[12] I. Masaki, R. R. Gorman, and B. H. Shulman, "Arc welding robot with vision," in *Proc. 11th Int. Symp. Industrial Robots*, 1981, pp. 813-817.
[13] R. W. Richardson, D. A. Gutow, R. A. Anderson, and D. F. Farson, "Coaxial arc pool viewing for process of monitoring and control," *Welding J.*, pp. 43-50, Mar. 1984.
[14] R. W. Richardson and R. A. Anderson, "Weld butt joint tracking with a coaxial viewer based weld vision system," in *Control of Manufacturing Processes and Robotic Systems*, D. E. Hardth and W. J. Book, Eds. New York: Amer. Soc. Mechanical Engineers, 1983.
[15] R. D. Richardson and. R. W. Richardson, "The measurement of two dimensional arc weld pool geometry by image analysis," in *Control of Manufacturing Processes and Robotic Systems*, D. E. Hardth and W. J. Book, Eds. New York: Amer. Soc. Mechanical Engineers, 1983.
[16] G. A. Slavin, "The planning of a multifactorial experiment for determining weld pool geometry and its influence on weld structure in the pulsed arc welding of heat resistant materials," *Automat. Welding*, vol. 27, no. 9, 1974.
[17] A. L. Yuille, D. S. Cohen, and P. W. Hallinan, "Feature extraction from faces using deformable templates," in *Proc. IEEE Conf. Computer Vision Pattern Recognition* (San Diego, CA), 1989.

Path Planning for Mobile Manipulators for Multiple Task Execution

Wayne F. Carriker, Pradeep K. Khosla, and Bruce H. Krogh

Abstract—**We consider the planning problem for a mobile manipulator system that must perform a sequence of tasks defined by position, orientation, force, and moment vectors at the end-effector. Each task can be performed in multiple configurations due to the redundancy introduced by mobility. We formulate the planning problem as an optimization problem in which the decision variables for mobility (base position) are separated from the manipulator joint angles in the cost function. The resulting numerical problem is nonlinear with nonconvex, unconnected feasible regions in the decision space. Simulated annealing is proposed as a general solution method for obtaining near-optimal results. The problem formulation and numerical solution by simulated annealing are illustrated for a manipulator system with three degrees of freedom mounted on a base with two degrees of freedom. These results are compared with results obtained by conventional nonlinear programming techniques customized for the particular example system.**

I. Introduction

Mobile manipulator systems are currently being proposed and developed for a variety of hazardous applications including planetary exploration, nuclear reactor maintenance, and construction [1], [2]. In these applications a common task description involves acquiring and moving material from one point to another over a large space that could not possibly be accomplished by a stationary manipulator. Thus, mobility is essential for accomplishing the task, and it is desirable to plan efficient point-to-point trajectories. The planning problem is complicated by task requirements that impose constraints on the system configurations for acquiring and depositing material.

The introduction of mobility creates redundancy that can be exploited to achieve a broader range of tasks. Although redundancy for manipulator systems has been studied extensively [3]-[9], little work has been done on formulating and solving task planning problems involving redundancy arising from mobility. A mobile system is capable of performing a number of tasks in widely separated locations. Thus, rather than using the redundancy to achieve optimal manipulator configurations for each task, one can use it to solve a more global optimization problem by finding optimal base trajectories for performing a sequence of tasks.

Our approach to the mobile manipulator path planning problem is to formulate a general optimization problem in which the degrees of freedom (DOF) introduced by mobility are separated from the manipulator motion in the cost function. The objective is to determine the best end-point positions and configurations for point-to-point motions with specified task requirements at each end of the trajectory. This leads to a global optimization problem with nonlinear and nonconvex constraints. Further, the problem may contain unconnected feasible regions in the optimization space. As an example, and for ease of illustration, we consider a system that consists of a 3-DOF manipulator mounted on a 2-DOF base, thus making the system redundant for tasks involving the positioning of the end-effector in three-dimensional Cartesian space.

Given the complex nature of the optimization problem, the solution obtained with a standard gradient descent optimization technique will depend on the initial guess. Our goal is to develop a more robust solution method. Since there has been limited success in the field of general global optimization, we consider the method of simulated annealing that has been effective in generating nearly optimal results for complex problems [10]-[12]. Simulated annealing is based on a numerical technique that was proposed in 1953 by Metropolis *et al.* [13]. We develop an algorithm based on this original work and compare the results of this technique with those from our earlier work using standard numerical optimization [14]. We then consider a more complex problem and examine the effectiveness of the simulated annealing technique.

The remainder of the paper is organized as follows. We begin in Section II by looking at the general problem of combining an n-DOF manipulator on an $m(m \leq 6)$ DOF freedom base. Here we consider nonredundant serial link manipulators, limiting n to six as well. In Section III, we examine the difficulties of solving the global optimization problem using standard gradient-based numerical techniques. We also examine the limitations of practical implementations of global optimization techniques. We then introduce the simulated annealing technique as an alternative method for obtaining near-optimal solutions for the mobile manipulator planning problem. In Section IV, we describe an example 5-DOF mobile manipulator system. We use a simple cost function that involves only the task base locations and task manipulator configurations. Then, in Section V we present numerical results from three example problems. The results of this work are summarized in Section VI.

II. Problem Formulation

We consider a mobile manipulator system consisting of a mobile base, controlled by a set of actuators that allow up to a full 6-DOF, and a serial link manipulator with at most 6-DOF performing a

Manuscript received March 23, 1990; revised December 1, 1990. This work was supported in part by an Engineering Creativity Award from the National Science Foundation under Grant ENG-8811580, by the U.S. Army AMCOM and DARPA under Contract DAAA-2189-C-0001, and by the National Science Foundation under Grants ECS-8404607 and CDA-8820575. This paper was presented at the IEEE International Conference on Robotics and Automation, Cincinnati, OH, May 13-18, 1990.

The authors are with the Laboratory for Automated Systems and Information Processing, Department of Electrical and Computer Engineering, Carnegie-Mellon University, Pittsburgh, PA 15213.

IEEE Log Number 9143687.

1042-296X/91/0600-0403$01.00 © 1991 IEEE

number of tasks within some workspace. The tasks of interest are described by their locations and the application of a desired force. The problem is to determine an optimal trajectory for the mobile manipulator system such that all the tasks can be performed given the constraints imposed by the physical limitations of the system.

Each task is defined by a desired position and orientation with respect to a global, inertial reference frame $\{I\}$, and a desired force and moment application. Without loss of generality, we assume that the force and moment vectors are also specified with respect to the frame $\{I\}$. Thus, the task definition is given as a 4×4 transformation matrix, ${}^{I}T_{T}$ and a vector $F_T \in R^{6\times 1}$ where

$$ {}^{I}T_{T} = \begin{bmatrix} n & o & a & p \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ F_T = \begin{bmatrix} f_x & f_y & f_z & m_x & m_y & m_z \end{bmatrix}^T \tag{1} $$

with the 3×3 rotation matrix $[n\ o\ a]$ and the vector p describing the orientation and position of the task frame and the vectors $[f_x\ f_y\ f_z]$ and $[m_x\ m_y\ m_z]$ describing forces and moments, respectively. A *task set* is then defined by an ordered collection of j tasks. That is,

$$ S = \{\{{}^{I}T_1, F_1\}, \{{}^{I}T_2, F_2\}, \ldots, \{{}^{I}T_j, F_j\}\}. \tag{2} $$

The mobile manipulator system is defined as a mobile base with m actuators producing at most 6-DOF and an n-DOF serial link manipulator. In the case of a wheeled mobile robot, the base can have at most 3-DOF representing linear translation and rotation in the plane. The position and orientation of the mobile base with respect to a global, inertial reference frame can be described by a 4×4 homogeneous transformation matrix ${}^{I}T_{B}$ of the form

$$ {}^{I}T_{B} = \begin{bmatrix} n & o & a & p \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3} $$

where n, o, a, and p are vectors representing the normal, orientation, approach, and position vectors of the base, respectively. The position and orientation of the base can be determined either by dead reckoning techniques [15] or by using on-board sensors [16], [17].

The position and orientation of the manipulator end-effector with respect to the base can also be represented by the 4×4 transformation matrix ${}^{B}T_{E}$, of the same form as ${}^{I}T_{B}$. The value of this matrix depends only on the values of the manipulator joint variable vector, $\theta \in R^{n\times 1}$. The position and orientation of the end-effector with respect to the global reference frame ${}^{I}T_{E}$ can then be expressed as follows:

$$ {}^{I}T_{E} = {}^{I}T_{B} \times {}^{B}T_{E}. \tag{4} $$

The motions of the mobile base and manipulator in Cartesian space are related to the motions of their actuators by system Jacobians such that

$$ \dot{x}_b = J_b(\phi)\dot{\phi} $$

$$ \dot{x}_m = J_m(\theta)\dot{\theta}. \tag{5} $$

The Jacobians also relate the forces and torques applied by the system to the forces and torques applied by each actuator [6], [7] such that

$$ \tau_b = J_b^T \times F_b $$

$$ \tau_m = J_m^T \times F_m \tag{6} $$

where $\dot{x}_b \in R^{6\times 1}$ and $\dot{x}_m \in R^{6\times 1}$ are the velocities of the base and manipulator end-effector in Cartesian space, J_b and J_m are the base and manipulator Jacobians with respect to the inertial reference frame $\{I\}$, $\dot{\phi}$ and $\dot{\theta}$ are the base and manipulator actuator velocities, $\tau_b \in R^{m\times 1}$ and $\tau_m \in R^{n\times 1}$ are the torques or forces generated by the base and manipulator actuators, and $F_b \in R^{6\times 1}$ and $F_m \in R^{6\times 1}$ are the base and manipulator forces and moments, respectively.

Performing all of calculations with respect to a common frame of reference, we can write

$$ \dot{x}_s = \dot{x}_b + \dot{x}_m $$

$$ \Phi = \begin{bmatrix} \theta^T & \phi^T \end{bmatrix}^T $$

$$ J_s(\Phi) = \begin{bmatrix} J_m(\theta)\, J_b(\phi) \end{bmatrix} $$

$$ \dot{x}_s = J_s(\Phi)\dot{\Phi} $$

$$ \tau_s = J_s^T(\Phi) \times F_s \tag{7} $$

where $\tau_s = [\tau_m\ \tau_b]^T$ and $F_s = F_b + F_m$.

The limits on the range of motion Φ of each actuator and the amount of force or torque τ that the actuator can apply are

$$ \Phi_{\text{lo}} \le \Phi \le \Phi_{\text{hi}} $$

$$ \tau_{\text{lo}} \le \tau \le \tau_{\text{hi}} \tag{8} $$

where the inequalities are applied componentwise. Additional constraints may be placed on Φ to prevent collisions with the base or the arm itself if necessary.

Thus, for any task in the set S, we want to find a range of actuator position values such that ${}^{I}T_{E} = {}^{I}T_{T}$, $\tau_s = J_s^T F_s$, and the torque/force limits of the actuators are not exceeded. We define this range of values to be a *solution set*. This set may be empty if the desired position or orientation cannot be achieved or if the desired force/moment is too large. When the set is not empty, we must determine the optimal positions and orientations for performing the tasks. To find a desirable solution, we define a cost C for performing the tasks in a task set S. We introduce a cost function that in some sense measures the energy or time required to perform the sequence of tasks. This is expressed as

$$ C = \alpha \sum_{i=1}^{n} \delta_b({}^{I}T_{B_{i-1}}, {}^{I}T_{B_i}) + \beta \sum_{i=1}^{n} \delta_m(\theta_{i-1}, \theta_i) \tag{9} $$

where ${}^{I}T_{BO}$ and θ_0 represent the initial configuration of the system, δ_b and δ_m are generalized distance functions representing the relative cost of moving the base and manipulator, respectively, between two tasks, and α and β are weighting factors. We now define the problem: Given a task set and a mobile manipulator system, determine a path for the system such that all tasks in the set are performed for some minimum cost, that is,

$$ \min C = \alpha \sum_{i=1}^{j} \delta_b({}^{I}T_{B_{i-1}}, {}^{I}T_{B_i}) + \beta \sum_{i=1}^{j} \delta_m(\theta_{i-1}, \theta_i) \tag{10} $$

subject to

$$ {}^{I}T_{E_i} = {}^{I}T_{T_i}, \qquad i = 1, 2, \ldots, j $$

$$ \tau_i = J_s^T(\Phi) \times F_{T_i}, \qquad i = 1, 2, \ldots, j $$

with the motion and torque constraints specified by (8).

III. Simulated Annealing

The problem that we propose is a nonlinear optimization problem with nonlinear, nonconvex constraints and unconnected feasible regions. Solutions to this problem obtained with any gradient descent method depend on the initial conditions and cannot be considered optimal. The best that one can hope for is an application-specific heuristic to produce *intelligent* initial guesses for the optimization routine. An example of such a solution is presented in [14].

The literature provides a limited number of nongradient techniques for solving the general, global optimization problem. Horst and Tuy [18] present a class of derivative-free optimization procedures and examine branch-and-bound techniques, Pinter's procedures; Pijavskii, Shubert, and Mladineo's algorithms; and Zheng and Galperin's technique. Though some of the proposed techniques can solve the general problem in theory, they are limited in their practical applications, requiring either limiting constraints to be placed on the problem, or excessive computation.

In order to solve our problem in the completely general case in reasonable time, we examine the simulated annealing technique. This technique does not guarantee an optimal solution, but has been successful in finding near-optimal solutions for a number of difficult problems. It has been particularly useful with large combinatorial problems. In simulated annealing algorithms, a *temperature* is assigned to the system, and the cost of the initial state is calculated. The system is then passed through a *cooling* process. At each temperature, new states are determined and their respective costs are evaluated. If the cost of a new state is lower than the current cost, the new state is accepted. This is very much like a gradient descent process. Unlike a strict descent process, however, if the new state produces a higher cost, there is a finite probability that the new state will be accepted despite the higher cost, allowing the algorithm to escape local minima. The probability P of accepting a higher cost state is related to both the change in cost, Δ cost, and the current temperature, T_c, by an expression $P = e^{(-\Delta \text{cost}/T_c)}$. Thus, the smaller the increase in cost and the higher the temperature, the more likely it is that the new state will be accepted.

New states are generated until the system reaches equilibrium at the current temperature. This condition is described by a steady-state probability distribution of the accessible states. Once a system has reached equilibrium, the temperature is reduced and the process is repeated. When the system has cooled sufficiently, it is unlikely that a higher cost solution will be accepted due to the low temperature. At this point, the technique behaves as a gradient descent algorithm and approaches the local minima. The simulated annealing process is then stopped and the current solution is taken to be the optimal result. The interested reader is directed to [12] and [19] for a more complete description of simulated annealing.

Attempts to improve the performance of simulated annealing routines [20], [21] typically involve complex cooling schedules and equilibrium determinations. In the interest of simplicity, our initial algorithm, shown in Fig. 1, used a constant cooling rate and constant-length Markov chains at each temperature. In order to determine an initial high temperature, T_{hi} is set to 1. A number of new states are then generated and the change in cost from the initial state to the new state and the value of T_{hi} are used to determine whether or not to accept the new state. If 60% or more of the new states are accepted, the current value of T_{hi} is used as the initial temperature. Otherwise, T_{hi} is replaced by $2 \times T_{\text{hi}}$ and the process is repeated.

Once T_{hi} has been calculated, the cooling process begins. We assume that the system is in equilibrium at a given temperature after we have made 15 $\times$ (number of tasks) random steps. These Markov chains are created by making random changes in the redundant manipulator joint values and calculating the remaining values using the inverse kinematics of the system. While this introduces extra calculations whose complexity depends on the size of the system, it confines the guesses to the small subset of the search space that meets the kinematic constraints of the robot. If the new values do not meet the torque limit constraints, the state is immediately removed and another state is generated. Once a valid state has been generated, it is evaluated for cost and the simulated annealing

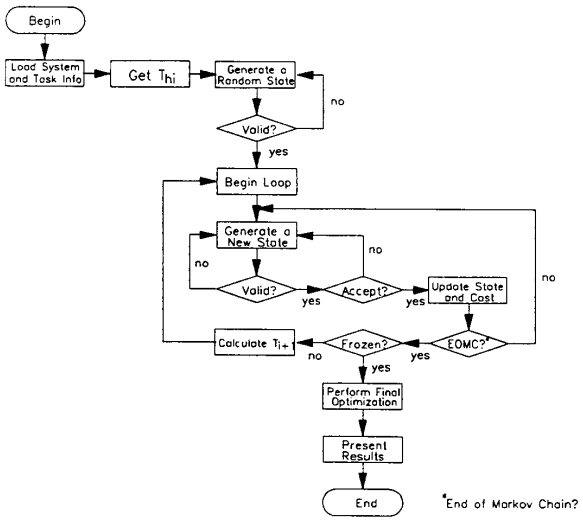

Fig. 1. Simulated annealing flowchart.

algorithm continues. When the system is considered to be in equilibrium, we calculate a new temperature, $T_{i+1} = 0.95 \times T_i$. Finally, the system is considered *frozen* when the cost function remains constant for ten sequential temperatures. The system temperature is then set to zero and one additional attempt to reach equilibrium is made.

To improve the quality of our solutions, we made two additions to the basic simulated annealing algorithm. The first involves a variable random step size. At high temperatures, we allow for the possibility of a very large change in value for each parameter in the decision space. This is done so that the system can reach all of the possible unconnected regions. As the system temperature is decreased, so is the maximum allowable random step size, so that $\text{step_size}_{i+1} = 0.99 \times \text{step_size}_i$. This prevents the generation of widely varying states at low temperatures when they are not likely to be accepted anyway. The second addition we made is to maintain a memory of the lowest cost state encountered during the optimization. Before the final attempt to reach equilibrium, we compare the frozen cost with the cost in memory. If the frozen cost is less than or equal to the remembered cost, we continue the process; otherwise, we replace the frozen cost and state with the better solution and perform the final calculations from this state. We included this operation in an attempt to better solve the unconnected regions problem. Experimentation showed that it was possible for the system to reach optimal regions early, but then escape and become trapped as the algorithm progressed. The extra chance to return to a more optimal solution greatly improved the performance of the algorithm.

IV. Application Example

As an example, we consider a simple 3-DOF arm mounted on a 2-DOF mobile base as shown in Fig. 2. The manipulator joints, θ_1, θ_2, and θ_3, are referred to as the trunk, shoulder, and elbow, respectively. The manipulator link lengths are $l_1 = 1.0$, $l_2 = 1.5$, and $l_3 = 0.75$, and the torque constraints for all three joint actuators is 5.0. The base is omnidirectional, that is, the base can achieve arbitrary point-to-point motion without regard to its orientation. (An example of such a wheelbase developed in our laboratory is described in [22].) For the purposes of this example, we make several simplifications to the general problem. Each task is characterized

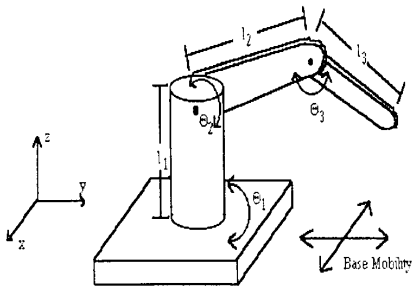

Fig. 2. Example mobile manipulator system.

only by a required position vector and minimum force application capability. Orientation and moments are not considered. In addition, the mobile base is not used to apply any of the desired force, but is assumed to apply sufficient force to remain stationary.

Since we are interested only in the position of the end-effector, the general transformation ${}^{I}T_{E}$ in (4) can be represented by the position relation

$$X_{\text{tip}} = X_{\text{base}} + X_{\text{man}}(\theta) \tag{11}$$

where $X_{\text{tip}} = [x_{\text{tip}}\ y_{\text{tip}}\ z_{\text{tip}}]^T$ is the position of the end-effector in the global inertial reference frame, $X_{\text{base}} = [x_{\text{base}}\ y_{\text{base}}\ 0]^T$ is the position of the base in the inertial frame, and $X_{\text{man}}(\theta)$ is as follows:

$$X_{\text{man}}(\theta) = \begin{bmatrix} \cos(\theta_1)(l_2\cos(\theta_2) + l_3\cos(\theta_2+\theta_3)) \\ \sin(\theta_1)(l_2\cos(\theta_2) + l_3\cos(\theta_2+\theta_3)) \\ l_1 + l_2\sin(\theta_2) + l_3\sin(\theta_2+\theta_3) \end{bmatrix}. \tag{12}$$

The manipulator Jacobian is

$$J_m = \begin{bmatrix} -S_1(l_2C_2 + l_3C_{23}) & -C_1(l_2S_2 - l_3S_{23}) & -l_3C_1S_{23} \\ C_1(l_2C_2 + l_3C_{23}) & -S_1(l_2S_2 + l_3S_{23}) & -l_3S_1S_{23} \\ 0 & l_2C_2 + l_3C_{23} & l_3C_{23} \end{bmatrix} \tag{13}$$

where $C_n = \cos(\theta_n)$, $S_n = \sin(\theta_n)$, $C_{nm} = \cos(\theta_n + \theta_m)$, and $S_{nm} = \sin(\theta_n + \theta_m)$.

The sequence of tasks to be performed is specified by the vector sets $\{\{X_{\text{tip},1}, F_1\}, \{X_{\text{tip},2}, F_2\}, \ldots, \{X_{\text{tip},n}, F_n\}\}$ which specify the location and force vector required to perform a sequence of n operations or tasks in the workspace. The constraints imposed by these vectors for the ith task can be expressed by the following equality and componentwise inequality constraints:

$$\begin{aligned} X_{\text{tip},i} &= X_{\text{base},i} + X_{\text{man}}(\theta^i) \\ \tau^i &= J_m(\theta^i)^T \times F_i \\ X_{\text{base,lo}} &\le X_{\text{base},i} \le X_{\text{base,hi}} \\ \theta_{\text{lo}} &\le \theta^i \le \theta_{\text{hi}} \\ \tau_{\text{lo}} &\le \tau^i \le \tau_{\text{hi}}. \end{aligned} \tag{14}$$

We assume the relative cost of executing a particular base/manipulator trajectory between two points can be expressed as a function of the trajectory endpoints. This assumption is reasonable when no on-line obstacle avoidance is required. In addition, we consider the cost of moving the base to be much more expensive than the cost of moving the arm. Then, for the n task situation, we define the distance function δ_b in (9) to be the square of the Euclidean distance between base positions, and the distance δ_m to be negligible. Further, we take the initial position of the system to be the desired

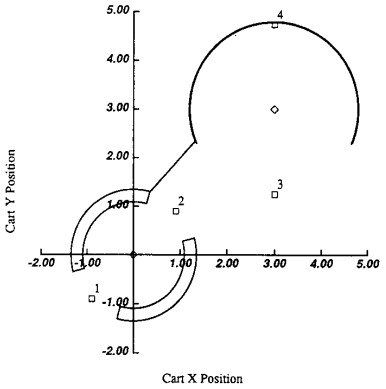

Fig. 3. Two-task-minimum base motion problem.

position for the first task. The cost function is then defined as

$$C = \sum_{i=2}^{n} (x_{\text{base},i} - x_{\text{base},i-1})^2 + (y_{\text{base},i} - y_{\text{base},i-1})^2. \tag{15}$$

If the robot is to return to task 1 after performing task n and repeat the tasks, then the cost is

$$C_{\text{cycle}} = C + (x_{\text{base},n} - x_{\text{base},1})^2 + (y_{\text{base},n} - y_{\text{base},1})^2. \tag{16}$$

V. Numerical Results

Fig. 3 shows the result of one simulation as seen from above, looking down at the floor. The diamonds represent the location of the desired tasks $X_{\text{tip},1} = [0.0\ 0.0\ 2.8]^T$, $X_{\text{tip},2} = [3.0\ 3.0\ 2.1]^T$, $F_1 = 2.25i + 2.25j$, and $F_2 = 2.25j - 2.25k$, projected onto the X-Y plane. The numbered boxes indicate four potential base positions, while the outlined regions represent all the valid base locations surrounding each task. In this problem, there is no reason to consider the cyclic cost function, and thus the cost function can be expressed as:

$$C = (x_{\text{base},2} - x_{\text{base},1})^2 + (y_{\text{base},2} - y_{\text{base},1})^2. \tag{17}$$

The line drawn between the regions indicates the trajectory chosen by our algorithm. While it is obvious that this represents nearly the shortest distance between the two task base regions, it may not be obvious that the regions represent all the valid base locations.

For any task location in the example, there will be a ring of base locations that allow the manipulator to reach the desired point. When we impose force constraints on the problem, some portion of this ring may become invalid. If we examine the first task, we see that two regions are missing about the line $y = x$. If we imagine that the base is sitting on this line and reaching toward the task point (positions 1 and 2 in Fig. 3), we see that most of the desired force vector must be generated by the shoulder joint of the manipulator. The trunk joint can supply no force in the desired direction, while the elbow is extended too far to aid the shoulder very much. As we move off this line, the trunk joint can assist the shoulder and the torque constraints are no longer violated.

The feasible base region for the second task also makes sense if we examine the geometry of the problem as shown in Fig. 4. We desire to apply a force in the positive Y direction and the negative Z direction. If we are in position 3 indicated in Fig. 3, the robot is configured as shown in Fig. 4(a). The desired force must be generated only by the shoulder joint. If instead, we are in position 4 indicated in Fig. 3 the robot will be as shown in Fig. 4(b). The desired force can now be generated by both the shoulder and the

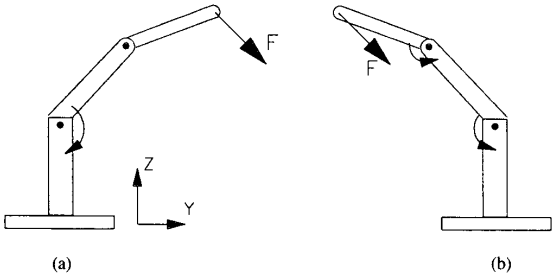

Fig. 4. Manipulator geometry during force application.

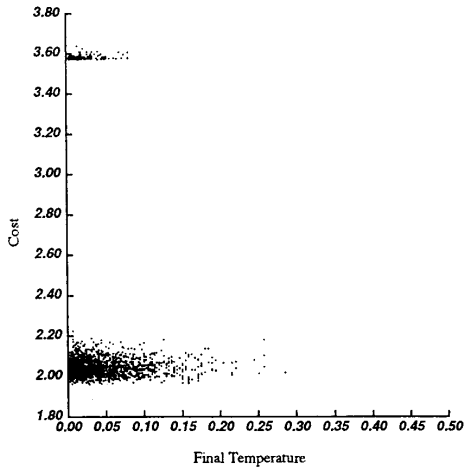

Fig. 5. Sample simulated annealing results.

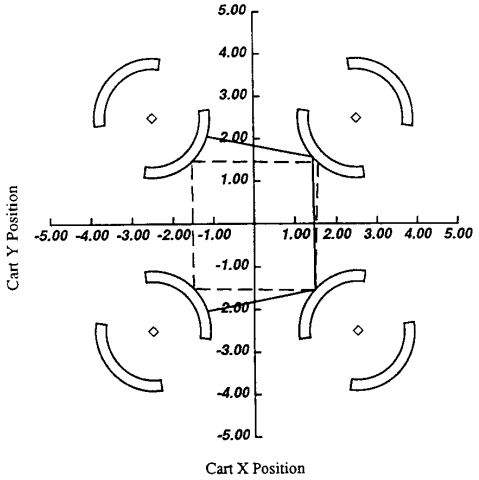

Fig. 6. Four-task-minimum base motion problem.

TABLE 1
RANGE OF SIMULATED ANNEALING SOLUTIONS

Cost Function Value	Upper Limit	% of Solutions in Range
optimal cost	1.9567	0.0
< 1% above optimal	1.9763	1.3
< 2% above optimal	1.9958	8.6
< 5% above optimal	2.0545	59.8
< 10% above optimal	2.1524	90.4
< 15% above optimal	2.2502	91.7
non-near-optimal		8.5

elbow and the torque constraints are not violated. The region about task 2 is very narrow because the desired task height is near the lower limit of the workspace for the manipulator. Thus, we can see that the regions indicated in Fig. 3 are reasonable and that the simulated annealing routine has returned a near-optimal solution for the given cost function.

The simulated annealing process is a Monte Carlo type solution, and thus we cannot determine the success of our solution from a single run. We use the results of our previous work [14] to determine the optimal solution to the above stated problem. This solution has a cost value of 1.9567, compared to the cost of 1.9942 produced by the simulated annealing run presented previously. If both of these solutions were mapped onto Fig. 3, it would be very difficult to distinguish one from the other. However, one cannot always expect such a close result.

Fig. 5 shows the results of 2000 simulated annealing runs for the example problem. The graph shows the final calculated cost versus the temperature at which the system was considered frozen. It is apparent that most of the solutions are near the optimal value, but that there is a set of results that are quite far above the optimal cost. If one examines this set, it is found that these solutions are trapped in a nonoptimal unconnected region against a joint space boundary. This problem might be avoided by customizing the algorithm for this particular system. With the current algorithm we obtained the results shown in Table I.

As a second example, we look at the four-task problem defined as follows: $X_1 = [-2.5\ 2.5\ 2.75]^T$, $F_1 = 2.5i + 2.5j$, $X_2 = [2.5\ 2.5\ 2.75]^T$, $F_2 = -2.5i + 2.5j$, $X_3 = [2.5\ -2.5\ 2.75]^T$, $F_3 = 2.5i + 2.5j$, $X_4 = [-2.5\ -2.5\ 2.75]^T$, $F_4 = -2.5i + 2.5j$. This definition produces the set of valid base locations shown in Fig. 6.

Two different costs were defined for this task set. The first result, represented by the solid line, shows the best path for the mobile base to follow to perform the four tasks. The second solution, shown with the dashed line, indicates the best path for the base to follow if the series of tasks is to be repeated. This path is more intuitively obvious, but both can be shown to be near-optimal solutions to the respective problems. If we examine the relative costs of the two solutions, we find that the cost of the three-link path is 23.78, while the cost of the cyclic path is 36.22. To make a useful comparison, however, we need to look at an equivalent number of path segments. If we extend the three-link path to include the fourth segment and make it become cyclic, we see the new path has a cost of 40.58: considerably worse than the cyclic path chosen by the simulated annealing algorithm. Likewise, if we use only the first three segments of the cyclic path, the cost is 27.32: again, worse than the noncyclic path originally chosen.

Though we present only a single solution here, all of the solutions generated for this problem were nearly optimal. The system was never trapped in a nonoptimal unconnected region in either the cyclic or noncyclic problems.

In addition to the examples presented in this work, we have examined simulated annealing solutions for three-task and ten-task problems. We have tested cyclic and noncyclic cost functions based on mobile base motion, manipulator joint motion, and a combination of the two. In all cases our algorithm was effective in finding satisfactory solutions without being tuned to the particular problem under study. Further, because we have not made use of any particular features of the example system to control the algorithm as

in our earlier work [14], there is reason to believe that the algorithm can be used with any mobile manipulator system.

A potential problem associated with using simulated annealing is that it may require an excessive amount of computer time. The data for Figs. 3, 5, and 6 were obtained on a Micro Vax II. We required an average of only 45 s of CPU time to solve the two-task problem, while the four-task problem required 140 to 170 s to solve depending on whether a cyclic solution was desired or not. These solution times are of the same order of magnitude as those achieved in [14] and were achieved without optimizing our simulated annealing algorithm for speed.

VI. Summary

In this work we have formulated a sequential task problem for a mobile manipulator as a global optimization problem. The mobile manipulator system redundancies allow a number of tasks to be performed for some minimum cost. The tasks are defined simply by desired positions and orientations and desired forces and moments, and the cost is defined by the length of the trajectory of the mobile base. This cost is representative of a measure of time or energy required to position and orient the system for each task. We note that these definitions produce a nonlinear cost function with nonlinear and nonconvex constraints and unconnected regions, and demonstrate that satisfactory solutions can be obtained through the use of simulated annealing. While the computational time required for this optimization increases with the complexity of the system and the number of tasks to be performed, it is not excessive, and the optimization, performed offline, could result in tremendous savings if a series of tasks is to be repeated many times.

References

[1] R. Mann, W. Hamel, and C. Weisbin, "The development of an intelligent nuclear maintenance robot," in *Proc. IEEE Int. Conf. Robotics Automat.*, 1988.

[2] R. Carlton and S. Bartholet, "The evolution of the application of mobile robotics to nuclear facility operations and maintenance," in *Proc. IEEE Int. Conf. Robotics Automat.*, 1987, pp. 720-726.

[3] T. Yoshikawa, "Analysis and control of robot manipulators with redundancy," in *Robotics Research: The First Int. Symp.* Cambridge, MA: MIT Press, 1984, pp. 735-748.

[4] R. Dubey and J. Luh, "Redundant robot control for higher flexibility," in *Proc. IEEE Int. Conf. Robotics Automat.*, 1987, pp. 1066-1071.

[5] S. Lee, "Dual redundant arm configuration optimization with task-oriented dual arm manipulabilty," *IEEE Trans. Robotics Automat.*, pp. 78-97, Feb. 1989.

[6] S. Chiu, "Control of redundant manipulators for task compatibility," in *Proc. IEEE Int. Conf. Robotics Automat.*, 1987, pp. 1718-1724.

[7] ——, "Task compatibility of manipulator postures," *Int. J. Robotics Res.*, vol. 7, no. 5, pp. 13-21, Oct. 1988.

[8] J. Baillieul, "Kinematic programming alternatives for redundant manipulators," in *Proc. Int. Conf. Robotics Automat.*, 1985.

[9] L. Kelmar and P. Khosla, "Automatic generation of kinematics for a reconfigurable modular manipulator system," in *Proc. IEEE Int. Conf. Robotics Automat.*, 1988, pp. 663-668.

[10] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 6, pp. 721-741, Nov. 1984.

[11] M. Vecchi and S. Kirkpatrick, "Global wiring by simulated annealing," *IEEE Trans. Computer-Aided Design*, pp. 215-222, Oct. 1983.

[12] S. Kirkpatrick, G. Gelett, and M. Vecchi, "Optimization by simulated annealing," *Science*, pp. 621-680, May 1983.

[13] N. Metropolis, A. Rosenbluth, M. Teller, and E. Teller, "Equations of state calculations by fast computing machines," *J. Chem. Phys.*, vol. 21, pp. 1087-1092, 1953.

[14] W. Carriker, P. Khosla, and B. Krogh, "An approach for coordinating mobility and manipulation," in *Proc. IEEE Int. Conf. Syst. Eng.*, Aug. 1989.

[15] R. Chatila and J. P. Laumond, "Position referencing and consistent world modelling for mobile robots," in *Proc. IEEE Int. Conf. Robotics Automat.*, 1985, pp. 138-145.

[16] R. Arkin, "Navigational path planning for a vision based mobile robot," *Robotica*, vol. 7, pp. 37-42, Jan. 1989.

[17] G. Miller and E. Wagner, "An optical rangefinder for autonomous robot," AT&T Bell Labs., Tech. Memo. TM 11228-870330-02, 1987.

[18] R. Horst and H. Tuy, "On the convergence of global methods in multiextremal optimization," *J. Opt. Theory Applications*, pp. 253-271, Aug. 1987.

[19] B. Hajek, "A tutorial survey of theory and applications of simulated annealing," in *Proc. 24th Conf. Decision and Control*, 1985, pp. 755-760.

[20] E. Aarts, J. Korst, and P. van Laarhoven, "A quantitative analysis of the simulated annealing algorithm: A case study of the traveling salesman problem," *J. Statistical Phys.*, vol. 50, no. 1/2, pp. 187-206, 1988.

[21] D. Mitra, F. Romeo, and A. Sangiovanni-Vancentelli, "Convergence and finite-time behavior of simulated annealing," in *Proc. IEEE Conf. Decision and Control*, 1985.

[22] D. Feng, M. Friedman, and B. Krogh, "The servo-control system for an omnidirectional mobile robot," in *Proc. IEEE Int. Conf. Robotics Automat.*, May 1989.

Impedance Control with Adaptation for Robotic Manipulations

W.-S. Lu and Q.-H. Meng

Abstract—**Motivated by the algorithms of Craig *et al.* and of Slotine and Li for unconstrained motion control, two adaptive versions of impedance control are presented in this work. In our treatment, it has been assumed that some parameters in manipulator dynamics may be uncertain, and the measurements from the wrist force sensor utilized are imprecise. By introducing the concept of target-impedance reference trajectory (TIRT), which characterizes a desired dynamic relation of the end-point with the environment and a refined Lyapunov approach, it is shown that the adaptation mechanisms suggested by Craig *et al.* and Slotine and Li can be injected into Hogan's conventional impedance control. The two resulting algorithms are compared in terms of implementation feasibility as well as computation efficiency. Simulation results are presented to illustrate the proposed algorithms.**

I. Introduction

Stable execution of contact tasks by mechanical manipulators has been identified as one of the major challenges in robotics [1]-[3], and a great deal of researchers' attention has been attracted to the problem during the past decade. Among several proposed control schemes, it appears that the impedance control of Hogan [4], [5] provides an unified approach to unconstrained motion control, obstacle avoidance, and constrained manipulation. In addition, it has

Manuscript received January 11, 1990; revised October 3, 1990. This work was supported by the Government of Canada under the Networks of Excellence Program. Portions of this paper were presented at the Third International Symposium on Robotics and Manufacturing, Vancouver, BC, Canada, July 1990.

The authors are with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC, V8W 3P6, Canada.

IEEE Log Number 9144463.

1042-296X/91/0600-0408$01.00 © 1991 IEEE