory that avoids the violation of j^{th} joint limit. The value of $\theta_j^*(k+1)$ is picked freely subject to the following constraint

$$\theta_{j,\min} \le \theta_j^*(k+1) \le \theta_{j,\max}. \tag{12}$$

Additional constraints to be considered are avoiding the violation of the other joint limits, due to the modification in the Cartesian space trajectory. It is desired to have as little deviation in the Cartesian space trajectory as possible, along one axis in Cartesian coordinates. The constrained optimization problem is formulated as follows

$$\min_{\theta_j^*} \|\Delta \mathbf{X}_i\|, \tag{13}$$

and

$$\Delta \mathbf{X}_i = J_{i,j}\left(\theta_j^*(k+1) - \theta_j(k)\right), \tag{14}$$

subject to

$$\theta_{j,\min} \le \theta_j^*(k+1) \le \theta_{j,\max},$$

where $j = 1,\ldots,N : N$ is number of joints in the manipulator. Here we minimize the absolute value of the deviation from the desired trajectory such that all the new joint values are within joint limit constraints. The algorithm for the mobile manipulator is summarized as follows (Table 1).

Each term of the Jacobian matrix for a three-link manipulator is nonlinear, so the problem defined by (13, 14) becomes a nonlinear minimization problem.

Table 1. Pseudo-code for global path re-planner.

1. Generate the Cartesian space trajectory from the initial position to the goal position through via-point
2. Discretize the path
3. Solve the inverse kinematics to obtain joint trajectories while taking care of obstacle avoidance and joint limits using (7)
4. If the joint limit is violated
   a. Pick a Jacobian element corresponding to the joint angle that violated the joint limit
   b. Optimize the problem defined above to obtain the desired deviation in the end-effector path
   c. Plan the trajectory from the deviated point to the next via or goal position
   d. Go to the Step 2
5. Solve the forward kinematics to get the re-planned end-effector trajectory
6. End if the Goal position is reached

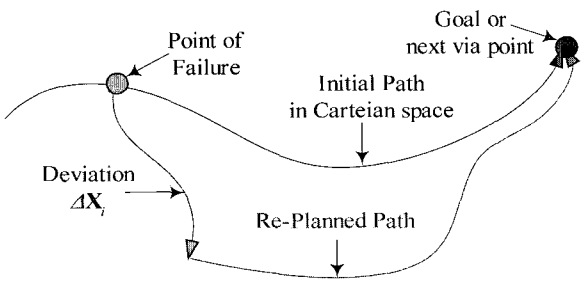


Fig. 4. Role of global *re-planner.*

There are many well-defined methods for getting a solution for it, such as the Newton-Lagrange method, Sequential Quadratic Programming and Multiplier Penalty method. The general strategy to approach such problem is to formulate it as Quadratic Programming problem using a Lagrange function.

4.2. Sequential quadratic programming

The solution of the constrained optimization problem (13,14) can be obtained by nonlinear programming methods, such as Sequential Quadratic Programming (SQP). In SQP, a Quadratic Programming (QP) problem is formulated and solved at each high-level iteration. Details on SQP can be obtained from [11-13]. The general method for solving constrained optimization problem is stated in [14].

The sequential quadratic programming finds a step away from the current point by minimizing a quadratic model of the problem, so this algorithm is considered as a generalization of Newton's method for unconstrained optimization. The efficiency, accuracy and chances of getting a solution by using SQP are tried and tested against other standard algorithms in [15].

A Lagrangian function is defined as:

$$L(\mathbf{\Theta}, \mathbf{\Lambda}) = \|\Delta \mathbf{X}_i\| + \sum_{i=1}^{M} \lambda_i c_i(\mathbf{\Theta}), \tag{15}$$

where M is the number of constraint functions, and λ_i are the Lagrange multipliers. M constraints on the problem are expressed as

$$c_i(\mathbf{\Theta}) \le 0. \tag{16}$$

The solution of the constrained optimization problem by SQP is carried out iteratively, in three steps (Table 2). If any of the new set of Lagrange multipliers is negative, solution to the constrained optimization (13) is obtained.

4. SIMULATION RESULTS

In order to verify the proposed algorithm, simulations are carried out for the three link mobile manipulator.

Table 2. Optimization with constraints by SQP.

2. Update the Hessian matrix (17) of the Lagrangian function.

$$H(m,n) = \frac{\partial^2 L}{\partial \theta_m \partial \theta_n} \tag{17}$$

As actual differentiation of Hessian matrix can be computationally intensive, quasi-Newton method is used. It starts with an initial guess and uses the Hessian matrix from previous step to update it.

3. Formulate a Quadratic Programming (QP) problem (18) using the updated Hessian matrix and solve it.

$$\min_d \frac{1}{2} d^T H d + \nabla f^T(x(k)) d, \tag{18}$$

where d is the direction in which the line search is performed and ∇f is the gradient function

4. Use d, the solution of QP, as search direction vector and perform the line search.

$$x(k+1) = x(k) + \alpha_k d, \tag{19}$$

where α_k is the step of iteration.

The position of the endeffector is derived from (21,22,23) assuming zero initial condition,

$$\Phi = \frac{r(\theta_r - \theta_l)}{b}, \tag{20}$$

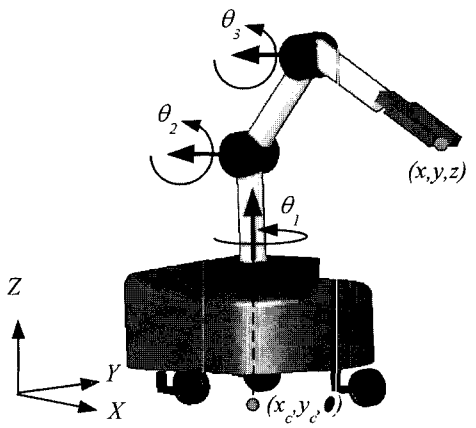


Fig. 5. 3-Link mobile manipulator.

Table 3. Kinametic parameters.

Parameter	Value
radius of wheels, r	0.058[m]
distance between wheels, b	0.23495[m]
link 1 length, L_1	0.5[m]
link 2 length, L_2	0.35[m]
link 3 length, L_3	0.3[m]
base height (offset in z-axis), z_o	0.12[m]

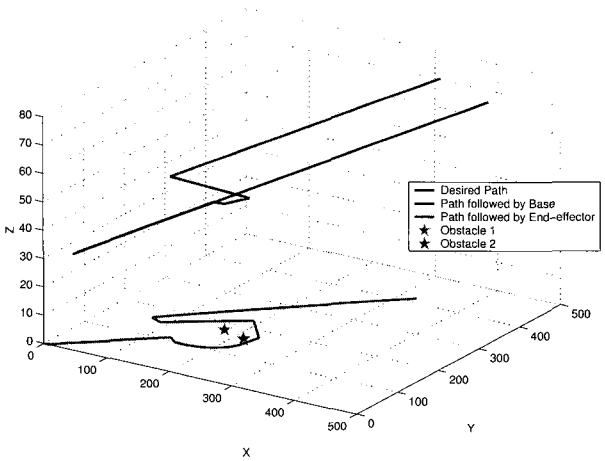


Fig. 6. Failure in path following by a mobile manipulator.

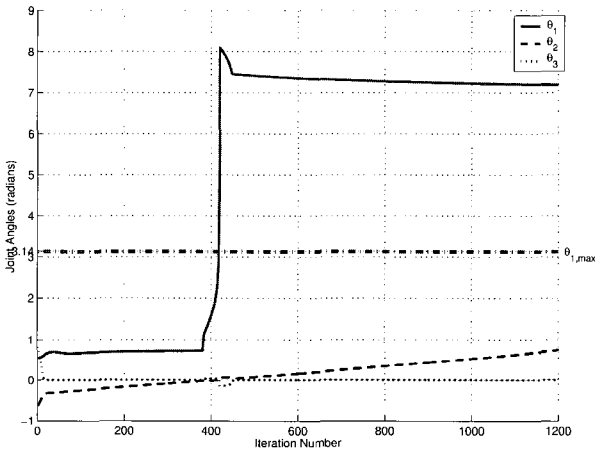


Fig. 7. Failed joint trajectory for a mobile manipulator.

$$x = \frac{r(\theta_r - \theta_l)\cos\Phi}{2} + L_2\cos\theta_2\cos\theta_1 + L_3\cos(\theta_2+\theta_3)\cos\theta_1, \quad (21)$$

$$y = \frac{r(\theta_r - \theta_l)\sin\Phi}{2} + L_2\cos\theta_2\sin\theta_1 + L_3\cos(\theta_2+\theta_3)\sin\theta_1, \quad (22)$$

$$z = z_o + L_1 + L_2\sin\theta_2 + L_3\sin(\theta_2+\theta_3), \quad (23)$$

where Φ is the orientation of the base.

A simplified Global Path Planner gives a straight line path from initial configuration to the goal position. If *via* points are given, they are sequentially connected by straight line paths. In this case, avoiding the point-obstacles is the second priority subtask and maintaining the joints within the limits is the third priority subtask. Fig. 7 shows a case in which two obstacles are very close to the path that would be followed by the base in order to successfully trace the straight line path. This results in joint 1 violating the maximum limit (Fig. 8) and the goal is not achieved

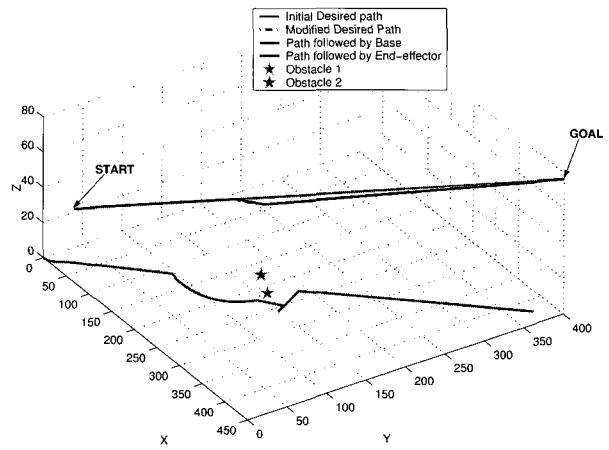


Fig. 8. Path followed by mobile manipulator using optimization.

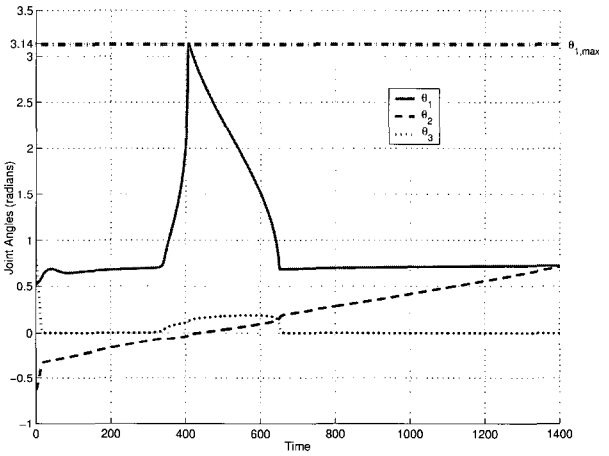


Fig. 9. Joint trajectory for mobile manipulator using optimization.

because the mobile manipulator goes into boundary singular configuration hence causing large error in inverse kinematics. Thus it is imperative to avoid joint limit violation.

To avoid the failure, a Global path re-planner is used. It gives a small deviation in the path and *re-plans* it till the goal position is reached (Fig. 6).

For the re-planner, the element of Jacobian matrix corresponding to Joint 1 and x-axis (*i.e.* $J_{1,1}$) is used. Although the *re-planner* suggests a small correction in x direction, subsequent iterations of inverse kinematics shows some deviation in y and z direction, too. This is due to the change in the $\boldsymbol{\Theta}$ vector, which changes the Jacobian matrix and induces deviations in y and z directions also.

5. CONCLUSIONS

Path planning is an important issue in the control of a mobile manipulator. It decides the Cartesian space path and generates the joint trajectories. These joint trajectories are used to drive the mobile manipulator.

The redundancy of the mobile manipulator is used to accomplish the auxiliary criterions in an order of priority. The Global Path Re-planner successfully avoids the violation of joint constraints and gives a path that reaches the goal position. Sequential Quadratic Programming method is used to sole a non-linear optimization problem.

APPENDIX

List of key symbols

$\mathbf{X}$	end-effector position vector
m	dimension of X
$\mathbf{\Theta}$	joint angle vector
Φ	orientation of the base
n	dimension of $\mathbf{\Theta}$
J	Jacobian matrix
J_b	Jacobian for the second order priority sub-task
$J_{i,j}$	i^{th} row, j^{th} column element of the Jacobian matrix J
k	proportional constant
I	identity matrix
P	linear combination of all the cost functions
p_j	j^{th} cost function
α_j	weighting factor for the j^{th} cost function
θ_j	j^{th} joint angle
$\theta_{j,\min}$	lower limit of the j^{th} joint
$\theta_{j,\max}$	upper limit of the j^{th} joint
N	number of joints of the manipulator
M	number of constraint functions
L	Lagrangian
c_i	i^{th} constraints
H	Hessian matrix
d	direction in which the line search is performed

REFERENCES

[1] Y. Shan and Y. Koren, "Obstacle accommodation motion planning," *IEEE Trans. on Robotics and Automation*, vol. 11, no. 1, pp. 36-49, 1995.

[2] Y. Yamamoto and X. Yun, "Control of mobile manipulators following a moving surface," *Proc. of IEEE International Conference on Intelligent Robots and Systems*, vol. 3, pp. 1-6, 1993.

[3] C. Altafini, "Inverse kinematics along a geometric spline for a holonomic mobile manipulator," *Proc. of IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1265-1270, 2001.

[4] H. G. Tanner and K. J. Kyriakopoulos, "Nonholonomic motion planning for mobile manipulators," *Proc. of IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1233-1238, 2000.

[5] A. Mohri, S. Furuno, and M. Yamamoto, "Trajectory planning of mobile manipulator with end-effector's specified path," *Proc. of IEEE International Conference on Intelligent Robots and Systems*, vol. 4, pp. 2264-2269, 2001.

[6] J. C. Latombe, *Robot Motion Planning*, Kluwer Academic Publishers, Boston, Massachusetts, 1991.

[7] S. G. Hong, S. W. Kim, K. B. Park, and J. J. Lee, "Local motion planner for nonholonomic mobile robots in the presence of the unknown obstacles," *Proc. of IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1212-1217, 1996.

[8] J. S. Zelek and M. D. Levine, "Local-global concurrent path planning and execution systems," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 30, no. 6, pp. 865-870, 2000.

[9] J. Craig, *Introduction to Robotics: Mechanics and Control*, Third edition, Prentice Hall, 2003.

[10] Y. Nakamura, *Advanced Robotics: Redundancy and Optimization*, Addison-Wesley Publishing Company, Inc., 1991.

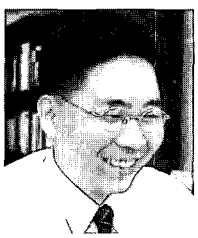
[11] M. C. Biggs, "Constrained minimization using recursive quadrative programming," *Towards Global Optimization*, vol. 1, pp. 341-349, 1975.

[12] S. P. Han, "A globally convergent method for nonlinear programming," *Optimization Theory and Applications*, vol. 22, pp. 297-303, 1977.

[13] M. J. D. Powell, "Variable metric methods for constrained optimization," *Mathematical Programming: The State of the Art*, Springer Verlag, pp. 288-311, 1983.

[14] R. Fletcher, *Practical Methods of Optimization*, 2nd ed., John Wiley & Sons, Inc., New York, 1987.

[15] K. Schittkowski, "NLQPL: A fortran-subroutine solving constrained nonlinear programming problems," *Annals of Operations Research*, vol. 5, pp. 485-500, 1985.



Sooyong Lee received the B.S. and M.S. degrees in Mechanical Engineering from Seoul National University, Seoul, Korea in 1989, and 1991, respectively, and the Ph.D degree from MIT, Cambridge, MA, in 1996. He worked as a Senior Research Scientist at KIST and then as an Assistant Professor in the Department of Mechanical Engineering at Texas A&M University. He joined Hongik University, Seoul, Korea in 2003 and is currently an Associate Professor in the Mechanical and System Design Engineering Department. His current research includes mobile robot localization and navigation, and active sensing.